TCS-TR-A-08-34

TCS Technical Report

On the Limits of Learning with Computational Models

by

Shane Legg, Jan Poland, and Thomas Zeugmann

Division of Computer Science

Report Series A

January 16, 2008

Hokkaido University
Graduate School of
Information Science and Technology

Email: thomas@ist.hokudai.ac.jp

Phone: +81-011-706-7684
Fax: +81-011-706-7684

On the Limits of Learning with Computational Models

Shane Legg
IDSIA
Galleria 2
CH-6928 Manno (TI), Switzerland
shane@idsia.ch

Jan Poland
ABB Switzerland Ltd. Corporate Research
Segelhof
CH-5405 Baden, Switzerland
jan.poland@ch.abb.com

Thomas Zeugmann
Division of Computer Science
Hokkaido University
N-14, W-9, Sapporo 060-0814, Japan
thomas@ist.hokudai.ac.jp

January 16, 2008

Abstract

This paper provides a short discussion concerning the state of the art in Bayesian learning theory with an emphasis on performance guarantees. In the second part of the paper, we outline some negative results indicating that there is no hope for a general learning algorithm that is computable and implementable, but powerful enough to learn any computable data.

> "When you have eliminated the impossible, whatever remains, . . . , must be the truth."
>
> Sherlock Holmes
> in *The Sign of Four* by Sir Arthur Conan Doyle

1. Introduction

Recent advances in technology have enormously increased our ability to collect, to gather and to store data in digital form, and to make this data available over networks. For example, the data collected in various fields such as biology, chemistry, finance, retail, telecommunications, astronomy, medicine or science in general, is growing extremely rapidly.

This data may be available as text, audio signals, digital images and video, molecular data among others, i.e., as digital media. However, these collections of data need processing, interpretation and information extraction in order to be useful. In particular, *learning*, extracting information from data, is fundamental for gaining *knowledge*. Clearly, powerful learning systems would be an enormous help in automatically extracting new interrelations, knowledge, patterns and the like from collections of data as described above. Consequently, there are many challenges to the field of machine learning and its foundations that require further efforts to develop the theories needed to provide, for example, performance guarantees.

One old and important principle of learning and discovery is the following: Consider different explanations of the data and rule out those explanations that are disproved by the data, i.e., that do not match the data. If this is implemented within a probabilistic framework, then a *Bayesian* learner is the result. The first part of this paper will discuss the state of the art in Bayesian learning theory and present performance guarantees.

The hypothesis set a learner works with should be sufficiently large to contain good explanations for the data, since a class containing only bad hypotheses cannot lead to successful learning. In particular, Bayesian learners can be endowed with so-called universal hypothesis classes that in principle allow any kind of computable data to be learned. On the other hand, learners themselves should be computable (and even efficiently computable) in order to be useful in practice, in contrast to universal Bayesian learners which are not computable. In the second part of the paper, we shall present some negative results on the learning capacities that are valid for *any* computable learner, irregardless of whether it uses hypotheses or not. These results indicate that there is no hope for a general learning algorithm that is computable and implementable, but powerful enough to learn any computable data.

The present work only covers a small part of the results obtained on "limits of learnability with computational models." Some topics are only briefly touched, such as "learning with expert advice," while many other important research areas are not mentioned at all, for instance "finite state predictability" (e.g. [10]) and the "learning of languages" (cf., e.g., [18, 23]).

2. Learning with Hypotheses: Bayes

For an introduction to Bayesian learning we refer the reader to Mitchell [22]. *Bayes' rule* is the key ingredient of Bayesian learning. It states that, for some observation, the current belief in each hypothesis should be updated by multiplying by the probability that the hypothesis assigns to the observation, after which the belief distribution is renormalized. We can write this as the famous equation

$$\text{Posterior}(\text{hypothesis}|\text{data}) = \frac{P(\text{data}|\text{hypothesis}) \cdot \text{Prior}(\text{hypothesis})}{P(\text{data})} \,. \qquad (1)$$

The reader should keep in mind that Bayes' rule is not a theorem in general. Under the assumption that the hypotheses and data are *both* sampled from a joint probability distribution that coincides with the prior $P(\text{hypothesis})$, Equation (1) would be a theorem. However, Bayes' rule is commonly not applied under such an assumption, in particular the distribution $P(\text{hypothesis})$ on the hypotheses is usually merely a *belief distribution*, there is no probabilistic sampling mechanism generating hypotheses assumed. Hence, Bayes' rule is motivated intuitively in the first place. Still, many optimality results and performance guarantees have been shown for Bayesian induction (e.g., in [5, 8, 3]), including the results presented in the following.

2.1. *What to learn? Hypotheses, history, inputs, observation spaces*

Let $\mathbb{N} = \{0, 1, 2, \dots\}$ denote the set of all natural numbers. We set $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. Furthermore, we use $\mathbb{R}$ to denote the set of all real numbers.

Let $\mathcal{X}$ be the observation space. We work in an *online prediction setup in discrete time*, that is, in each time step $t = 1, 2, \dots$, an observation $x_t \in \mathcal{X}$ is revealed to the learner. The task for the learner will be to predict x_t *before* it is observed. One question of fundamental technical impact concerns the structure of the observation space $\mathcal{X}$. We restrict our attention to the two most important cases of (a) $\mathcal{X}$ being discrete (finite or countable) and (b) continuous $\mathcal{X} \subset \mathbb{R}^d$ for suitable dimension $d \in \mathbb{N}^+$. As any discrete space can be mapped to a subset of $\mathbb{R}^d$, it is technically sufficient to restrict to $\mathcal{X} \subset \mathbb{R}^d$, which we shall do in the following (except for a few places where we explicitly deal with finite observation spaces).

A *hypothesis* ν specifies a probability distribution on the observation space $\mathcal{X}$. In the simplest case, when it does not depend on any input, these hypotheses represent the assumption that the observed data are independently identically distributed (i.i.d.). In all other cases, there is some space of *inputs* or *side information* $\mathcal{Z}$, and a hypothesis maps inputs to distributions on $\mathcal{X}$. In fact, technically, the inputs play no role at all, as we shall see in the following. We therefore may assume the existence of an arbitrary input space $\mathcal{Z}$ without any structure (which may consist of just one point, meaning that there are no inputs at all), and inputs are generated by an arbitrary process. This covers, among others, two of the most important learning setups: *Classification*, where the data is conditionally i.i.d. given the inputs, and *prediction of non-i.i.d. sequences*, where in each time step t we may define the input $z_t = (x_1, \dots, x_{t-1})$ to be the observation history seen so far. Generally, we shall denote the history of inputs *and* observations by

$$h_{1:t-1} = h_{<t} = (z_1, x_1, z_2, x_2, \dots, z_{t-1}, x_{t-1})$$

(observe that two pieces of notation have been introduced here).

Now, a hypothesis is formally defined as a function

$$\nu : \mathcal{Z} \to \mathcal{M}^1_{\mathrm{D+C}}(\mathcal{X}) \ .$$

Here, $\mathcal{M}^1_{\mathrm{D+C}}(\mathcal{X})$ denotes the set of all probability distributions on $\mathcal{X} \subset \mathbb{R}^d$, that are mixtures of discrete distributions (with nonzero mass concentrated on single points) and distributions with continuous density functions. This is expressed by the lower indices D and C in $\mathcal{M}^1_{\mathrm{D+C}}(\mathcal{X})$, respectively. We make this restriction mainly because we wish to be able to define all subsequent quantities, in particular Bayesian posteriors, effortlessly[1] and uniquely (except perhaps on a set of measure zero).[2] Furthermore, the upper index 1 in $\mathcal{M}^1_{\mathrm{D+C}}(\mathcal{X})$ refers to the fact that we only consider normalized distributions. That is, we have

$$\int d\nu(\cdot|z) = 1 \text{ for all } z \in \mathcal{Z} \ .$$

Note that we consistently use this integral notation, also for discrete observation spaces (in which case the integral is in reality a sum).

A Bayesian learner is always based on a *hypothesis class* $\mathcal{C} = \{\nu_1, \nu_2, \ldots\}$. In this work we restrict ourselves to discrete, i.e., finite or countable, hypothesis classes (and in the notation we assume a countable hypothesis class from now on, without loss of generality). Before the learning process starts, each hypothesis $\nu \in \mathcal{C}$ is endowed with a prior weight $w_\nu \in (0, 1)$, such that $\sum_{\nu \in \mathcal{C}} w_\nu = 1$, i.e., we assume Prior(hypothesis) = w_ν in (1).

Hypothesis classes considered in statistics are usually *continuously parameterized*. One motivation to study discrete classes is that they are technically simpler, so they can serve as a basis for the more advanced continuous case. In the continuous case, some Bayesian predictors such as MAP (see below) are not consistent at all, while others such as MML (minimum message length) [30, 31] and MDL (minimum description length) [27] require appropriate discretization. Also, countable hypothesis classes always admit stronger performance guarantees than possible for their continuously parameterized counterparts. In particular, we shall be able to show almost sure consistency, whereas only convergence in probability holds in the continuous case (e.g. in [2]).

Another particular motivation to consider discrete hypothesis classes arises in Algorithmic Information Theory. General continuous hypothesis classes are computationally not tractable. The largest hypothesis class which can be manipulated in the limit by a computer, is the class of all computable hypotheses on some fixed universal Turing machine, more precisely a prefix machine [20]. Thus each hypothesis corresponds to a program, and there are countably many programs. Each hypothesis has a natural description length, namely the length of the corresponding program. If we

[1] For instance, for a measure defined by a "devil's staircase," one has to spend additional effort in order to define everything properly, which is not the aim of the present work. However, this and other cases can be treated with the methods described here.

[2] The continuity assumption is technical. It can be immediately lifted and replaced by "uniform piecewise continuity," which means that there is a single partition of $\mathcal{X}$ such that the continuous parts of all distributions $\nu \in \mathcal{C}$ and for all $z \in \mathcal{Z}$ are continuous on each of the elements of the partition. Maybe it can be even further lifted.

agree that programs are binary strings, then a natural prior is defined by two to the power of the negative description length.

If we are dealing with a *universal* hypothesis class as defined in Algorithmic Information Theory, we need to be careful about the phenomenon of *probability leaks*: A hypothesis, that is a program on our universal Turing machine, may not produce output for certain inputs. Because of our inability to decide the halting problem, we cannot generally detect this case. As a consequence, there is no limit-computable way of defining hypotheses that are proper probability distributions, they are rather *semimeasures*. In this paper, we shall not address this issue further, instead we point to the references: Consistency theorems for the semimeasure case are known for marginalization [28, 16] and for MAP predictions [25], but not for stochastic model selection. All of the probability distributions considered in this paper will be proper measures.

We rewrite Bayes' rule (1) using new notation: For a hypothesis $\nu \in \mathcal{C}$, current prior weights $w_{\nu'}(h_{<t})$ of all hypotheses $\nu' \in \mathcal{C}$ depending on the history $h_{<t}$, input z_t, and observation x_t, we set the posterior weight of ν to

$$w_\nu(h_{1:t}) = \frac{\nu(x_t|z_t) \cdot w_\nu(h_{<t})}{\sum_{\nu' \in \mathcal{C}} \nu'(x_t|z_t) \cdot w_{\nu'}(h_{<t})} \ . \tag{2}$$

Note that we actually need to distinguish three variants of Bayes' rule (not to be confused with the three variants of Bayesian prediction discussed below): In the case of a discrete observation space, the quantities $\nu'(x|z)$ (and therefore also the sum in the denominator) are probabilities, while for continuous observation space, they are densities. Finally, if at least one hypothesis $\nu \in \mathcal{C}$ is a mixture of a discrete and a continuous distribution, then *all* $\nu'(x|z)$ must be treated as mixtures in the following way: If for an observation $x \in \mathcal{X}$, there is a hypothesis assigning non-zero mass to x, then the $\nu'(x|z)$ are treated as probabilities (and all hypotheses assigning merely a non-zero density to that particular x will get posterior weight 0). Otherwise, the $\nu'(x|z)$ are treated as densities.

2.2. *How to learn? Three fundamental variants of Bayesian prediction*

Given a set of hypotheses $\mathcal{C}$ and some observed data $h_{1:t} = (z_1, x_1, \ldots, z_t, x_t)$, a legitimate question is asking which of the hypotheses in $\mathcal{C}$ actually generated the data. It is clear that this question might not be well-defined if the process generating the data, which we shall call μ in the sequel, is *not* a member of $\mathcal{C}$. Actually, one can immediately construct examples where any Bayesian learner produces very undesirable results in this non-realizable learning setup (see [14] for sophisticated examples). In this work, we shall restrict to the *realizable* case, where the true distribution generating the observations is contained in the class, that is, $\mu \in \mathcal{C}$. (Note that Sherlock Holmes always deals with the realizable case, as one of the people involved must be the bad guy. Recall that this only refers to the distribution of the observation *given* the inputs,

we do not need any assumption on the generation of the inputs z_t). Of course, the learner does not know in advance which element of $\mathcal{C}$ is the true distribution μ.

However, hypothesis identification has technical difficulties. For instance, consider the case where two hypotheses are in $\mathcal{C}$ that make (almost) identical predictions, one of them being the true one. Then it is (almost) impossible to identify the right one, but if we just want to make predictions, we do not need to take care: Choosing any of the two will yield (almost) perfect predictions.

So from now on, we restrict our focus to prediction. That is, for a given history $h_{<t}$ and current input $z_t \in \mathcal{Z}$, we are interested in a *predictive distribution*[3] on the observation space $\mathcal{X}$ that comes as close to the truth as possible. Our hypothesis class endowed with the Bayesian posterior $\big(w_{\nu'}(h_{<t})\big)_{\nu'\in\mathcal{C}}$ offers us *three fundamental* ways to obtain such a prediction. Before giving a formal description, we shortly explain the three ways of making predictions. In the first setting, the actual prediction is made by taking the weighted mean of all hypotheses (marginalization). The second setting is actually the oldest and perhaps most intuitive setting. Here we make the prediction in accordance with the hypothesis that has actually the highest belief value (maximum a posteriori). In the third setting, we again make the prediction in accordance with a single hypothesis but in difference to the second setting, this hypothesis is chosen randomly. Intuitively, we can imagine this random choice as throwing a dice having as much sides as there are hypotheses, and the probability of each side is equal to its actual belief value (stochastic model selection).

Next, we proceed formally.

1. **Marginalization**. If we apply Bayes' rule (1) to the modified setting where the next observation x_t takes the place of the hypothesis, then, as an easy computation shows, we get a predictive distribution $\xi(x_t|z_t, h_{<t})$ by integrating the predictions of *all* hypotheses w.r.t. the current posterior:

$$\xi(x|z_t, h_{<t}) = \sum_{\nu'\in\mathcal{C}} w_{\nu'}(h_{<t})\nu'(x|z_t) \ . \tag{3}$$

2. **Maximum a posteriori (MAP)**. If we are interested in a *single hypothesis'* prediction, then we may choose the *hypothesis with maximal a-posteriori belief value*, abbreviated as the MAP hypothesis:

$$\nu^*_{h_{<t}} = \arg\max_{\nu\in\mathcal{C}}\{w_\nu(h_{<t})\} \text{ and} \tag{4}$$

$$m(x_t|z_t, h_{<t}) = \nu^*_{h_{<t}}(x_t|z_t) \ , \tag{5}$$

where the latter $m(x_t|z_t, h_{<t})$ is the MAP prediction.

[3]In many prediction tasks, a single value is required as prediction, rather than a distribution. Such a single prediction can be derived from a predictive distribution, e.g. by minimizing a risk function, compare Corollary 4 below.

3. **Stochastic model selection**. The third possibility is to randomize and sample a hypothesis according to the probability distribution defined by the current posterior. This *stochastic model selection* can be formally written as

$$\Xi(x_t|z_t, h_{<t}) = \tilde{N}(x_t|z_t) \text{ where } \tilde{N} \in \mathcal{C} \quad \text{and } \mathbf{P}(\tilde{N} = \nu') = w_{\nu'}(h_{<t}) \text{ for all } \nu' \in \mathcal{C} \ . \tag{6}$$

Note that for a given history $h_{<t}$, the first two methods are deterministic, resulting in a fixed predictive distribution. Stochastic model selection uses additional randomness.

There are also other ways to use a Bayesian hypothesis class for prediction. MAP is tightly related to MML and MDL, but the terms MML and MDL are (also) used for (slightly, in the case of discrete hypothesis class) different concepts [9, 27]. Also, there is a "dynamic" variant of MAP defined in [25], where a MAP hypothesis is chosen for each possible outcome x_t and used for prediction. Nevertheless many, if not most, Bayesian prediction methods can be roughly grouped into the three fundamental approaches "integrate over all hypothesis," "take the hypothesis with the best current score," and "select one hypothesis at random according to the current belief distribution." Furthermore, we hold (but this is a matter of taste) that the three listed above are the simplest and most natural of the prediction methods to consider.

2.3. *Performance guarantees for Bayesian learners*

We are now ready to state the performance guarantees for the three Bayesian learners defined in (3), (5), and (6). We start with the technically easiest case of marginalization (3). Actually, this result has been originally discovered by Solomonoff [28] within the context of Algorithmic Information Theory.

Recall that $\mu \in \mathcal{C}$ is the true distribution generating the data, and ξ is the marginalization predictor. The quadratic *Hellinger distance* between the ξ-predictions and μ-predictions at time t is given by

$$\Delta_t^2(\mu, \xi) := \int d\big(\sqrt{\mu(\cdot|z_t)} - \sqrt{\xi(\cdot|z_t, h_{<t})}\big)^2. \tag{7}$$

It clearly depends on the history $h_{<t}$ and the current input z_t. Our main technical results are all stated as cumulative (i.e., over $t = 1, \ldots, \infty$) bounds on the Hellinger distance (that is, *errors*) of the predictive probabilities to the truth.

Theorem 1. *If $\mu \in \mathcal{C}$, then for* any *sequence of inputs $z_1, z_2, \ldots$,*

$$\sum_{t=1}^{\infty} \mathbf{E}_\mu \Delta_t^2(\mu, \xi) \leq \log w_\mu^{-1} \tag{8}$$

holds, where log *denotes the natural logarithm and* w_μ *is the prior weight of the true distribution.* $\mathbf{E}_\mu$ *refers to the fact that the expectation is taken w.r.t. the true distribution* μ*, i.e., all observations are generated w.r.t.* μ *conditional to the inputs, and this expectation is computed.*

It should not be surprising that the quantity w_μ appears on the r.h.s. and therefore has an impact on how large the error on the l.h.s. can grow. After all, if the Bayesian learner assigns a high prior weight to the true distribution, the error should be small. The remarkable fact is the *logarithmic* dependence in w_μ. As by Kraft's inequality, the logarithm of a weight can be interpreted as its description length, (8) is a very strong result asserting that the cumulative error never exceeds the description length of the true distribution. In a sense: When finding the truth single-handed, our error is at most the number of bits a teacher needs to tell us the truth.

Corresponding results for the MAP predictor (5) and stochastic model selection (6) have been proved in [25] and [24], respectively. They read as follows

Theorem 2. *Assume* $\mu \in \mathcal{C}$*. Suppose that, for any history with nonzero probability density, the hypotheses always admit the specification of a (not necessarily unique) MAP hypothesis* ν^**. This is satisfied for instance if all hypotheses correspond to continuous probability densities that are uniformly bounded. Then*

$$\sum_{t=1}^{\infty} \mathbf{E}_\mu \Delta_t^2(\mu, m) \leq 21 w_\mu^{-1}. \tag{9}$$

Theorem 3. *Assume* $\mu \in \mathcal{C}$*. Then, for any sequence of inputs* $z_1, z_2, \ldots$*,*

$$\sum_{t=1}^{\infty} \mathbf{E}_\mu \mathbf{E}_\Xi \Delta_t^2(\mu, \Xi) = O\big(w_\mu^{-1} + \Pi(\log \mathcal{H} + \log w_\mu^{-1})\big) = O(\Pi \log w_\mu^{-1}) \tag{10}$$

holds. The quantities $\mathcal{H}$ *and* Π*, the Shannon entropy and the* μ*-entropy potential of the hypothesis class, are defined below.* $\mathbf{E}_\Xi$ *serves as a reminder that the* Ξ*-predictor is randomized.*

The quantity $\mathcal{H}$ in the theorem is the Shannon entropy of the hypothesis class w.r.t. the current posterior distribution,

$$\mathcal{H}(h_{<t}) = \mathcal{H}\big([w_\nu(h_{<t})]_{\nu\in\mathcal{C}}\big) = -\sum_{\nu\in\mathcal{C}} w_\nu(h_{<t}) \log w_\nu(h_{<t}) \ .$$

Moreover, we define the current *entropy potential of the hypothesis class relative to the true distribution* μ as

$$\Pi\big((w_\nu)_{\nu\in\mathcal{C}}\big) = \sup\big\{\mathcal{H}\big((\tfrac{\tilde{w}_\nu}{\sum_{\nu'}\tilde{w}_{\nu'}})_{\nu\in\mathcal{C}}\big) : \tilde{w}_\mu = w_\mu \wedge \tilde{w}_\nu \leq w_\nu \ \forall \nu \in \mathcal{C}\setminus\{\mu\}\big\} \tag{11}$$

and $\Pi(h_{<t}) = \Pi\big([w_\nu(h_{<t})]_{\nu\in\mathcal{C}}\big)$. This can be paraphrased as "worst-case entropy of the class under all possible Bayesian updates where the true distribution always has evidence value 1."

Although we shall be only interested in consistency, that is asymptotic behavior, in the following, we briefly discuss the r.h.s. of the bounds (8), (9) and (10). The first one, $O(\log w_\mu^{-1})$, is excellent and non-improvable, as already discussed. The second one $O(w_\mu^{-1})$ is exponentially larger and can be huge in general. One can construct examples where this bound is sharp [26]. Fortunately, this does not necessarily imply that the MAP predictions are bad, the actual error (and also the bound) is smaller in many important cases. Still, there are situations where MAP predictions tend to be "unbalanced" and therefore unfavorable compared to marginalization. Stochastic model selection often gives better results in such cases, as long as the entropy potential Π is reasonably small, for instance of order $\log w_\mu^{-1}$. One can show [24] that this holds if the prior $(w_\nu)_{\nu\in\mathcal{C}}$ has sufficiently *light tails*, while in general, Π can grow as large as $\Omega(\mathcal{H} w_\mu^{-1})$.

2.4. *Implications: almost sure consistency and loss bounds*

One important consequence of any finite bound on the expected cumulative Hellinger error is *almost sure consistency* of the predictor in the Hellinger sense. That is, the Hellinger distance of the predictive to the true distribution tends to zero almost surely. In the case of a finite or countable observation space $\mathcal{X}$, this implies in particular convergence of all predictive probabilities $\xi(x_t|z_t, h_{<t})$ to the true probabilities $\mu(x_t|z_t)$. In the case of a continuous observation space, the predicted probability masses of any measurable subset of $\mathcal{X}$ converges to the true mass. However, we cannot establish the convergence of moments, e.g. the expectation, without making further assumptions.

Other implications of Theorems 1–3 are *loss bounds* on a Bayes-optimal decision maker based on the predictive distribution, w.r.t. arbitrary loss functions. The proof of the following corollary proceeds as that of [25, Theorem 27].

Corollary 4. *For each input z, let $\ell(\cdot,\cdot|z) : (\hat{x}, x) \mapsto \ell(\hat{x}, x|z) \in [0,1]$ be a loss function known to the learner, depending on the true outcome x and the prediction $\hat{x}$ (ℓ may also depend on the time, but we do not complicate notation by making this explicit). Let $\ell^\mu_{<\infty}$ be the cumulative loss of a predictor knowing the true distribution μ, where the predictions are made in a Bayes optimal way (i.e., choosing the prediction $\arg\min_{\hat{x}} \mathbf{E}_{x\sim\mu}\ell(\hat{x}, x|z_t)$ for current input z_t), and $\ell^\xi_{<\infty}$, $\ell^m_{<\infty}$, $\ell^\Xi_{<\infty}$ be the corresponding quantities for the respective Bayesian learners. Then the loss of each learner is bounded by*

$$\mathbf{E}\ell^\xi_{<\infty} \leq \mathbf{E}\ell^\mu_{<\infty} + O\left(\log w_\mu^{-1}\right) + O\left(\sqrt{\log w_\mu^{-1}\mathbf{E}\ell^\mu_{<\infty}}\right) \;, \tag{12}$$

$$\mathbf{E}\ell^m_{<\infty} \leq \mathbf{E}\ell^\mu_{<\infty} + O\left(w_\mu^{-1}\right) + O\left(\sqrt{w_\mu^{-1}\mathbf{E}\ell^\mu_{<\infty}}\right) \;, \text{ and} \tag{13}$$

$$\mathbf{E}\ell^\Xi_{<\infty} \leq \mathbf{E}\ell^\mu_{<\infty} + O\left(\Pi\log w_\mu^{-1}\right) + O\left(\sqrt{\Pi(\log w_\mu^{-1})\mathbf{E}\ell^\mu_{<\infty}}\right) \;, \tag{14}$$

respectively.

The bound may seem weak to a reader familiar with another learning model, *prediction with expert advice*, which has received quite some attention since [21, 29].

Algorithms of this type are based on a class of experts rather than hypotheses, and proceed by randomly selecting experts according to a (non-Bayesian) posterior based on past performance of the experts. It is straightforward to use a hypothesis as an expert. Thus the experts theorems (for instance [17, Theorem 8(i)]) imply a bound similar to (14), but *without any assumption on the data generating process* μ, instead the bounds are relative to the best expert (hypothesis) in hindsight $\hat{\nu}$ (and moreover with $\Pi \log w_\mu^{-1}$ replaced by $\log w_{\hat{\nu}}^{-1}$). So the experts bounds are stronger, which does not necessarily imply that the experts algorithms are better: bounds like (14) are derived in the worst case over all loss functions, and in this worst case Bayesian learning is not better than experts learning, even under the proper learning assumption. However, experts algorithms do not provide estimates for the probabilities, which Bayesian algorithms do provide: in many practically relevant cases learning probabilities does yield superior performance.

3. Computability and Computable Learners

The above performance guarantees for the Bayesian learners only hold in the realizable case. And, as indicated above, Bayesian learning can dramatically fail if this condition is violated. It is therefore natural to consider large hypothesis classes that are likely to contain the correct hypotheses even under very weak assumptions.

From now on, we restrict ourselves to the binary observation space, i.e., $x_t \in \mathbb{B}$ for all t, where $\mathbb{B} := \{0, 1\}$. The limit of hypothesis classes that are computationally tractable has been studied in Algorithmic Information Theory [20]. Of the many different possible constructions, the class of *all lower-semicomputable semi-probability distributions* has especially attractive properties: It is the smallest set which (a) contains all computable hypotheses and (b) is enumerable by a computational process (the class of only computable hypotheses lacks this second property). This means that, in principle, predictions according to Bayesian learners can be computed in some sense. Specifically, they can be only *lower semi-computed*, that is, approximated from below, where the learner has no information about the quality of the current approximation. We mention that in the context of sequence prediction and with this universal model class, the marginalization predictor ξ defines a semimeasure on binary strings that coincides with the algorithmic a-priori probability induced by the underlying universal Turing machine (see [20]).

Given that the Bayesian predictors are only lower semi-computable, are computable predictors possible at all? In the remainder of this paper we shall give a largely negative answer to this question. To this aim we restrict to the sequence prediction case, i.e., apart from the complete history, there is no side information available. Also, we restrict ourselves to the prediction of deterministic sequences. This is a special case of the stochastic setup considered so far (see, e.g., Mitchell [22]). In contrast to the first part of the paper, the following discussion includes the shorter proofs. For a more complete analysis we refer the reader to [19].

Before proceeding we have to introduce some notation. By $\mathbb{B}^*$ we denote the free monoid over $\mathbb{B}$ (cf. Hopcroft and Ullman [15]). We refer to the elements of $\mathbb{B}^*$ as strings. The empty string is denoted by λ. The length *lexicographical* ordering is a total order on $\mathbb{B}^*$ defined as $\lambda < 0 < 1 < 00 < 01 < 10 < 11 < 000 < 001 < \cdots$. A *substring* of x is defined $x_{j:k} := x_j x_{j+1} \ldots x_k$ where $1 \leq j \leq k \leq n$. By convention, $x_{1:0} := \lambda$. By $|x|$ we mean the length of the string x, for example, $|x_{j:k}| = k - j + 1$. Sometimes we need to encode a natural number as a string. Using simple encoding techniques it can be shown that there exists a computable injective function $f : \mathbb{N} \to \mathbb{B}^*$ where no string in the range of f is a prefix of any other, and $\forall n \in \mathbb{N} : |f(n)| \leq \log_2 n + 2 \log_2 \log_2 n + 1 = O(\log n)$.

Unlike strings which always have finite length, a *sequence* ω is an infinite list of symbols $x_1 x_2 x_3 \ldots$. We use $\mathbb{B}^\infty$ to denote the set of all infinite sequences over $\mathbb{B}$. Of particular interest to us will be the class of sequences which can be generated by an algorithm executed on a universal Turing machine. In particular, we say that a sequence $\omega \in \mathbb{B}^\infty$ is a *computable binary sequence* if there exists a program $q \in \mathbb{B}^*$ that writes ω to a one-way output tape when run on a monotone universal Turing machine $\mathcal{U}$, that is, $\exists q \in \mathbb{B}^* : \mathcal{U}(q) = \omega$. We denote the set of all computable sequences by $\mathcal{C}$. A similar definition for strings is not necessary as all strings have finite length and are therefore trivially computable.

We call a program $p \in \mathbb{B}^*$ that on a universal Turing machine $\mathcal{U}$ computes a total function $\mathbb{B}^* \to \mathbb{B}$ a *computable binary predictor*. For simplicity of notation we shall write $p(x)$ to mean the function computed by the program p when executed on $\mathcal{U}$ along with the input string x. Having $x_{1:n}$ as input, the objective of a predictor is for its output, called its *prediction*, to match the next symbol in the sequence. Formally, $p(x_{1:n}) = x_{n+1}$.

We shall only consider whether or not a predictor can learn to predict in the limit. Specifically, we say that a predictor p can *learn to predict* a sequence $\omega := x_1 x_2 \ldots \in \mathbb{B}^\infty$ if there exists $m \in \mathbb{N}$ such that $\forall n \geq m : p(x_{1:n}) = x_{n+1}$. This is essentially "next value" prediction as characterized by Barzdin [4], which has interesting connections to Gold's [13] notion of *identifiability in the limit*. We refer the interested reader to Freivalds, Bārzdiņš and Podnieks [11] and Ambainis *et al.* [1] for further information and more relations to the present work.

Central to our analysis will be the set of all predictors able to learn to predict ω, which we shall denote by $P(\omega)$. Similarly for sets of sequences $S \subset \mathbb{B}^\infty$, define $P(S) := \bigcap_{\omega \in S} P(\omega)$.

Kolmogorov complexity is a standard measure of complexity for sequences and is defined to be the length of the shortest program which generates the sequence. More formally, for any sequence $\omega \in \mathbb{B}^\infty$ the *Kolmogorov complexity* of the sequence is,

$$K(\omega) := \min_{q \in \mathbb{B}^*} \{|q| : \mathcal{U}(q) = \omega\},$$

where $\mathcal{U}$ is a universal Turing machine. If no such q exists, we define $K(\omega) := \infty$.

It can be shown that this measure of complexity depends on our choice of universal Turing machine $\mathcal{U}$, but only up to an additive constant that is independent of ω. This is due to the fact that a universal Turing machine can simulate any other universal Turing machine with a fixed length program. In essentially the same way as the definition above we can define the Kolmogorov complexity of a string $x \in \mathbb{B}^n$, written $K(x)$, by requiring that $\mathcal{U}(q)$ halts after generating x on the output tape. For an extensive treatment of Kolmogorov complexity and some of its applications see [20] or [6]. As many of our results will have the above property of holding within an additive constant that is independent of the variables in the expression, we shall indicate this by placing a small plus above the equality or inequality symbol. For example, $f(x) \stackrel{+}{<} g(x)$ means that that $\exists c \in \mathbb{R}, \forall x : f(x) < g(x) + c$.

3.1. *Prediction of computable sequences*

The most elementary result is that every computable sequence can be predicted by at least one predictor, and that this predictor need not be significantly more complex than the sequence to be predicted.

Lemma 5. $\forall \omega \in \mathcal{C}, \exists p \in P(\omega) : K(p) \stackrel{+}{<} K(\omega)$.

Proof. Consider a "predictor" p that only ever predicts ω, no matter what it has observed so far. Clearly p can be just a trivial modification of the shortest program that actually generates ω, and thus the result follows. ∎

Not only can any computable sequence be predicted, there also exist very simple predictors able to predict arbitrarily complex sequences:

Lemma 6. *There exists a predictor p such that $\forall n \in \mathbb{N}, \exists\, \omega \in \mathcal{C} : p \in P(\omega)$ and $K(\omega) > n$.*

Proof. Take a string x such that $K(x) = |x| \geq 2n$, and from this define a sequence $\omega := x0000\ldots$. Clearly $K(\omega) > n$ and yet a simple predictor p that always predicts 0 can learn to predict ω. ∎

Unfortunately, no universal predictor exists, indeed for every predictor there exists a sequence which it cannot predict at all:

Lemma 7. *For any predictor p there constructively exists a sequence $\omega := x_1x_2\ldots \in \mathcal{C}$ such that $\forall n \in \mathbb{N} : p(x_{1:n}) \neq x_{n+1}$ and $K(\omega) \stackrel{+}{<} K(p)$.*

Proof. For any computable predictor p there constructively exists a computable sequence $\omega = x_1x_2x_3\ldots$ computed by an algorithm q defined as follows: Set $x_1 = 1 - p(\lambda)$, then $x_2 = 1 - p(x_1)$, then $x_3 = 1 - p(x_{1:2})$ and so on. Clearly $\omega \in \mathcal{C}$ and $\forall n \in \mathbb{N} : p(x_{1:n}) = 1 - x_{n+1}$.

Let p^* be the shortest program that computes the same function as p and define a sequence generation algorithm q^* based on p^* using the procedure above. By construction, $|q^*| = |p^*| + c$ for some constant c that is independent of p^*. Because q^* generates ω, it follows that $K(\omega) \leq |q^*|$. By definition $K(p) = |p^*|$ and so $K(\omega) \stackrel{+}{<} K(p)$. ∎

Allowing the predictor to be probabilistic does not fundamentally avoid the problem of Lemma 7. In each step, rather than generating the opposite to what will be predicted by p, instead q attempts to generate the symbol which p is least likely to predict given $x_{1:n}$. To do this q must simulate p in order to estimate $\Pr\big(p(x_{1:n}) = 1\big|x_{1:n}\big)$. With sufficient simulation effort, q can estimate this probability to any desired accuracy for any $x_{1:n}$. This produces a computable sequence ω such that $\forall n \in \mathbb{N}$: $\Pr\big(p(x_{1:n}) = x_{n+1}\big|x_{1:n}\big)$ is not significantly greater than $\frac{1}{2}$, that is, the performance of p is no better than a predictor that makes completely random predictions.

As probabilistic prediction complicates things without avoiding this fundamental problem, we shall consider only deterministic predictors as this will highlight the key results most clearly.

3.2. *Prediction of simple computable sequences*

Given that the computable prediction of any computable sequence is impossible, a weaker goal is to be able to predict all "simple" computable sequences. To formalize this, for $n \in \mathbb{N}$, let $\mathcal{C}_n := \{\omega \in \mathcal{C} : K(\omega) \leq n\}$. Further, let $P_n := P(\mathcal{C}_n)$ be the set of predictors able to learn to predict all sequences in $\mathcal{C}_n$.

Firstly we note that prediction algorithms exist that can learn to predict all sequences up to a given complexity, and that these predictors need not be significantly more complex than the sequences they can predict:

Lemma 8. $\forall n \in \mathbb{N}, \exists p \in P_n : K(p) \stackrel{+}{<} n + O(\log n)$.

Proof. Omitted.

The question now is: Can we do better than this? Lemma 6 shows us that there exist predictors able to predict at least some sequences vastly more complex than themselves. This suggests that there might exist simple predictors able to predict arbitrary sequences up to a high complexity. Formally, could there exist $p \in P_n$ where $n \gg K(p)$? Unfortunately, these simple but powerful predictors are not possible:

Theorem 9. $\forall n \in \mathbb{N} : p \in P_n \Rightarrow K(p) \stackrel{+}{>} n$.

Proof. For any $n \in \mathbb{N}$ let $p \in P_n$, that is, $\forall \omega \in \mathcal{C}_n : p \in P(\omega)$. By Lemma 7 we know that $\exists\, \omega' \in \mathcal{C} : p \notin P(\omega')$. As $p \notin P(\omega')$ it must be the case that $\omega' \notin \mathcal{C}_n$, that is, $K(\omega') \geq n$. From Lemma 7 we also know that $K(p) \stackrel{+}{>} K(\omega')$ and so the result follows. ∎

Intuitively the reason for this is as follows: Lemma 7 guarantees that every simple predictor fails for at least one simple sequence. Thus if we want a predictor that can learn to predict all sequences up to a moderate level of complexity, then clearly the predictor cannot be simple. Likewise, if we want a predictor that can predict all sequences up to a high level of complexity, then the predictor itself must be very complex. Thus, even though we have made the generous assumption of unlimited computational resources and data to learn from, only very complex algorithms can be truly powerful predictors.

These results easily generalize to notions of complexity that take computation time into consideration. As sequences are infinite, the appropriate measure of time is the time needed to generate or predict the next symbol in the sequence. Under any reasonable measure of time complexity, the operation of inverting a single output from a binary valued function can be performed with little cost. If C is any complexity measure with this property, it is trivial to see that the proof of Lemma 7 still holds for C. From this, an analogue of Theorem 9 for C easily follows.

With similar arguments these results also generalize in a straightforward way to complexity measures that take space or other computational resources into account. Thus, the fact that extremely powerful predictors must be very complex, holds under any measure of complexity for which inverting a single bit is inexpensive.

3.3. *The limits of mathematical analysis*

Naturally, highly complex theories of prediction will be very difficult to mathematically analyze, if not practically impossible. Thus at some point the development of very general prediction algorithms must become mainly an experimental endeavor due to the difficulty of working with the required theory. Interestingly, an even stronger result can be proved showing that beyond some point the mathematical analysis is in fact impossible, even in theory:

Theorem 10. *In any consistent formal axiomatic system $\mathcal{F}$ that is sufficiently rich to express statements of the form "$p \in P_n$", there exists $m \in \mathbb{N}$ such that for all $n > m$ and for all predictors $p \in P_n$ the true statement "$p \in P_n$" cannot be proved in $\mathcal{F}$.*

In other words, even though we have proved that very powerful sequence prediction algorithms exist, beyond a certain complexity it is impossible to find any of these algorithms using mathematics. The proof has a similar structure to Chaitin's information theoretic proof [7] of Gödel incompleteness theorem for formal axiomatic systems [12].

Proof. For each $n \in \mathbb{N}$ let T_n be the set of statements expressed in the formal system $\mathcal{F}$ of the form "$p \in P_n$", where p is filled in with the complete description of some algorithm in each case. As the set of programs is denumerable, T_n is also denumerable and each element of T_n has finite length. From Lemma 8 and Theorem 9 it follows that each T_n contains infinitely many statements of the form "$p \in P_n$" which are true.

Fix n and create a search algorithm s that enumerates all proofs in the formal system $\mathcal{F}$ searching for a proof of a statement in the set T_n. As the set T_n is recursive, s can always recognize a proof of a statement in T_n. If s finds any such proof, it outputs the corresponding program p and then halts.

By way of contradiction, assume that s halts, that is, a proof of a theorem in T_n is found and p such that $p \in P_n$ is generated as output. The size of the algorithm s is a constant (a description of the formal system $\mathcal{F}$ and some proof enumeration code) as

well as an $O(\log n)$ term needed to describe n. It follows then that $K(p) \stackrel{+}{<} O(\log n)$. However from Theorem 9 we know that $K(p) \stackrel{+}{>} n$. Thus, for sufficiently large n, we have a contradiction and so our assumption of the existence of a proof must be false. That is, for sufficiently large n and for all $p \in P_n$, the true statement "$p \in P_n$" cannot be proved within the formal system $\mathcal{F}$. ∎

The exact value of m depends on our choice of formal system $\mathcal{F}$ and which reference machine $\mathcal{U}$ we measure complexity with respect to. However for reasonable choices of $\mathcal{F}$ and $\mathcal{U}$ the value of m would be in the order of 1000. That is, the bound m is certainly not so large as to be vacuous.

4. Conclusions

Within the present paper we have provided a short discussion concerning the state of the art in Bayesian learning theory with an emphasis on *performance guarantees*. Then we outlined some negative results indicating that there is no hope for a general learning algorithm that is computable and implementable, but powerful enough to learn any computable data.

We consider these results to be important for a deeper understanding of the capabilities and limitations of automated knowledge extraction from digital data sets. Therefore, the insight obtained should be considered as a step toward the development of a theory of knowledge and media.

References

[1] A. Ambainis, K. Apsītis, C. Calude, R. Freivalds, M. Karpinski, T. Larfeldt, I. Sala, and J. Smotrovs. Effects of Kolmogorov complexity present in inductive inference as well. In *Algorithmic Learning Theory, 8th International Workshop, ALT '97, Sendai, Japan, October 1997, Proceedings*, volume 1316 of *Lecture Notes in Artificial Intelligence*, pages 244–259. Springer, 1997.

[2] A. R. Barron and T. M. Cover. Minimum complexity density estimation. *IEEE Trans. Inform. Theory*, 37(4):1034–1054, 1991.

[3] A. R. Barron, J. J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Trans. Inform. Theory*, 44(6):2743–2760, 1998.

[4] J. M. Barzdin. Prognostication of automata and functions. In C. V. Freiman, J. E. Griffith, and J. L. Rosenfeld, editors, *Information Processing 71, Proceedings of IFIP Congress 71, Volume 1 - Foundations and Systems, Ljubljana, Yugoslavia, August 23-28, 1971*, pages 81–84. North-Holland, 1972.

[5] D. Blackwell and L. Dubins. Merging of opinions with increasing information. *Annals of Mathematical Statistics*, 33:882–887, 1962.

[6] C. S. Calude. *Information and Randomness*. Springer, Berlin, 2nd edition, 2002.

[7] G. J. Chaitin. Gödel's theorem and information. *International Journal of Theoretical Physics*, 22:941–954, 1982.

[8] B. S. Clarke and A. R. Barron. Information-theoretic asymptotics of Bayes methods. *IEEE Trans. Inform. Theory*, 36(3):453–471, 1990.

[9] J. W. Comley and D. L. Dowe. Minimum message length and generalized Bayesian nets with asymmetric languages. In P. D. Grünwald, I. J. Myung, and M. A. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*, pages 265–294. The MIT Press, 2005.

[10] M. Feder, N. Merhav, and M. Gutman. Universal prediction of individual sequences. *IEEE Trans. Inform. Theory*, 38(4):1258–1270, 1992.

[11] R. Freivalds, J. Bārzdiņš, and K. Podnieks. Inductive inference of recursive functions: Complexity bounds. In *Baltic Computer Science*, volume 502 of *Lecture Notes in Computer Science*, pages 111–155. Springer, 1991.

[12] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Matematik und Physik*, 38:173–198, 1931. [English translation by E. Mendelsohn: "On undecidable propositions of formal mathematical systems". In M. Davis, editor, *The undecidable*, pages 39–71, New York, 1965. Raven Press, Hewlitt].

[13] E. M. Gold. Language identification in the limit. *Inform. Control*, 10(5):447–474, 1967.

[14] P. Grünwald and J. Langford. Suboptimal behaviour of Bayes and MDL in classification under misspecification. In *Learning Theory: 17th Annual Conference on Learning Theory, COLT 2004, Banff, Canada, July 1-4, 2004. Proceedings*, volume 3120 of *Lecture Notes in Artificial Intelligence*, pages 331–347. Springer, 2004.

[15] J. Hopcroft and J. Ullman. *Formal Languages and their Relation to Automata*. Addison-Wesley, Reading, Mass., 1969.

[16] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2004.

[17] M. Hutter and J. Poland. Adaptive online prediction by following the perturbed leader. *Journal of Machine Learning Research*, 6:639–660, 2005.

[18] S. Jain, D. Osherson, J. S. Royer, and A. Sharma. *Systems that Learn: An Introduction to Learning Theory, second edition.* MIT Press, Cambridge, Massachusetts, 1999.

[19] S. Legg. Is there an elegant universal theory of prediction? In *Algorithmic Learning Theory, 17th International Conference, ALT 2006, Barcelona, Spain, October 2006, Proceedings*, volume 4264 of *Lecture Notes in Artificial Intelligence*, pages 274–287. Springer, oct 2006.

[20] M. Li and P. M. B. Vitányi. *An introduction to Kolmogorov complexity and its applications.* Springer, 2nd edition, 1997.

[21] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. In *30th Annual Symposium on Foundations of Computer Science*, pages 256–261, Research Triangle Park, North Carolina, 1989. IEEE.

[22] T. M. Mitchell. *Machine Learning.* WCB/McGraw-Hill, Boston, Massachusetts, 1997.

[23] D. N. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, Cambridge, Massachusetts, 1986.

[24] J. Poland. The missing consistency theorem for bayesian learning: Stochastic model selection. In *Algorithmic Learning Theory, 17th International Conference, ALT 2006, Barcelona, Spain, October 2006, Proceedings*, volume 4264 of *Lecture Notes in Artificial Intelligence*, pages 259–273. Springer, oct 2006.

[25] J. Poland and M. Hutter. Asymptotics of discrete MDL for online prediction. *IEEE Transactions on Information Theory*, 51(11):3780–3795, 2005.

[26] J. Poland and M. Hutter. MDL convergence speed for Bernoulli sequences. *Statistics and Computing*, 16(2):161–175, 2006.

[27] J. J. Rissanen. Fisher Information and Stochastic Complexity. *IEEE Trans. Inform. Theory*, 42(1):40–47, Jan. 1996.

[28] R. J. Solomonoff. Complexity-based induction systems: comparisons and convergence theorems. *IEEE Trans. Inform. Theory*, 24(4):422–432, 1978.

[29] V. G. Vovk. Aggregating strategies. In *Proc. Third Annual Workshop on Computational Learning Theory*, pages 371–383, Rochester, New York, 1990. Morgan Kaufmann.

[30] C. S. Wallace and D. M. Boulton. An information measure for classification. *Computer Journal*, 11(2):185–194, Aug. 1968.

[31] C. S. Wallace and D. L. Dowe. Minimum Message Length and Kolmogorov Complexity. *Computer Journal*, 42(4):270–283, 1999.