Machine Learning, 9, 165-203 (1992)

A Learning Criterion for Stochastic Rules

KENJI YAMANISHI YAMANISHI@IBL.CL.NEC.CO.JP
C&C Information Technology Research Labs., NEC Corporation, 1-1, Miyazaki 4-chome, Miyamaeku, Kawasaki, Kanagawa 216, Japan

Abstract. This paper proposes a learning criterion for stochastic rules. This criterion is developed by extending Valiant's PAC (Probably Approximately Correct) learning model, which is a learning criterion for deterministic rules. Stochastic rules here refer to those which probabilistically asign a number of classes, $\{Y\}$, to each attribute vector $\mathbf{X}$. The proposed criterion is based on the idea that learning stochastic rules may be regarded as probably approximately correct identification of conditional probability distributions over classes for given input attribute vectors. An algorithm (an MDL algorithm) based on the MDL (Minimum Description Length) principle is used for learning stochastic rules. Specifically, for stochastic rules with finite partitioning (each of which is specified by a finite number of disjoint cells of the domain and a probability parameter vector associated with them), this paper derives target-dependent upper bounds and worst-case upper bounds on the sample size required by the MDL algorithm to learn stochastic rules with given accuracy and confidence. Based on these sample size bounds, this paper proves polynomial-sample-size learnability of stochastic decision lists (which are newly proposed in this paper as a stochastic analogue of Rivest's decision lists) with at most k literals (k is fixed) in each decision, and polynomial-sample-size learnability of stochastic decision trees (a stochastic analogue of decision trees) with at most k depth. Sufficient conditions for polynomial-sample-size learnability and polynomial-time learnability of any classes of stochastic rules with finite partitioning are also derived.

Keywords. Learning from examples, stochastic rules, PAC model, MDL principle, stochastic decision lists, stochastic decision trees, sample complexity

1. Introduction

In 1984, Valiant introduced the PAC (Probably Approximately Correct) learning model (Valiant, 1984), which is a complexity-theoretic criterion for learning functions (or we call them *deterministic rules*). The goal in the PAC model is to find a good approximation of an unknown "function" by drawing random examples of it independently. Let random examples in the form of $(\mathbf{X}, Y)$ ($Y = f^*(\mathbf{X})$, f^* is called a *target function* or a *target rule*), $\mathbf{X} \in \mathcal{X} = \{0, 1\}^n$, n is a positive integer), of an unknown Boolean function on $\mathcal{X}$, be independently drawn according to the probability distribution $Q(\mathbf{X})$ over $\mathcal{X}$ ($\mathcal{X}$ is called the *domain*, and $\mathcal{Y} = \{0, 1\}$, the set of possible values which the target function takes, is called the *range*. The random variable $\mathbf{X} \in \mathcal{X}$ is called an *attribute vector*, and the variable $Y \in \mathcal{Y}$ is called a *class*). A *learning algorithm* takes these examples as input and outputs, with probability at least $1 - \delta$, a function g (called a *hypothesis*) that approximates f^* in the sense that the probability of $f^*(\mathbf{X}) \neq g(\mathbf{X})$ is at most ϵ for $\mathbf{X}$ drawn independently according to $Q(\mathbf{X})$. Here ϵ and δ are some suitably small positive real numbers, and are respectively called the *accuracy parameter* and *confidence parameter*. The learning algorithm

An extended abstract of this paper appeared in Proceedings of the 3rd Annual Workshop on Computational Learning Theory.

is required to run efficiently in the sense that the needed sample size and the computation time are both polynomial in $1/\epsilon$, $1/\delta$, and n.

In the original PAC model, examples were assumed to be perfectly noise-free. In the real world, however, examples will be afflicted with noise or other uncertainty (e.g., uncertainty which is due to lack of relevant attributes, etc.), and some examples with identical attributes may in fact differ in class values. That is, randomly drawn examples are not always in the form of $(\mathbf{X}, f^*(\mathbf{X}))$ for some target function f^*. "0" is sometimes possibly assigned to a given $\mathbf{X}$, and "1" is also sometimes possibly assigned to $\mathbf{X}$ with identical attribute values. The question of how to deal with such a kind of uncertainty in classification has been one of the main obstacles to the application of the conventional PAC model to actual problems.

Several noise models have been proposed within the PAC model: See, for example, Valiant (1985), Kearns and Li (1988), Angluin and Laird (1988), and Sloan (1988). In all of these, a rule to be learned is still assumed to be a function, and uncertainty in classification is dealt with as "noise" filtered out of the target function. For example, Angluin and Laird proposed a misclassification noise model (Angluin & Laird, 1988) as follows: With probability $1 - \nu$ the class value for an example of the target function f^* is correctly reported, and with probability ν it is incorrectly reported, where ν is an unknown noise probability. This kind of explanation for uncertainty is, however, too restricted since the rate of misclassification noise is uniform over all attribute vectors. Thus, we need to model the misclassification uncertainty itself without using the notion of noise filtered out through the target function.

Haussler generalizes the PAC model in a novel approach to a model of learning under uncertainty (Haussler, 1989) by letting $\mathcal{Y}$ be a continuous range. In Haussler's model, examples are assumed to be generated from a fixed probability distribution on $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{X}$: domain, $\mathcal{Y}$: range), where the class $Y \in \mathcal{Y}$ for a given $\mathbf{X} \in \mathcal{X}$ is not always determined by a "target function." The goal of Haussler's model is to find a good approximation of the function with minimal loss rather than that of the target function.

In this paper, we propose a new computational model of learning under uncertainty. It has been developed from the viewpoint of statistical estimation theory rather than from the decision-theoretic viewpoint on which Haussler's model is based. In our model, a rule refers not to a function but to a conditional probability distribution over the range $\mathcal{Y}$ for a given input attribute vector $\mathbf{X}$ in $\mathcal{X}$. We refer to such rules as *stochastic rules* in this paper. The notion of stochastic rules provides a natural way of dealing with uncertainty in classification. Thus, the goal of learning stochastic rules is to learn from examples not only the deterministic classification structure but also the uncertainty itself in the classification. This uncertainty is represented by 'probability.' A model of learning stochastic rules has also been considered by Kearns and Schapire (1990) independently of this work and they call such rules *probabilistic concepts*.

The purpose of this paper is twofold. First, we develop a learning criterion for stochastic rules by extending the PAC model in a manner different from that of Haussler's approach. Our proposed criterion is predicted on the basic idea that learning stochastic rules may be regarded as estimation of a conditional probability distribution over the range $\mathcal{Y}$ for a given attribute vector $\mathbf{X}$ in $\mathcal{X}$. Here we assume that the "target rule," which generates examples, belongs to some known class. Second, we derive upper bounds on the sample complexity of learning *stochastic rules with finite partitioning* (each of which is specified

by a countable model, belonging to a finite set, and a specific type class-assignment probability parameter vector) and also derive sufficient conditions for polynomial-sample-size learnability and polynomial-time learnability of any given class of stochastic rules with finite partitioning. Here the sample complexity refers to the smallest sample size over all learning algorithms required for our criterion to be satisfied.

Let us now briefly summarize our technical approach and discuss some of the previously reported research upon which we base our work.

Our proposed learning criterion is a generalization of Valiant's PAC model in the following three senses: 1) The objects to be learned are classes of stochastic rules rather than those of functions. Functions can be regarded as specific types of stochastic rules whose class-assignment probabilities are all 0 or 1. 2) Learning confidence (the probability that the hypothesis lies within some accuracy of the target rule) is measured in terms of a product probability distribution on $\mathcal{X} \times \mathcal{Y}$. 3) The difference between a hypothesis and the target rule is measured in terms of some notions of deviation between two probability distributions (e.g., the Hellinger distance, the variation distance, the Kullback-Leibler divergence, and the quadratic distance, etc.) rather than the probability of an erroneous prediction, i.e., the probability of a symmetric difference $(Q[\mathbf{X} : \hat{f}(\mathbf{X}) \neq f^*(\mathbf{X})]$ $(Q(\mathbf{X})$: probability distribution on the domain, f^*: a target function, $\hat{f}$: a hypothesis), which was used in the original PAC model.

Let ϵ be an accuracy parameter and let δ be a confidence parameter. In our definition of learnability of stochastic rules, a learning algorithm is required to take as input independent random examples generated independently according to $Q(\mathbf{X})P^*(Y|\mathbf{X})$ and to output, with probability at least $1 - \delta$, a hypothesis $\hat{P}$ such that $d(P^*, \hat{P}) < \epsilon$ for the target rule P^* with respect to some distance measure d, with sample size and computation time which are both polynomial in $1/\epsilon$, $1/\delta$, and n. This model of learnability contains Valiant's PAC learning model as a special case where targets are conditional probability distributions which take values in $\{0, 1\}$ only. We also stress here that Haussler's model (Haussler, 1989; 1990), which inspired Kearns and Schapire's model (Kearns & Schapire, 1990), is essentially different from our proposed model in the sense that the former requires that the expected loss for the hypothesis come within ϵ of the minimum one, but the latter requires that a hypothesis come within ϵ of the target rule itself with respect to some distance measure.

The learning algorithm used to derive our upper bounds on the sample complexity is different from those which have been used in the conventional PAC model or modified PAC models. The following types of algorithms have proven to perform well in the context of learning functions. 1) Consistent algorithms; algorithms outputting a hypothesis consistent with the given examples (Valiant, 1984; Rivest, 1987, etc.). 2) Minimum disagreement algorithms; algorithms outputting a hypothesis that best or approximately best fits the given examples (Kearns & Li, 1988; Sloan, 1988; Angluin & Laird, 1988, etc.). Examples of the type 1) algorithms include Blumer et al.'s Occam algorithm (Blumer, et al., 1987), which outputs an approximately minimum hypothesis which is consistent with the given examples. Examples of the type 2) algorithms include Kearns and Schapire's Occam algorithm (Kearns & Schapire, 1990), which outputs a hypothesis that approximately best fits the given examples and typically must be shorter than the sample size.

However, in the presence of uncertainty, one cannot simply consider that an optimal hypothesis is that which best fits the given examples because such a hypothesis may be

affected by statistical irregularities of the examples. That is, it may be "overfitting" the given examples. We use an algorithm based on the *MDL (Minimum Description Length) principle* as a learning algorithm for stochastic rules in order to avoid the overfitting problem and to make the hypotheses converge to the target rule faster. We refer to this algorithm as an *MDL algorithm*.

The MDL principle, on which our learning algorithm is based, was developed by Rissanen (1978; 1983; 1984; 1986; 1989), Wallace and Boulton (1968) and Solomonoff (1964), etc. This principle gives a strategy for selecting the best hypothesis from the class of hypotheses. The MDL principle asserts that the best hypothesis is that which requires the least code length in bits for the encoding of itself and the given examples observed through it. Intuitively, the best hypothesis is selected on the basis of the trade-off between its simplicity and its goodness of fit to the given examples.

The validity of the MDL principle in inductive learning has been reported in several empirical studies, e.g., inferring decision trees (Quinlan & Rivest, 1989), shape reconstruction (Pednault, 1989), shape recognition (Segen, 1989), classification rules with hierarchical parameter structures (Yamanishi, 1989; 1990a), etc. This paper specifically presents an MDL algorithm for learning stochastic decision lists (a stochastic analogue of Rivest's decision lists (Rivest, 1987). Further, this paper introduces a general family of stochastic rules, called *stochastic rules wtih finite partitioning*, each of which is specified by a countable model and a probability parameter vector, and presents an MDL algorithm for learning them. Our method for applying the MDL principle to learning from examples is basically predicated on Quinlan and Rivest's method (Quinlan & Rivest, 1989), but our algorithm gives a general strategy for learning general stochastic rules with finite partitioning.

In this paper, we derive upper bounds on the sample size required by the MDL algorithm to output, with probability at least $1 - \delta$, a hypothesis which lies within ϵ of the target rule with respect to the *Hellinger distance* or the *variation distance*. We derive them applying the type of the proof techniques used in Barron (1985) and Barron and Cover (1991). The sample-complexity estimation techniques used in this paper are different from the uniform convergence method used by Haussler (1989; 1990). He considers the case in which the distance between a hypothesis and the target function can be reduced to the difference of the expectations of some appropriate loss function. In his model, the uniform convergence method is used to estimate the rate at which the minimum empirical loss converges to the minimum expected loss and thereby give upper bounds on the sample complexity of learning functions. Unlike the uniform convergence method, in this paper, we directly estimate the probability with which the distance between the target rule and the MDL rule is less than a fixed accuracy parameter, making use of some properties of the MDL rules.

Before our investigation, Barron and Cover have proved that the Hellinger distance between the target probability distribution and the one estimated by the MDL principle is asymptotically bounded by a statistical quantity, what we call the 'index of resolvability,' which quantifies the complexity of the hypothesis space and its approximation error to the target distribution (Barron & Cover, 1991). Unlike Barron and Cover's results, our result gives a new aspect to the convergence for the MDL algorithm in that, in our approach, we measure the rate of convergence in terms of sample complexity as a function of the accuracy parameter, the confidence parameter and the parameters specifying the target rule. Further notice that our results are concerned with only the cases where the target rule is parametric.

Let $N_0(1/\epsilon, 1/\delta, P^*)$ be the smallest number of examples required by the MDL algorithm to produce, with probability at least $1 - \delta$, a hypothesis that lies within ϵ of the target rule, P^* with respect to the Hellinger distance. Ignoring time complexity, we prove that

$$N_0\left(\frac{1}{\epsilon}, \frac{1}{\delta}, P^*\right) = O\left(\frac{m^*}{\epsilon}\log\frac{m^*}{\epsilon} + \frac{\ell(M^*)}{\epsilon} + \frac{1}{\epsilon}\log\frac{1}{\delta}\right),$$

where m^* is the dimension of the probability parameter vector space of the target rule, and $\ell(M^*)$ is the description length for the countable model of the target rule. Notice that this sample complexity bound depends on the target rule.

Further, for stochastic rules with finite partitioning, we also derive an upper bound on the sample complexity of learning by the ML (Maximum Likelihood) algorithm, and compare it with that by the MDL algorithm. Here the ML algorithm is an algorithm that, from given examples, outputs a hypothesis that maximizes the likelihood for the examples, equivalently, minimizes the description length for them, ignoring rule complexities. We prove that the upper bound on the sample size required by the ML algorithm is of the same order as that for the MDL algorithm for the case in which the target rule has the largest-dimensional probability parameter vector over all hypotheses and, for the encoding of the countable models, we use a code-length function based on the uniform distribution over the set of countable models.

For a given class C of stochastic rules with finite partitioning, satisfying some appropriate conditions, we also derive an upper bound on the smallest sample size $N_0(1/\epsilon, 1/\delta, n)$ required by the MDL algorithm to produce, "for all" distribution Q on $\mathcal{X}$, "for all" target rule $P^* \in C$, with probability at $1 - \delta$, a hypothesis that lies within ϵ of P^*. $N_0(1/\epsilon, 1/\delta, n)$ is estimated as follows.

$$N_0\left(\frac{1}{\epsilon}, \frac{1}{\delta}, n\right) = O\left(\frac{\xi(C)}{\epsilon}\log\frac{\xi(C)}{\epsilon} + \frac{\log|\mathcal{M}|}{\epsilon} + \frac{1}{\epsilon}\log\frac{1}{\delta}\right).$$

$\xi(C)$ is the maximum value of the dimension of a probability parameter vector specifying a rule in C. $\mathcal{M}$ is the set of all countable models, each of which specifies a rule in C. ϵ and δ are respectively accuracy and confidence parameters. This sample complexity estimation implies that if $\xi(C)$ and $\log|\mathcal{M}|$ are polynomial in n, then C is statistically learnable (=learnable with sample size polynomial in $1/\epsilon$, $1/\delta$, and n) with respect to the Hellinger distance, the variation distance, and the quadratic distance. Using this sufficient condition for statistical learnability, we prove that the class of stochastic decision lists with at most k literals in each term and the class of stochastic decision trees (a stochastic analogue of decision trees) with at most k depth are statistically learnable, where k is fixed. Further, we derive a sufficient condition for polynomial learnability (learnability in time polynomial in $1/\epsilon$, $1/\delta$, and n) of any given class of stochastic rules with finite partitioning. This condition is characterized in terms of the existence of a polynomial-time algorithm that outputs a hypothesis which lies within a small accuracy of an output of the MDL algorithm.

The organization of the rest of the paper is as follows: Section 2 gives a formal definition of stochastic rules and a number of their examples. Section 3 gives a learning criterion for stochastic rules. Section 4 gives the MDL algorithm. Section 5 gives bounds on the sample sizes required by the MDL algorithm and the ML algorithm. Section 6 discusses learnability of classes of stochastic rules with finite partitioning.

2. Definition of stochastic rules and some examples

In this section, a formal definition of stochastic rules and a number of their examples are presented.

2.1. *General definition of stochastic rules*

Let n be a positive integer. Let $\mathcal{X}_i$ $(i = 1, \ldots, n)$ be finite or countably infinite and $\mathcal{Y}$ be finite. Here $\mathcal{X} \stackrel{\text{def}}{=} \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ is a measurable set which we call the *domain* and $\mathcal{Y}$ is a set which we call the *range*. $\mathbf{X} = (x_1, \ldots, x_n) \in \mathcal{X}$ is called an *attribute vector*, and $Y \in \mathcal{Y}$ is called a *class*. Let $\mathcal{P}$ be a family of probability distributions on $\mathcal{X} \times \mathcal{Y}$.

Let $\mathcal{F}$ be a family of functions from $\mathcal{X}$ to $\mathcal{Y}$. We call $f \in \mathcal{F}$ a *deterministic rule*. Let $Q(\mathbf{X})$ be a probability distribution over $\mathcal{X}$. The conventional learning problem dealt with in Valiant's PAC model (Valiant, 1984) is as follows: Given finite training examples (a sequence of examples is called a *sample*) $D^N = D_1 \ldots D_N$, $(D_i = (\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, $y_i = f^*(\mathbf{x}_i), f^* \in \mathcal{F}, i = 1, \ldots, N)$ drawn independently according to some unknown probability distribution $\pi \in \mathcal{P}$, find a good approximation of an unknown deterministic rule $f^* \in \mathcal{F}$ which we call a *target function*. Notice that in this learning framework, $\mathbf{X}$ is generated according to the distribution $Q(\mathbf{X})$, but Y is assumed to be uniquely determined by $\mathbf{X}$; i.e., $Y = f^*(\mathbf{X})$. We will extend this learning problem to the case in which Y is not uniquely determined by $\mathbf{X}$ but all classes in $\mathcal{Y}$ are probabilistically assigned to $\mathbf{X}$.

First note that for each $\pi \in \mathcal{P}$, π may be resolved as follows:

$$\pi(\mathbf{X}, Y) = Q(\mathbf{X})P(Y \mid \mathbf{X}), \tag{1}$$

where $\mathbf{X}$ is a random variable on $\mathcal{X}$ and Y is a random variable on $\mathcal{Y}$. In the sequel, $\mathbf{X} \in \mathcal{X}$, $Y \in \mathcal{Y}$ denote random variables, and $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$ denote observed values. $Q(\mathbf{X})$ is a probability distribution over $\mathcal{X}$ and $P(Y \mid \mathbf{X})$ is a conditional probability distribution over $\mathcal{Y}$ for a given $\mathbf{X} \in \mathcal{X}$. We call $P(Y \mid \mathbf{X})$ a *stochastic rule*. In other words, a stochastic rule assigns all classes in $\mathcal{Y}$ to each attribute vector $\mathbf{X}$, giving each class a specific probability for each individual attribute vector.

In the discussion below, we consider only classes of stochastic rules with finite partitioning, each of which is specified by a finite number of partitions of the domain $\mathcal{X}$ and real-valued probability parameter vectors associated with the partitions.

Definition 1 (Stochastic rules with finite partitioning). A finite set $\{S_i\}_{i=1,\ldots,m}$ $(m < \infty)$ of subsets of $\mathcal{X}$ is called (*finite*) *partitioning of* $\mathcal{X}$ if $\mathcal{X} = \cup_{i=1}^{m} S_i$, $S_i \cap S_j = \phi$ $(i \neq j)$. Each S_i $(i = 1, \ldots, m)$ is called a *disjoint cell of* $\mathcal{X}$. *The probabilities* $\{p_i(j)\}_{i=1,\ldots,m,j=1,\ldots,s}$ *associated with* $\{S_i\}_{i=1,\ldots,m}$ are defined by $p_i(j) \stackrel{\text{def}}{=}$ the probability that $Y = j$ for $\mathbf{X} \in S_i$, where $p_i(j) \in [0, 1]$ $(i = 1, \ldots, m, j = 1, \ldots, s)$. A *stochastic rule with finite partitioning* (specified by a finite number of partitions $\{S_i\}_{i=1,\ldots,m}$ and a set of associated probabilities, $\{p_i(j)\}_{i=1,\ldots,m,j=1,\ldots,s}$) is defined by the following type of stochastic rule:

If $\mathbf{x} \in S_1$,
  then $Y = 1$ with probability $p_1(1)$, $\ldots$, $Y = s$ with probability $p_1(s)$
else if $\mathbf{x} \in S_2$,
  then $Y = 1$ with probability $p_2(1)$, $\ldots$, $Y = s$ with probability $p_2(s)$
  . . . . . .
  . . . . . .
else if $\mathbf{x} \in S_{m-1}$,
  then $Y = 1$ with probability $p_{m-1}(1)$, $\ldots$, $Y = s$ with probability $p_{m-1}(s)$
else $Y = 1$ with probability $p_m(1)$, $\ldots$, $Y = s$ with probability $p_m(s)$ (2)

Letting C_{FP} be the class of all possible stochastic rules with finite partitioning, any class C of stochastic rules with finite partitioning is defined as a subset of C_{FP}. □

Explicitly, we denote a class C of stochastic rules with finite partitioning as

$$C = \{P(Y \mid \mathbf{X} : \theta \prec M) \in C_{FP} : M \in \mathcal{M}, \theta \in \Theta(M)\}.$$

Here $P(Y \mid \mathbf{X}: \theta \prec M)$ is a stochastic rule specified by a real-valued vector $\theta = (p_1(1), \ldots, p_1(s-1), \ldots, p_m(1), \ldots, p_m(s-1))$ which we call a *probability parameter vector* and by countable parameter M which we call a *countable model*. Here M specifies finite partitioning of the domain $\mathcal{X}$ with m disjoint cells. $\Theta(M) \subset [0, 1]^{m(s-1)}$ is a set of probability parameter vectors associated with M, and $\mathcal{M}$ is a set of all countable models, each of which specifies finite partitioning of the domain $\mathcal{X}$. We define dim $\Theta(M)$ as the dimension of $\Theta(M)$, i.e., dim $\Theta(M) = m(s-1)$. Notice that $p_i(s)$ is determined by $p_i(s) = 1 - \Sigma_{j=1}^{s-1} p_i(j)$ $(i = 1, \ldots, m)$.

Let C be a class of stochastic rules. In our formulation, the learning problem can be stated simply: "Given finite training examples $D^N = D_1 \ldots D_N$, $(D_i = (\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ $i = 1, \ldots, N)$ drawn independently according to some unknown product probability distribution on $\mathcal{X} \times \mathcal{Y}$; $Q(\mathbf{X})P^*(Y \mid \mathbf{X})(P^*(Y \mid \mathbf{X})$ is assumed to belong to $C)$, find a good approximation of the unknown stochastic rule $P^*(Y \mid \mathbf{X}) \in C$ efficiently." $P^*(Y \mid \mathbf{X})$ is called a *target rule*.

Angluin and Laird's misclassification noise model (Angluin & Laird, 1988) can be thought of as a model of learning specific type stochastic rules. Angluin and Laird's misclassification noise model is formulated as follows: With probability $1 - \nu$ the class value for an example of the target function f^* (Boolean function) is correctly reported, and with probability ν it is incorrectly reported, where ν is an unknown noise probability. Letting

$\mathcal{Y} = \{0, 1\}$, learning the target function f^* and estimating the noise probability ν in such a noise model can be regarded as learning a stochastic rule $P(Y \mid \mathbf{X})$ such that $Y = f^*(\mathbf{X})$ with probability $1 - \nu$ and $Y = 1 - f^*(\mathbf{X})$ with probability ν, where the probability parameter associated with the rule is only ν. That is, Angluin and Laird are trying to deal with classification uncertainty by filtering out the misclassification noise, but such a kind of classification uncertainty can be modeled using a specific conditional probability distribution in our stochastic setting.

2.2. *Stochastic decision lists and stochastic decision trees*

In this subsection, let us introduce a class of stochastic rules with finite partitioning, called *stochastic decision lists*. The stochastic decision lists are regarded as a stochastic analogue of Rivest's decision lists (Rivest, 1987).

Let $\mathcal{X}_1 = \ldots = \mathcal{X}_n = \mathcal{Y} = \{0, 1\}$. Let n be the number of attributes and let k be a given positive integer.

Each variable x_i takes a value in $\{0, 1\}$, and is associated with two literals: x_i itself and its negation $\bar{x}_i$. A natural assignment gives a value to $\bar{x}_i$: $\bar{x}_i = 0$ if $x_i = 1$, otherwise $\bar{x}_i = 1$. We denote the set of $2n$ literals as $L_n \stackrel{\text{def}}{=} \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$. A *term* is a conjunction of literals and can be interpreted a mapping from assignments into $\{0, 1\}$. For example, the term $x_1 \wedge \bar{x}_2 \wedge x_3$ is 1 if and only if $x_2 = 0$ and both x_1 and x_3 are 1. The *size* of a term is the number of literals. We denote the set of all terms of size at most k as T_k^n; i.e., $T_k^n \stackrel{\text{def}}{=} \{z_{i_1} \wedge z_{i_2} \wedge \ldots \wedge z_{i_d}; 0 \leq d \leq k, z_{i_j} \in L_n, z_{i_j} \neq z_{i_k} (j \neq k), \bar{z}_{i_j} \neq z_{i_k} (j \neq k)\}$. Here T_k^n includes the constant function "1," which is regarded as a term with $d = 0$. We denote an ordered set of m terms: $t_1, \ldots, t_m$ as $\langle t_1, \ldots, t_m \rangle$, where t_m is the constant function 1 and $t_i \neq t_j$ for $i \neq j$. We denote a set of all ordered sets in the form $\langle t_1, \ldots, t_m \rangle$ as Γ_k^n; i.e., $\Gamma_k^n \stackrel{\text{def}}{=} \{M = \langle t_1, \ldots, t_m \rangle : t_i \in T_k^n - \{1\}\ (i = 1, \ldots, m-1), t_i \neq t_j\ (i \neq j), t_m = 1, 1 \leq m \leq |T_k^n|\}$. p_i denotes the probability that $Y = 1$ for $\mathbf{X}$ such that $t_1(\mathbf{X}) = \ldots = t_{i-1}(\mathbf{X}) = 0$ and $t_i(\mathbf{X}) = 1$ $(i = 1, \ldots, m)$, where $0 \leq p_i \leq 1$ $(i = 1, \ldots, m)$. We let $\theta \stackrel{\text{def}}{=} (p_1, \ldots, p_m)$ be a probability parameter vector.

We define a *stochastic decision list* specified by $\theta = (p_1, \ldots, p_m)$, $M = \langle t_1, \ldots, t_m \rangle$, and k as a stochastic rule which gives a class-assignment; $Y = 1$ with probability p_i and $Y = 0$ with probability $1 - p_i$ for arbitrary $\mathbf{X}$, where i is the least index such that $t_i(\mathbf{X}) = 1$. That is, a stochastic decision list specified by θ and M has the following semantics:

For an attribute vector $\mathbf{X} \in \mathcal{X}$,
if $t_1 = 1$ for $\mathbf{X}$ then $Y = 1$ with probability p_1 ($Y = 0$ with $1 - p_1$)
else if $t_2 = 1$ for $\mathbf{X}$ then $Y = 1$ with probability p_2 ($Y = 0$ with probability $1 - p_2$)
. . . . . .
. . . . . .
else if $t_{m-1} = 1$ for $\mathbf{X}$ then $Y = 1$ with probability p_{m-1} ($Y = 0$ with probability $1 - p_{m-1}$)
else $Y = 1$ with probability p_m for all $\mathbf{X}$ ($Y = 0$ with probability $1 - p_m$).

We call M a *decision form* and k a *degree*. For a given decision form $M = \langle t_1 \ldots t_m \rangle$, we call m the *depth* of M and t_i the *i-th decision*. k specifies the level of fineness of the partitioning. M specifies the partitioning of $\mathcal{X}$. To define a stochastic decision list, we must first fix k, and then determine M and having fixed k and M, determine θ. Thus, θ, M, and k form a hierarchical structure. We may write this structure as:

$$\theta \prec M \prec k. \tag{3}$$

In this paper, for a fixed k, $P(Y \mid \mathbf{X} : \theta \prec M)$ denotes a stochastic decision list specified by θ and M.

Definition 2 (Class of stochastic decision lists). Let C_{FP} be the set of all stochastic rules with finite partitioning on $\{0, 1\}^n \times \{0, 1\}$. For a fixed k, a *class of all stochastic decision lists with degree at most k*, which we denote as C_{DL}^k, is defined as: For a given n, $C_{DL}^k = \{P(Y \mid \mathbf{X} : \theta \prec M) \in C_{FP} : M \in \Gamma_k^n, \theta \in \Theta(M)\}$, where $\Theta(M) = [0, 1]^m$ is a set of probability parameter vectors associated with M, and $m = \dim\Theta(M)$. When we emphasize the number n of attributes, we will indicate this in parentheses, as in $C_{DL}^k(n)$. □

The definition of stochastic decision lists is easily extended to the case where $\mathcal{Y} = \{1, \ldots, s\}$. Each $\theta \in \Theta(M)$ is written as: $\theta = (p_1(1), \ldots, p_1(s-1), \ldots, p_m(1), \ldots, p_m(s-1))$ where $p_i(j)$ denotes the probability with which $Y = j$ for $\mathbf{X}$ such that $t_1(\mathbf{X}) = \ldots = t_{i-1}(\mathbf{X}) = 0$ and $t_i(\mathbf{X}) = 1$, where $\Theta(M) \subset [0, 1]^{m(s-1)}$.

Let us take one more example of classes of stochastic rules with finite partitioning, called *stochastic decision trees*. Let $\mathcal{X} = \{0, 1\}^n$ and $\mathcal{Y} = \{1, \ldots, s\}$. Let $S_n = \{x_1, \ldots, x_n\}$ where x_i is a variable which takes a values in $\{0, 1\}$ $(i = 1, \ldots, m)$. A *stochastic decision tree* is a binary tree with probabilistic class-assignment where each internal node is labeled with a variable in S_n, and at the i-th leaf (that we assume all leaves are properly ordered) the class j is assigned with probability $p_i(j)$ $(i = 1, \ldots, m, j = 1, \ldots, s)$. m is the number of leaves. The *depth* of a stochastic decision tree is the length of the longest path from the root to a leaf. The *tree form* of a stochastic decision tree is the binary tree where the class-assignment is ignored. That is, the structure of a stochastic decision tree is decomposed into its tree form and its probabilities for class-assignment.

Each stochastic decision tree defines a conditional probability distribution over $\mathcal{Y}$ for a given attribute vector $\mathbf{X}$ as follows: At each internal node the left edge to a child is taken if the variable at the internal node is 0, otherwise the right edge is taken. This assignment determines a unique path from a root to a leaf when an attribute vector $\mathbf{X}$ is given. The probability with which each class is assigned to $\mathbf{X}$ is the probability given at the leaf reached. That is, stochastic decision trees are regarded as a stochastic analogue of decision trees (see, for example, Rivest (1987, p. 233–234) and Quinlan and Rivest (1989)).

For a fixed positive integer k, let Ω_k^n be the set of binary tree forms with depth at most k. Let $P(Y \mid \mathbf{X} : \theta \prec M)$ denote a stochastic decision tree specified by a tree form $M \in \Omega_k^n$ and a probability parameter vector $\theta = (p_1(1), \ldots, p_1(s-1), \ldots, p_m(1), \ldots, p_m(s-1))$. m is the number of leaves of the stochastic decision tree. Notice that $p_i(s)$ is determined by $1 - \Sigma_{j=1}^{s-1} p_i(j)$ $(i = 1, \ldots, m)$.

Definition 3 (Class of stochastic decision trees). Let $\mathcal{C}_{FP}$ be a set of all stochastic rules with finite partitioning defined on $\{0, 1\}^n \times \{1, \ldots, s\}$ $(s < \infty)$. For a fixed k, a *class of stochastic decision trees with depth at most k and s classes*, which we denote as $\mathcal{C}^k_{DT(s)}$, is defined as: For a given n, $\mathcal{C}^k_{DT(s)} \stackrel{\text{def}}{=} \{P(Y \mid \mathbf{X} : \theta \prec M) \in \mathcal{C}_{FP} : M \in \Omega^n_k, \theta \in \Theta(M)\}$, where $\Theta(M)(\subset [0, 1]^{m(s-1)})$ is a set of probability parameter vectors associated wtih M and $\dim\Theta(M) = m(s - 1)$. Specifically, we denote $\mathcal{C}^k_{DT(2)}$ as $\mathcal{C}^k_{DT}$. When we emphasize the number n of attributes, we will indicate this in parentheses as in $\mathcal{C}^k_{DT(s)}(n)$. □

3. A learning criterion for stochastic rules

In this section, a learning criterion for stochastic rules is proposed.

Before giving a general definition, let us review Valiant's PAC model to clarify the significance of our new criterion. In the PAC model, an attribute vector $\mathbf{X}$ is assumed to be generated according to some probability distribution $Q(\mathbf{X})$ over the domain $\mathcal{X}$. PAC learning refers to probably approximately correct identifying an unknown target function which is known to belong to some specific class. The following definition of PAC learnability appears widely in the literature of computational learning theory. The following definition essentially follows the definiton in Valiant (1984).

Definition 4 (PAC learning criterion for deterministic rules). Let $\mathcal{F}$ be a class of functions from $\mathcal{X}$ to $\mathcal{Y}$. $\mathcal{F}$ is called *polynomially learnable* if there exists an algorithm $\mathcal{A}$ such that, for some polynomial $N_0(.,.,.)$, for all n, for all $0 < \epsilon < 1$, for all $0 < \delta < 1$, for all $N \geq N_0(1/\epsilon, 1/\delta, n)$, for all $Q(\mathbf{X})$ on $\mathcal{X}$, for all $f^*(\mathbf{X}) \in \mathcal{F}$, when given N independently drawn examples $D^N = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ $(y_i = f^*(\mathbf{x}_i), i = 1, \ldots, N$, each $\mathbf{x}_i$ is generated according to $Q(\mathbf{X}))$ as input, $\mathcal{A}$ outputs $\hat{f}_{[D^N]}(\mathbf{X}) \in \mathcal{F}$ satisfying (4)

$$Prob[Q[f^*(\mathbf{X}) \neq \hat{f}_{[D_N]}(\mathbf{X})] \geq \epsilon] \leq \delta, \tag{4}$$

and $\mathcal{A}$ runs in time $T_0(1/\epsilon, 1/\delta, n)$ which is polynomial in $1/\epsilon$, $1/\delta$, and n. Here *Prob* is a probability taken with respect to N-product distribution $Q(\mathbf{X}_1 \ldots \mathbf{X}_N) = \Pi^N_{i=1} Q(\mathbf{X}_i)$ (which measures the occurrence probability of D^N) on $\mathcal{X}^N$ and over any coin tosses $\mathcal{A}$ may make. n is the number of attributes. $Q[f^*(\mathbf{X}) \neq \hat{f}_{[D^N]}(\mathbf{X})]$ denotes the probability that $\mathbf{X}$ such that $f^*(\mathbf{X}) \neq \hat{f}_{[D^N]}(\mathbf{X})$ occurs with respect to the distribution $Q(\mathbf{X})$. □

In the above definition, $\hat{f}_{[D^N]}$ is called a *hypothesis* (of f^*) made by the algorithm $\mathcal{A}$.

We extend Valiant's learnability criterion to the stochastic setting, on the basis of the following three viewpoints:

1) The target objects to be learned are classes of stochastic rules rather than those of deterministic rules.
2) Learning confidence (i.e., the probability with which a hypothesis lies within ϵ of the target rule) must be evaluated with respect to the product probability distribution over $\mathcal{X} \times \mathcal{Y}$.
3) An error of a hypothesis must be measured in terms of some variations of statistical deviation between two probability distributions: the hypothesis and the target rule.

Taking these three points into consideration, we are led to a new learning model for stochastic rules. The goal of this model is probably approximately correct identification of the target stochastic rule belonging to some known specific class.

Definition 5 (PAC learning criterion for stochastic rules). Let C be a class of stochastic rules. C is called *statistically learnable (with repsect to the distance measure d) by an algorithm* $\mathcal{A}$ if, for some polynomial $N_0(.,.,.)$, for all n, for all $\epsilon > 0$, for all $0 < \delta < 1$, for all $N \geq N_0(1/\epsilon, 1/\delta, n)$, for all $Q(\mathbf{X})$ on $\mathcal{X}$, for all $P^*(Y \mid \mathbf{X}) \in C$, $\mathcal{A}$ takes as input N examples $D^N = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ drawn independently according to $Q(\mathbf{X})P^*(Y \mid \mathbf{X})$ and outputs $\hat{P}_{[D^N]}(Y \mid \mathbf{X}) \in C$ satisfying

$$Prob[d(P^*, \hat{P}_{[D^N]}) \geq \epsilon] \leq \delta. \tag{5}$$

Here *Prob* is the probability taken with respect to N-product probability distribution (which measures the occurrence probability of D^N); $Q(\mathbf{X}_1 \ldots \mathbf{X}_N)P^*(Y_1 \ldots Y_N \mid \mathbf{X}_1 \ldots \mathbf{X}) = \Pi_{i=1}^N Q(\mathbf{X}_i)P^*(Y_i \mid \mathbf{X}_i)$ (which we denote, for short, as $(QP^*)(D^N)$) on $(\mathcal{X} \times \mathcal{Y})^N$ and over any coin tosses $\mathcal{A}$ makes. $P_{[D^N]}(Y|\mathbf{X})$ is called a *hypothesis* of $\mathcal{A}$. n is the number of attributes. Here ϵ is called an *accuracy parameter*, and δ is called a *confidence parameter*. $d(P^*, \hat{P}_{[D^N]})$ is one of the following deviations: d_H, d_V, d_{KL}, and d_Q:

$$d_H(P^*, \hat{P}_{[D^N]}) = \sum_{\mathbf{X} \in \mathcal{X}} Q(\mathbf{X}) \sum_{Y \in \mathcal{Y}} | \sqrt{P^*(Y \mid \mathbf{X})} - \sqrt{\hat{P}_{[D^N]}(Y \mid \mathbf{X})} |^2, \tag{6}$$

$$d_V(P^*, \hat{P}_{[D^N]}) = \sum_{\mathbf{X} \in \mathcal{X}} Q(\mathbf{X}) \sum_{Y \in \mathcal{Y}} | P^*(Y \mid \mathbf{X}) - \hat{P}_{[D^N]}(Y \mid \mathbf{X}) |, \tag{7}$$

$$d_{KL}(P^*, \hat{P}_{[D^N]}) = \sum_{\mathbf{X} \in \mathcal{X}} Q(\mathbf{X}) \sum_{Y \in \mathcal{Y}} P^*(Y \mid \mathbf{X}) \log_2 \frac{P^*(Y \mid \mathbf{X})}{\hat{P}_{[D^N]}(Y \mid \mathbf{X})}, \tag{8}$$

$$d_Q(P^*, \hat{P}_{[D^N]}) = \sum_{\mathbf{X} \in \mathcal{X}} Q(\mathbf{X}) \sum_{Y \in \mathcal{Y}} | P^*(Y \mid \mathbf{X}) - \hat{P}_{[D^N]}(Y \mid \mathbf{X}) |^2. \tag{9}$$

C is said to be *statistically learnable* if there exists an algorithm $\mathcal{A}$ such that C is statistically learnable by $\mathcal{A}$.

C is said to be *polynomially learnable* (with respect to the distance measure d), if there exists an algorithm $\mathcal{A}$ such that C is statistically learnable (with respect to the distance measure d) by $\mathcal{A}$, and, for all n, for all $\epsilon > 0$, for all $0 < \delta < 1$, for all $Q(\mathbf{X})$ on $\mathcal{X}$, for all $P^*(Y \mid \mathbf{X}) \in C$, $\mathcal{A}$ runs in time $T_0(1/\epsilon, 1/\delta, n)$ which is polynomial in $1/\epsilon$, $1/\delta$, and n. □

For fixed ϵ and δ, we call (5) an (ϵ, δ)-*criterion*. We may weaken the definition of statistical or polynomial learnability by allowing sample size and computation time to depend on the target rule $P^*(Y \mid \mathbf{X})$, the distribution $Q(\mathbf{X})$ or both. For fixed ϵ and δ, define the *sample complexity* of learning a class C of stochastic rules by the minimum samples size satisfying the (ϵ, δ)-criterion over all learning algorithms.

The deviations d_H, d_V, d_{KL}, and d_Q are known as, respectively, the *Hellinger distance*, the *variation distance*, the *Kullback-Leibler divergence*, and the *quadratic distance*. All of them are statistical notions of 'distances' between two probability distributions, which have often been used in the literature of statistical inference and information theory. Only d_V is a metric. d_H and d_Q are symmetric but do not satisfy the triangle inequality. d_{KL} is asymmetric and does not satisfy the triangle inequality. d_H, d_V, and d_Q are bounded but d_{KL} is not bounded. Here it should be noted that, as will be seen in Section 5, the sample size with which an algorithm satisfies the (ϵ, δ)-criterion depends on the choice of the deviation from among d_H, d_V, d_{KL}, and d_Q. In this paper, sample size bounds are derived with respect to d_H and d_V only.

Lemma 1. For any two conditional probability distributions; P_1 and P_2, the following inequalities hold.

$$(d_V(P_1, P_2))^2/4 \leq d_H(P_1, P_2) \leq d_{KL}(P_1, P_2), \tag{10}$$

$$(d_V(P_1, P_2))^2/(2 \ln 2) \leq d_{KL}(P_1, P_2), \tag{11}$$

$$d_H(P_1, P_2) \leq d_V(P_1, P_2), \tag{12}$$

$$d_Q(P_1, P_2) \leq d_V(P_1, P_2). \tag{13}$$

□

The first inequality in (10) follows Pitman (1979, p. 7). The second inequality in (10) follow Barron and Cover (1991, p. 1047). (11) follows Kullback (1967). (12) follows Kraft (1955). (13) is trivial.

Lemma 1 shows that d_H and d_V are polynomially related to each other, and both are polynomially bounded by d_{KL}.

As mentioned in Introduction, a learning criterion for stochastic rules have also been developed in Kearns and Schapire (1990) for the special case $Y = \{0, 1\}$ independently of this work. The learning criterion proposed in this paper follows the definition in Yamanishi (1990b). Also related are learning criteria for "density estimation" which have first been developed by Laird (1988), followed by Abe and Warmuth (1990) and Cesa-Bianchi (1990) independently. Here the density estimation refers to the problem of learning distributions on the set $\mathcal{X}$ itself. Various distances have been used as error measures for hypotheses in the context of density estimation; the Kullback-Leibler divergence (Laird, 1988; Abe & Warmuth, 1990), the variation distance (Laird, 1988), and the difference of expected variation distances (Cesa-Bianchi, 1990). Kearns and Schapire used the expected quadratic distance, which can be reduced to the difference of the expected quadratic loss, as an error measure of hypotheses over the binary range.

4. Learning strategy based on the MDL principle

In this section, an algorithm for learning stochastic rules with finite partitioning is presented. The rule selection strategy in this algorithm is based on the Minimum Description Length

(MDL) principle (Rissanen, 1978; 1983, 1984; 1986; 1989; Wallace & Boulton, 1968; Solomonoff, 1964). Hereafter, "log" denotes the logarithm of base 2, and "ln" denotes the natural logarithm.

4.1. Learning stochastic rules based on the MDL principle

Learning stochastic rules can be thought of as being basically estimation of the target rule from the given examples. The MDL principle is employed as a criterion for selecting the best hypothesis from among possible ones.

Let N examples $D^N = D_1 \ldots D_N$ be independently drawn according to an unknown probability distribution. Let $\mathcal{H}$ be a class of hypotheses. The MDL principle asserts that, when given D^N, the hypothesis h that one should select from $\mathcal{H}$ is the one that minimizes the following sum:

$$\ell(h) + \ell(D^N \mid h). \tag{14}$$

Here $\ell(h)$ is the description length for h and effectively measures the complexity of h. $\ell(D^N \mid h)$ is the description length for D^N with respect to h and measures the goodness of fit of h to D^N, where a shorter description length indicates a better fit. In other words, h is selected taking into account the trade-off between its simplicity and its goodness of fit to given examples. Here the "description length" is the code-length in bits needed for the encoding of h or D^N under the condition that no codeword is a prefix of another codeword. This condition, which we call the *prefix condition*, is sufficient for the requirement that the code string be uniquely decodable; i.e., every codeword can be exactly decodable even if commas are not allowed. It is further known (Gallager, 1986, pp. 45–49, 514) that there exists a code satisfying the prefix condition with codeword length $\ell(h)$ if and only if the length function ℓ: $\mathcal{H} \to \mathbf{R}^+ \cup \{0\}$ ($\mathbf{R}^+$ denotes the set of all positive real numbers) satisfies *Kraft's inequality* (Kraft 1949):

$$\sum_{h \in \mathcal{H}} 2^{-\ell(h)} \leq 1, \tag{15}$$

where $\ell(h)$ is the code-length for h. Hereafter, non-integer code lengths are allowed.

Let $D^N = D_1, \ldots, D_N$ $D_i = (\mathbf{x}_i, y_i)$ $(i = 1, \ldots, N)$ be examples drawn independently according to a product distribution on $\mathcal{X} \times \mathcal{Y}$. In general, when learning stochastic rules using the MDL principle, we select a hypothesis that minimizes the total description length for the hypothesis space itself plus the description length for $y^N = y_1 \ldots y_N$ for given $\mathbf{x}^N = \mathbf{x}_1 \ldots \mathbf{x}_N$. Here notice that we need not take the description length for $\mathbf{x}^N$ into consideration because the hypothesis to be outputted is a conditional distribution of Y for given $\mathbf{X}$, which is independent of the distribution over $\mathcal{X}$.

Let each hypothesis be specified by a real-valued probability parameter vector θ and a countable model M. $\ell(y^N \mid \mathbf{x}^N : \theta \prec M)$ denotes the description length for y^N for given $\mathbf{x}^N$ relative to the distribution $P(Y \mid \mathbf{X} : \theta \prec M)$ specified by fixed θ and M. Let $\mathcal{M}$ be a set of all countable models and let $\Theta_N(M)$ be a set of real-valued parameter vectors

associated with a fixed M, such that $\Theta_N(M)$ is a finite set whose size depends on sample size N. We describe y^N in $\ell(y^N \mid \mathbf{x}^N : \theta \prec M)$ bits, letting θ and M be fixed. $\ell(\theta \mid M)$ is the description length for θ for a fixed M and satisfies Kraft's inequality: $\Sigma_{\theta \in \Theta_N(M)} 2^{-\ell(\theta \mid M)} \leq 1$. $\ell(M)$ denotes the description length for M and satisfies Kraft's inequality: $\Sigma_{M \in \mathcal{M}} 2^{-\ell(M)} \leq 1$. The total description length to be minimized with respect to θ and M, which we denote as $\ell_\lambda(y^N : \theta \prec M \mid \mathbf{x}^N)$, is calculated as follows.

$$\ell_\lambda(y^N : \theta \prec M \mid \mathbf{x}^N) = \ell(y^N \mid \mathbf{x}^N : \theta \prec M) + \lambda\{\ell(\theta \mid M) + \ell(M)\}, \tag{16}$$

where $\ell(\theta \mid M) + \ell(M)$ denotes the description length for the hypothesis, and λ is an adjustment parameter which is not less than 1. Note that if $\ell(\theta \mid M) + \ell(M)$ satisfies Kraft's inequality: $\Sigma_{\theta \in \Theta_N(M)} \Sigma_{M \in \mathcal{M}} 2^{-\{\ell(\theta \mid M) + \ell(M)\}} \leq 1$, then $\Sigma_{\theta \in \Theta_N(M)} \Sigma_{M \in \mathcal{M}} 2^{-\lambda\{\ell(\theta \mid M) + \ell(M)\}} \leq 1$ also holds for $\lambda \geq 1$.

4.2. MDL algorithm for learning stochastic decision lists

Details of the methods for learning stochastic decision lists based on the MDL principle are described in the discussion to follow. Hereafter the degree k is assumed to be fixed.

Let N independent random examles $D^N = D_1 \ldots D_N$, $D_i = (\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ $(i = 1, \ldots, N)$ be given. Let $S = \{D_1, \ldots, D_N\}$ be a multiset of examples. Let $\mathbf{x}^N \stackrel{\text{def}}{=} \mathbf{x}_1 \ldots \mathbf{x}_N$, $y^N \stackrel{\text{def}}{=} y_1 \ldots y_N$. N_j denotes the number of examples in S in such that $t_1(\mathbf{x}) = \ldots = t_{j-1}(\mathbf{x}) = 0$, and $t_j(\mathbf{x}) = 1$. N_j^+ denotes the number of examples in S such that $t_1(\mathbf{x}) = \ldots = t_{j-1}(\mathbf{x}) = 0$, $t_j(\mathbf{x}) = 1$ and $y = 1$. N_j^- denotes the number of examples in S such that $t_1(\mathbf{x}) = \ldots = t_{j-1}(\mathbf{x}) = 0$, $t_j(\mathbf{x}) = 1$ and $y = 0$. Notice here that $N_1 + \ldots + N_m = N$, $N_j^+ + N_j^- = N_j$, $(j = 1, \ldots, m)$.

For a fixed k, let $P(Y \mid \mathbf{X} : \theta \prec M) \in C_{DL}^k(n)$ be the conditional probability distribution defined by a stochastic decision list specified by θ and M such that $M \in \Gamma_k^n$. For given examples D^N, the *conditional likelihood* (hereafter, for short, called *likelihood*) of $P(Y \mid \mathbf{X} : \theta \prec M)$ for y^N when given $\mathbf{x}^N$ is defined as

$$P(y^N \mid \mathbf{x}^N : \theta \prec M) = \prod_{i=1}^{N} P(y_i \mid \mathbf{x}_i : \theta \prec M)$$

$$= \prod_{j=1}^{m} p_j^{N_j^+}(1 - p_j)^{N_j^-},$$

where m is the depth of M. Notice here that the description length for y^N for given $\mathbf{x}^N$ relative to θ and M; $\ell(y^N \mid \mathbf{x}^N : \theta \prec M)$, is calculated as $\ell(y^N \mid \mathbf{x}^N : \theta \prec M) = -\log P(y^N \mid \mathbf{x}^N : \theta \prec M)$, since $\ell(y^N \mid \mathbf{x}^N : \theta \prec M)$ satisfies Kraft's inequality: $\Sigma_{y^N} 2^{-\ell(y^N : \theta \prec M)} = \Sigma_{y^N} P(y^N \mid \mathbf{x}^N : \theta \prec M) = 1$. The code-length defined by the minus logarithm of the occurrence probability is called *Shannon complexity* (see e.g., Rissanen (1989, p. 38)). Thus, the description length for y^N for given $\mathbf{x}^N$ relative to θ and M; $\ell(y^N \mid \mathbf{x}^N : \theta \prec M)$, is calculated as follows.

$$\ell(y^N \mid \mathbf{x}^N : \theta \prec M) \stackrel{\text{def}}{=} -\log P(y^N \mid \mathbf{x}^N : \theta \prec M)$$

$$= -\log \prod_{j=1}^{m} p_j^{N_j^+}(1 - p_j)^{N_j^-}$$

$$= \sum_{j=1}^{m} N_j\{H(\tilde{p}_j) + D(\tilde{\mathbf{p}}_j \parallel \mathbf{p}_j)\}, \tag{17}$$

where $H(x) \stackrel{\text{def}}{=} -x \log x - (1 - x) \log (1 - x)$ is the entropy function, and $\tilde{p}_j = N_j^+/N_j$. $D(\tilde{\mathbf{p}}_j \parallel \mathbf{p}_j) = \tilde{p}_j \log (\tilde{p}_j/p_j) + (1 - \tilde{p}_j) \log ((1 - \tilde{p}_j)/(1 - p_j))$ is the Kullback-Leibler divergence between $\tilde{\mathbf{p}}_j = (\tilde{p}_j, 1 - \tilde{p}_j)$ and $\mathbf{p}_j = (p_j, 1 - p_j)$ $(j = 1, \ldots, m)$.

Since $\theta = (p_1, \ldots, p_m)$ is unknown, we must estimate θ from D^N. Let $\hat{\theta} = (\hat{p}_1, \ldots, \hat{p}_m)$ be a maximum likelihood estimator of θ. Here the *maximum likelihood estimator* is the estimator whose value maximizes the likelihood for the given examples, or equivalently minimizes the minus logarithm of the likelihood for the given examples. Notice here that $D(\tilde{\mathbf{p}}_j \parallel \mathbf{p}_j) \geq 0$ and that $D(\tilde{\mathbf{p}}_j \parallel \mathbf{p}_j) = 0$ if and only if $\tilde{p}_j = p_j$. Thus the maximum likelihood estimator $\hat{\theta} = (\hat{p}_1, \ldots, \hat{p}_m)$ is given by $\hat{p}_j = \tilde{p}_j = N_j^+/N_j$ $(j = 1, \ldots, m)$.[1]

We must describe $\hat{\theta}$ with finite precision in order to apply the MDL principle to hypothesis selection. Letting $\bar{\theta} \stackrel{\text{def}}{=} \hat{\theta} + \delta$ be a truncated vector for $\hat{\theta}$ and $\delta = (\delta_1, \ldots, \delta_m)$ be an m dimensional precision, let us derive the optimal size of precision by considering the minimization of the description length for y^N for given $\mathbf{x}^N$ plus the description length for the precision δ; $-\log P(y^N \mid \mathbf{x}^N : \bar{\theta} \prec M) - \Sigma_{j=1}^m \log \delta_j$, where $-\Sigma_{j=1}^m \log \delta_j$ is the description length for the precision δ (Rissanen, 1989, p. 55). When $0 \leq \hat{p}_j \leq 1/2$ $(j = 1, \ldots, m)$, assume that $0 < \delta_j < 1/2$ $(j = 1, \ldots, m)$. Then, from (17), we have

$$-\log P(y^N \mid \mathbf{x}^N : \hat{\theta} + \delta \prec M)$$

$$= \sum_{i=1}^{m} N_j H(\hat{p}_j) + \sum_{j=1}^{m} N_j \left\{ \hat{p}_j \log \frac{\hat{p}_j}{\hat{p}_j + \delta_j} - (1 - \hat{p}_j) \log \frac{1 - \hat{p}_j}{1 - \hat{p}_j - \delta_j} \right\}$$

$$= -\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M)$$

$$+ \sum_{j=1}^{m} N_j \left\{ -\hat{p}_j \log \left(1 + \frac{\delta_j}{\hat{p}_j}\right) - (1 - \hat{p}_j) \log \left(1 - \frac{\delta_j}{1 - \hat{p}_j}\right) \right\}$$

$$< -\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M)$$

$$+ \sum_{j=1}^{m} N_j \left\{ -\hat{p}_j \left(\frac{\delta_j}{\hat{p}_j} - \left(\frac{\delta_j}{\hat{p}_j} \right)^2 \right) + (1 - \hat{p}_j) \left(\frac{\delta_j}{1 - \hat{p}_j} + \left(\frac{\delta_j}{1 - \hat{p}_j} \right)^2 \right) \right\} \frac{1}{\ln 2}$$

$$= -\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M) + \sum_{j=1}^{m} N_j \left(\frac{\delta_j^2}{\hat{p}_j(1 - \hat{p}_j) \ln 2} \right).$$

where we have used general inequalities: $\log(1 + x) \geq (x - x^2)/\ln 2$ $(x \geq 0)$ and $\log(1 - x) \geq (-x - x^2)/(\ln 2)$ $(0 \leq x < 1/2)$. Let $\hat{\delta} = (\hat{\delta}_1, \ldots, \hat{\delta}_m)$ be the value of δ minimizing $-\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M) + \Sigma_{j=1}^m N_j(\delta^2/(\hat{p}_j(1 - \hat{p}_j) \ln 2)) - \Sigma_{j=1}^m \log \delta_j$, then we obtain that $\hat{\delta}_j = \sqrt{\hat{p}_j(1 - \hat{p}_j)/2N_j}$. Notice here that the following inequality holds.

$$-\log P(y^N \mid \mathbf{x}^N : \hat{\theta} + \hat{\delta} \prec M) < -\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M) + m. \tag{18}$$

When for some j, $1/2 \leq \hat{p}_j \leq 1$, we assume that $\delta_j < 0$. As with the case where $0 \leq \hat{p}_j \leq 1/2$, we can easily obtain an optimal precision as follows: $\hat{\delta}_j = -\sqrt{\hat{p}_j(1 - \hat{p}_j)2N_j}$.

To realize this optimal precision, we quantize $\Theta(M)$ to obtain $\Theta_N(M)$, which is a rectangular grid of truncated vectors, as follows: For each direction of $[0, 1]^m$, the set of truncated vectors consists of two sets: $\{\bar{p}_k\}$ $(1/2N\sqrt{N} \leq \bar{p}_k \leq 1/2)$ and $\{\bar{p}_l'\}$ $(1/2 < \bar{p}_l' \leq 1 - 1/(2N\sqrt{N}))$, which are defined by

$$\bar{p}_1 = \frac{1}{2N\sqrt{N}},$$

$$\bar{p}_k = \bar{p}_{k-1} + \sqrt{\frac{\bar{p}_{k-1}(1 - \bar{p}_{k-1})}{2N}} \quad (k = 2, 3, \ldots s_1),$$

$$\bar{p}_1' = 1 - \frac{1}{2N\sqrt{N}},$$

$$\bar{p}_l' = \bar{p}_{l-1}' - \sqrt{\frac{\bar{p}_{l-1}'(1 - \bar{p}_{l-1}')}{2N}} \quad (l = 2, 3, \ldots s_2), \tag{19}$$

where s_1 satisfies $\bar{p}_{s_1} \leq 1/2$, $\bar{p}_{s_1+1} > 1/2$ and s_2 satisfies $\bar{p}_{s_2}' > 1/2$, $\bar{p}_{s_2+1}' \leq 1/2$. Notice here that while the optimal precision is proportional to $1/\sqrt{N_j}$, which depends on examples, the precision for the above quantization method is proportional to $1/\sqrt{N}$, which depends only on sample size. This precision is accurate enough because $1/N_j > 1/N$. The method for truncation of each component of a maximum likelihood estimator is as follows: If $0 \leq \hat{p} \leq 1/2$, then we let the truncated value for $\hat{p}$ be $\bar{p}_k$ such that $\bar{p}_{k-1} < \hat{p} \leq \bar{p}_k$. If $1/2 < \hat{p} \leq 1$, then we let the truncated value for $\hat{p}$ be $\bar{p}_l$ such that $\bar{p}_l \leq \hat{p} < \bar{p}_{l-1}$. Hereafter, we fix the method for quantization (19), by which we define $\Theta_N(M)$.

For this method for quantization of $\Theta(M)$, we see that the precision is not larger than $\hat{\delta}$ and thus $\delta^2/(\hat{p}_j(1 - \hat{p}_j) \ln 2) \leq m$. Thus, from (18), it is verified that if $\bar{\theta} \in \Theta_N(M)$ is a truncated vector for the maximum likelihood estimator $\hat{\theta} \in \Theta(M)$, the following inequality holds.

$$-\log P(y^N \mid \mathbf{x}^N : \bar{\theta} \prec M) < -\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M) + m. \tag{20}$$

Further notice that for a fixed M, $\arg \min_{\theta \in \Theta_N(M)} \{-\log P(y^N \mid \mathbf{x}^N : \theta \prec M)\}$ is calculated as a truncated vector (in $\Theta_N(M)$) of the maximum likelihood estimator $\hat{\theta}$.

Let $\ell(\theta \mid M)$ be the description length for any point $\theta \in \Theta_N(M)$ for fixed M and k. The total number of elements in $\Theta_N(M)$ for a fixed M is bounded as

$$\left(2 \times \int_{1/2N\sqrt{N}}^{1/2} \sqrt{\frac{2N}{x(1-x)}}\, dx\right)^m < (4\sqrt{2N})^m,$$

where $m = \dim \Theta(M)$. It follows that $\log(4\sqrt{2N})^m$ bits are sufficient for encoding any $\theta \in \Theta_N(M)$. Thus, $\ell(\theta \mid M)$ is given by

$$\ell(\theta \mid M) = \frac{m \log N}{2} + \frac{5m}{2}. \tag{21}$$

Notice here that $\Sigma_{\theta \in \Theta_N(M)} 2^{-\ell(\theta|M)} \leq 1$.

Let $\ell(M)$ be the description length for M. The description for the i-th decision t_i requires $\log(|T_k^n| - i + 1)$ bits because $t_i \in T_k^n - \{t_1, \ldots, t_{i-1}\}$. Thus, $\ell(M)$ is calculated as follows.

$$\ell(M) = \log \#T_k^n[m], \tag{22}$$

where $\#T_k^n[m] \stackrel{\text{def}}{=} |T_k^n| \, (|T_k^n| - 1) \ldots (|T_k^n| - m + 1)$.

$$|T_k^n| = \sum_{i=1}^{k} 2^i \binom{n}{i} + 1$$

is the total number of elements in T_k^n. Notice that the m-th decision is always the constant function "1." Thus the number of all possible ordered sets in the form of $\langle t_1, \ldots, t_m \rangle$ is $(|T_k^n| - 1) \ldots (|T_k^n| - m + 1) = \#T_k^n[m]/|T_k^n|$. Hence, $\Sigma_{M \in \Gamma_k^n} 2^{-\ell(M)} = \Sigma_{m=1}^{|T_k^n|} (\#T_k^n[m]/|T_k^n|) \, 2^{-\log \#T_k^n[m]} = 1$. Therefore the code-length function (22) satisfies Kraft's inequality. Of course, any other code-length functions can also be used provided they satisfy Kraft's inequality.

If an algorithm $\mathcal{A}$ always outputs a hypothesis h belonging to $\mathcal{H}$, we say that $\mathcal{A}$ uses a *hypothesis space* $\mathcal{H}$. Notice that a hypothesis space depends on the number of examples because, for each M, θ is selected from $\Theta_N(M)$ with finite precision of $O(1/\sqrt{N})$. For $C_{DL}^k(n)$, we define $\mathcal{H}_N(C_{DL}^k(n))$, which is a subset of $C_{DL}^k(n)$, as

$$\mathcal{H}_N(C_{DL}^k(n)) \stackrel{\text{def}}{=} \{P(Y \mid \mathbf{X} : \theta \prec M) \in C_{DL}^k(n) : M \in \Gamma_k^n, \, \theta \in \Theta_N(M)\}, \tag{23}$$

where $\Theta_N(M)$ is a rectangular grid of probability parameter vectors with cell of width $\delta = O(1/\sqrt{N})$ in each direction of the m-dimensional space $[0, 1]^m$ where m is the depth of M.

An *MDL algorithm (with the adjustment parameter* λ *and the code-length function* ℓ*)* for learning $C_{DL}^k(n)$ is an algorithm which takes independent random examples D^N as input and outputs a rule $\hat{P}$ in $\mathcal{H}_N(C_{DL}^k(n))$ that attains the total minimum description length over $\mathcal{H}_N(C_{DL}^k(n))$ by letting the adjustment parameter be λ and by using the code-length function $\ell : \Gamma_k^n \to \mathbf{R}^+ \cup \{0\}$ (such that $\Sigma_{M \in \Gamma_k^n} 2^{-\ell(M)} \leq 1$) in (16).

Below, we more precisely describe the MDL algorithm for learning $C_{DL}^k(n)$, which we denote as $\mathcal{A}_{MDL}$. $\mathcal{A}_{MDL}$ uses a hypothesis space $\mathcal{H}_N(C_{DL}^k(n))$, which depends on sample size N. Fix an adjustment parameter λ and a code-length function $\ell : \Gamma_k^n \to \mathbf{R}^+ \cup \{0\}$ such that $\Sigma_{M \in \Gamma_k^n} 2^{-\ell(M)} \leq 1$.

1° $\mathcal{A}_{MDL}$ draws N independent examples D^N.
2° For each $M \in \Gamma_k^n$, $\mathcal{A}_{MDL}$ calculates $\bar{\theta} = \arg\min_{\theta \in \Theta_N(M)} \ell(y^N \mid \mathbf{x}^N : \theta \prec M)$ and $\ell_\lambda(y^N : \bar{\theta} \prec M \mid \mathbf{x}^N) \stackrel{\text{def}}{=} \ell(y^N \mid \mathbf{x}^N : \bar{\theta} \prec M) + \lambda(\ell(\bar{\theta} \mid M) + \ell(M))$.
3° $\mathcal{A}_{MDL}$ chooses one countable model $\hat{M} \in \Gamma_k^n$ which attains the minimum of $\ell_\lambda(y^N : \bar{\theta} \prec M \mid \mathbf{x}^N)$. If there exist more than one M that attain the minimum, $\mathcal{A}_{MDL}$ choose $\hat{M}$ from among them such that $\ell(\bar{\theta} \mid \hat{M}) + \ell(\hat{M})$ is shortest.
4° $\mathcal{A}_{MDL}$ outputs $P(Y \mid \mathbf{X} : \bar{\theta} \prec \hat{M}) \in \mathcal{H}_N(C_{DL}^k(n))$.

We call an output of $\mathcal{A}_{MDL}$ an *MDL rule* and the optimal countable model $\hat{M}$ an *MDL estimator.*

Maximizing the likelihood for given examples is equivalent to minimizing the description length for the examples relative to a rule, ignoring the description length for the rule itself. An *ML algorithm* is an algorithm which takes independent random examples as input and outputs a rule which maximizes the likelihood (we refer to such a rule as an *ML rule*). We refer to a countable model that the ML algorithm chooses as an *ML estimator.*

The ML algorithm is an analogue to the "minimum disagreement algorithms" (Kearns & Li, 1988; Sloan, 1988; Angluin & Laird, 1988, etc.) used in the conventional PAC model in the sense that both algorithms output a rule which best fits the given examples. The "Occam algorithm" (Blumer, et al, 1987) proven to perform well in the conventional PAC model, can be thought of as an algorithm outputting a rule minimizing the description length for the rule, having minimized the description length for given examples. This kind of minimization of description lengths (for given examples and a rule) would result in outputting nothing but a ML rule in our stochastic setting.

Notice that both of the MDL and the ML algorithms are not efficient because, in computation of an MDL estimator (an ML estimator) $\hat{M}$, the number of countable models for which the total description lengths (likelihoods) should be compared is $|\Gamma_k^n| = O(3^{|T_k^n|} |T_k^n|!)$ and $|T_k^n| = O(n^k)$ (see Rivest (1987, p. 243)), which is exponential in n. However, in this study we will confine our investigation to the question of sample complexity alone and ignore time complexity in the subsequent sections.

4.3. MDL algorithm for learning general stochastic rules with finite partitioning

Letting $\mathcal{Y} = \{0, 1\}$, the MDL algorithm for general classes of stochastic rules with finite partitioning is stated below.

Let C_{FP} be a set of all stochastic rules with finite partitioning defined on $\mathcal{X} \times \mathcal{Y}$. Let C be a subset of C_{FP}. That is, $C = \{P(Y \mid \mathbf{X} : \theta \prec M) \in C_{FP} : M \in \mathcal{M}, \theta \in \Theta(M)\}$, where $\mathcal{M}$ is the finite set of all countable models specifying partitioning of the domain $\mathcal{X}$, and $\Theta(M) = [0, 1]^m$ is the set of all real-valued probability parameter vectors associated with M. m is the number of disjoint cells partitioned by M.

As with the class of stochastic decision lists, the MDL algorithm for learning C uses the hypothesis space $\mathcal{H}_N(C)$, which is defined as

$$\mathcal{H}_N(C) \stackrel{\text{def}}{=} \{P(Y \mid \mathbf{X} : \theta \prec M) \in C : M \in \mathcal{M}, \theta \in \Theta_N(M)\}, \tag{24}$$

where $\Theta_N(M)$ is a rectangular grid of probability vectors with cell of width of an order $O(1/\sqrt{N})$ in each direction of the m-dimensional space $[0, 1]^m$ where each direction of $[0, 1]^m$ is quantized according to (19), and $|\Theta_N(M)| \leq (4\sqrt{2N})^m$.

When given N independently drawn examples; $D^N = (\mathbf{x}_1, y_1) \ldots (\mathbf{x}_N, y_1)$ ($\mathbf{x}^N \stackrel{\text{def}}{=} \mathbf{x}_1 \ldots \mathbf{x}_N$, $y^N \stackrel{\text{def}}{=} y_1 \ldots y_N$), the total description length for D^N to be minimized with respect to θ and M is calculated as follows.

$$\ell(y^N \mid \mathbf{x}^N : \theta \prec M) + \lambda \left\{ \frac{m \log N}{2} + \frac{5m}{2} + \ell(M) \right\}. \tag{25}$$

$\ell(y^N \mid \mathbf{x}^N : \theta \prec M)$ is the description length for y^N for given $\mathbf{x}^N$ with respect to fixed $\theta = (p_1, \ldots, p_m) \in \Theta_N(M)$ and $M \in \mathcal{M}$, where p_i is the probability that any example that falls in the i-th cell belongs to the class 1. It is calculated as the minus logarithm of the likelihood as follows.

$$\ell(y^N \mid \mathbf{x}^N : \theta \prec M) = -\log \prod_{j=1}^{m} p_j^{N_j^+} (1 - p_j)^{N_j - N_j^+}$$

$$= \sum_{j=1}^{m} N_j \{H(\tilde{p}_j) + D(\tilde{\mathbf{p}}_j \parallel \mathbf{p}_j)\}, \tag{26}$$

where $H(\tilde{p}_j) \stackrel{\text{def}}{=} -\tilde{p}_j \log \tilde{p}_j - (1 - \tilde{p}_j) \log (1 - \tilde{p}_j)$, $\tilde{p}_j \stackrel{\text{def}}{=} N_j^+/N_j$ ($j = 1, \ldots m$). N_j is the number of examples whose attribute vector belongs to the j-th cell, and N_j^+ is the number of examples whose attribute vector belongs to the j-th cell and whose class is 1 ($j = 1, \ldots, m$). $D(\tilde{\mathbf{p}}_j \parallel \mathbf{p}_j) \stackrel{\text{def}}{=} \tilde{p}_j \log (\tilde{p}_j/p_j) + (1 - \tilde{p}_j) \log ((1 - \tilde{p}_j)/(1 - p_j))$.

$(m \log N)/2 + 5m/2$ is the description length for an m-dimensional real-valued probability parameter vector $\theta \in \Theta_N(M)$. This description length is derived as with stochastic decision lists.

$\ell(M)$ is the description length for $M \in \mathcal{M}$, which is calculated using a code-length function satisfying Kraft's inequality: $\Sigma_{M \in \mathcal{M}} 2^{-\ell(M)} \leq 1$. λ is an adjustment parameter such that $\lambda \geq 1$.

Fix an adjustment parameter λ and a code-length function $\mathcal{M} \to \mathbf{R}^+ \cup \{0\}$. The MDL algorithm (with the adjustment parameter λ and the code-length function ℓ) for learning C takes examples D^N as input, and then chooses one hypothesis in $\mathcal{H}_N(C)$ that attains the minimum of (25) over $\mathcal{H}_N(C)$, and outputs it. That is, an MDL rule is written as

$$P(Y \mid \mathbf{X} : \bar{\theta} \prec \hat{M}),$$

where for each $M \in \mathcal{M}$,

$$\bar{\theta} \stackrel{\text{def}}{=} \arg \min_{\theta \in \Theta_N(M)} \ell(y^N \mid \mathbf{x}^N : \theta \prec M),$$

and

$$\hat{M} \stackrel{\text{def}}{=} \arg \min_{M \in \mathcal{M}} \{\ell(y^N \mid \mathbf{x}^N : \bar{\theta} \prec M) + \lambda(\ell(\bar{\theta} \mid M) + \ell(M))\}.$$

Notice here that when for a fixed M, for the truncated vector $\bar{\theta}$ for the maximum likelihood estimator $\hat{\theta}$, the following inequality holds (the proof can be done as with (20)).

$$-\log P(y^N \mid \mathbf{x}^N : \bar{\theta} \prec M) < -\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M) + m. \tag{27}$$

Further notice $\bar{\theta} = \arg \min_{\theta \in \Theta_N(M)} \ell(y^N \mid \mathbf{x}^N : \theta \prec M)$ is calculated as a truncated vector of the maximum likelihood estimator $\hat{\theta}$.

5. Sample complexity bounds

In this section, ignoring time complexity, we derive upper bounds on the sample size required by the MDL algorithm to learn stochastic rules with finite partitioning within the (ϵ, δ)-criterion. These bounds are derived by applying the type of information theoretic proof techniques used in Barron (1985, pp. 92–93) and Barron and Cover (1991) (proof of Theorem 1). Next, we derive upper bounds on the sample size required by some other information-criteria based learning algorithms and compare them with the upper bound for the MDL algorithm. Hereafter, we assume that $\mathcal{Y} = \{0, 1\}$. Results will be easily extended to classes of stochastic rules with finite partitioning over the continuous domain.

5.1. Target-dependent sample size bounds for the MDL algorithm

Let C_{FP} be a set of all stochastic rules with finite partitioning defined on $\mathcal{X} \times \mathcal{Y}$. Let $C = \{P(Y \mid \mathbf{X} : \theta \prec M) \in C_{FP} : M \in \mathcal{M}, \theta \in \Theta(M)\}$ be a class of stochastic rules with finite partitioning where $\mathcal{M}$ is a finite set of countable models specifying partitioning of the domain, and, for each $M \in \mathcal{M}$, $\Theta(M)$ is a set of all probability parameter vectors associated with M. Hereafter, assume that the target rule is in C; i.e., there exist a true countable model $M^* \in \mathcal{M}$ and a true probability parameter vector $\theta^* \in \Theta(M^*)$.

Let $\hat{M}$ be the MDL estimator from N independently drawn examples D^N and let $\hat{\theta}$ be a maximum likelihood estimator from D^N for the fixed $\hat{M}$. It is known from the theory of maximum likelihood estimation (Fisher, 1956), that $\hat{\theta}$ converges to θ^*. The main concern is whether or not $P(Y \mid \mathbf{X} : \bar{\theta} \prec \hat{M})$ converges to $P(Y \mid \mathbf{X} : \theta^* \prec M^*)$. The almost sure convergence of the MDL estimators has been shown in Rissanen (1978; 1983; 1984; 1986; 1989); Barron (1985); and Barron and Cover (1991). In the literature of computational learning theory, however, it is a more important concern for sample complexity evaluation how fast the MDL rule converges to the target rule. In this subsection, we evaluate the rate of convergence of the MDL rule to the target rule and also derive target-dependent sample size bounds for the MDL algorithm.

Theorem 1 (Target-dependent upper bound on the sample size required by the MDL algorithm). Let C be a class of stochastic rules with finite partitioning in the form; $C = \{P(Y \mid \mathbf{X} : \theta \prec M) \in C_{FP} : M \in \mathcal{M}, \theta \in \Theta(M)\}$ where $\mathcal{M}$ is a finite set of countable models, and $\Theta(M)$ is a set of probability parameter vectors associated with M. Let a code-length function $\ell : \mathcal{M} \to \mathbf{R}^+ \cup \{0\}$ such that $\Sigma_{M \in \mathcal{M}} 2^{-\ell(M)} \leq 1$ be fixed. Let $\hat{P}_{[D^N]} \in \mathcal{H}_N(C)$ be a hypothesis that the MDL algorithm (with the adjustment parameter $\lambda = 2$ and the code-length function ℓ) outputs from N independent random samples D^N drawn according to the product probability distribution $Q(\mathbf{X})P(Y \mid \mathbf{X} : \theta^* \prec M^*)$ on $\mathcal{X} \times \mathcal{Y}$. Letting $P^*(Y \mid \mathbf{X}) = P(Y \mid \mathbf{X} : \theta^* \prec M^*)$, assume that the target rule $P^*(Y \mid \mathbf{X})$ belongs to C; i.e., $M^* \in \mathcal{M}$ and $\theta^* \in \Theta(M^*)$. Then, for any $\epsilon > 0$, for any $Q(\mathbf{X})$ on $\mathcal{X}$, for any $P^*(Y \mid \mathbf{X}) \in C$, the following inequality holds.

$$(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon] < \exp\left[-\frac{N\epsilon}{2} + \left(g_N(M^*) + \frac{m^*}{2} \right) \ln 2 \right]. \tag{28}$$

Here $g_N(M^*) \stackrel{\text{def}}{=} (m^* \log N)/2 + 5m^*/2 + \ell(M^*)$ and $m^* = \dim \Theta(M^*)$. $\ell(M^*)$ denotes the description length for $M^* \in \mathcal{M}$. d_H is the Hellinger distance (see (6)).

For $0 < \delta < 1$, and sample size

$$N \geq \frac{e}{\epsilon(e-1)} \left(m^* \ln \frac{64m^*}{\epsilon} + (2 \ln 2)\ell(M^*) + 2 \ln \frac{1}{\delta} \right), \tag{29}$$

the probability in the lefthand side in (28) is at most δ. □

Note that, without loss of generality, we assume that P^* is a rule with the minimum dimensional probability parameter vector over all the equivalent rules. Here we say that for $P_1, P_2 \in C$, P_1 is *equivalent* to P_2 if and only if for all $(\mathbf{X}, Y)$, $P_1(Y \mid \mathbf{X}) = P_2(Y \mid \mathbf{X})$.

The following lemma, which was used in Haussler and Long (1990), is also employed in the proof of Theorem 1.

Lemmas 2. Let $x, y \in \mathbf{R}^+$. Then

$$\ln x \leq xy - \ln ey.$$

(See Haussler and Long (1990, p. 16) for the proof.) □

Proof of Theorem 1. Let $g_N(\theta, M) \stackrel{\text{def}}{=} \ell(\theta \mid M) + \ell(M) = (m \log N)/2 + 5m/2 + \ell(M)$ where $m = \dim \Theta(M)$. Specifically, we let $g_N(M^*) \stackrel{\text{def}}{=} g_N(\theta, M^*) = (m^* \log N)/2 + 5m^*/2 + \ell(M^*)$. Let $\mathcal{H}_N(C)$ (for short, $\mathcal{H}_N$) be the hypothesis space which the MDL algorithm uses for learning C, where the notation follows the previous section.

First, by the definition of the MDL estimator in the case of $\lambda = 2$ (see 4.2), notice that the following inequality holds for given examples $D^N = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \in (\mathcal{X} \times \mathcal{Y})^N$ (we let $\mathbf{x}^N \stackrel{\text{def}}{=} \mathbf{x}_1 \ldots \mathbf{x}_N$, $y^N \stackrel{\text{def}}{=} y_1 \ldots y_N$).

$$\min_{P \in \mathcal{H}_N} \{\ell(y^N \mid \mathbf{x}^N : \theta \prec M) + 2g_N(\theta, M)\}$$

$$= \min_{M \in \mathcal{M}} \min_{\theta \in \Theta_N(M)} \{-\log P(y^N \mid \mathbf{x}^N : \theta \prec M) + 2g_N(\theta, M)\}$$

$$\leq -\log P(y^N \mid \mathbf{x}^N : \bar{\theta} \prec M^*) + 2g_N(M^*), \tag{30}$$

where $\bar{\theta}$ is the truncated vector for θ^*.

Let $\hat{\theta}$ be the non-truncated maximum likelihood estimator from D^N. From the relationship between the likelihoods for $\hat{\theta}$ and its truncation $\bar{\theta}$ (see (27)), the following inequality holds.

$$-P(y^N \mid \mathbf{x}^N : \bar{\theta} \prec M^*) + 2g_N(M^*)$$

$$< -\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M^*) + m^* + 2g_N(M^*). \tag{31}$$

Since $\hat{\theta}$ is the maximum likelihood estimator for M^*, the following inequality holds.

$$-\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M^*) + m^* + 2g_N(M^*)$$

$$\leq -\log P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*) + m^* + 2g_N(M^*). \tag{32}$$

By combining (30), (31), with (32), we have

$$\min_{M \in \mathcal{M}} \min_{\theta \in \Theta_N(M)} \{-\log P(y^N \mid \mathbf{x}^N : \theta \prec M) + 2g_N(\theta, M)\}$$

$$< -\log P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*) + m^* + 2g_N(M^*). \tag{33}$$

Notice that if $d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon$, then the minimum value of the total description length is attained by one of rules Ps such that $P \in \mathcal{H}_N$, $d_H(P^*, P) \geq \epsilon$. It follows that $-\log P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*) + m^* + 2g_N(M^*) > \min_{P \in \mathcal{H}_N, d_H(P^*, P) \geq \epsilon} \{\ell(y^N \mid \mathbf{x}^N : \theta \prec M) + 2g_N(\theta, M)\}$. Hence we have the following inequalities using (33).

$$(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon]$$

$$\leq (QP^*)[-\log P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*) + m^* + 2g_N(M^*)$$

$$> \min_{P \in \mathcal{H}_N, d_H(P^*, P) \geq \epsilon} \{\ell(y^N \mid \mathbf{x}^N : \theta \prec M) + 2g_N(\theta, M)\}]$$

$$= (QP^*)[P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*) 2^{-2g_N(M^*) - m^*}$$

$$< \max_{\substack{M \in \mathcal{M}, \theta \in \Theta_N(M) \\ d_H(P^*, P) \geq \epsilon}} P(y^N \mid \mathbf{x}^N : \theta \prec M) 2^{-2g_N(\theta, M)}] \tag{34}$$

$$\leq \sum_{\substack{M\in\mathscr{M}\\ d_H(P^*,P)\geq\epsilon}} \sum_{\theta\in\Theta_N(M)} (QP^*)[Q(\mathbf{x}^N)P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*)2^{-2g_N(M^*)-m^*}$$

$$\leq Q(\mathbf{x}^N)P(y^N \mid \mathbf{x}^N : \theta \prec M)2^{-2g_N(\theta,M)}]. \quad (35)$$

For the sake of simplicity, we write $P(Y \mid \mathbf{X} : \theta^* \prec M^*)$ as $P^*(Y \mid \mathbf{X})$ and $P(Y \mid \mathbf{X} : \theta \prec M)$ as $P(Y \mid \mathbf{X})$. For each $P(Y \mid \mathbf{X})$ satisfying $d_H(P^*, P) \geq \epsilon$, we have the following inequalities using a Chernoff style inequality. Below, $(\Sigma_{\mathbf{x}^N}\Sigma_{y^N})^\dagger$ denotes the summation with respect to $\mathbf{x}^N$ and y^N under the constraint that $Q(\mathbf{x}^N)P^*(y^N \mid \mathbf{x}^N)2^{-2g_N(M^*)-m^*} \leq Q(\mathbf{x}^N)P(y^N \mid \mathbf{x}^N)2^{-2g_N(\theta,M)}$.

$$(QP^*)[Q(\mathbf{x}^N)P^*(y^N \mid \mathbf{x}^N)2^{-2g_N(M^*)-m^*} \leq Q(\mathbf{x}^N)P(y^N \mid \mathbf{x}^N)2^{-2g_N(\theta,M)}]$$

$$= \left(\sum_{\mathbf{x}^N}\sum_{y^N}\right)^\dagger Q(\mathbf{x}^N)P^*(y^N \mid \mathbf{x}^N)$$

$$< \left(\sum_{\mathbf{x}^N}\sum_{y^N}\right)^\dagger Q(\mathbf{x}^N)P^*(y^N \mid \mathbf{x}^N)\left(\frac{2^{2g_N(M^*)-2g_N(\theta,M)+m^*}P(y^N \mid \mathbf{x}^N)}{P^*(y^N \mid \mathbf{x}^N)}\right)^{1/2}$$

$$= 2^{g_N(M^*)-g_N(\theta,M)+m^*/2}\left(\sum_{\mathbf{x}^N}\sum_{y^N}\right)^\dagger Q(\mathbf{x}^N)(P^*(y^N \mid \mathbf{x}^N)P(y^N \mid \mathbf{x}^N))^{1/2}$$

$$\leq 2^{g_N(M^*)-g_N(\theta,M)+m^*/2}\sum_{\mathbf{x}^N}\sum_{y^N} Q(\mathbf{x}^N)(P^*(y^N \mid \mathbf{x}^N)P(y^N \mid \mathbf{x}^N))^{1/2}$$

$$= 2^{g_N(M^*)-g_N(\theta,M)+m^*/2}\sum_{\mathbf{x}^N}\sum_{y^N}\prod_{i=1}^{N}(Q(\mathbf{x}_i)(P^*(y_i \mid \mathbf{x}_i)P(y_i \mid \mathbf{x}_i))^{1/2})$$

$$= 2^{g_N(M^*)-g_N(\theta,M)+m^*/2}\sum_{\mathbf{x}_N} Q(\mathbf{x}_N)\sum_{y_N}(P^*(y_N \mid \mathbf{x}_N)P(y_N \mid \mathbf{x}_N))^{1/2}$$

$$\cdots\sum_{\mathbf{x}_1} Q(\mathbf{x}_1)\sum_{y_1}(P^*(y_1 \mid \mathbf{x}_1)P(y_1 \mid \mathbf{x}_1))^{1/2}$$

$$= 2^{g_N(M^*)-g_N(\theta,M)+m^*/2}\left(\sum_{(\mathbf{x},y)} Q(\mathbf{x})(P^*(y \mid \mathbf{x})P(y \mid \mathbf{x}))^{1/2}\right)^N$$

$$\leq 2^{g_N(M^*)-g_N(\theta,M)+m^*/2}\, e^{-N\epsilon/2}. \quad (36)$$

The last inequality follows: For P such that $d_H(P^*, P) \geq \epsilon$,

$$\sum_{(\mathbf{x},y)} Q(\mathbf{x})(P^*(y \mid \mathbf{x})P(y \mid \mathbf{x}))^{1/2} = 1 - (1/2)d_H(P^*, P)$$

$$\leq e^{-d_H(P^*,P)/2}$$

$$\leq e^{-\epsilon/2}$$

(see Barron (1985, pp. 92–93)), where we have used the property of the Hellinger distance: $d_H(P^*, P) = \Sigma_{\mathbf{x}} Q(\mathbf{x}) \Sigma_y | \sqrt{P^*(y \mid \mathbf{x})} - \sqrt{P(y \mid \mathbf{x})} |^2 = 2 - 2\Sigma_{(\mathbf{x},y)} Q(\mathbf{x})(P^*(y \mid \mathbf{x}) P(y \mid \mathbf{x}))^{1/2}$.

Therefore, by plugging (36) into (35), we obtain the following inequalities:

$$(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon] < \Sigma_{M \in \mathcal{M}} \Sigma_{\theta \in \Theta_N(M)} 2^{g_N(M^*) - g_N(\theta,M) + m^*/2} e^{-N\epsilon/2}$$
$$d_H(P^*,P) \geq \epsilon$$

$$\leq \sum_{M \in \mathcal{M}} \sum_{\theta \in \Theta_N(M)} 2^{g_N(M^*) - g_N(\theta,M) + m^*/2} e^{-N\epsilon/2}$$

$$\leq \exp\left[- \frac{N\epsilon}{2} + \left(g_N(M^*) + \frac{m^*}{2} \right) \ln 2 \right].$$

In the last inequality, we have used Kraft's inequality:

$$\sum_{M \in \mathcal{M}} \sum_{\theta \in \Theta_N(M)} 2^{-g_N(\theta,M)} = \sum_{M \in \mathcal{M}} 2^{-\ell(M)} \sum_{\theta \in \Theta_N(M)} 2^{-(m \log N)/2 - 5m/2}$$

$$\leq 1.$$

Thus we have (28).

The inequality (28) yields the following equivalent expression:

$$N \geq \frac{m^* \ln N}{\epsilon} + \frac{2}{\epsilon} \left((3m^* + \ell(M^*)) \ln 2 + \ln \frac{1}{\delta} \right). \tag{37}$$

Since by Lemma 2 for any $\nu \in \mathbf{R}^+$ such that $0 < \nu < 1$,

$$\ln N \leq \frac{\nu\epsilon N}{m^*} + \ln \frac{m^*}{\nu\epsilon e}.$$

Thus the following is sufficient to guarantee (37).

$$N \geq \frac{m^*}{\epsilon} \left(\frac{\nu\epsilon N}{m^*} + \ln \frac{m^*}{\nu\epsilon e} \right) + \frac{2}{\epsilon} \left((3m^* + \ell(M^*)) \ln 2 + \ln \frac{1}{\delta} \right).$$

Choosing $\nu = 1/e$ for readability and solving for N yield[2]

$$N \geq \frac{e}{\epsilon(e-1)}\left[m^* \ln \frac{64m^*}{\epsilon} + (2 \ln 2)\ell(M^*) + 2 \ln \frac{1}{\delta}\right],$$

which gives (29). This completes the proof of Theorem 1. □

Here notice that the righthand side of (28) exponentially goes to zero as N increases and that it depends on the target complexities; m^* and $\ell(M^*)$. For example, for stochastic decision lists, m^* denotes the depth of the target rule and $\ell(M^*) = \log \#T_k^n[m^*]$ (see (22)). Assume that the code-length function for $\mathscr{M}$ has the property that $\ell(M_1) \geq \ell(M_2)$ for M_1, $M_2 \in \mathscr{M}$ if $\dim \Theta(M_1) \geq \dim \Theta(M_2)$. Then it follows from Theorem 1 that the simpler the target rule is (i.e., the smaller both of m^* and $\ell(M^*)$ are), the faster the MDL rule converges to it.

Next, let us derive a target-dependent upper bound on the sample size with which the MDL algorithm, with probability at least $1 - \delta$, outputs a hypothesis that lies within ϵ of the target rule with respect to the variation distance.

Theorem 2 (Target-dependent upper bound on the sample size with respect to the variation distance). Under the same assumption and notation as Theorem 1, for any $\epsilon > 0$, for any $Q(\mathbf{X})$ on $\mathcal{X}$, for any $P^*(Y \mid \mathbf{X}) \in C$, the following inequality holds.

$$(QP^*)[D^N : d_V(P^*, \hat{P}_{[D^N]}) \geq \epsilon] < \exp\left[-\frac{N\epsilon^2}{8} + \left(g_N(M^*) + \frac{m^*}{2}\right) \ln 2\right]. \tag{38}$$

Here d_V is the variation distance (see (7)).

For $0 < \delta < 1$, and sample size

$$N \geq \frac{4e}{\epsilon^2(e-1)}\left[m^* \ln \frac{256m^*}{\epsilon^2} + (2 \ln 2)\ell(M^*) + 2 \ln \frac{1}{\delta}\right], \tag{39}$$

the probability in the lefthand side in (38) is at most δ. The notation follows Theorem 1. □

Proof. First notice the general relationship (10) between the Hellinger distance d_H and the variation distance d_V: for all P_1, P_2: stochastic rules, $(d_V(P_1, P_2))^2/4 \leq d_H(P_1, P_2)$. Using this relationship and (28) in Theorem 1, we have the following inequalities.

$$(QP^*)[D^N : d_V(P^*, \hat{P}_{[D^N]}) \geq \epsilon] \leq (QP^*)\left[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \left(\frac{\epsilon}{2}\right)^2\right]$$

$$< \exp\left[-\frac{N\epsilon^2}{8} + \left(g_N(M^*) + \frac{m^*}{2}\right) \ln 2\right].$$

Thus we have (38).

As with the proof of Theorem 1, setting the bound (38) to δ and solving for N yield the result on the sample size. This completes the proof of Theorem 2. □

Theorem 1 shows that letting the target rule be P^*, the smallest sample size N_H ($1/\epsilon$, $1/\delta$, P^*) required by the MDL algorithm (with $\lambda = 2$ and the code-length function ℓ) to satisfy the (ϵ, δ)-criterion (5) with respect to the Hellinger distance is estimated as

$$N_H\left(\frac{1}{\epsilon}, \frac{1}{\delta}, P^*\right) = O\left(\frac{m^*}{\epsilon}\log\frac{m^*}{\epsilon} + \frac{\ell(M^*)}{\epsilon} + \frac{1}{\epsilon}\log\frac{1}{\delta}\right). \tag{40}$$

Theorem 2 shows that letting the target rule be P^*, the sample size N_V ($1/\epsilon$, $1/\delta$, P^*) required by the MDL algorithm (with $\lambda = 2$ and the code-length function ℓ) to satisfy the (ϵ, δ)-criterion (5) with respect to the variation distance is estimated as

$$N_V\left(\frac{1}{\epsilon}, \frac{1}{\delta}, P^*\right) = O\left(\frac{m^*}{\epsilon^2}\log\frac{m^*}{\epsilon} + \frac{\ell(M^*)}{\epsilon^2} + \frac{1}{\epsilon^2}\log\frac{1}{\delta}\right). \tag{41}$$

Here it should be noticed that both bounds depend on ϵ, δ and the target complexities; m^* and $\ell(M^*)$, only. Further notice that the sample complexity bounds depend on the choice of a distance measure. Using the relationship (13) between the quadratic distance and the variations distance, it is easily verified that the smallest sample size with respect to the quadratic distance is also bounded as (39).

5.2. Sample size bounds for the ML algorithm

In this subsection, for classes of stochastic rules with finite partitioning, let us give an upper bound on the sample size required by the ML algorithm to satisfy the (ϵ, δ)-criterion and let us compare it with that required by the MDL algorithm. We stress that the two upper bounds are obtained via a similar technique and thus admit a fair comparison although they are both upper bounds.

Theorem 3 (Upper bounds on the sample size required by the ML algorithm). Let C be a class of stochastic rules with finite partitioning; i.e., $C = \{P(Y \mid \mathbf{X} : \theta \prec M) \in C_{FP} : M \in \mathcal{M}, \theta \in \Theta(M)\}$ where $\mathcal{M}$ is a finite set of countable models and $\Theta(M)$ is a set of probability parameter vectors associated with M. Further we assume that $\mathcal{M}$ is written as $\mathcal{M} = \bigcup_{m=1}^{m_{max}} \mathcal{M}^{(m)}$ where $\mathcal{M}^{(m)}$ is a set of $M(\in \mathcal{M})$ such that $\dim \Theta(M) = m$. m_{max} is an upper bound on m. Let a hypothesis space $\mathcal{H}_N(C)$ (for short, we denote it as $\mathcal{H}_N$) be the same as that in Theorem 1. Let $\hat{P}_{[D^N]} \in \mathcal{H}_H(C)$ be a hypothesis that the ML algorithm outputs from N independent random examples D^N drawn according to the product probability distribution $Q(\mathbf{X})P(Y \mid \mathbf{X} : \theta^* \prec M^*)$ on $\mathcal{X} \times \mathcal{Y}$. That is, $\hat{P}_{[D^N]}(Y \mid \mathbf{X}) = P(Y \mid \mathbf{X} : \hat{\theta} \prec \hat{M}) \in \mathcal{H}_H(C)$, where $\hat{M}$ is an ML estimator from D^N i.e., $\hat{M} = \arg\min_{M \in \mathcal{M}}\{-\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M)\}$ and $\hat{\theta} = \arg\min_{\theta \in \Theta_N(M)}\{-\log P(y^N \mid \mathbf{x}^N : \theta \prec M)\}$ for each $M \in \mathcal{M}$. Letting $P^*(Y \mid \mathbf{X}) = P(Y \mid \mathbf{X} : \theta^* \prec M^*)$, assume that the target rule P^* belongs to C; i.e., $M^* \in \mathcal{M}$ and $\theta^* \in \Theta(M^*)$, where we let $m^* = \dim \Theta(M^*)$. Then, for any $\epsilon > 0$, for any $Q(\mathbf{X})$ on $\mathcal{X}$, for any $P^*(Y \mid \mathbf{X}) \in C$, the following inequality holds.

$$(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon]$$

$$< \exp\left[-\frac{N\epsilon}{2} + \ln\left(\sum_{m=1}^{m_{max}} (4\sqrt{2N})^m \mid \mathscr{M}^{(m)} \mid \right) + \frac{m^* \ln 2}{2} \right]. \quad (42)$$

Here d_H is the Hellinger distance.

For $0 < \delta < 1$, the ML algorithm, with probability at least $1 - \delta$, outputs a hypothesis that lies within ϵ of the target rule with respect to the Hellinger distance, with sample size

$$N = O\left(\frac{m_{max}}{\epsilon} \ln \frac{m_{max}}{\epsilon} + \frac{\log \mid \mathscr{M} \mid}{\epsilon} + \frac{1}{\epsilon} \log \frac{1}{\delta} \right). \quad (43)$$

The ML algorithm also outputs, with probability at least $1 - \delta$, a hypothesis that lies within ϵ of the target rule with respect to the variation distance (and also with respect to the quadratic distance), with sample size

$$N = O\left(\frac{m_{max}}{\epsilon^2} \ln \frac{m_{max}}{\epsilon} + \frac{\log \mid \mathscr{M} \mid}{\epsilon^2} + \frac{1}{\epsilon^2} \log \frac{1}{\delta} \right). \quad (44)$$

□

Proof. The outline of the proof is the same as that in Theorem 1. Let $\hat{M}$ be the ML estimator of the true countable model. We use the following property of the ML estimator instead of that of the MDL estimator: For given examples $D^N = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \in (\mathcal{X} \times \mathcal{Y})^N$ (we let $\mathbf{x}_N \stackrel{\text{def}}{=} \mathbf{x}_1 \ldots \mathbf{x}_N$, $y^N \stackrel{\text{def}}{=} y_1 \ldots y_N$), the following inequalities hold.

$$\min_{M \in \mathscr{M}} \min_{\theta \in \Theta_N(M)} \{-\log P(y^N \mid \mathbf{x}^N : \theta \prec M)\} \leq -\log P(y^N \mid \mathbf{x}^N : \bar{\theta} \prec M^*)$$

$$< -\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M^*) + m^*$$

$$\leq -\log P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*) + m^*,$$

where $\hat{\theta}$ is the non-truncated maximum likelihood estimator from D^N for M^*, and $\bar{\theta}$ is the truncated vector for $\hat{\theta}$ in $\Theta_{N(M^*)}$. In the second inequality, we have used (27). The process for estimating $(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon]$ is essentially the same as Theorem 1. However, (34) in the proof of Theorem 1 is replaced by (45), and (35) is also replaced by (46).

$$(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon]$$

$$\leq (QP^*)[-\log P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*) + m^*$$

$$> \min_{\substack{M \in \mathscr{M}, \theta \in \Theta_N(M) \\ d_H(P^*, P) \geq \epsilon}} \{-\log P(y^N \mid \mathbf{x}^N : \theta \prec M)\}]$$

$$= (QP^*)[P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*)2^{-m^*}$$

$$< \max_{\substack{M \in \mathscr{M}, \theta \in \Theta_N(M) \\ d_H(P^*, P) \geq \epsilon}} P(y^N \mid \mathbf{x}^N : \theta \prec M)] \quad (45)$$

$$\leq \sum_{\substack{M \in \mathscr{M} \\ d_H(P^*, P) \geq \epsilon}} \sum_{\theta \in \Theta_N(M)} (QP^*)[Q(\mathbf{x}^N)P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*)2^{-m^*}$$

$$< Q(\mathbf{x}^N)P(y^N \mid \mathbf{x}^N : \theta \prec M)]. \quad (46)$$

As with the proof of Theorem 1, for each $P(Y \mid \mathbf{X} : \theta \prec M) \in \mathscr{H}_N$ satisfying $d_H(P^*, P) \geq \epsilon$, the following inequality holds (see also (36)).

$$(QP^*)[Q(\mathbf{x}^N)P(y^N \mid \mathbf{x}^N : \theta^* \prec M^*)2^{-m^*} \leq Q(\mathbf{x}^N)P(y^N \mid \mathbf{x}^N : \theta \prec M)] < 2^{m^*/2}e^{-N\epsilon/2}. \quad (47)$$

Thus, plugging (47) into (46) yields

$$(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon]$$

$$< \sum_{\substack{M \in \mathscr{M} \\ d_H(P^*, P) \geq \epsilon}} \sum_{\theta \in \Theta_N(M)} 2^{m^*/2}e^{-N\epsilon/2}$$

$$\leq \sum_{M \in \mathscr{M}} \sum_{\theta \in \Theta_N(M)} 2^{m^*/2}e^{-N\epsilon/2}$$

$$= \exp\left[-\frac{N\epsilon}{2} + \ln\left(\sum_{m=1}^{m_{max}} (4\sqrt{2N})^m \mid \mathscr{M}^{(m)} \mid \right) + \frac{m^* \ln 2}{2} \right],$$

where we have used the fact (see the derivation of (21) with regard to the size of $\Theta_N(M)$):

$$\sum_{M \in \mathscr{M}} \sum_{\theta \in \Theta_N(M)} 1 \leq \sum_{m=1}^{m_{max}} (4\sqrt{2N})^m \mid \mathscr{M}^{(m)} \mid.$$

Hence we obtain (42).

Notice that $\ln(\Sigma_{m=1}^{m_{max}}(4\sqrt{2N})^m \mid \mathscr{M}^{(m)} \mid) \leq m_{max} \ln N + (5m_{max}/2) \ln 2 + \ln \mid \mathscr{M} \mid$. Thus, we have

$$(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon]$$

$$\leq \exp\left[-\frac{N\epsilon}{2} + \left(m_{max} \log N + \log \mid \mathscr{M} \mid + \frac{5m_{max}}{2} + \frac{m^*}{2} \right) \ln 2 \right]. \quad (48)$$

As with the proof of Theorem 1, setting the bound (48) to δ and solving for N yield (43).

As with Theorem 2, we have (44) using the bound (43) and the relationship (10) between the variation distance and the Hellinger distance. Using the relationship (13) between the variation distance and the quadratic distance, it is easily verified that (44) is also an upper bound on the sample size with respect to the quadratic distance. This completes the proof of Theorem 3. □

Theorem 3 shows that the sample size bounds (43) and (44) depend on not the target complexities; m^* and $\ell(M^*)$, but the parameters of the hypothesis space; m_{max} and $\log \mid \mathscr{M} \mid$.

Let us consider the "worst-case" where the target rule is $P(Y \mid \mathbf{X} : \theta^* \prec M^*)$ such that $\dim \Theta(M^*) = m_{max}$. We use the following code-length function over $\mathscr{M}$: For all $M \in \mathscr{M}$,

$$\ell(M) = \log \mid \mathscr{M} \mid, \tag{49}$$

which is derived by $\ell(M) = -\log P(M)$ where $P(M)$ is a uniform distribution over $\mathscr{M}$; $P(M) \stackrel{\text{def}}{=} 1/\mid \mathscr{M} \mid$ for all $M \in \mathscr{M}$ (we call the code-length function (49) the *uniform code-length function*). Then the bound (40) for the MDL algorithm is equivalent with the bound (43) for the ML algorithm. Similarly, in this worst-case, (41) for the MDL algorithm (with $\lambda = 2$ and the uniform code-length function) is of the same order as (44) for the ML algorithm. This implies that the upper bound on the sample size for the ML algorithm is of the same order as the worst-case upper bound for the MDL algorithm with $\lambda = 2$ and the uniform code-length function. Except for this worst-case, the ML algorithm requires larger sample size to satisfy the (ϵ, δ)-criterion (5) than the MDL algorithm.

Let us consider more general cases where learning algorithms are based on information criteria which select P specified by M that minimize the following statistic:

$$-\log P(y^N \mid \mathbf{x}^N : \hat{\theta} \prec M) + f_N(\hat{\theta}, M), \tag{50}$$

where $\hat{\theta}$ is a maximum likelihood estimator for a fixed M, and $f_N(\hat{\theta}, M)$ is a function of $\hat{\theta}$ and M, which depends on N.

Let $f_N(\hat{\theta}, M) = O(\log^\alpha N)$ $(0 \leq \alpha < 1)$ with respect to N. For example, we have the ML algorithm by letting $f_N(\hat{\theta}, M) = 0$, and we have Akaike's information criterion (Akaike, 1974) by letting $f_N(\hat{\theta}, M) = m \log e$, where $m = \dim \Theta(M)$. In this case, using the same type proof technique as that used in the proof of Theorem 3, we obtain the following upper bound on the sample size with respect to the Hellinger distance.

$$N = O\left(\frac{m_{max}}{\epsilon} \ln \frac{m_{max}}{\epsilon} + \frac{\log \mid \mathscr{M} \mid}{\epsilon} + \frac{1}{\epsilon} \log \frac{1}{\delta}\right), \tag{51}$$

which is of the same order as the bound for the ML algorithm.

Next, let us consider the case where $f_N(\theta, M) = O(\log^\alpha N)$ $(\alpha > 1)$. Using the same type proof technique as that used in the proof of Theorem 1, we obtain an upper bound on the rate of convergence:

$$(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon] < \exp\left[-\frac{N\epsilon}{2} + O(\log^\alpha N)\right], \tag{52}$$

where $\hat{P}_{[D^N]}$ is an output of the algorithm. For some finite positive integer N_0, the right-hand side of (52) is larger than that in (28) for all $N \geq N_0$. It follows that the upper bound for this case is also larger than that for the MDL algorithm for sufficiently small ϵ and δ.

Therfore, the upper bound (40) on the sample size required by the MDL algorithm gives the least upper bound on the sample complexity (which is defined by the 'smallest' sample size needed for the (ϵ, δ)-criterion (5)) of learning stochastic rules with finite partitioning with respect to the Hellinger distance.

6. Results on learnability of stochastic rules with finite partitioning

In this section, we first give upper bounds on the worst-case sample complexities of learning stochastic rules with finite partitioning and also give a sufficient condition for statistical learnability of any given class. The worst-case sample complexity is the sample complexity of learning the target rule such that the probability that a hypothesis cannot lie within ϵ of the rule is largest over all rules in the target class C. Based on the results, we prove statistical learnability of C_{DL}^k and C_{DT}^k and derive worst-case sample complexity bounds of learning them. Further, we derive a sufficient condition for polynomial learnability of any given class of stochastic rules with finite partitioning.

6.1. Worst-case sample complexity bounds and statistical learnability

We introduce a notion of 'sound code-length function' in order to derive bounds on worst-case sample complexities.

Definition 6 (Sound code-length function). Let C_{FP} be a set of all stochastic rules with finite partitioning defined on $\mathcal{X} \times \mathcal{Y}$. Let C be a subset of C_{FP} i.e., $C = \{P(Y \mid \mathbf{X} : \theta \prec M) \in C_{FP} : M \in \mathcal{M}, \theta \in \Theta(M)\}$ where $\mathcal{M}$ is a finite set of countable models and $\Theta(M)$ is a set of probability parameter vectors associated with $M \in \mathcal{M}$. We define $\xi(C)$ by $\xi(C) \stackrel{\text{def}}{=} \max_{M \in \mathcal{M}} \dim \Theta(M)$. Let ℓ be a code-length function over $\mathcal{M}$: i.e., a function $\ell : \mathcal{M} \to \mathbf{R}^+ \cup \{0\}$, satisfying Kraft's inequality: $\Sigma_{M \in \mathcal{M}(C)} 2^{-\ell(M)} \leq 1$. We define the *maximum complexity* of C associated with ℓ as $\max_{M \in \mathcal{M}} \ell(M)$, and we denote it as $\ell_{max}(C)$. We say that ℓ is *sound* if $\ell_{max}(C) = \ell(M)$ holds for M such that $\dim \Theta(M) = \xi(C)$. □

The following lemma gives a lower bound on the maximum complexity for any given C.

Lemma 3. Let C be a class of stochastic rules with finite partitioning. Let $\mathcal{M}$ be a finite set of countable models specifying C. Then for any code-length function ℓ, the maximum complexity of C; $\ell_{max}(C)$, satisfies the following inequality.

$$\ell_{max}(C) \geq \log \mid \mathcal{M} \mid. \tag{53}$$

The equality holds only for the uniform code-length function: $\ell(M) = \log \mid \mathcal{M} \mid$, for all $M \in \mathcal{M}$. □

Proof. Let $\ell(M)$ be the code length for $M \in C$. First note

$$\sum_{M \in \mathcal{M}} 2^{-\ell(M)} \geq \mid \mathcal{M} \mid 2^{-\ell_{max}(C)}.$$

Here the equality holds only for the uniform code-length function: $\ell(M) = \log \mid \mathcal{M} \mid$, for all $M \in \mathcal{M}$. Notice that the uniform code-length function is sound.

On the other hand, by Kraft's inequality, we have

$$\sum_{M \in \mathcal{M}} 2^{-\ell(M)} \leq 1.$$

Notice that the equality holds for the uniform code-length function.

Hence, from the above two inequalities, we have

$$\ell_{max}(C) \geq \log \mid \mathcal{M} \mid.$$

Here the equality holds only for the uniform code-length function. This completes the proof of Lemma 3. □

The following theorem gives a sufficient condition for statistical learnability of stochastic rules with finite partitioning and worst-case upper bounds on the sample complexities of learning them.

Theorem 4 (Worst-case sample complexity bounds and sufficient condition for statistical learnability). Let C be a class of stochastic rules with finite partitioning and $\mathcal{M}$ be a finite set of countable models, each of which specifies partitioning of the domain. Fix a sound code-length function $\ell : \mathcal{M} \to \mathbf{R}^+ \cup \{0\}$ such that $\Sigma_{M \in \mathcal{M}} 2^{-\ell(M)} \leq 1$. Whenever sample size N satisfies

$$N \geq \frac{e}{\epsilon(e-1)} \left(\xi(C) \ln \frac{64\xi(C)}{\epsilon} + (2 \ln 2)\ell_{max}(C) + 2 \ln \frac{1}{\delta} \right), \tag{54}$$

the MDL algorithm (with $\lambda = 2$ and the code-length function ℓ), with probability at least $1 - \delta$, produces a hypothesis $\hat{P}$ such that $d_H(P^*, \hat{P}) < \epsilon$ for all Q on $\mathcal{X}$ and for all $P^* \in C$. Here d_H is the Hellinger distance.

Further, there exists an algorithm which learns C within the (ϵ, δ)-criterion with respect to the Hellinger distance for sample size

$$N \geq \frac{e}{\epsilon(e-1)}\left(\xi(C)\ln\frac{64\xi(C)}{\epsilon} + (2\ln 2)\log|\mathscr{H}| + 2\ln\frac{1}{\delta}\right). \tag{55}$$

That is, an upper bound on the worst-case sample complexity $N_H(1/\epsilon, 1/\delta, n)$ of learning C with respect to the Hellinger distance is given by

$$N_H\left(\frac{1}{\epsilon}, \frac{1}{\delta}, n\right) = O\left(\frac{\xi(C)}{\epsilon}\log\frac{\xi(C)}{\epsilon} + \frac{\log|\mathscr{H}|}{\epsilon} + \frac{1}{\epsilon}\log\frac{1}{\delta}\right). \tag{56}$$

An upper bound on the worst-case sample complexity $N_V(1/\epsilon, 1/\delta, n)$ with respect to the variation distance (and also with respect to the quadratic distance) is given by

$$N_V\left(\frac{1}{\epsilon}, \frac{1}{\delta}, n\right) = O\left(\frac{\xi(C)}{\epsilon^2}\log\frac{\xi(C)}{\epsilon} + \frac{\log|\mathscr{H}|}{\epsilon^2} + \frac{1}{\epsilon^2}\log\frac{1}{\delta}\right). \tag{57}$$

If both of $\xi(C)$ and $\log|\mathscr{H}|$ are polynomial in n, then C is statistically learnable with respect to the Hellinger distance, the variation distance and the quadratic distance. □

Proof. First, notice that if there exists an algorithm which takes independent examples D^N, drawn according to the target rule as input, and outputs $\hat{P}_{[D^N]}$ such that

$$\max_{P^* \in C}(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon] \leq \delta,$$

then the (ϵ, δ)-criterion (5) holds even for the worst-case where the target rule P^* attains $\max_{P^* \in C}(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon]$. That is, (5) holds for all $P^* \in C$.

Next, notice that, by Theorem 1, the following inequality holds for the MDL algorithm with $\lambda = 2$ and the code-length function ℓ.

$$\max_{P^* \in C}(QP^*)[D^N : d_H(P^*, \hat{P}_{[D^N]}) \geq \epsilon] < \max_{M^* \in \mathscr{H}} \exp\left[-\frac{N\epsilon}{2} + \left(\frac{m^* \log N}{2} + \ell(M^*) + 3m^*\right)\ln 2\right],$$

where $\hat{P}_{[D^N]}$ is an output of the MDL algorithm from D^N. We define $F_N(M^*)$ by $F_N(M^*) \stackrel{\text{def}}{=} \exp[-(N\epsilon)/2 + ((m^* \log N)/2 + \ell(M^*) + 3m^*)\ln 2]$, where $m^* = \dim \Theta(M^*)$.

Since, by the assumption, the code-length function ℓ is sound, $\max_{M^* \in \mathscr{H}} F_N(M^*)$ is attained by M^* such that $\dim \Theta(M^*) = \xi(C)$. Hence, $\max_{M^* \in \mathscr{H}} F_N(M^*)$ is given by

$$\max_{M^* \in \mathscr{H}} F_N(M^*) = \exp\left[-\frac{N\epsilon}{2} + \left(\frac{\xi(C)\log N}{2} + \ell_{max}(C) + 3\xi(C)\right)\ln 2\right].$$

As with the proof of Theorem 1, setting $\max_{M^* \in \mathscr{H}} F_N(M^*)$ to δ and solving for N give the sample complexity bound (54). By Lemma 3, $\ell_{max}(C) \geq \log|\mathscr{H}|$ and the equality holds only for the uniform code-length function. Therefore, an upper bound on the sample

complexity, which is defined by the "smallest" sample size (required for the (ϵ, δ)-criterion to be satisfied) over all algorithms, is obtained by plugging $\ell_{max}(C) = \log \mid \mathscr{M} \mid$ into (54). This bound is attained by the MDL algorithm with $\lambda = 2$ and the uniform code-length function. Thus we have (55).

The upper bound (55) on the sample complexity is polynomial in $1/\epsilon$ and $1/\delta$. (55) implies that if both $\xi(C)$ and $\log \mid \mathscr{M} \mid$ are polynomial in n, then the upper bound on the sample complexity of learning any $P^* \in C$ is also polynomial in n, and thus C is statistically learnable with respect to the Hellinger distance. Since the Hellinger distance and the variation distance are polynomially related each other and the quadratic distance is bounded by the variation distance (see Lemma 1), C is also statistically learnable with respect to the variation distance and the quadratic distance if C is statistically learnable with respect to the Hellinger distance.

As with Theorem 2, (55) and the relationship (10) between the Hellinger distance and the variation distance yield (57). Using the relationship (13) between the variation distance and the quadratic distance, it is easily verified that (57) is also an upper bound with respect to the quadratic distance. This completes the proof of Theorem 4. □

The worst-case sample complexity bound (55) is attained by the MDL algorithm using the uniform code-length function. However, if we like to let the MDL rule converge to the target rule faster when the target rule is simple, it is better to use any other encoding scheme such that $\ell(M_1) \leq \ell(M_2)$ if $\dim \Theta(M_1) \leq \dim \Theta(M_2)$. Because, when we consider not worst-case but the target-dependent case, the rate of convergence depends on the description length for the countable model of the target rule (see (28)).

As corollaries of Theorem 4, we have results on statistical learnability of C_{DL}^k and C_{DT}^k and upper bounds on the sample complexities of learning them.

Corollary 1 (Statistical learnability of C_{DL}^k and sample complexity bound). For a fixed k, C_{DL}^k is statistically learnable with respect to the Hellinger distance, the variation distance, and the quadratic distance. The following sample size N is sufficient for learning $C_{DL}^k(n)$ within the (ϵ, δ)-criterion with respect to the Hellinger distance.

$$N \geq \frac{e}{\epsilon(e-1)}\left(\mid T_k^n \mid \ln \frac{64 \mid T_k^n \mid}{\epsilon} + 2 \ln \mid T_k^n \mid ! + 2 \ln \frac{1}{\delta}\right), \tag{58}$$

where $\mid T_k^n \mid$ denotes the number of all elements in T_k^n;

$$\mid T_k^n \mid = \sum_{i=0}^{k} 2^i \binom{n}{i}.$$

That is, an upper bound on the sample complexity $N_H(1/\epsilon, 1/\delta, n)$ of learning $C_{DL}^k(n)$ with respect to the Hellinger distance is given by

$$N_H\left(\frac{1}{\epsilon}, \frac{1}{\delta}, n\right) = O\left(\frac{n^k}{\epsilon} \log \frac{n}{\epsilon} + \frac{1}{\epsilon} \log \frac{1}{\delta}\right). \tag{59}$$

Proof. For $C_{DL}^k(n)$, the code-length function (22) over $\mathscr{H} = \Gamma_k^n$ is sound. Notice

$$\xi(C_{DL}^k(n)) = |T_k^n| = \sum_{i=0}^{k} 2^i \binom{n}{i} = O(n^k),$$

$$\ell_{max}(C_{DL}^k(n)) = \log |T_k^n|! = O(n^k \log n).$$

Thus, we obtain (58) and (59) using (54). The upper bound (59) on the sample complexity is polynomial in $1/\epsilon$, $1/\delta$ and n. Hence C_{DL}^k is statistically learnable with respect to the Hellinger distance. □

Corollary 2 (Statistical learnablity of C_{DT}^k and sample complexity bound). For a fixed k, C_{DT}^k is statistically learnable with respect to the Hellinger distance, variation distance and the quadratic distance. The following sample size N is sufficient for learning $C_{DT}^k(n)$ within the (ϵ, δ)-criterion with respect to the Hellinger distance.

$$N \geq \frac{e}{\epsilon(e-1)}\left(2^k \ln \frac{2^{k+6}}{\epsilon} + (2^{k-1} \ln 2)(\log n\langle k\rangle) + 2 \ln \frac{1}{\delta}\right), \tag{60}$$

where $n\langle k\rangle \stackrel{\text{def}}{=} (n+1) \cdot n \cdot \ldots \cdot (n-k+1)$. □

Proof. Let us define a *terminal node* in a path (from a root to a leaf) by a node which is the farthest node (except a leaf) from the root in the path. For a given $P \in C_{DT}^k(n)$, let the length (=the number of inner nodes) of the path from the root to the i-th terminal node be t_i $(i = 1, \ldots, m)$, where m is the number of terminal nodes. Each node is selected from $\{x_1, \ldots, x_n, *\}$ where * is a symbol indicating a terminal node, and the j-th node from the root can be described in $\log(n+2-j)$ bits. Thus, $\log n\langle t_i\rangle \stackrel{\text{def}}{=} \Sigma_{j=1}^{t_i} \log(n+2-j) + \log(n-t_i+1) = \log(n+1) + \log n + \ldots + \log(n-t_i+1)$ bits are sufficient to encode the path from the root to the i-th terminal node, where the last $\log(n-t_i+1)$ bits are necessary for describing "being the terminal node." Thus, $\Sigma_{i=1}^m (\log n\langle t_i\rangle)$ bits are sufficient to encode a tree form $M \in \Omega_k^n$ with m terminal nodes. See Quinlan and Rivest (1989) for more details of methods for encoding decision trees. It is easily verified that this code-length function scheme is sound. Also it is easily verified that $\ell_{max}(C_{DT}^k(n)) = 2^{k-1}(\log n\langle k\rangle)$ and $\xi(C_{DT}^k(n)) = 2^k$. Thus the worst-case sample complexity bound (60) is obtained using (54). Since this sample complexity bound is polynomial in $1/\epsilon$, $1/\delta$, and n, C_{DT}^k is statistically learnable with respect to the Hellinger distance. □

For any class $\mathscr{F}$ of deterministic rules, it is known (see Blumer, et al. (1989)) that the smallest sample size $N_0(1/\epsilon, 1/\delta, n)$ that any consistent algorithm, with probability at least $1-\delta$, outputs a hypothesis such that $Q(f^*(\mathbf{X}) \neq \hat{f}(\mathbf{X})) < \epsilon$ is bounded as $N_0(1/\epsilon, 1/\delta, n) = O((VCdim(\mathscr{F})/\epsilon)\log(1/\epsilon) + (1/\epsilon)\ln(1/\delta))$, where $VCdim(\mathscr{F})$ is the Vapnik-Chervonenkis dimension of $\mathscr{F}$ (Vapnik & Chervonenkis, 1971; Blumer, et al., 1989). It is further known (Blumer, et al., 1987) that when $\mathscr{F}$ is finite, the sample complexity of learning $\mathscr{F}$ is given by $N_0(1/\epsilon, 1/\delta, n) = O((1/\epsilon)\log|\mathscr{F}| + (1/\epsilon)\log(1/\delta))$. For the class of deterministic decision lists with degree at most k (i.e., the class of stochastic rules in C_{DL}^k whose probability

parameters are all 0 or 1), which we denote $\mathcal{F}^k_{DL}(n)$, it is known that $VCdim(\mathcal{F}^k_{DL}(n)) = \Theta(n^k)$ (Ehrenfeucht, et al. 1989) and $|\mathcal{F}^k_{DL}(n)| = O(3^{|T^n_k|} \mid T^n_k \mid !)$ (Rivest, 1987). Thus we have the following upper bound on the sample complexity of learning $\mathcal{F}^k_{DL}(n)$ (Ehrenfeucht, et al., 1989).

$$N_0\left(\frac{1}{\epsilon}, \frac{1}{\delta}, n\right) = O\left(\frac{n^k}{\epsilon} \min\left\{\log\frac{1}{\epsilon}, \log n\right\} + \frac{1}{\epsilon}\log\frac{1}{\delta}\right). \tag{61}$$

Comparing the bound (61) with the bound (59), we can see that $\log(n/\epsilon)$ in (59) is replaced by $\min\{\log(1/\epsilon), \log n\}$ in (61). This sample complexity comparison with respect to the upper bounds suggests that, when we ignore time complexity, learning stochastic decision lists with respect to the Hellinger distance may require only slightly more examples than learning deterministic decision lists with respect to the symmetric erroneous measure. Notice here that there exists an efficient algorithm for learning deterministic decision lists in the noise-free case (Rivest, 1987), but the MDL algorithm is not efficient.

6.2. Sufficient condition for polynomial learnability of stochastic rules

In this subsection, we derive a sufficient condition for polynomial learnability of any given class of stochastic rules with finite partitioning. In learning deterministic rules, the problem of finding the shortest rules among those that are consistent with given examples is often computationally intractable (Pitt & Valiant, 1988). We may also conjecture that the problem of finding the MDL rule from $\mathcal{H}_N(C^k_{DL}(n))$ (see 4.2) is computationally intractable. Thus, in order to learn stochastic rules within our criterion efficiently, we need an algorithm that outputs, in polynomial time, not an MDL rule itself but a rule that approximates the MDL rule. We call such an algorithm an *approximately-MDL algorithm*. Let a class C of stochastic rules be given. The following theorem shows that if there exists a polynomial-time approximately-MDL algorithm which outputs a rule that lies within some accuracy of the MDL rule $\hat{P}_{[D^N]} \in \mathcal{H}_N(C)$ with respect to the variation distance, then C is polynomially learnable with respect to the variation distance.

Theorem 5 (Sufficient condition for polynomial learnability of general stochastic rules with finite partitioning). Let C be a class of stochastic rules with finite partitioning. Let $\mathcal{TC}$ be a set of all countable models, each of which specifies a target rule belonging to C. We assume that C satisfies the sufficient condition for statistical learnability given in Theorem 4. If there exists an algorithm $\mathcal{A}$ such that, when given N input examples D^N, for some constants $\alpha \geq 0$ and $\beta > 0$, for any n, for any $\epsilon > 0$, for any $0 < \delta < 1$, for all Q on $\mathcal{X}$, for all $P^* \in C$, $\mathcal{A}$ outputs, in time polynomial in n and N, a hypothesis $\tilde{P}_{[D^N]} \in \mathcal{H}_N(C)$ satisfying

$$d_V(\tilde{P}_{[D^N]}, \hat{P}_{[D^N]}) < n^\alpha N^{-\beta}, \tag{62}$$

for all random examples D^N drawn according to $Q(\mathbf{X})P^*(Y \mid \mathbf{X})$ and for any MDL rule $\hat{P}_{[D^N]} \in \mathcal{H}_N(C)$ (which the MDL algorithm (with $\lambda = 2$ and the uniform code-length function) outputs from D^N), then C is polynomially learnable. Here d_V is the variation distance. Whenever sample size N satisfies

$$N \geq \max \left\{ \left(\frac{2n^\alpha}{\epsilon} \right)^{1/\beta}, \right.$$

$$\left. \frac{16e}{\epsilon^2(e-1)} \left(\xi(C) \ln \frac{1024\xi(C)}{\epsilon^2} + (2 \ln 2) \log \mid \mathcal{M} \mid + 2 \ln \frac{1}{\delta} \right) \right\}, \tag{63}$$

$\mathcal{A}$ outputs, for all Q on $\mathcal{X}$ and for all $P^* \in C$, with probability at least $1 - \delta$, a hypothesis that lies within ϵ of the target rule with respect to the variation distance. Here $\xi(C) = \max_{M \in \mathcal{M}} \dim \Theta(M)$ ($\mathcal{M}$ is a finite set of countable models specifying C). □

Proof. Assume that there exists an algorithm $\mathcal{A}$ satisfying (62) for all Q on $\mathcal{X}$, for all $P^* \in C$, for all random examples D^N drawn according to $Q(\mathbf{X})P^*(Y \mid \mathbf{X})$, and for any MDL rule $\hat{P}_{[D^N]} \in \mathcal{H}_N(C)$. Then, whenever N satisfies $N \geq (2n^\alpha/\epsilon)^{1/\beta}$, $d_V(\tilde{P}_{[D^N]}, \hat{P}_{[D^N]}) < \epsilon/2$ holds for all Q on $\mathcal{X}$, for all $P^* \in C$, for all random examples D^N, where $\tilde{P}_{[D^N]}$ is an output of $\mathcal{A}$.

Next, notice that, for such an algorithm $\mathcal{A}$, whenever N satisfies $N \geq (2n^\alpha/\epsilon)^{1/\beta}$, the following inequalities hold.

$$(QP^*)[D^N : d_V(P^*, \tilde{P}_{[D^N]}) \geq \epsilon]$$

$$\leq (QP^*)[D^N : d_V(P^*, \hat{P}_{[D^N]}) + d_V(\hat{P}_{[D^N]}, \tilde{P}_{[D^N]}) \geq \epsilon]$$

$$\leq (QP^*)\left[D^N : d_V(P^*, \hat{P}_{[D^N]}) \geq \frac{\epsilon}{2} \right].$$

Here in the second inequality, we have used the triangle inequality with respect to d_V metric; $d_V(P^*, \tilde{P}_{[D^N]}) \leq d_V(P^*, \hat{P}_{[D^N]}) + d_V(\hat{P}_{[D^N]}, \tilde{P}_{[D^N]})$ for any MDL rule $\hat{P}_{[D^N]}$ inferred from D^N. By Theorem 2 and Theorem 4, we see that, whenever N satisfies $N \geq (16e/\epsilon^2(e-1))((\xi(C) \ln (1024\xi(C))/\epsilon^2 + (2 \ln 2) \log \mid \mathcal{M} \mid + 2 \ln (1/\delta))$, the MDL algorithm (with $\lambda = 2$ and the uniform code-length function) outputs $\hat{P}_{[D^N]}$ satisfying $(QP^*)[D^N : d_V(P^*, \hat{P}_{[D^N]}) \geq \epsilon/2] \leq \delta$ for all Q on $\mathcal{X}$ and for all $P^* \in C$. Thus, whenever N satisfies (63), the following inequality holds for all Q on $\mathcal{X}$ and for all $P^* \in C$.

$$(QP^*)[D^N : d_V(P^*, \tilde{P}_{[D^N]}) \geq \epsilon] \leq \delta.$$

Since C is assumed to satisfy the sufficient condition for statistical learnability (see Theorem 4), the sample size N which is given by the righthand side in (63) is polynomial in $1/\epsilon$, $1/\delta$, and n. Therefore, if $\mathcal{A}$ runs in time polynomial in n and N, C is polynomially learnable. This completes the proof of Theorem 5. □

Theorem 5 is a criterion for determining whether or not any given class of stochastic rules with finite partitioning is polynomially learnable. The condition (62) implies that the output hypothesis must come within accuracy $O(N^{-\beta})$ of the MDL rule with respect to the variation distance, over the class of hypotheses.

7. Conclusion

In this paper, we have developed a learning criterion for stochastic rules by extending Valiant's PAC learning criterion to the stochastic setting. We have presented the MDL algorithm for learning stochastic rules with finite partitioning. Further, we have derived upper bounds on the sample complexity of learning stochastic rules with finite partitioning and have given sufficient conditions for statistical learnability and polynomial learnability of any given class. From target-dependent sample-size bound analysis, it has turned out that the MDL algorithm performs well within our learning model in the sense that it is more sample-size efficient than other known information-criteria-based learning algorithms.

In our proposed learning framework, the following problems remain open for future study.

(1) *The necessity for improving the sample complexity bounds.*
We feel that our upper bounds on the sample complexities are still in need of refining and that they should be compared to lower bound estimators, which we as yet have no method of determining. Further, the sample complexity estimation with respect to the Kullback-Leibler divergence or other distances looks very challenging.

(2) *The necessity for developing efficient approximately-MDL algorithms.*
We need efficient approximately-MDL algorithms in order to generate nearly optimal (in the sense of the MDL principle) stochastic rules in time polynomial in n. It would be interesting if we could characterize what approximately-MDL algorithms satisfy the sufficient condition for polynomial learnability (which is given in Theorem 5).

(3) *The necessity for investigating learnability of general classes of stochastic rules.*
We have focused on learnability of classes of stochastic rules with finite partitioning. It would be interesting if we could characterize statistical learnability and polynomial learnability of more general classes, e.g., classes of stochastic rules specified by infinite countable models, etc.

(4) *The necessity for extending our learning model to an agnostic one.*
In this paper, we have assumed that the target rule according to which examples are generated belongs to some known parametric class. However, this assumption seems too strong for the practical application. It would be interesting if we could extend our learning criterion to that in which no assumption is made about the nature of the target rule. We call such a learning model an *agnostic* one (see Haussler (1990, p. 32)). Haussler, from the decision theoretic viewpoint, has developed an agnostic learning model (Haussler, 1989; 1990) in which the uniform convergence technique is available for the sample complexity estimation. However, it has not yet been clarified how the computational complexity of the MDL algorithm is related to his agnostic approach.

These questions will be dealt with in future papers.

Acknowledgments

The author especially wishes to express his sincere gratitude to Dr. Abe and Mr. Takeuchi of C&C Information Technology Research Laboratories, NEC corporation, for his helpful commentary on the paper. The author greatly appreciates the anonymous reviewer's helpful suggestions. He also thanks Mr. Nakamura and Mr. Kaneko of C&C Information Technology Research Laboratories, NEC Corporation, for their encouragement and support.

Notes

1. When $N_j^+ = 0$, or $N_j^+ = N_j$, in order to avoid setting $\hat{p}_j = 0$, or 1, we often use the Laplace estimator (see for example, Schreiber (1985)); $\hat{p}_j = (N_j^+ + 1)/(N_j + 2)$ $(j = 1, \ldots, m)$ rather than the maximum likelihood estimator.
2. The technique for solving N is due to Haussler and Long (1990, p. 17).

References

Abe, N. & Warmuth, M. (1990). On the computational complexity of approximating distributions by probabilistic automata. *Proceedings of the Third Workshop on Computational Learning Theory* (pp. 52–66), Rochester, NY: Morgan Kaufmann.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Trans. Autom. Contr., AC-19*, 716–723.

Angluin, D. & Laird, P. (1988). Learning from noisy examles. *Machine Learning*, 343–370.

Barron, A.R. (1985). *Logically smooth density estimation*. Ph.D. dissertation, Dept. of Electrical Eng., Stanford Univ.

Barron, A.R. & Cover, T.M. (1991). Minimum complexity density estimation. *IEEE Trans. on IT, IT-37*, 1034–1054.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M.K. (1987). Occam's razor. *Information Processing Letters, 24*, 377–380.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M.K. (1989). Learnability and Vapnik-Chervonenkis dimension. *Journal of ACM, 36*, 929–965.

Cesa-Bianchi, N. (1990). Learning the distribution in the extended PAC model. *Proceedings of the First International Workshop on Algorithmic Learning Theory* (pp. 236–246), Tokyo, Japan: Japanese Society for Artificial Intelligence.

Ehrenfeucht, A., Haussler, D., Kearns, M., & Valiant, L. (1989). A general lower bound on the number of examples needed for learning. *Information and Computation, 82*, 247–251.

Fisher, R.A. (1956). *Statistical Methods and Scientific Inference*. Olyver and Boyd.

Gallager, R.G. (1986). *Inforamtion theory and reliable communication*. New York: Wiley, 1986.

Haussler, D. (1989). *Generalizing the PAC model for neural net and other learning applications*. Technical Report UCSC CRL-89-30, Univ. of California at Santa Cruz.

Haussler, D. (1990). Decision theoretic generalizations of the PAC learning model. *Proceedings of the First International Workshop on Algorithmic Learning Theory* (pp. 21–41), Tokyo, Japan: Japanese Society for Artificial Intelligence.

Haussler, D. & Long, P. (1990). *A generalization of Sauer's lemma*. Technical Report UCSC CRL-90-15, Univ. of California at Santa Cruz.

Kearns, M. & Li, M. (1988). Learning in the presence of malicious errors. *Proceedings of the 20th Annual ACM Symposium on Theory of Computing* (pp. 267–279), Chicago, IL.

Kearns, M. & Schapire, R. (1990). Efficient distribution-free learning of probabilistic concepts. *Proceedings of the 31st Symposium on Foundations of Computer Science* (pp. 382–391), St. Louis, Missouri.

Kraft, C. (1949). *A device for quantizing, grouping, and coding amplitude modulated pulses*. M.S. Thesis, Department of Electrical Engineering, MIT, Cambridge, MA.

Kraft, C. (1955). Some conditions for consistency and uniform consistency of statistical procedures. *University of California Publications in Statistics*, *2*, 125–141.

Kullback, S. (1967). A lower bound for discrimination in terms of variation. *IEEE Trans. on IT*, *IT-13*, 126–127.

Laird, P.D. (1988). Efficient unsupervised learning. *Proceedings of the First Annual Workshop on Computational Learning Theory* (pp. 91–96), Cambridge, MA: Morgan Kaufmann.

Pednault, E.P.D. (1989). Some experiments in applying inductive inference principles to surface reconstruction. *Proceedings of the 11th International Joint Conference on Artificial Intelligence* (pp. 1603–1609), Morgan Kaufmann.

Pitman, E.J.G. (1979). *Some Basic Theory for Statistical Inference*. London: Chapman and Hall.

Pitt, L. & Valiant, L.G. (1988). Computational limitation on learning from examples. *Journal of ACM*, *35*, 965–984.

Quinlan, J.R. & Rivest, R.L. (1989). Inferring decision trees using the minimum description length criterion. *Information and Computation*, *80*, 227–248.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, *11*, 416–431.

Rissanen, J. (1984). Universal coding, information, prediction, and estimation. *IEEE Trans. on IT*, *IT-30*, 629–636.

Rissanen, J. (1986). Stochastic complexity and modeling. *Annals of Statistics*, *14*, 1080–1100.

Rissanen, J. (1989). *Stochastic complexity in statistical inquiry*. World Scientific, Series in Computer Science, 15.

Rivest, R.L. (1987). Learning decision lists. *Machine Learning*, *2*, 229–246.

Schreiber, F. (1985). The Bayes Laplace statistic of the multinomial distributions. *AEU*, *39*, 293–298.

Segen, J. (1989). *From features to symbols: Learning relational shape*. In J.C. Simon, (Ed.), *Pixels to features*. Elsevier Science Publishers B.V.

Sloan, R. (1988). Types of noise in data for concept learning. In *Proceedings of the First Annual Workshop on Computational Learning Theory* (pp. 91–96), Cambridge, MA, CA: Morgan Kaufmann.

Solomonoff, R.J. (1964). A formal theory of inductive inference. *Part 1. Information and Control*, *7*, 1–22.

Valiant, L.G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 1134–1142.

Valiant, L.G. (1985). Learning disjunctions of conjunctions. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 560–566), Los Angeles, CA: Morgan Kaufmann.

Vapnik, V.N. & Chervonenkis, A.Ya. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, *XVI(2)*, 264–280.

Wallace, C.S. & Boulton, D.M. (1968). An information measure for classification. *Computer Journal*, 185–194.

Yamanishi, K. (1989). Inductive inference and learning criterion of stochastic classification rules with hierarchical parameter structures. *Proceedings of the 12th Symposium of Information Theory and Its Applications*, *2* (pp. 707–712) (in Japanese), Inuyama, Japan.

Yamanishi, K. (1990a). Inferring optimal decision lists from stochastic data using the minimum description length criterion. Presented at *1990 IEEE International Symposium on Information Theory*, San Diego, CA.

Yamanishi, K. (1990b). A learning criterion for stochastic rules. *Proceedings of the Third Annual Workshop on Computational Learning Theory* (pp. 67–81), Rochester, NY: Morgan Kaufmann.