PAC Learning with Constant-Partition Classification Noise and Applications to Decision Tree Induction

Scott E. Decatur*

MIT Laboratory for Computer Science
and
DIMACS

Abstract

We consider the problem of concept learning in Valiant's PAC learning model in which the data used for learning is noisy. Specifically, we introduce a new model of noise called *constant-partition classification noise* (CPCN) which generalizes the standard model of classification noise to allow different examples to have different rates of random misclassification. One example of CPCN type noise is data with differing rates of false positives and false negatives. We then show how to learn in the presense of CPCN for any concept class learnable by statistical queries. This set of classes includes *every* concept class known to be learnable in the presense of standard classification noise. Our model is the first such non-uniform generalization of the standard classification noise model that allows efficient learning of this wide range of concept classes.

We then examine standard methods of decision tree induction in the context of noisy data. We observe that the core of commonly used algorithms such as ID3, CART and C4.5 are not robust to CPCN noise, or even to standard classification noise. We therefore propose a simple modification to these algorithms in order to make them robust against CPCN. The modification is based on the statistical query techniques for CPCN described above.

1 Introduction

We consider learning from examples in the *probably approximately correct* (PAC) model as defined by Valiant [13]. Formal models such as this allow one to pose, and hopefully answer, precise questions regarding the nature of learning. In doing so, we hope that we then shed light on more applied problems in learning.

In the PAC model, an adversary chooses, from a specified function class, a hidden $\{0,1\}$-valued function defined over a specified domain of examples and chooses a distribution over this domain. The goal of the learner is to output, with high probability, an hypothesis with the following property: the probability is small that the hypothesis disagrees with the target function on an example chosen

*Supported by a grant from the Reed Fund through the MIT School of Science and a DIMACS postdoctoral research fellowship. Current Address: Center for Discrete Math and Computer Science (DIMACS), Rutgers University, Piscataway, NJ 08855. email: sed@dimacs.rutgers.edu randomly according to the hidden distribution. The learner gains information about the target function and distribution by interacting with an example oracle. At each request by the learner, this oracle draws an example randomly according to the hidden distribution, labels it according to the hidden target function, and returns the labelled example to the learner. There has been extensive research in providing algorithms and characterizing learnability in this model [1]. We describe this and other relevant models in more detail in Section 2.

One criticism of the PAC model has been that it assumes all of the data used for learning is noise-free. In order to combat this deficiency, variations of PAC learning have been introduced which model the types of noise that might occur in a real learning environment. One of the most widely studied models of noise in this setting has been *classification noise* [2] in which examples are mislabelled by an i.i.d. random process.

Although the classification noise model is more realistic than the noise-free model, it makes the strong assumption that every example has the same probability of being misclassified. For example, it does not capture a setting in which a random false positive is more or less likely than a random false negative. Models which relax this uniformity assumption have been investigated, but computationally efficient algorithms have not been possible in these models except for isolated concept classes. Two such models are non-uniform classification noise [5] and malicious misclassification noise [12]. The former model allows a different noise rate for each example and learning appears to be difficult for worst case assignments of the noise rates, however it becomes as easy as learning with standard classification noise if the noise rates are selected at *random*. The latter model has a fixed probability with which an adversary may affect the label of each example, but if allowed to play on a given example, the adversary may choose *not* to misclassify it. Thus, this adversary could effectively simulate a non-uniform classification noise example oracle with a different noise rate (no more than its play rate) for each example. As noted above, learning algorithms exist for these models, but in most cases they do not run in polynomial time.

In order to relax the assumption of uniform noise rates while still retaining the ability to learn efficiently, we consider settings with partial non-uniformity and propose a new model called *constant-partition classification noise* (CPCN). In this model, the labelled example space is partitioned into a constant number of regions, each of which may have a different noise rate. We consider cases in which the learner either knows the partition, or instead knows that it is one of a polynomial number of partitions. In Section 3, we precisely define the new model and give some examples of learning settings it captures. One such example, described above, is that of different rates of false positive and false negative. After defining the CPCN model, we highlight the following important "hypothesis testing" property of the model: an hypothesis with small error on *noise-free* data can be selected from a set of hypotheses by selecting the one with the fewest disagreements on a set of CPCN corrupted examples.

In order to provide efficient learning algorithms for the CPCN model, we are unable to rely solely on the above hypothesis testing property since most classes of concepts contain an exponential number of concepts and in most cases it is not known how to select the member of such an exponential sized class that has the fewest disagreements with a sample. We instead make use of a tool called the *statistical query* (SQ) learning model [6]. Kearns introduced this restricted version of noise-free PAC learning in order to address standard classification noise learning in a unified manner. In this setting, the learner may not view labelled examples, but instead may ask for estimates of the values of various statistics based on the distribution of noise-free labelled examples. Since such statistics could be accurately estimated with high probability using a sample of noise-free labelled examples, one can view this model as restricting the way in which the learner may use the PAC example oracle. Kearns showed that a statistical query algorithm for a concept class yields an example-based *classification noise tolerant* algorithm for the class. This result is quite powerful since the vast majority of classes for which noise-free PAC algorithms are known also have SQ algorithms which can be easily derived from their PAC algorithms [6]. In Section 4, we show how to learn all of these concept classes in the presence of CPCN noise by showing our main result: how to take any SQ algorithm for a class and convert it into a PAC algorithm for that class which tolerates CPCN noise. After proving this result, we highlight how varying degrees of knowledge of the noise process affect the computation time of the CPCN algorithms.

Finally, in Section 5 we examine standard methods of decision tree induction in the context of noisy data. We observe that the core of commonly used algorithms such as ID3, CART and C4.5 are not robust to CPCN noise, or even to standard classification noise. We therefore propose a simple modification to these algorithms in order to make them robust against CPCN. The modification is based on the SQ techniques for CPCN described in Section 4.

2 PAC, CN and SQ Models

In Valiant's PAC model of learning from examples [13], an adversary selects both the hidden target $\{0,1\}$-valued function f from a specified class of functions $\mathcal{F}$ and the hidden distribution D which is defined over X, the domain of f. X constitutes the set of possible examples. This set is often the Boolean hypercube $\{0,1\}^n$, in which case n is the number of Boolean attributes in each example. The learner for class $\mathcal{F}$ is given access to an example oracle $EX(f,D)$ which, when polled by the learner, returns $\langle x, f(x)\rangle$, an example x drawn randomly according to D and its correct labelling with respect to f. The learner is also given accuracy parameter $\varepsilon \in (0,1]$ and confidence parameter $\delta \in (0,1]$. In time polynomial in n, $1/\varepsilon$ and $1/\delta$, the learner must output an hypothesis h which, with probability at least $1-\delta$, has the following property: given an example x drawn randomly according to D, the probability that $f(x) \neq h(x)$ is at most ε. If h has this property, we say that it is ε-close to f on D.

Angluin and Laird introduced a variant of PAC learning called the *classification noise* (CN) model [2] in which the learner has access to a noisy example oracle $EX_{\mathrm{CN}}^{\eta}(f,D)$ instead of $EX(f,D)$. When a labelled example is requested from this oracle, an example is chosen according to the hidden distribution D, and returned. In addition, with probability $1-\eta$ the correct labelling of the example according to f is returned, while with probability η the incorrect classification is returned. While this model differs from the standard PAC model in the oracle used to interact with f and D, the learner is still required to output an hypothesis h such that with probability less than ε, h disagrees with a labelled example $\langle x, f(x)\rangle$ drawn randomly from the *noise-free* oracle $EX(f,D)$. The learner is given a bound η_b such that $0 \leq \eta \leq \eta_b < 1/2$ and the running time of the learning algorithm is allowed to be polynomial in $\frac{1}{1/2-\eta_b}$ in addition to the usual parameters. We say that a class is learnable with classification noise if it is learnable with every fixed classification noise rate η less than the information theoretic limit of $1/2$.

Learning in the PAC model may depend upon specific properties of individual examples. In contrast, learning in the *statistical query* model [6] is based only on statistical properties of large sets of examples. In the SQ model, the PAC example oracle $EX(f,D)$ is replaced by a statistics oracle $STAT(f,D)$. The learner interacts with $STAT(f,D)$ by asking it queries of the form $[\chi,\tau]$ where *query predicate* χ is a $\{0,1\}$-valued function on labelled examples and $\tau \in (0,1]$ is the *tolerance of the query*. The query is a request for the value P_χ, the probability that $\chi(x,\ell) = 1$ when $\langle x,\ell\rangle$ is a labelled example drawn randomly from the noise-free example oracle $EX(f, D)$. The statistics oracle returns an approximation $\hat{P}_\chi$ such that $|\hat{P}_\chi - P_\chi| \leq \tau$. Once again, the goal of learning remains to find an accurate hypothesis with respect to noise-free examples. The *tolerance of an SQ algorithm* is defined to be the smallest τ of all queries it makes. An important restriction of the model is that the inverse of the tolerance of an SQ algorithm must be bounded by some polynomial the PAC parameters.

3 Constant-Partition Classification Noise

As described above, we wish to relax the uniformity assumption of the standard classification noise model, but not at the expense of efficiency of learning. Thus, we consider a setting in which there is partial, but not full, non-uniformity of noise rates among examples. We formalize this setting as the *constant-partition classification noise* model (CPCN).

Central to the CPCN model is the actual partition of the noise-free labelled example space. We represent the i-th partition region by predicate π_i which evaluates to 1 when applied to noise-free labelled examples from this region. In addition to using π_i to denote the predicate over labelled examples, we shall also use it to denote the set of examples which satisfy the predicate.

Definition 1 *For partition $\Pi = \{\pi_1, \pi_2, \dots, \pi_k\}$ and noise rates $\eta = \{\eta_1, \eta_2, \dots, \eta_k\}$, we define the operation of the CPCN example oracle $EX_{CPCN}^{\Pi,\eta}(f, D)$ as follows: (1) an example x is chosen at random from X according to D, (2) i denotes the index of the partition region π_i to which $\langle x, f(x)\rangle$ belongs, and (3) with probability $1 - \eta_i$, $\langle x, f(x)\rangle$ is returned, while with probability η_i, $\langle x, \neg f(x)\rangle$ is returned.*

As mentioned above, a simple but realistic example of a setting which is captured by the CPCN model but not by the standard classification noise model is when learning occurs from examples that have probability η_- of being false positives and a different probability η_+ of being false negatives. In this case, $\eta = \{\eta_-, \eta_+\}$ and $\Pi = \{\pi_-, \pi_+\}$ where π_- is the set of all negatively labelled examples and π_+ is the set of all positively labelled examples. Specifically, $\pi_-(x, \ell) = 1 - \ell$ and $\pi_+(x, \ell) = \ell$.

Note that the learner may not know the partition region to which a given noisy labelled example belongs, since the partition may depend (as in the above example) on the true noise-free label. Nonetheless, the noise process is well-defined, and furthermore, the learner can know the partition *functions*: $\{\pi_i(x, \ell)\}$. We also consider cases where the learner does not know Π (*i.e.* the partition functions of Π), but does know that Π belongs to some known polynomial sized set of partitions. An example of such a setting is where learning occurs from examples whose error rate depends on whether the number of Boolean attributes that are TRUE in the example is few, medium or many. This setting may arise when more TRUE attributes corresponds to more complex examples, resulting in more likely errors. If the cutoffs for few, medium and many are known in advance, then the partition is clear. Otherwise, there are at most $O(n^2)$ pairs of cutoffs possible when there are n Boolean attributes, which in turn define the partition functions for $O(n^2)$ partitions. We first define learnability in the CPCN model as taking the partition function as input, and address the relaxation to unknown partitions in Section 4.

Definition 2 *A class $\mathcal{F}$ is said to be* CPCN learnable *if there exists an algorithm which in addition to the standard PAC inputs takes as input partition predicates Π of the labelled example space and learns*

$\mathcal{F}$ by drawing labelled examples from $EX^{\Pi,\eta}_{CPCN}(f,D)$. The algorithm may run in time polynomial in the standard PAC parameters as well as $1/\gamma$ where $\gamma = \min_i(1/2 - \eta_i)$.[1]

Although we may have reason to believe that a partition Π describes which examples have similar noise properties, it is less likely that we will know the particular noise rate in each of these regions (*e.g.* data has a different rate of false positives than false negatives, but these two rates are not known *a priori*). For that reason, we assume that the learner does not know the noise rates, but instead knows only an upper bound on all of the noise rates $\eta_b < 1/2$, or equivalently a lower bound $\gamma_b > 0$ on γ.

A basic, but important, property of the CPCN model is the relative performance of hypotheses on data drawn from $EX^{\Pi,\eta}_{CPCN}$. Specifically, *hypothesis testing* in the CPCN model refers to the use of a sample of CPCN data to determine which hypothesis from a specified set of hypotheses has relatively small error on *noise-free* data. Sloan [12] shows the following theorem for hypothesis testing in the malicious misclassification noise model:

Theorem 1 (Sloan) *Given N hypotheses, one of which has true error at most $\varepsilon/2$, then with probability at least $1-\delta$, the hypothesis with the smallest empirical error rate on a sample of size $\frac{2}{\varepsilon^2(1-2\eta)^2}\ln\frac{2N}{\delta}$ from a malicious misclassification oracle will have true error at most ε.*

As the CPCN example oracle could be simulated by the adversarial example oracle of Sloan's model, the hypothesis testing result for that model applies to the CPCN model. Thus, we may draw a sample of size $O(\frac{\log(N/\delta)}{\varepsilon^2(1-2\eta_b)^2}) = O(\frac{1}{\varepsilon^2\gamma^2}\log(N/\delta))$ from $EX^{\Pi,\eta}_{CPCN}$ and simply choose the hypothesis that has the fewest disagreements on this sample. In fact, using an analysis similar to that used by Laird for standard classification noise [8], we can show that hypothesis testing can be performed in the CPCN model using a sample with only a linear dependence on $1/\varepsilon$. This improved hypothesis testing is stated formally in the following theorem and is proven in the full paper.

Theorem 2 *Given N hypotheses, one of which has true error at most $\varepsilon/2$, then with probability at least $1-\delta$, the hypothesis with the smallest empirical error rate on a sample of size $O(\frac{1}{\varepsilon\gamma^2}\log\frac{N}{\delta})$ from a CPCN oracle will have true error at most ε.*

In Section 4, the ability to perform hypothesis testing will be relied on to learn without knowledge of the individual noise rates. As described at the end of Section 4, it is also useful in settings where one does not know the exact partition Π, but does know that Π belongs to some polynomial sized set of partitions.

4 CPCN Learning from SQ Algorithms

This section contains our main result on the use of statistical query algorithms to construct CPCN algorithms.

Theorem 3 *If concept class $\mathcal{F}$ is learnable by statistical queries, then $\mathcal{F}$ is PAC learnable in the presense of known constant-partition classification noise.*

[1] The dependence on $1/\gamma$ is required since lower bounds quadratic in $1/\gamma$ on the number of examples needed for learning with *standard* classification noise [11] also hold in the CPCN model.

Proof: The proof of this theorem is constructive and entails a simulation of the statistical query algorithm in a way that is robust to the constant-partition classification noise. Let the noise process be defined by: (1) $\Pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$, the set of predicates on labelled examples which partition the labelled example space into a constant k regions, and (2) $\eta = \{\eta_1, \eta_2, \ldots, \eta_k\}$, the respective noise rates in these regions. Let $\gamma = \min_i(1/2 - \eta_i)$.

Our strategy is to simulate the SQ algorithm by simulating each query to the statistics oracle using only data from $EX_{CPCN}^{\Pi,\eta}$. In doing so, we output the same hypothesis that would have been output had one interacted directly with the statistics oracle. We begin by describing how one would estimate P_χ for a single query $[\chi, \tau]$ in the SQ algorithm.

In a sample drawn from $EX_{CPCN}^{\Pi,\eta}$, a labelled example $\langle x, \ell \rangle$ which satisfies χ might be due to the selection of $\langle x, \ell \rangle$ to which noise was not applied, or the selection of $\langle x, \neg\ell \rangle$ to which noise was applied. In order to determine the relative likelihood of these two events, it is important to know the partition to which $\langle x, \ell \rangle$ belongs, as well as the partition to which $\langle x, \neg\ell \rangle$ belongs. We therefore define

$$\overline{\pi}_i(x, \ell) = \pi_i(x, \neg\ell)$$

and consider Π', the set of k^2 predicates $\{(\pi_i \wedge \overline{\pi}_j)\}_{i,j}$. Note that if Π constitutes a partition of the labelled example space, then so does Π'. By refining the query χ in terms of Π':

$$\chi_{i,j}(x, \ell) = \chi(x, \ell) \wedge \pi_i(x, \ell) \wedge \overline{\pi}_j(x, \ell),$$

we may decompose P_χ as follows:

$$P_\chi = \sum_{i=1}^{k} \sum_{j=1}^{k} P_{\chi_{i,j}}. \tag{1}$$

Our strategy is therefore to determine each $P_{\chi_{i,j}}$ with tolerance at most τ/k^2, and use their sum as an estimate of P_χ that is within $\pm\tau$. In order to estimate each $P_{\chi_{i,j}}$ using data from $EX_{CPCN}^{\Pi,\eta}$, we further define

$$\begin{aligned} \overline{\chi}(x, \ell) &= \chi(x, \neg\ell) \\ \overline{\chi}_{i,j}(x, \ell) &= \overline{\chi}(x, \ell) \wedge \overline{\pi}_i(x, \ell) \wedge \pi_j(x, \ell). \end{aligned}$$

We know that a labelled example which satisfies $\chi_{i,j}$ belongs to partition π_i and therefore has noise rate η_i, while a labelled example which satisfies $\overline{\chi}_{i,j}$ belongs to partition π_j and therefore has noise rate η_j. We use the notation P_z^* to denote the probability of drawing a *noisy example from* $EX_{CPCN}^{\Pi,\eta}$ which satisfies predicate z. By the definition of $EX_{CPCN}^{\Pi,\eta}$, we then have:

$$\begin{aligned} P^*_{\chi_{i,j}} &= (1 - \eta_i) P_{\chi_{i,j}} + \eta_j P_{\overline{\chi}_{i,j}} \\ P^*_{\overline{\chi}_{i,j}} &= (1 - \eta_j) P_{\overline{\chi}_{i,j}} + \eta_i P_{\chi_{i,j}}. \end{aligned}$$

Rearranging terms and solving for $P_{\chi_{i,j}}$, we have:

$$P_{\chi_{i,j}} = \frac{P^*_{\chi_{i,j}}(1 - \eta_j) - P^*_{\overline{\chi}_{i,j}} \eta_j}{1 - \eta_i - \eta_j}. \tag{2}$$

Simple algebra may then be used to verify that in order to use Equation (2) to estimate $P_{\chi_{i,j}}$ to within $\pm\tau/k^2$, it is sufficient to have estimates of $P^*_{\chi_{i,j}}$, $P^*_{\overline{\chi}_{i,j}}$, η_i and η_j each to within plus or minus:

$$\tau'_{i,j} = \frac{\tau}{k^2} \cdot \frac{1 - \eta_i - \eta_j}{3} \cdot \frac{1}{2} \cdot \frac{1}{3} = \frac{\tau(1 - \eta_i - \eta_j)}{18k^2}.$$

The estimates of each $P^*_{\chi_{i,j}}$ and $P^*_{\overline{\chi}_{i,j}}$ are obtained by sampling from $EX^{\Pi,\eta}_{CPCN}$. Using standard Chernoff bounds, we see that each can be estimated to within $\pm\tau'_{i,j}$ with high probability using $O((\tau'_{i,j})^{-2})$ such examples.

Our estimates of noise rates $\{\eta_i\}$ come from "guesses." Since $(1 - \eta_i - \eta_j) \geq 2\gamma$ for all i, j, we have $\tau'_{i,j} \geq \tau_0\gamma/9k^2 \triangleq \tau_*$ where τ_0 is the smallest τ used in the SQ algorithm. Clearly, there exists a set of $O(1/\tau_*)$ guesses evenly spaced between 0 and $1/2$ such that for any η_i, at least one of these guesses is within τ_* of η_i. Thus, there exist $G = O(1/\tau_*^k)$ k-tuples of noise rate guesses for $(\eta_1, \ldots, \eta_k)$ such that for at least one of these k-tuples, each component of the k-tuple is within τ_* of its respective true noise rate. We therefore simulate the SQ algorithm G times, each with a different k-tuple of noise rate guesses which are used throughout that simulation. For at least one such simulation, all components of every application of Equation (2) are sufficiently accurate with high probability, resulting in a good hypothesis. Finally, we use hypothesis testing to select an ε-good hypothesis from the set of G candidate hypotheses.

The number of examples from $EX^{\Pi,\eta}_{CPCN}$ required for the above described procedure can be determined as in other uses of statistical queries [4]. For example, if the SQ algorithm being simulated makes N queries and has minimum tolerance τ_0, then

$$O(\frac{N}{\tau_0^2\gamma^2}\log(\frac{N}{\delta}) + \frac{1}{\varepsilon\gamma^2}\log\log\frac{1}{\gamma})$$

examples suffices. Still more efficient sample complexities often can be achieved by considering the class $\mathcal{Q}$ of queries χ used by the SQ algorithm, as opposed to the number of queries used [6]. In the full paper, we define the relevant measures on the class $\mathcal{Q}$ and the sample complexities achieved. □

As mentioned above, we also make use of hypothesis testing to learn when the partition is not known exactly, but instead known to belong to a polynomial sized set of partitions. Given an algorithm for CPCN learning with known partition, one can use this algorithm as a subroutine in which each call to the algorithm uses a different one of the polynomially many candidate partitions. The run using the correct partition will succeed with high probability and output a good hypothesis. Hypothesis testing may then be performed on the set of hypotheses returned by the calls of the algorithm in order to determine a sufficiently good output hypothesis.

In order to characterize the complexity of the SQ simulation in terms of the noise process, we let $P = \{\Pi_i\}$ be the set of candidate partitions, each of which divides the labelled example space into k regions, and let $j \leq k$ be the number of noise rates which are completely unknown or for which we do not have a sufficiently accurate estimate (*i.e.* within $\pm\tau_* = \tau_0\gamma/9k^2$). Then our SQ simulation runs the base SQ algorithm $H = |P| \cdot (1/\tau_*)^j$ times and performs hypothesis testing on the H hypotheses returned. This exponential dependence on j leads to the limitation of *constant*-sized partitions.

5 Noisy Decision Tree Induction

In this section, we examine the behavior of learning algorithms that construct decision trees and we do so with respect to both standard classification noise and CPCN. In particular, we examine the *information gain* criteria central to common applied machine learning algorithms for decisions trees such as ID3, CART and C4.5 [10].

A decision tree is a classification rule which has attribute tests at its internal nodes and classifications at its leaves. The classification of an example by such a tree is the classification of the leaf reached by starting at the root and following the branches that correspond to the values the example has for each attribute test. When refining a leaf into an internal node during the building of a decision tree, the information gain criteria attempts to find the literal that when split upon provides the largest information gain, or equivalently the literal with the largest mutual information between itself and the labelling at the current leaf. While the components of the information gain formula are usually written as cardinalities of various sets, one could instead write them as probabilities with respect to uniform selection from these sets. We do so in Equations (3) through (5) below, where x denotes an example, ℓ denotes a label, S and T denote sets of examples, z denotes an m-valued attribute, and $z(x) = i$ denotes that attribute z of example x has value i.

$$\mathsf{info}(S) = -\sum_{b=0,1} \Pr(\ell = b | x \in S) \cdot \log_2(\Pr(\ell = b | x \in S)) \tag{3}$$

$$\mathsf{info}_z(T) = -\sum_{i=1}^{m} \Pr(z(x) = i | x \in T) \cdot \mathsf{info}(\{x : x \in T \wedge z(x) = i\}) \tag{4}$$

$$\mathsf{gain}(z; T) = \mathsf{info}(T) - \mathsf{info}_z(T) \tag{5}$$

By examining the behavior of the information gain function in classification noise models such as standard CN and CPCN, we show that it has the following property: there exist labelling rules and distributions of examples in which the attribute test selected in the presence of noisy data is *different* than the attribute test that would have been selected in the absence of noise.

In Figure 1, we give an example of a distribution of labelled examples within a set T which exhibits this property. The regions of T denote whether each attribute is true or false for those examples. Within each region, the values shown are the probabilities of drawing an example that belongs to this region and is labelled positive (or respectively, negative). Note that for any fixed node with examples T, minimizing $\mathsf{info}_z(T)$ over z is identical to maximizing $\mathsf{gain}(z; T)$ over z. Thus, we can focus on the values of $\mathsf{info}_z(T)$ in order to determine which attribute test is selected. In the absence of noise, the above formulae yield $\mathsf{info}_{X_1}(T) = 0.705$ and $\mathsf{info}_{X_2}(T) = 0.707$ and therefore the preferred attribute to test on is X_1. But in the presence of standard classification noise with misclassification rate $\eta = 0.05$, we have $\mathsf{info}_{X_1}(T) = 0.773$ and $\mathsf{info}_{X_2}(T) = 0.767$, and therefore the attribute test selected is X_2.

In order to make algorithms that employ the information gain criteria robust against CPCN (and therefore standard classification noise as well), we simply observe that such algorithms are SQ-like in that they base their decisions on the values of statistics. Thus, we may perform the simulation described in the formal PAC setting of Section 4. In this simulation, if we know the partition and the noise rates, then we can exactly reverse the noise process by taking the available CPCN data and using it along with Equations (1) and (2) to compute the noise-free statistics as input to the information gain formulae. If instead, we do not know the noise rates and/or the partition, then we can wrap a procedure for guessing them around the decision tree algorithm and perform hypothesis testing. This technique of wrapping guess-and-test around a primary algorithm is common in machine learning: the determination of the partition and noise rates is a form of *model selection* and hypothesis testing is often performed using *cross-validation* or a related method (see for example [3, 7]).

Although this technique for adapting real algorithms such as C4.5 *provably* reverses the effects of standard classification noise or even CPCN, it is not clear whether these types of noise are significant components of the noise found in real data. In order to answer this question, we have adapted the C4.5 algorithm [10] for CPCN noise and are using it on classification problems in the UCI Machine Learning Database [9]. In the full paper, we will report results from these experiments.

	X_2	$\overline{X}_2$
X_1	+: 0.16 −: 0.64	+: 0.05 −: 0.05
$\overline{X}_1$	+: 0.00 −: 0.10	+: 0.00 −: 0.00

Figure 1: Distribution on T where gain is tricked by classification noise.

Acknowledgements

Thanks to Javed Aslam and Dana Ron for helpful discussions.

References

[1] Dana Angluin. Computational learning theory: Survey and selected bibliography. In *Proceedings of the* 24^{th} *Annual ACM Symposium on the Theory of Computing*, 1992.

[2] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[3] L. Breiman, J. H. Freidman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

[4] Scott Decatur. *Efficient Learning from Faulty Data*. PhD thesis, Harvard University, 1995.

[5] Scott Decatur. PAC learning with random non-uniform classification noise. Manuscript, 1995.

[6] Michael Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the* 25^{th} *Annual ACM Symposium on the Theory of Computing*, pages 392–401, San Diego, 1993.

[7] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the* 14^{th} *International Joint Conference on Artificial Intelligence*, pages 1137–1143. Morgan Kaufmann, 1995.

[8] Philip D. Laird. *Learning from Good and Bad Data*. Kluwer international series in engineering and computer science. Kluwer Academic Publishers, Boston, 1988.

[9] C.J. Merz and P.M. Murphy. UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1996.

[10] J. Ross Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[11] Hans Ulrich Simon. General bounds on the number of examples needed for learning probabilistic concepts. In *Proceedings of the Sixth Annual ACM Workshop on Computational Learning Theory*, pages 402–411. ACM Press, 1993.

[12] Robert H. Sloan. Four types of noise in data for PAC learning. *Information Processing Letters*, 54(3):157–162, 12 May 1995.

[13] Leslie Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.