CN=CPCN

Liva Ralaivola LIVA.RALAIVOLA@LIF.UNIV-MRS.FR
François Denis FRANCOIS.DENIS@LIF.UNIV-MRS.FR
Christophe N. Magnan CHRISTOPHE.MAGNAN@LIF.UNIV-MRS.FR
Laboratoire d'Informatique Fondamentale de Marseille
UMR 6166 CNRS, 39, rue F. Joliot-Curie, F-13453 Marseille cedex 13, France

Abstract

We address the issue of the learnability of concept classes under three classification noise models in the *probably approximately correct* framework.

After introducing the Class-Conditional Classification Noise (CCCN) model, we investigate the problem of the learnability of concept classes under this particular setting and we show that concept classes that are learnable under the well-known uniform classification noise (CN) setting are also CCCN-learnable, which gives CN=CCCN.

We then use this result to prove the equality between the set of concept classes that are CN-learnable and the set of concept classes that are learnable in the Constant Partition Classification Noise (CPCN) setting, or, in other words, we show that CN=CPCN.

1. Introduction

This paper presents a study in the probably approximately correct (PAC) framework. In particular, we investigate the equality of concept classes in different classification noise settings from the learnability standpoint.

More precisely, we study three different noise settings: the *uniform classification noise setting* CN (Angluin & Laird, 1988), the *class-conditional classification noise setting* CCCN and the *constant partition classification noise setting* CPCN (Decatur, 1997). The second setting is a particular case of the latter and it is characterized by a by a noise process that flips the label of an example according to unifom classification noise processes defined on each (positive or negative) class. This setting is therefore a generalization of the uniform classification noise setting where noise is added independently of the class of the examples.

Our first contribution is the formal proof that CN = CCCN, that is, the concept classes that are learnable (in the PAC sense) under the CN framework (these classes are said to be CN-learnable) are also learnable under the CCCN (they are therefore CCCN-learnable) framework, and conversely. The idea to prove this result is that it is possible to bring a CCCN learning problem down to a CN learning problem by an appropriate addition of noise to the labeled examples of the CCCN problem.

Our second contribution is the proof that CN = CPCN, that is, the concept classes that are CN-learnable are CPCN-learnable, and conversely. The underlying idea of the proof is that a CPCN learning problem can be decomposed into several CCCN learning problems.

The paper is organized as follows. Section 2 briefly recalls the notion of PAC-learnability and formally presents and/or recalls the different noise settings together with the corresponding definitions of CN-learnability, CCCN-learnability and CPCN-learnability. Section 3 gives the proof of CN=CCCN while section 4 develops that of CN=CPCN. A short discussion on possible relaxation of noise constraints is provided in section 5.

2. Preliminaries

2.1. Learning in the PAC Framework

In the classical PAC learning framework, the problem of concept learning can be stated as follows (Valiant, 1984). Let $\mathcal{X}$ be a space (e.g. $\mathbb{R}^n$ or $\{0,1\}^d$), subsequently referred to as the *input space*. Let c be some *concept* from a *concept class* $\mathcal{C}$ (basically, $\mathcal{C}$ is a subset

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

of $\mathcal{X}$) and D some fixed but unknown distribution on $\mathcal{X}$ from $\mathcal{D}$, the set of all the distributions on $\mathcal{X}$. The task of learning is that of identifying c given access only to a *sampling oracle* $EX(c, D)$, such that each call to $EX(c, D)$ outputs a pair $\langle \mathbf{x}, t(\mathbf{x}) \rangle$, with $\mathbf{x} \in \mathcal{X}$ drawn randomly according to D and $t(\mathbf{x}) = 1$ if $\mathbf{x} \in c$ and $t(\mathbf{x}) = 0$ otherwise (i.e. t is the indicator function of c). $\mathcal{C}$ is said to be *efficiently PAC-learnable*, if, there is an algorithm $\mathcal{A}$ such that for every concept c in $\mathcal{C}$, for every distribution D over $\mathcal{X}$, for every $\varepsilon > 0$ and for every $\delta > 0$, $\mathcal{A}$, when given access to $EX(c, D)$, outputs with probability at least $1 - \delta$ a hypothesis $h \in \mathcal{H}$, where $\mathcal{H}$ is a representation class over $\mathcal{X}$, such that the probability $\text{err}_D(h) := P_{\mathbf{x} \sim D}(h(\mathbf{x}) \neq t(\mathbf{x}))$ of disagreement between h and t on instances randomly drawn from D is lower than ε (Kearns & Vazirani, 1994); $\delta > 0$ is referred to as the *confidence* parameter (although the confidence is actually $1 - \delta$), $\varepsilon > 0$ as the *precision* and $\text{err}_D(h)$ is the *error* of h. There must be two polynomials $p(\cdot, \cdot)$ and $q(\cdot, \cdot)$, such that in order to draw a hypothesis h, $\mathcal{A}$ needs at most $p(\frac{1}{\varepsilon}, \frac{1}{\delta})$ training examples and it runs in at most $q(\frac{1}{\varepsilon}, \frac{1}{\delta})$ time. These two polynomials should also take as another argument the size of the concept c to be learned, but as it will not play any explicit role in our discussion, we have decided for sake of clarity not to mention it in the sample size and time requirements.

2.2. CN, CPCN and CCCN Learnability

In the framework of uniform Classification Noise (CN) concept learning (Angluin & Laird, 1988), the oracle to which the learning procedure has access is defined as follows (Angluin & Laird, 1988).

Definition 1 (CN oracle $EX_{\text{CN}}(c, D)$). *Let $\eta \in [0; 1]$. Given $c \in \mathcal{C}$ and $D \in \mathcal{D}$, the uniform Classification Noise oracle $EX_{\text{CN}}(c, D)$ outputs a pair $\langle \mathbf{x}, t^\eta(\mathbf{x}) \rangle$ according to the following procedure:*

(a) $\mathbf{x}$ *is drawn randomly according to D;*

(b) $t^\eta(\mathbf{x})$ *is set as*

$$t^\eta(\mathbf{x}) := \begin{cases} t(\mathbf{x}) \text{ with prob. } 1 - \eta \\ \neg t(\mathbf{x}) \text{ with prob. } \eta. \end{cases}$$

The notion of CN-learnability, defined by Angluin and Laird (1988) readily follows.

Definition 2 (CN-learnability). *A concept class $\mathcal{C}$ is efficiently CN-learnable by representation class $\mathcal{H}$* iff *there exist an algorithm $\mathcal{A}$ and polynomials $p(\cdot, \cdot, \cdot)$ and $q(\cdot, \cdot, \cdot)$ such that for any $c \in \mathcal{C}$, for any $D \in \mathcal{D}$, for any $\varepsilon > 0$, for any $\delta \in]0; 1]$ and for any $\eta \in [0; 0.5[$, when given access to $EX^\eta_{\text{CN}}(c, D)$ and given inputs ε, δ and an upper bound $\eta_b < 0.5$ on η, $\mathcal{A}$ outputs with probability at least $1 - \delta$ a hypothesis $h \in \mathcal{H}$ such that $err_D(h) \leq \varepsilon$.*

To output such an hypothesis $\mathcal{A}$ requires at most $p(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta_b})$ training samples and it runs in $q(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta_b})$ time.

Remark 1. *Here, we have assumed the knowledge of an upper bound η_b on the actual noise η. As stated in (Angluin & Laird, 1988) and (Kearns & Vazirani, 1994), this assumption is not restrictive since it is possible to guess a value for η_b when none is provided. The classes of concepts that can be CN-learned with a provided bound η_b are therefore exactly the same as the ones that can be CN-learned without any knowledge on an upperbound on η.*

As for CPCN-learnability, introduced in (Decatur, 1997), this setting assumes a set of partition functions $\Pi = \{\pi_1, \ldots, \pi_k\}$ defined on the labeled space $\mathcal{X} \times \mathcal{Y}$ and taking values in $\{0, 1\}$ such that $\sum_{i=1}^k \pi_i(\langle \mathbf{x}, y \rangle) = 1$ for any pair $\langle \mathbf{x}, y \rangle$ from $\mathcal{X} \times \mathcal{Y}$ and it assumes a CPCN oracle as defined by Decatur (1997).

Definition 3 (CPCN oracle $EX^{\Pi, \boldsymbol{\eta}}_{\text{CPCN}}(c, D)$). *Let $\Pi = \{\pi_1, \ldots, \pi_k\}$ a set of partition functions over $\mathcal{X} \times \mathcal{Y}$ and $\boldsymbol{\eta} = [\eta_1 \ldots \eta_k]$, with $\eta_i \in [0; 1]$. Given $c \in \mathcal{C}$ and $D \in \mathcal{D}$, the CPCN oracle $EX^{\Pi, \boldsymbol{\eta}}_{\text{CPCN}}(c, D)$ outputs a labeled example $\langle \mathbf{x}, t^{\boldsymbol{\eta}}(\mathbf{x}) \rangle$ as follows:*

(a) $\mathbf{x}$ *is drawn according to D;*

(b) if i is the index such that $\pi_i(\langle \mathbf{x}, t(\mathbf{x}) \rangle) = 1$ then

$$t^{\boldsymbol{\eta}}(\mathbf{x}) := \begin{cases} t(\mathbf{x}) \text{ with prob. } 1 - \eta_i \\ \neg t(\mathbf{x}) \text{ with prob. } \eta_i. \end{cases}$$

The next definition is that of CPCN-learnability (Decatur, 1997).

Definition 4 (CPCN-learnability). *A concept class $\mathcal{C}$ is efficiently CPCN-learnable by representation class $\mathcal{H}$* iff *there exist an algorithm $\mathcal{A}$ and polynomials $p(\cdot, \cdot, \cdot)$ and $q(\cdot, \cdot, \cdot)$ such that for any set $\Pi = \{\pi_1, \ldots, \pi_k\}$ of partition functions, for any $\boldsymbol{\eta} = [\eta_1 \cdots \eta_k]$, with $\eta_i \in [0; 1/2[$, for any $c \in \mathcal{C}$, for any $D \in \mathcal{D}$, for any $\varepsilon > 0$ and for any $\delta \in]0; 1]$, when given access to $EX^{\Pi, \boldsymbol{\eta}}_{\text{CPCN}}(c, D)$ and given inputs ε, δ and an upper bound $\eta_b < 0.5$ on the noise rates η_i, $\mathcal{A}$ outputs with probability at least $1 - \delta$ a hypothesis $h \in \mathcal{H}$ such that $err_D(h) \leq \varepsilon$.*

To output such an hypothesis $\mathcal{A}$ requires at most $p(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta_b})$ training samples and it runs in $q(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta_b})$ time.

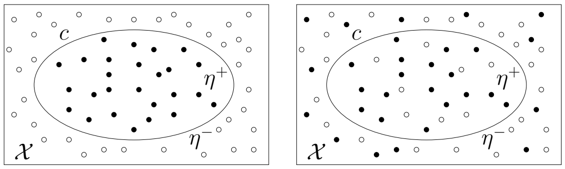

Figure 1. Left: classical (noise free) concept learning setting showing 26 positive examples (black discs) and 37 negative examples (white discs); $\eta^1 = 0$ and $\eta^0 = 0$. Right: Class-Conditional Noise concept learning setting; the values of η^1 and η^0 might be $\eta^1 = 8/26$ and $\eta^0 = 13/37$.

In order to prove our main result, that is, CN = CPCN, we will focus on the specific CPCN case where $\Pi = \{\pi_+, \pi_-\}$ and $\boldsymbol{\eta} = [\eta^1\ \eta^0]$ with $\pi_+(\mathbf{x}, y) = y$ and $\pi_-(\mathbf{x}, y) = 1 - y$. A CPCN oracle $EX_{\text{CPCN}}^{\Pi,\boldsymbol{\eta}}$ defined along this setting corresponds to the case where different classification noises are applied to positive and negative examples, as illustrated on Figure 1. From now on, we refer to the problem of learning in this particular framework, i.e., with $\Pi = \{\pi_+, \pi_-\}$, $\boldsymbol{\eta} = [\eta^1\ \eta^0]$ and the corresponding CPCN oracle, as the problem of learning under Class-Conditional Classification Noise (CCCN); the corresponding oracle will hence be denoted as $EX_{\text{CCCN}}^{\boldsymbol{\eta}}$. CCCN-learnability is obviously defined as in Definition 4.

3. CN=CCCN

The main theorem of this section states that the class CN of concepts that are learnable under the uniform classification noise model is the same as the class CCCN of concepts that are learnable under the class-conditional classification noise model:

Theorem 1. *CCCN=CN*

CCCN $\subseteq$ CN is obvious: if $c \in \mathcal{C}$ is a concept from a class $\mathcal{C}$ that is CCCN-learnable with any noise vector $\boldsymbol{\eta} = [\eta^1\ \eta^0]$ given a noise upper bound on η_b then it is still learnable when $\eta^1 = \eta^0$, i.e. it is CN-learnable (with the same noise upper bound η_b).

3.1. Sketch of the Proof

The proof of Theorem 1 proceeds in three steps. First, we show (Lemma 1) that from noisy oracle $EX_{\text{CCCN}}^{\boldsymbol{\eta}}$, it is possible to construct another CCCN noisy oracle $EX_{\text{CCCN}}^{\bar{\boldsymbol{\eta}}}$ whose noise vector $\bar{\boldsymbol{\eta}} = [\bar{\eta}^1\ \bar{\eta}^0]$ depends on two 'renoising' control parameters ρ and s. In addition, we observe that there exists a specific pair (ρ_{opt}, s_{opt}) of values that allows to turn a CCCN learning problem into CN learning problem.

Secondly, we show (Lemma 2) that it suffices to know a sufficiently accurate approximation ρ to ρ_{opt} (with the correct setting of the corresponding s) to 'almost' meet the requirements for PAC-learnability from the CCCN-oracle.

Algorithm 1 FlipLabel

```
Input: $\rho \in [0;1]$, $s \in \{0,1\}$, $t \in \{0,1\}$
Output: $t^{\rho,s} \in \{0,1\}$

  Draw a random number $r$ uniformly in $[0;1]$
  if $s = t$ then
    $t^{\rho,s} := t$
  else
    if $r \leq \rho$ then
      $t^{\rho,s} := 1 - t$
    else
      $t^{\rho,s} := t$
    end if
  end if
  return $t^{\rho,s}$
```

Then, it is proved that knowing $\eta_b < 0.5$ such that $\eta^1, \eta^0 \leq \eta_b$ makes it possible to learn any CCCN concept that is CN-learnable (Proposition 1). This concludes the proof of Theorem 1.

3.2. Formal Proof

Lemma 1. *Let $c \in \mathcal{C}$ and $D \in \mathcal{D}$. Let $EX_{\text{CCCN}}^{\boldsymbol{\eta}}(c, D)$ be the CCCN oracle with noise vector $\boldsymbol{\eta} = [\eta^1\ \eta^0]$ with $\eta^1, \eta^0 \in [0;1]$. Given parameters $\rho \in [0;1]$ and $s \in \{0,1\}$, the procedure that returns a pair $\langle \mathbf{x}, t^{\bar{\boldsymbol{\eta}}}(\mathbf{x})\rangle$ by (1) polling a labeled example $\langle \mathbf{x}, t^{\boldsymbol{\eta}}(\mathbf{x})\rangle$ from $EX_{\text{CCCN}}^{\boldsymbol{\eta}}$ and (2) setting $t^{\bar{\boldsymbol{\eta}}}(\mathbf{x})$ to* FlipLabel$(\rho, s, t^{\boldsymbol{\eta}}(\mathbf{x}))$ *(cf. Algorithm 1), simulates a call to a CCCN-oracle $EX_{\text{CCCN}}^{\bar{\boldsymbol{\eta}}}(c, D)$ of noise vector $\bar{\boldsymbol{\eta}} = [\bar{\eta}^1\ \bar{\eta}^0]$ with $\bar{\eta}^1, \bar{\eta}^0 \in [0;1]$ and such that*

$$\bar{\eta}^1 = (1-\rho)\eta^1 + (1-s)\rho \ \ \textit{and} \ \ \bar{\eta}^0 = (1-\rho)\eta^0 + s\rho.$$

Proof. Let $c \in \mathcal{C}$, $D \in \mathcal{D}$, $\rho \in [0;1]$ and, for sake of exposition, suppose that $s = 1$.

The procedure described in the lemma together with the way FlipLabel is defined (cf. Algorithm 1) are such that $P(t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}) = 1 | t^{\boldsymbol{\eta}}(\mathbf{x}) = 1) = 1$ and $P(t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}) = 1 | t^{\boldsymbol{\eta}}(\mathbf{x}) = 0) = \rho$. Therefore, the probabilities of flipping the class $t(\mathbf{x})$ of a random example $\mathbf{x}$ to the opposite class $1 - t(\mathbf{x})$ are given by (dropping the dependence on $\mathbf{x}$)

$$\begin{aligned}
\bar{\eta}^1 &= P(t^{\bar{\boldsymbol{\eta}}} = 0 | t = 1) \\
&= P(t^{\bar{\boldsymbol{\eta}}} = 0, t^{\boldsymbol{\eta}} = 0 | t = 1) + P(t^{\bar{\boldsymbol{\eta}}} = 0, t^{\boldsymbol{\eta}} = 1 | t = 1) \\
&= P(t^{\bar{\boldsymbol{\eta}}} = 0 | t^{\boldsymbol{\eta}} = 0) P(t^{\boldsymbol{\eta}} = 0 | t = 1) \\
&\qquad + P(t^{\bar{\boldsymbol{\eta}}} = 0 | t^{\boldsymbol{\eta}} = 1) P(t^{\boldsymbol{\eta}} = 1 | t = 1) \\
&= (1-\rho)\eta^1,
\end{aligned}$$

and

$$\begin{aligned}
\bar{\eta}^0 &= P(t^{\bar{\boldsymbol{\eta}}} = 1 | t = 0) \\
&= P(t^{\bar{\boldsymbol{\eta}}} = 1, t^{\boldsymbol{\eta}} = 1 | t = 0) + P(t^{\bar{\boldsymbol{\eta}}} = 1, t^{\boldsymbol{\eta}} = 0 | t = 0) \\
&= P(t^{\bar{\boldsymbol{\eta}}} = 1 | t^{\boldsymbol{\eta}} = 1) P(t^{\boldsymbol{\eta}} = 1 | t = 0) \\
&\qquad + P(t^{\bar{\boldsymbol{\eta}}} = 1 | t^{\boldsymbol{\eta}} = 0) P(t^{\boldsymbol{\eta}} = 0 | t = 0) \\
&= \rho + \eta^0 (1 - \rho),
\end{aligned}$$

which corresponds to the expressions for $\bar{\eta}^1$ and $\bar{\eta}^0$ provided in the lemma. It is straightforward to check that when $s = 0$ we do also recover these expressions.

Checking that $\bar{\eta}^1, \bar{\eta}^0$ are in $[0; 1]$ is straightforward: both $\bar{\eta}^1$ and $\bar{\eta}^0$ are bounded from above by $(1 - \rho) \max(\eta^1, \eta^0) + \rho$, which, since $1 - \rho \geq 0$ and $\max(\eta^1, \eta^0) \in [0; 1]$, is upper bounded by $(1 - \rho) + \rho = 1$.

□

Remark 2. *This lemma has the direct consequence that it is possible to get a CN oracle from a CCCN oracle as soon as the noise parameters of the CCCN oracle are known. Indeed, if $EX^{\boldsymbol{\eta}}_{\text{CCCN}}(c, D)$ is a CCCN oracle of known noise vector $\boldsymbol{\eta} = [\eta^1 \ \eta^0]$ then using $\rho_{opt} := \frac{|\eta^1 - \eta^0|}{1 + |\eta^1 - \eta^0|}$ and setting $s_{opt} := 1$ if $\eta^1 > \eta^0$ and 0 otherwise allows to obtain a CN oracle. This CN oracle has its noise equal to $\eta_{opt} := \bar{\eta}^1 = \bar{\eta}^0 = \frac{\max(\eta^1, \eta^0)}{1 + |\eta^1 - \eta^0|}$.*

Remark 3. *If only an upper bound $\eta_b < 0.5$ is known on the noise rates η^1 and η^0 of a CCCN oracle $EX^{\boldsymbol{\eta}}_{\text{CCCN}}(c, D)$ then it is straightforward to see that $\rho_{opt} \leq \eta_b$ and that the noise of the CN oracle obtained from $EX^{\boldsymbol{\eta}}_{\text{CCCN}}(c, D)$ by adding noise to one of the classes is also upper bounded by η_b.*

Lemma 2. *Let $\mathcal{C}$ be a concept class on $\mathcal{X}$ that is CN-learnable by representation class $\mathcal{H}$. Let $\mathcal{A}^\eta$ be an algorithm that CN-learns $\mathcal{C}$ (with any noise rate $\eta \in [0; 1/2[$); $p(\cdot, \cdot, \cdot)$ and $q(\cdot, \cdot, \cdot)$ are polynomials (in $1/\varepsilon, 1/\delta, 1/(1 - 2\eta)$, respectively) for $\mathcal{A}^\eta$'s sample size and time requirements.*

Let $\eta^1, \eta^0 \in [0; 0.5[$ be the (actual) unkown *noise levels for the positive class and the negative class. Assume that we know a value $\eta_b < 0.5$ such that $\eta^1 \leq \eta_b$ and $\eta^0 \leq \eta_b$ and that we know whether $\eta^1 \geq \eta^0$.*

There exists an algorithm $\mathcal{A}$ such that for any $c \in \mathcal{C}$, for any $D \in \mathcal{D}$, for any $\varepsilon > 0$, for any $\delta > 0$, for any $\Delta \in]0; 1]$, for $\ell := p(1/\varepsilon, 1/\delta, 1/(1 - 2\eta_b)$ and $\tau := \frac{\Delta}{2\ell}$, for any $\rho \in [0; 1]$ verifying $|\rho - \rho_{opt}| < \tau$, for $s := \mathbf{1}_{(\eta^1 \geq \eta^0)}$, $\mathcal{A}$, when given inputs ε, δ, ρ, s, $\mathcal{A}^\eta$, ℓ and η_b and given access to $EX^{\boldsymbol{\eta}}_{\text{CCCN}}(c, \mathcal{D})$, outputs with probability $1 - \delta - \Delta$ a hypothesis $h \in \mathcal{H}$ verifying

$$err_D(h) \leq \varepsilon.$$

Algorithm 2 ApproximateLearn

Input: $\varepsilon > 0$, $\delta > 0$, $\rho \in [0; 1]$, $s \in \{0, 1\}$, $\mathcal{A}^\eta$ that CN-learns $\mathcal{C}$ with $q(\cdot, \cdot, \cdot)$ running time, $\ell \in \mathbb{N}$, $\eta_b \in [0; 1/2[$

Output: $h \in \mathcal{H}$

```
Build the CCCN oracle $EX^{\bar{\boldsymbol{\eta}}}_{\text{CCCN}}(c, D)$ using $\rho$ and $s$ as in Lemma 1
Draw a sample $\mathcal{S} = \{\langle \mathbf{x}_1, t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}_1) \rangle, \ldots, \langle \mathbf{x}_\ell, t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}_\ell) \rangle\}$ of size $\ell$ from $EX^{\bar{\boldsymbol{\eta}}}_{\text{CCCN}}(c, D)$
Input $\mathcal{S}$, $\varepsilon$ and $\delta$ to $\mathcal{A}^\eta$ with the upper bound on $\eta$ set to $\eta_b$
if the running time of $\mathcal{A}^\eta$ gets longer than $q(1/\varepsilon, 1/\delta, 1/(1 - 2\eta_b))$ then
    stop $\mathcal{A}^\eta$ and return $\emptyset$
else
    return the hypothesis $h \in \mathcal{H}$ output by $\mathcal{A}^\eta$
end if
```

In order to output h, $\mathcal{A}$ requires a polynomial number of labeled data and runs in polynomial time.

Proof. The idea of the proof is that if ρ is not too far from ρ_{opt} and s is set to the correct value (either 0 or 1) then the oracle resulting from the procedure specified in Lemma 1 is 'almost' a CN oracle and c can therefore be learned under D by $\mathcal{A}^\eta$.

Let us fix $\varepsilon > 0$, $\delta > 0$, $c \in \mathcal{C}$ and $D \in \mathcal{D}$. We assume, without loss of generality, that $\eta^1 \geq \eta^0$ and that, as a consequence, the indicator function $\mathbf{1}_{(\eta^1 \geq \eta^0)}$ takes the value 1. We also fix ρ such that $|\rho - \rho_{opt}| < \tau$.

We show that a call to ApproximateLearn (cf. Algorithm 2) with the inputs ε, δ, ρ, s, $\mathcal{A}^\eta$, ℓ and η_b outputs with probably $1 - \delta - \Delta$ a hypothesis $h \in \mathcal{H}$ such that $\text{err}(h) \leq \varepsilon$.

We know from Remark 3 that η_{opt}, the noise of the CN oracle obtained when using the procedure of Lemma 1 with ρ_{opt} and s_{opt}, is bounded from above by η_b. Therefore, ℓ set as in the lemma ensures that if $\mathcal{A}^\eta$ is provided with ℓ labeled sample data from a CN-oracle with noise lower than η_b, then it outputs with probability $1 - \delta$ a hypothesis having error not larger than ε. In addition, the running time to output such an hypothesis will not exceed $q(1/\varepsilon, 1/\delta, 1/(1 - 2\eta_b))$. The following analysis, which builds on an idea of Goldberg (2005), shows that it is possible with high probability to draw from $EX^{\boldsymbol{\eta}}_{\text{CCCN}}$ samples of size ℓ that can be interpreted as samples drawn from the CN oracle having noise η_{opt}.

Given a sample $\mathcal{S} = \{(\mathbf{x}_1, t(\mathbf{x}_1)), \ldots, (\mathbf{x}_\ell, t(\mathbf{x}_\ell))\}$ drawn from noise free oracle $EX(c, D)$ and param-

Algorithm 3 CorruptSample

```
Input: $\mathcal{S} = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_\ell, t_\ell)\} \in (\mathcal{X} \times \{0,1\})^\ell$, $\eta$,
  $\bar{\boldsymbol{\eta}} = [\bar\eta^1 \; \bar\eta^0]$
Output: $\mathcal{S}^\eta = \{(\mathbf{x}_1, t_1^\eta), \dots, (\mathbf{x}_\ell, t_\ell^\eta)\}$,
        $\mathcal{S}^{\bar{\boldsymbol{\eta}}} = \{(\mathbf{x}_1, t_1^{\bar{\boldsymbol{\eta}}}), \dots, (\mathbf{x}_\ell, t_\ell^{\bar{\boldsymbol{\eta}}})\}$
  for $i = 1, \dots, \ell$ do
    Draw a random number $u$ uniformly in $[0;1]$
    /* noise process 1 */
    if $u \leq \eta$ then
      $t_i^\eta := 1 - t_i$
    else
      $t_i^\eta := t_i$
    end if
    /* noise process 2 */
    if $u \leq \bar\eta^{t_i}$ then
      $t^{\bar{\boldsymbol{\eta}}} := 1 - t_i$
    else
      $t^{\bar{\boldsymbol{\eta}}} := t_i$
    end if
  end for
  return $(\mathcal{S}^\eta, \mathcal{S}^{\bar{\boldsymbol{\eta}}})$
```

eters η_{opt} and $\bar{\boldsymbol{\eta}}$ as inputs, algorithm CorruptSample (cf. Algorithm 3) produces two labeled samples $\mathcal{S}^{\eta_{opt}} = \{(\mathbf{x}_1, t^{\eta_{opt}}(\mathbf{x}_1)), \dots, (\mathbf{x}_\ell, t^{\eta_{opt}}(\mathbf{x}_\ell)\}$ and $\mathcal{S}^{\bar{\boldsymbol{\eta}}} = \{(\mathbf{x}_1, t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}_1)), \dots, (\mathbf{x}_\ell, t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}_\ell)\}$, which may be noisy versions of $\mathcal{S}$. It is important to note that

(a) the two processes that $\mathcal{S}$ undergoes in order to give rise to $\mathcal{S}^{\eta_{opt}}$ and $\mathcal{S}^{\bar{\boldsymbol{\eta}}}$ (cf. Algorithm 3) precisely simulate a CN noise process with noise parameter η_{opt} and a CCCN noise process with noise parameter $\bar{\boldsymbol{\eta}} = [\bar\eta^1 \; \bar\eta^0]$, respectively (see Definitions 1 and 3);

(b) the only way for $\mathcal{S}^{\eta_{opt}}$ and $\mathcal{S}^{\bar{\boldsymbol{\eta}}}$ to differ is that there exists (at least) an i such that $t^{\eta_{opt}}(\mathbf{x}_i) \neq t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}_i)$: this means that during the i-th iteration of the main loop of CorruptSample, the random uniform variate u fell either between $\bar\eta^1$ and η_{opt} if $t(\mathbf{x}_i)$ was equal to 1 (hence, only one of $t^{\eta_{opt}}(\mathbf{x}_i)$ and $t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}_i)$ would be a flipped version of $t(\mathbf{x}_i)$) or between $\bar\eta^0$ and η_{opt} if $t(\mathbf{x}_i)$ was equal to 0; dropping the dependence on i and using $p := P(t(\mathbf{x}) = 1)$, the probability of observing $t^{\eta_{opt}}(\mathbf{x}) \neq t^{\bar{\boldsymbol{\eta}}}(\mathbf{x})$ is therefore

$$P(t^{\eta_{opt}}(\mathbf{x}) \neq t^{\bar{\boldsymbol{\eta}}}(\mathbf{x})) = p|\bar\eta^1 - \eta_{opt}| + (1-p)|\bar\eta^0 - \eta_{opt}|. \tag{1}$$

From a broader perspective, $EX(c, D)$ in combination with CorruptSample – applied on labeled samples of size ℓ drawn from $EX(c, D)$ –, define a distribution $D_{\eta_{opt}, \bar{\boldsymbol{\eta}}}(\mathcal{S}^{\eta_{opt}}, \mathcal{S}^{\bar{\boldsymbol{\eta}}})$ on $(\mathcal{X} \times \{0,1\})^\ell \times (\mathcal{X} \times \{0,1\})^\ell$. According to what we have just noted, the marginal distributions $D_{\eta_{opt}}(\mathcal{S}^{\eta_{opt}})$ and $D_{\bar{\boldsymbol{\eta}}}(\mathcal{S}^{\bar{\boldsymbol{\eta}}})$ derived from $D_{\eta_{opt}, \bar{\boldsymbol{\eta}}}(\mathcal{S}^{\eta_{opt}}, \mathcal{S}^{\bar{\boldsymbol{\eta}}})$ are *exactly* the distributions of samples of size ℓ from $EX_{\text{CN}}^{\eta_{opt}}(c, D)$ and $EX_{\text{CCCN}}^{\bar{\boldsymbol{\eta}}}(c, D)$, respectively.

If we define $\omega(\mathcal{S})$ as the binary random variable that is equal to 1 if $\mathcal{S}$ allows $\mathcal{A}^\eta$ to output an hypothesis having error less than ε within the running time specified in ApproximateLearn (cf. Algorithm 2) and 0 otherwise, we have:

$$\begin{aligned}
&P_{\mathcal{S}^{\bar{\boldsymbol{\eta}}} \sim D_{\bar{\boldsymbol{\eta}}}} \left(\omega(\mathcal{S}^{\bar{\boldsymbol{\eta}}}) = 0\right) \\
&\qquad = P_{(\mathcal{S}^\eta, \mathcal{S}^{\bar{\boldsymbol{\eta}}}) \sim D_{\eta_{opt}, \bar{\boldsymbol{\eta}}}} \left(\omega(\mathcal{S}^{\bar{\boldsymbol{\eta}}}) = 0, \omega(\mathcal{S}^{\eta_{opt}}) = 1\right) \\
&\qquad\quad + P_{(\mathcal{S}^\eta, \mathcal{S}^{\bar{\boldsymbol{\eta}}}) \sim D_{\eta_{opt}, \bar{\boldsymbol{\eta}}}} \left(\omega(\mathcal{S}^{\bar{\boldsymbol{\eta}}}) = 0, \omega(\mathcal{S}^{\eta_{opt}}) = 0\right) \\
&\qquad \leq P_{(\mathcal{S}^\eta, \mathcal{S}^{\bar{\boldsymbol{\eta}}}) \sim D_{\eta_{opt}, \bar{\boldsymbol{\eta}}}} \left(\omega(\mathcal{S}^{\bar{\boldsymbol{\eta}}}) = 0, \omega(\mathcal{S}^{\eta_{opt}}) = 1\right) + \delta \\
&\qquad \leq P_{(\mathcal{S}^\eta, \mathcal{S}^{\bar{\boldsymbol{\eta}}}) \sim D_{\eta_{opt}, \bar{\boldsymbol{\eta}}}} (\mathcal{S}^{\bar{\boldsymbol{\eta}}} \neq \mathcal{S}^{\eta_{opt}}) + \delta.
\end{aligned}$$

However, for $(\mathcal{S}^\eta, \mathcal{S}^{\bar{\boldsymbol{\eta}}})$ distributed according to $D_{\eta_{opt}, \bar{\boldsymbol{\eta}}}$, the following holds true:

$$\begin{aligned}
P(\mathcal{S}^{\eta_{opt}} \neq \mathcal{S}^{\bar{\boldsymbol{\eta}}}) &= P(t_1^{\eta_{opt}} \neq t_1^{\bar{\boldsymbol{\eta}}} \vee \ldots \vee t_\ell^{\eta_{opt}} \neq t_\ell^{\bar{\boldsymbol{\eta}}}) \\
&\leq \ell P(t^{\eta_{opt}}(\mathbf{x}) \neq t^{\bar{\boldsymbol{\eta}}}(\mathbf{x})) \quad \text{(union bound)} \\
&= \ell \left(p|\eta_{opt} - (1-\rho)\eta^1| \right. \qquad \text{(cf. (1))} \\
&\qquad \left. +(1-p)|\eta_{opt} - \eta^0 - \rho(1-\eta^0)|\right) \\
&\leq \ell \left(|\rho - \rho_{opt}|\eta^1 + |\rho - \rho_{opt}|(1-\eta^0)\right) \\
&\qquad\qquad \text{(Remark 2)} \\
&\leq \ell\,(\tau + \tau) \qquad \text{(assumption)} \\
&\leq \ell \cdot 2\frac{\Delta}{2\ell} \qquad \text{(definition of } \tau) \\
&= \Delta,
\end{aligned}$$

where $t_i^{\eta_{opt}} := t^{\eta_{opt}}(\mathbf{x}_i)$, $t_i^{\bar{\boldsymbol{\eta}}} := t^{\bar{\boldsymbol{\eta}}}(\mathbf{x}_i)$.

The probability with which a random sample $\mathcal{S}^{\bar{\boldsymbol{\eta}}}$ of size ℓ generated from $D_{\bar{\boldsymbol{\eta}}}$, that is, polled from $EX_{\text{CCCN}}^{\bar{\boldsymbol{\eta}}}(c, D)$, prevents a successful learning is therefore bounded by $\delta + \Delta$. In other words, when given access to a sample of size ℓ from $EX_{\text{CCCN}}^{\bar{\boldsymbol{\eta}}}(c, D)$, as well as other input parameters, $\mathcal{A}^\eta$ has henceforth a probability at least $1 - \delta - \Delta$ to output a hypothesis having error lower than ε. □

Proposition 1. *Any concept class that is efficiently CN-learnable is also efficiently CCCN-learnable: CN ⊆ CCCN.*

More precisely, for every CN-learnable class there is an algorithm $\mathcal{A}$ such that for any concept c and any distribution D, for any $\epsilon > 0$ and $\delta > 0$, for any noise vector $\boldsymbol{\eta} = [\eta^1 \; \eta^0]$ with $\eta^1, \eta^0 \leq \eta_b < 0.5$, when given access to $EX_{\text{CCCN}}^{\boldsymbol{\eta}}(c, D)$, $\mathcal{A}$ outputs with probability $1-\delta$ a hypothesis h such that $err_D(h) \leq \epsilon$.

Proof. We note that this proposition closes the proof of Theorem 1.

Algorithm 4 LearnWithUpperBound

```
Input: $\varepsilon > 0$, $\delta > 0$, $\eta_b < 0.5$, $\mathcal{A}^\eta$ that CN-learns $\mathcal{C}$ with $p(\cdot,\cdot,\cdot)$ sample size
Output: a hypothesis $h \in \mathcal{H}$
  $H := \emptyset$
  $\Delta := \frac{\delta}{4}$
  $\varepsilon' := \frac{\varepsilon}{4}(1-2\eta_b)$
  $\delta' := \frac{\delta}{4}$
  $\ell := p(\frac{1}{\varepsilon'}, \frac{1}{\delta'}, \frac{1}{1-2\eta_b})$
  $\tau := \frac{\Delta}{2\ell}$
  for all $s \in \{0,1\}$ and $i \in \mathbb{N}$ such that $i\tau < \eta_b$ do
    $\rho_i := i\tau$
    $H := H \cup \{\mathsf{ApproximateLearn}(\varepsilon', \delta', \rho_i, s, \mathcal{A}^\eta, \ell, \eta_b)\}$
  end for
  $m := \frac{8}{\varepsilon^2(1-2\eta_b)^2} \ln \frac{16\ell}{\delta^2}$
  draw a sample $\mathcal{S}_m^{\boldsymbol{\eta}}$ of $m$ labeled examples from $EX_{\text{CCCN}}^{\boldsymbol{\eta}}(c, D)$
  return $\text{argmin}_{h\in H} \text{err}_{\mathcal{S}_m^{\boldsymbol{\eta}}}(h)$
```

In order to prove this lemma, it suffices to see that algorithm LearnWithUpperBound (cf. Algorithm 4) can CCCN-learn any concept c from a class that is CN-learnable under any distribution D when given an upper bound $\eta_b < 0.5$ on η^1 and η^0.

Let us fix $c \in \mathcal{C}$, $D \in \mathcal{D}$, $\epsilon > 0$, $\delta > 0$ and let us assume that we know η_b.

From the way τ is set, the double loop of LearnWithUpperBound ensures that there is a pair (ρ^*, s^*) of values such that ρ^* is within a distance of τ from ρ_{opt} and $s^* = s_{opt}$. Hence, by applying lemma 2, we know that there is with probability at least $1-\delta/4-\delta/4 = 1-\delta/2$ a hypothesis h^* such that $\text{err}(h^*) \leq \frac{\varepsilon}{4}(1-2\eta_b)$.

There is a need for a strategy capable with high probability to pick from H a hypothesis that has error lower than ε. Simple calculations give, for any h:

$$\begin{aligned} P(h(\mathbf{x}) \neq t^{\boldsymbol{\eta}}(\mathbf{x})) = p\eta^1 &+ (1-p)\eta^0 \\ &+ (1-2\eta^1)P(h(\mathbf{x})=1, t(\mathbf{x})=0) \\ &+ (1-2\eta^0)P(h(\mathbf{x})=0, t(\mathbf{x})=1), \end{aligned}$$

where, as earlier, p stands for $P(t(\mathbf{x})=1)$. Consequently, for any ε-bad hypothesis h, that is, any hypothesis having error larger than ε, we have

$$P(h(\mathbf{x}) \neq t^{\boldsymbol{\eta}}(\mathbf{x})) > p\eta^1 + (1-p)\eta^0 + \varepsilon(1-2\eta_b). \quad (2)$$

Besides, $P(h^*(\mathbf{x}) \neq t(\mathbf{x})) \leq \frac{\varepsilon}{4}(1-2\eta_b)$ implies

$$P(h^*(\mathbf{x}) = 1 \neq t(\mathbf{x}) = 0) \leq \frac{\varepsilon}{4}(1-2\eta_b)$$
$$P(h^*(\mathbf{x}) = 0 \neq t(\mathbf{x}) = 1) \leq \frac{\varepsilon}{4}(1-2\eta_b),$$

and, therefore

$$\begin{aligned} &P(h^*(\mathbf{x}) \neq t^{\boldsymbol{\eta}}(\mathbf{x})) \\ &\quad \leq p\eta^1 + (1-p)\eta^0 + \frac{\varepsilon}{4}(1-2\eta_b)\cdot 2(1-\eta^1-\eta^0) \\ &\quad \leq p\eta^1 + (1-p)\eta^0 + \frac{\varepsilon}{2}(1-2\eta_b). \quad (3) \end{aligned}$$

Equations (2) and (3) say that there is a gap of at least $\frac{\varepsilon}{2}(1-2\eta_b)$ between the error (on noisy patterns) of any ε-bad hypothesis and the error (on noisy patterns) of h^*. There is henceforth a size m of test sample $\mathcal{S}_m^{\boldsymbol{\eta}}$ such that the empirical errors measured on $\mathcal{S}_m^{\boldsymbol{\eta}}$ of all ε-bad hypotheses are far enough from the empirical error of h^*, i.e., for any ε_{cut} (strictly) within the bounds of (2) and (3), there is a size m of test sample that guarantees (with high probability) that the empirical errors on $\mathcal{S}_m^{\boldsymbol{\eta}}$ of all ε-bad hypotheses are above ε_{cut} while the empirical error of h^* on $\mathcal{S}_m^{\boldsymbol{\eta}}$ is below ε_{cut}.

Letting $\varepsilon_{\text{cut}} := p\eta^1 + (1-p)\eta^0 + \frac{3\varepsilon}{4}(1-2\eta_b)$, $H_{bad} := \{h \in H : \text{err}(h) > \varepsilon\}$ and $G_{m,\varepsilon_{cut}}^{\boldsymbol{\eta}} := \{h \in H : \text{err}_{\mathcal{S}_m^{\boldsymbol{\eta}}}(h) \leq \varepsilon_{\text{cut}}\}$, we have

$$\begin{aligned} P(\exists h \in H_{bad} \cap G_{m,\varepsilon_{cut}}^{\boldsymbol{\eta}}) &\leq |H_{bad}| P(h \in H_{bad} \cap G_{m,\varepsilon_{cut}}^{\boldsymbol{\eta}}) \\ &\qquad \text{(union bound)} \\ &\leq |H| \exp\left(-\frac{m\varepsilon^2(1-2\eta_b)^2}{8}\right) \\ &\qquad \text{(Chernoff bound)} \\ &\leq \frac{1}{2\tau} \exp\left(-\frac{m\varepsilon^2(1-2\eta_b)^2}{8}\right). \end{aligned}$$

In order to have $P(\exists h \in H_{bad} \cap G_{m,\varepsilon_{cut}}^{\boldsymbol{\eta}}) \leq \delta/4$, it suffices to choose m so that the right-hand side of the last inequation is bounded from above by $\delta/4$, i.e., it suffices to have

$$m = \frac{8}{\varepsilon^2(1-2\eta_b)^2} \ln \frac{16\ell}{\delta^2}$$

as it is set in LearnWithUpperBound.

Likewise, for h^*, we have

$$\begin{aligned} P(h^* \notin G_{m,\varepsilon_{cut}}^{\boldsymbol{\eta}}) &\leq \exp\left(-\frac{m\varepsilon^2(1-2\eta_b)^2}{8}\right) \\ &\qquad \text{(Chernoff bound)} \\ &\leq 2\tau\frac{\delta}{4} = 2\cdot\frac{\delta}{8\ell}\cdot\frac{\delta}{4} \\ &\leq \frac{\delta}{4} \end{aligned}$$

for the specific choice of m made.

It directly follows that the hypothesis $h_{\min}$ from H that minimizes the empirical error on $\mathcal{S}_m^{\boldsymbol{\eta}}$ – for the given value of m – is, with probability at least $1 - \delta/4 - \delta/4 = 1 - \delta/2$, a hypothesis that has *true* error lower than ε. (We note that, though it may possibly be the case, $h_{\min}$ need not be h^*.)

All in all, we have that when given an upper bound on η^1 and η^0, and given access to a polynomial number of labeled data, LearnWithUpperBound outputs with probability at least $1-\delta$ a hypothesis with error at most ε. In addition, since ApproximateLearn controls its running time the running time of LearnWithUpperBound is polynomial as well. This closes the proof of Proposition 1. □

4. CPCN=CCCN=CN

In this section we provide a result showing the equality between CPCN and CCCN. This directly gives the main result of this paper, namely CN = CPCN.

The idea of the proof is that it is possible to build a partition of the input space $\mathcal{X}$ from the partition functions of a CPCN oracle (which define a partition over $\mathcal{S}\times\{0,1\}$): this partition is such that the noise process that corrupts the data within each part is a CCCN noise process. Given a CCCN learning algorithm $\mathcal{A}$ and some condition as for the number of data to draw from the CPCN oracle to be sure that each part contains enough (or no) data to be CCCN-learned, hypotheses are learned on each part. These hypotheses are used to relabel a CPCN sample of an appropriate size, which is in turn input to $\mathcal{A}$ to output with high probability a hypothesis having small error.

Lemma 3. *Let $c \in \mathcal{C}$ and $D \in \mathcal{D}$. Let h be a classifier that has error $err_D(h) \leq \varepsilon$. Then, for any $\alpha \in]0;1]$ and any integer $\ell \leq \alpha/\varepsilon$, the probability that h correctly predicts the labels of the elements of a sample of size ℓ drawn according to D is greater than $1-\alpha$.*

Proof. The probability that h correctly predicts the class of ℓ elements independently drawn according to D is greater than $(1-\varepsilon)^\ell$. It can be easily checked that for any $0 \leq \varepsilon \leq 1$, $(1-\varepsilon)^\ell \geq 1-\ell\varepsilon \geq 1-\alpha$. □

Lemma 4. *Let $D \in \mathcal{D}$. Let $\overline{\pi}_1,\ldots,\overline{\pi}_k$ be a partition of $\mathcal{X}$, let $0<\varepsilon,\delta\leq 1$ be two parameters, let m be an integer and let $\ell \geq \max(2m/\varepsilon, -2\log(\delta/k)/\varepsilon^2)$. Then, with a probability greater than $1-\delta$, any sample $\mathcal{S}$ of $\mathcal{X}$ containing ℓ examples independently drawn according to D will contain at least m elements of each part $\overline{\pi}_i$ that satisfies $D(\overline{\pi}_i)\geq\varepsilon$, with $D(\overline{\pi}_i) := P_{\mathbf{x}\sim D}(\overline{\pi}_i(\mathbf{x}) = 1)$.*

Proof. We note that the partition $\overline{\pi}_1,\ldots,\overline{\pi}_k$ is defined with respect to the unlabeled space $\mathcal{X}$.

Let $\ell \geq \max(2m/\varepsilon, -2\log(\delta/k)/\varepsilon^2)$, let $\mathcal{S}$ be a sample containing ℓ examples independently drawn according to D, and let $m_i := |\mathcal{S}\cap\overline{\pi}_i)|$. It comes from Chernoff bound and the way ℓ is chosen that, for any $1\leq i\leq k$,

$$P\left(\frac{m_i}{\ell} \leq D(\overline{\pi}_i) - \frac{\varepsilon}{2}\right) \leq \exp\left(-\frac{\ell\varepsilon^2}{2}\right) \leq \frac{\delta}{k}.$$

Hence, if $\overline{\pi}_i$ is such that $D(\overline{\pi}_i)\geq\varepsilon$,

$$P(m_i \leq m) \leq P\left(m_i \leq \ell\frac{\varepsilon}{2}\right) \leq \frac{\delta}{k}.$$

By the union bound

$$\begin{aligned} P(\exists i : m_i \leq m, D(\overline{\pi}_i)\geq\varepsilon) &\leq kP((m_i\leq m), D(\overline{\pi}_i)\geq\varepsilon) \\ &= k\cdot\frac{\delta}{k} = \delta. \end{aligned}$$

Therefore, with probability greater than $1-\delta$, any part $\overline{\pi}_i$ such that $D(\overline{\pi}_i)\geq\varepsilon$ satisfies $m_i > m$. □

Proposition 2. *Let $\mathcal{C}$ be a class of concepts over $\mathcal{X}$ which is in CCCN. Then $\mathcal{C}$ is in CPCN. Stated otherwise: CCCN $\subseteq$ CPCN.*

Proof. Let $\mathcal{A}$ be a CCCN learning algorithm for $\mathcal{C}$ and let $p(\cdot,\cdot,\cdot)$ be a polynomial such that for any target concept c in $\mathcal{C}$, any distribution $D\in\mathcal{D}$, any accuracy parameter ε, any confidence parameter δ and any noise rate bound η_b, if $\mathcal{A}$ is given as input a sample $\mathcal{S}$ drawn according to $EX^{\boldsymbol{\eta}}_{\text{CCCN}}(c,D)$ (where $\boldsymbol{\eta} = [\eta^1\ \eta^0]$ and $\eta^1,\eta^0\leq\eta_b$) and satisfying $|S|\geq p(1/\varepsilon,1/\delta,1/(1-2\eta_b))$, then $\mathcal{A}$ outputs a hypothesis whose error rate is lower than ε with probability at least $1-\delta$.

Let $\Pi = \{\pi_1,\ldots,\pi_k\}$ be a partition of $\mathcal{X}\times\{0,1\}$ and let $\eta = [\eta_1\cdots\eta_k]$ be a vector of noise rates satisfying $0\leq\eta_i\leq\eta_b$ for $1\leq i\leq k$. We deduce from Π a partition $\overline{\Pi} = (\overline{\pi}_1,\ldots,\overline{\pi}_l)$ of $\mathcal{X}$ based on the parts π_{ij} defined for $1\leq i,j\leq k$, by $\pi_{ij} = \{\mathbf{x}\in\mathcal{X}|\langle\mathbf{x},1\rangle\in\pi_i$ and $\langle\mathbf{x},0\rangle\in\pi_j\}$. (It is straightforward to check that for any $\mathbf{x}\in\mathcal{X}$, there exist i and j such that $\mathbf{x}\in\pi_{ij}$ and that $\pi_{ij}\cap\pi_{uv}\neq\emptyset$ implies $i=u$ and $j=v$.) For any $\overline{\pi}_i\in\overline{\Pi}$ such that $\overline{\pi}_i = \pi_{uv}$, define $\eta^1_i := \eta_u$ and $\eta^0_i := \eta_v$.

Let $c\in\mathcal{C}$, let $D\in\mathcal{D}$ and let $0<\varepsilon,\delta\leq 1$ be accuracy and confidence parameters.

Let $n_1\geq p(1/\varepsilon,4/\delta,1/(1-2\eta_b))$, let $\varepsilon_1 := \delta/(4ln_1)$, let $m := p(1/\varepsilon_1,4l/\delta,1/(1-2\eta_b))$ and let $n_2\geq\max(2m/\varepsilon_1,-2\log(\delta/(4l))/\varepsilon_1^2)$. Note that n_2 is polynomial in $1/\varepsilon,1/\delta$ and $1/(1-2\eta_b)$.

Let $\mathcal{S}_2$ be a sample of size n_2 drawn according to $EX^{\Pi,\boldsymbol{\eta}}_{\text{CPCN}}(c,D)$. From Lemma 4, with probability at least $1-\delta/4$, any part $\overline{\pi}_i$ such that $D(\overline{\pi}_i)\geq\varepsilon_1$ satisfies $|\mathcal{S}_2\cap\overline{\pi}_i| > m$. Let $I := \{i : |\mathcal{S}_2\cap\overline{\pi}_i| > m\}$.

For each $i\in I$, run algorithm $\mathcal{A}$ on each sample $\mathcal{S}_2\cap\overline{\pi}_i$ and let h_i be the output classifier. With a probability greater than $1-\delta/4$, each h_i is such that $P_{D|\pi_i(\mathbf{x})=1}(h_i(\mathbf{x}) \neq t(\mathbf{x})) \leq \varepsilon_1$.

Now, let $\mathcal{S}_1$ be a new sample of size n_1 drawn according to $EX^{\Pi,\boldsymbol{\eta}}_{\text{CPCN}}(c, D)$.

- Let $\overline{\pi}_i$ be a part such that $D(\overline{\pi}_i) < \varepsilon_1$. The probability that $\mathcal{S}_1$ contains no element of $\overline{\pi}_i$ is $\geq (1-\varepsilon_1)^{n_1} \geq 1-\delta/(4l)$.
- Let $\overline{\pi}_i$ be a part such that $D(\overline{\pi}_i) \geq \varepsilon_1$. From Lemma 3, the probability that h_i computes the correct label of each example of $\mathcal{S}_1 \cap \overline{\pi}_i$ is greater than $1-\delta/(4l)$.

That is (using the union bound) the probability that $\mathcal{S}_1$ contains no element of a part $\overline{\pi}_i$ satisfying $D(\overline{\pi}_i) < \varepsilon_1$ and that all elements of $\mathcal{S}_1$ are correctly labeled by hypotheses $(h_i)_{i\in I}$ is greater than $1-\delta/4$.

Finally, relabel the examples of $\mathcal{S}$ using the predictions given by hypotheses $(h_i)_{i\in I}$ and run algorithm $\mathcal{A}$ on the relabeled sample $\tilde{\mathcal{S}}_1$. With a probability greater than $1-\delta/4$, it will output a hypothesis h such that $\text{err}_D(h) \leq \varepsilon$.

Taking everything together, the overall procedure outputs with probability $1 - 4\cdot\delta/4 = 1-\delta$ a hypothesis h that has error $\text{err}_D(h) \leq \varepsilon$. □

We can therefore state the main result of this paper:

Theorem 2. *CN=CCCN=CPCN.*

Proof. From the previous section, we know that CN = CCCN. In this section, we showed that CCCN $\subseteq$ CPCN and, since CPCN $\subseteq$ CCCN (the CCCN framework is a particular case of the CPCN framework), CPCN = CCCN. This trivially gives CN = CPCN. □

5. Bounds on the Noise Rates

In this study, we have restricted ourselves to the case where the upper bound η_b on the noise rates is stricly lower than 1/2. It can be shown that it is possible to address the case where instead of having an upper bound on the noise rates, a lower bound $\eta_b^l > 0.5$ is provided. Whichever the oracle to which the learning procedure is given access, it suffices to flip all the labels of the labeled examples it produces. Doing that brings the learning problem considered back to the classical setting where the upper bound on the noise rates is now $1-\eta_b^l$. The question of the learnability in the CPCN (or CCCN) framework in the more general case where some noise rates may be above 0.5 and some other below is still an open problem.

Another open question is that of the need of having an upper bound on the noise rates. Even though it is known that in the CN framework such a bound can be estimated (in polynomial time) through the learning process, it is not clear whether such an (distribution free) estimation can be carried out for the CCCN and CPCN cases.

6. Conclusion and Outlook

This paper presents a particular case of the learnability in the PAC-framework, where classes of examples are subject to various classification noise settings and we give two important results, namely, CN=CCCN and CN=CPCN.

An interesting outlook of this work is that of its application to the learning of noisy perceptrons (Blum et al., 1996; Cohen, 1997) in the CPCN framework. The question of the consequences of our work on the problem of learning optimal separating hyperplanes (in finite dimension as well as in infinite dimensional spaces) with soft-margins is also a problem we are interested in and that we plan to investigate.

References

Angluin, D., & Laird, P. (1988). Learning from Noisy Examples. *Machine Learning*, *2*.

Blum, A., Frieze, A. M., Kannan, R., & Vempala, S. (1996). A Polynomial-Time Algorithm for Learning Noisy Linear Threshold Functions. *Proc. of 37th IEEE Symposium on Foundations of Computer Science* (pp. 330–338).

Cohen, E. (1997). Learning Noisy Perceptrons by a Perceptron in Polynomial Time. *Proc. of 38th IEEE Symposium on Foundations of Computer Science* (pp. 514–523).

Decatur, S. E. (1997). Pac Learning with Constant-Partition Classification Noise and Applications to Decision Tree Induction. *Proc. of the 14th Int. Conf. on Machine Learning*.

Goldberg, P. (2005). Some Discriminant-based PAC Algorithm. Personal communication.

Kearns, M. J., & Vazirani, U. V. (1994). *An Introduction to Computational Learning Theory*. MIT Press.

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