Learning Kernel Perceptrons on Noisy Data Using Random Projections

Guillaume Stempfel and Liva Ralaivola

Laboratoire d'Informatique Fondamentale de Marseille, UMR CNRS 6166
Université de Provence, 39, rue Joliot Curie, 13013 Marseille, France
{guillaume.stempfel,liva.ralaivola}@lif.univ-mrs.fr

Abstract. In this paper, we address the issue of learning nonlinearly separable concepts with a kernel classifier in the situation where the data at hand are altered by a uniform classification noise. Our proposed approach relies on the combination of the technique of random or deterministic projections with a classification noise tolerant perceptron learning algorithm that assumes distributions defined over finite-dimensional spaces. Provided a sufficient separation margin characterizes the problem, this strategy makes it possible to envision the learning from a noisy distribution in any separable Hilbert space, regardless of its dimension; learning with any appropriate Mercer kernel is therefore possible. We prove that the required sample complexity and running time of our algorithm is polynomial in the classical PAC learning parameters. Numerical simulations on toy datasets and on data from the UCI repository support the validity of our approach.

1 Introduction

For a couple of years, it has been known that kernel methods [1] provide a set of efficient techniques and associated models for, among others, classification supported by strong theoretical results (see, e.g. [2,3]), mainly based on *margin* criteria and the fact they constitute a generalization of the well-studied class of linear separators.

Astonishingly enough however, there is, to our knowledge, very little work on the issue of learning noisy distributions with kernel classifiers, a problem which is of great interest if one aims at using kernel methods on real-world data. Assuming a *uniform classification noise* process [4], the problem of learning from noisy distributions is a key challenge in the situation where the *feature space* associated with the chosen kernel is of *infinite dimension*, knowing that approaches to learn noisy linear classifiers in finite dimension do exist [5,6,7,8].

In this work, we propose an algorithm to learn noisy distributions defined on general Hilbert spaces (not necessarily finite dimensional) from a reasonable number of data (where reasonable is specified later on); this algorithm combines the technique of random projections with a known finite-dimensional noise-tolerant linear classifier.

The paper is organized as follows. In Section 2, the problem setting is depicted together with the assumed classification noise model. Our strategy to learn kernel classifiers from noisy distributions is described in Section 3. Section 4 reports some contributions related to the questions of learning noisy perceptrons and learning kernel classifiers using projections methods. Numerical simulations carried out on synthetic

M. Hutter, R.A. Servedio, and E. Takimoto (Eds.): ALT 2007, LNAI 4754, pp. 328–342, 2007.

datasets and on benchmark datasets from the UCI repository proving the effectiveness of our approach are presented in Section 5.

2 Problem Setting and Main Result

Remark 1 (Binary classification in Hilbert spaces, zero-bias separating hyperplanes). From now on, $\mathcal{X}$ denotes the input space, assumed to be a *Hilbert space* equipped with an inner product denoted by $\cdot$. In addition, we will restrict our study to the binary classification problem and the target space $\mathcal{Y}$ will henceforth always be $\{-1, +1\}$.

We additionally make the simplifying assumption of the existence of zero-bias separating hyperplanes (i.e. hyperplanes defined as $\mathbf{w} \cdot \mathbf{x} = 0$).

2.1 Noisy Perceptrons in Finite Dimension

The Perceptron algorithm [9] (cf. Fig. 1) is a well-studied greedy strategy to derive a linear classifier from a sample $\mathcal{S} = \{(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_m, y_m)\}$ of m labeled pairs $(\mathbf{x}_i, y_i)$ from $\mathcal{X} \times \mathcal{Y}$ assumed to be drawn independently from an *unknown* and *fixed* distribution D over $\mathcal{X} \times \mathcal{Y}$. If there exists a separating hyperplane $\mathbf{w}^* \cdot \mathbf{x} = 0$ according to which the label y of $\mathbf{x}$ is set, i.e. y is set to $+1$ if $\mathbf{w}^* \cdot \mathbf{x} \geq 0$ and -1 otherwise[1], then the Perceptron algorithm, when given access to $\mathcal{S}$, converges towards a hyperplane $\mathbf{w}$ that correctly separates $\mathcal{S}$ and might with high probability exhibit good generalization properties [10].

Input: $\mathcal{S} = \{(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_m, y_m)\}$
Output: a linear classifier $\mathbf{w}$

$t \leftarrow 0, \mathbf{w}_0 \leftarrow \mathbf{0}$
while there is i s.t. $y_i \mathbf{w}_t \cdot \mathbf{x}_i \leq 0$ **do**
  $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_i \mathbf{x}_i / \|\mathbf{x}_i\|$
  $t \leftarrow t + 1$
end while
return $\mathbf{w}$

Fig. 1. Perceptron algorithm

We are interested in the possibility of learning linearly separable distributions on which a random *uniform classification noise*, denoted as CN [4], has been applied, that is, distributions where correct labels are flipped with some given probability η. In order to tackle this problem, [5] has proposed a simple algorithmic strategy later exploited by [6]: it consists in an iterative learning process built upon the Perceptron algorithm where update vectors are computed as sample averages of training vectors fulfilling certain properties. The expectations of those update vectors guarantee the convergence of the learning process and, thanks in part to Theorem 1 stated just below, it is guaranteed with probability $1 - \delta$ ($\delta \in (0, 1)$) that whenever the dimension n of $\mathcal{X}$ is *finite* and there exists a separating hyperplane of margin $\gamma > 0$, a polynomial number of training data is sufficient for the sample averages to be close enough to their expectations; this, in turn implies a polynomial running time complexity of the algorithm together with a $1 - \delta$ guarantees for a generalization error of ε. Here, *polynomiality* is defined with respect to n, $1/\delta$, $1/\varepsilon$, $1/\gamma$ and $1/(1 - 2\eta)$. Note that despite the availability of generalization bounds for soft-margin SVM expressed in terms of margin and the values of

[1] We assume a deterministic labelling of the data according to the target hyperplane $\mathbf{w}^*$, i.e. $Pr(y = 1|\mathbf{x}) = 1$ or $Pr(y = 1|\mathbf{x}) = 0$; a nondeterministic setting can be handled as well.

Algorithm 1. RP-classifier

Input: • $\mathcal{S} = \{(\mathbf{x}_1, y_1) \dots (\mathbf{x}_m, y_m)\}$ in $\mathcal{X} \times \{-1, +1\}$
• n, projection dimension
Output: • a random projection $\pi = \pi(\mathcal{S}, n) : \mathcal{X} \to \mathcal{X}'$, $\mathcal{X}' = \text{span}\langle \mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_n} \rangle$
• projection classifier $f(\mathbf{x}) = \mathbf{w} \cdot \pi(\mathbf{x})$, $\mathbf{w} \in \mathcal{X}'$

learn an orthonormal random projection $\pi : \mathcal{X} \to \mathcal{X}'$
learn a linear classifier $\mathbf{w}$ from $\mathcal{S} = \{(\pi(\mathbf{x}_1), y_1) \dots (\pi(\mathbf{x}_m), y_m)\}$
return π, $\mathbf{w}$

slack variables, which account for possible classification errors, there is no result, to our knowledge, which characterizes the solution obtained by solving the quadratic program when the data is uniformly corrupted by classification noise. It is therefore not possible to control beforehand the values of the slack variables, and, hence, the non-triviality of the bounds (i.e. bounds with values lower than 1).

Theorem 1 ([11]). If $\mathcal{F} = \{f_\varphi(\mathbf{x}) | \varphi \in \Phi\}$ *has a pseudo-dimension of* h *and a range* R *(i.e.* $|f_\varphi(\mathbf{x})| \leq R$ *for any* φ *and* $\mathbf{x}$*), and* if *a random sample of* $M \geq m_0(h, R, \delta, \varepsilon) = \frac{8R^2\left(2h \ln \frac{4R}{\varepsilon} + \ln \frac{2}{\delta}\right)}{\varepsilon^2}$ *i.i.d examples are drawn from a fixed distribution,* then *with probability* $1 - \delta$*, the sample average of every indicator function* $f_\varphi(\mathbf{x}) > \alpha$ *is within* $\frac{\varepsilon}{R}$ *of its expected value, and the sample average of every* f_φ *is within* ε *of its expected value. (The pseudo-dimension of* $\mathcal{F}$ *is the VC dimension of* $\{f_\varphi(\mathbf{x}) > \alpha | \varphi \in \Phi \wedge \alpha \in \mathbb{R}\}$*.)*

2.2 Main Result: RP Classifiers and Infinite-Dimensional Spaces

h The question that naturally arises is whether it is possible to learn linear classifiers from noisy distributions defined over *infinite dimensional spaces* with similar theoretical guarantees with respect to the polynomiality of sample and running time complexities. We answer to this question positively by exhibiting a family of learning algorithm called *random projection classifiers* capable of doing so. Classifiers of this family learn from a training sample $\mathcal{S}$ according to Algorithm 1: given a finite projection dimension n, they first learn a projection π from $\mathcal{X}$ to a space $\mathcal{X}'$ spanned by n (randomly chosen) vectors of $\mathcal{S}$ dimensional space and then, learn a finite dimensional noisy perceptron from the labeled data projected according to π. An instantiation of RP-classifiers simply consists in a choice of a random projection learning algorithm and of a (noise-tolerant) linear classifier.

Let us more formally introduce some definitions and state our main result.

Remark 2 (Labeled Examples Normalization). In order to simplify the definitions and the writing of the proofs we will use the handy transformation that consists in converting every labeled example $(\mathbf{x}, y)$ to $y\mathbf{x}/\|\mathbf{x}\|$. From now on, we will therefore consider distributions and samples defined on $\mathcal{X}$ (instead of $\mathcal{X} \times \mathcal{Y}$).

Note that the transformation does not change the difficulty of the problem and that the search for a separating hyperplane between +1 and -1 classes boils down to the search for a hyperplane $\mathbf{w}$ verifying $\mathbf{w} \cdot \mathbf{x} > 0$.

Definition 1 ((γ, ε)-separable distributions $\mathcal{D}^{\gamma,\varepsilon}$). *For* $\gamma > 0, \varepsilon \in [0, 1)$*,* $\mathcal{D}^{\gamma,\varepsilon}$ *is the set of distributions on* $\mathcal{X}$ *such that for any* D *in* $\mathcal{D}^{\gamma,\varepsilon}$*, there exists a unit vector* $\mathbf{w}$ *in* $\mathcal{X}$ *such that* $Pr_{\mathbf{x} \sim D}[\mathbf{w} \cdot \mathbf{x} < \gamma] \leq \varepsilon$.

Definition 2 (CN distributions $\mathcal{U}^{\gamma,\eta}$ [4]). *For $\eta \in [0, 0.5)$, let the random transformation U^η map $\mathbf{x}$ to $-\mathbf{x}$ with probability η and leave it unchanged with probability $1-\eta$. The set of distributions $\mathcal{U}^{\gamma,\eta}$ is defined as $\mathcal{U}^{\gamma,\eta} := U^\eta(\mathcal{D}^{\gamma,0})$.*

Uniform classification noise may appear as a very limited model but learnability results int this framework can be easily extended to more general noise model [12]. We can now state our main result.

Theorem 2 (Dimension-Independent Learnability of Noisy Perceptrons). *There are an algorithm $\mathcal{A}$ and polynomials $p(\cdot,\cdot,\cdot,\cdot)$ and $q(\cdot,\cdot,\cdot,\cdot)$ such that the following holds.*

$\forall \varepsilon \in (0,1), \forall \delta \in (0,1), \forall \gamma > 0, \forall \eta \in [0, 0.5), \forall D \in \mathcal{D}^{\gamma,0}$, if *a random sample $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ with $m \geq p(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta}, \frac{1}{\gamma})$ is drawn from $U^\eta(D)$,* then *with probability at least $1-\delta$, $\mathcal{A}$ runs in time $q(\frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{1-2\eta}, \frac{1}{\gamma})$ and the classifier $f := \mathcal{A}(\mathcal{S})$ output by $\mathcal{A}$ has a generalization error $Pr_{\mathbf{x}\sim D}(f(\mathbf{x}) \leq 0)$ bounded by ε.*

3 Combining Random Projections and a Noise-Tolerant Algorithm

This section gives a proof of Theorem 2 by showing that an instance of RP-classifier using a linear learning algorithm based on a specific perceptron update rule, Cnoise-update, proposed by [8] and on properties of simple random projections proved by [13] is capable of efficiently learning CN distributions (see Definition 2) independently of the dimension of the input space.

The proof works in two steps. First (section 3.1) we show that Cnoise-update (Algorithm 2) in finite dimension can tolerate a small amount of *malicious noise* and still returns relevant update vectors. Then (section 3.2) thanks to properties of random projections (see [13]) we show that they can be efficiently used to transform a CN problem into one that meets the requirements of Cnoise-update (and Theorem 4 below).

3.1 Perceptron Learning with Mixed Noise

We suppose in this subsection that $\mathcal{X}$ is of finite dimension n. We make use of the following definitions.

Definition 3 (Sample and population accuracies). *Let $\mathbf{w}$ be a unit vector, D be a distribution on $\mathcal{X}$ and $\mathcal{S}$ be a sample drawn from D. We say that $\mathbf{w}$ has* sample accuracy $1-\varepsilon$ *on $\mathcal{S}$ and* (population) accuracy $1-\varepsilon'$ if*:*

$$Pr_{\mathbf{x}\in S}\left[\mathbf{w}\cdot\mathbf{x} < 0\right] = \varepsilon, \quad \textit{and} \quad Pr_{\mathbf{x}\sim D}\left[\mathbf{w}\cdot\mathbf{x} < 0\right] = \varepsilon'.$$

Definition 4 (CN-consistency). *A unit vector $\mathbf{w}^*$ is CN-consistent on $D \in \mathcal{U}^{\gamma,\eta}$* if $Pr_{\mathbf{x}\sim D}\left[\mathbf{w}^*\cdot\mathbf{x} < \gamma\right] = \eta$. *It means $\mathbf{w}^*$ makes* no error *on the noise free version of D.*

We recall that according to the following theorem [8], Cnoise-update, depicted in Algorithm 2, when used in a perceptron-like iterative procedure, renders the learning of CN-distributions possible in finite dimension.

Algorithm 2. Cnoise-Update [8]

Input: $\mathcal{S}$: training data, $\mathbf{w}$: current weight vector, ν a nonnegative real value
Output: an update vector $\mathbf{z}$

$$\boldsymbol{\mu} \leftarrow \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}\in\mathcal{S}}\mathbf{x}, \qquad \boldsymbol{\mu}' \leftarrow \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}\in\mathcal{S}\wedge\mathbf{w}\cdot\mathbf{x}\leq 0}\mathbf{x}$$

if $\mathbf{w}\cdot\boldsymbol{\mu} \leq \nu\|\mathbf{w}\|$ **then**
  $\mathbf{z} \leftarrow \boldsymbol{\mu}$
else
  $a \leftarrow \dfrac{\mathbf{w}\cdot\boldsymbol{\mu} - \nu\|\mathbf{w}\|}{\mathbf{w}\cdot\boldsymbol{\mu} - \mathbf{w}\cdot\boldsymbol{\mu}'}, \qquad b \leftarrow \dfrac{-\mathbf{w}\cdot\boldsymbol{\mu}' + \nu\|\mathbf{w}\|}{\mathbf{w}\cdot\boldsymbol{\mu} - \mathbf{w}\cdot\boldsymbol{\mu}'}, \qquad \mathbf{z} \leftarrow a\boldsymbol{\mu}' + b\boldsymbol{\mu}$
end if
if $\mathbf{w}\cdot\mathbf{z} > 0$ **then**
  $\mathbf{z} \leftarrow \mathbf{z} - \mathbf{w}\dfrac{\mathbf{w}\cdot\mathbf{z}}{\mathbf{w}\cdot\mathbf{w}}$ */* projection step */*
end if
return $\mathbf{z}$

Theorem 3 ([8]). *Let $\gamma \in [0,1], \eta \in [0, 0.5), \varepsilon \in (0, 1-2\eta]$. Let $D \in \mathcal{U}^{\gamma,\eta}$.* If $\mathbf{w}^*$ *is CN-consistent on D,* if *a random sample $\mathcal{S}$ of $m \geq m_0\left(10(n+1), 2, \delta, \frac{\varepsilon\gamma}{4}\right)$ examples are drawn from D and* if *the perceptron algorithm uses update vectors from* Cnoise-Update*$(\mathcal{S}, \mathbf{w}_t, \frac{\varepsilon\gamma}{4})$ for more than $\frac{16}{(\varepsilon\gamma)^2}$ updates on these points, then the $\mathbf{w}_t$ with the highest sample accuracy has accuracy at least $1-\eta-\varepsilon$ with probability $1-\delta$*[2].

The question that is of interest to us deals with a little bit more general situation than simple CN noise. We would like to show that Cnoise-update is still applicable when, in addition to being CN, the distribution on which it is called is also corrupted by *malicious noise* [14], i.e. a noise process whose statistical properties cannot be exploited in learning (this is an 'incompressible' noise). Envisioning this situation is motivated by the projection step, which may introduce some amount of *projection noise* (cf. Theorem 5), that we treat as malicious noise.

Of course, a limit on the amount of malicious noise must be enforced if some reasonable generalization error is to be achieved. Working with distributions from $\mathcal{U}^{\gamma,\eta}$ we therefore set $\theta_{\max}(\gamma,\eta) = \frac{\gamma(1-2\eta)}{8}$ as the maximal amount tolerated by the algorithm. For $\theta \leq \theta_{\max}$, a minimal achievable error rate $\varepsilon_{\min}(\gamma,\eta,\theta) = \frac{64\theta}{\gamma(1-\eta)\left(\frac{1}{8}-\theta\right)}$ will be our limit[3]. Provided that the amount of malicious noise is lower than $\theta_{\max}$, we show that learning can be achieved for any error $\varepsilon \geq \varepsilon_{\min}(\gamma,\eta,\theta)$. The proof non trivially extends that of [8] and roughly follows its lines.

Definition 5 (Mixed-Noise distributions, $\mathcal{U}^{\gamma,\eta\,\theta}$). *For $\theta \in [0,1)$, let the random transformation U^θ leave an input $\mathbf{x}$ unchanged with probability $1-\theta$ and change it to any arbitrary $\mathbf{x}'$ with probability θ (nothing can be said about $\mathbf{x}'$). The set of distributions $\mathcal{U}^{\gamma,\eta,\theta}$ is defined as $\mathcal{U}^{\gamma,\eta,\theta} := U^\theta\left(U^\eta(\mathcal{D}^{\gamma,0})\right)$.*

[2] For the remaining of the paper, ε is not the usual error parameter ε' used in PAC, but $\varepsilon'(1-2\eta)$.
[3] Slightly larger amount of noise and smaller error rate could be theoretically targeted. But the choices we have made suffice to our purpose.

Remark 3 (CN and MN decomposition). For $\gamma > 0, \eta \in [0, 0.5), \theta \in [0, 1)$, the image distribution $D^{\gamma,\eta,\theta} := U^{\theta}\left(U^{\eta}(D^{\gamma,0})\right)$ of $D^{\gamma,0} \in \mathcal{D}^{\gamma,0}$ is therefore a mixture of two distributions: the first one, of weight $1-\theta$, is a CN distribution with noise η and margin γ while nothing can be said about the second, of weight θ. This latter distribution will be referred to as the malicious part (MN) of $D^{\gamma,\eta,\theta}$. In order to account for the malicious noise, we introduce the random variable $\theta : \mathcal{X} \to \{0, 1\}$ such that $\theta(\mathbf{x}) = 1$ if $\mathbf{x}$ is altered by malicious noise and $\theta(\mathbf{x}) = 0$ otherwise.

From now on, we will use $E[f(\mathbf{x})]$ for $E_{\mathbf{x}\sim D}[f(\mathbf{x})]$ and $\hat{E}[f(\mathbf{x})]$ for $E_{\mathbf{x}\in S}[f(\mathbf{x})]$.

Lemma 1. *Let $\gamma > 0$, $\eta \in [0, 0.5)$ and $\delta \in (0, 1)$. Let $\theta \in [0, \theta_{\max}(\gamma, \eta))$ such that $\varepsilon_{min}(\gamma, \eta, \theta) < 1$, $\varepsilon \in (\varepsilon_{\min}(\gamma, \eta, \theta), 1]$ and $D \in \mathcal{D}^{\gamma,\eta,\theta}$. Let $m' > 1$.* If *a sample $\mathcal{S}$ of size $m \geq m_1(m', \gamma, \theta, \varepsilon, \delta) = m' \frac{64^2}{2\left(1-\theta-\frac{\varepsilon\gamma}{64}\right)(\varepsilon\gamma)^2} \ln \frac{2}{\delta}$ is drawn from D* then*, with probability $1-\delta$:*

$$1. \left| \frac{1}{m} \sum_{x\in\mathcal{S}} \theta(\mathbf{x}) - E[\theta(\mathbf{x})] \right| \leq \frac{\varepsilon\gamma}{64} \qquad 2. \ |\{\mathbf{x} \in \mathcal{S} | \theta(\mathbf{x}) = 0\}| > m'.$$

Proof. Simple Chernoff bounds arguments prove the inequalities. (It suffices to observe that $\frac{1}{m}\sum_{x\in\mathcal{S}} \theta(\mathbf{x}) = \hat{E}[\theta(\mathbf{x})]$ and $\sum_{x\in\mathcal{S}} \theta(\mathbf{x}) = m - |\{\mathbf{x} \in \mathcal{S} | \theta(\mathbf{x}) = 0\}|$.) □

Definition 6 (CN-consistency on Mixed-Noise distributions). *Let $\gamma > 0, \eta \in [0, 0.5)$, $\theta \in [0, \theta_{\max}(\gamma, \eta))$. Let $D \in \mathcal{U}^{\gamma,\eta,\theta}$. Let $\mathbf{w}^* \in \mathcal{X}$.* If $Pr_{\mathbf{x}\sim D}[\mathbf{w}^* \cdot \mathbf{x} \leq \gamma | \theta(\mathbf{x}) = 0] = \eta$ then $\mathbf{w}^*$ *is said to be CN-consistent.*

The next lemma says how much the added malicious noise modify the sample averages on the CN part of a distribution.

Lemma 2. *Let $\gamma > 0, \eta \in [0, 0.5)$ and $\delta \in (0, 1]$. Let $\theta \in [0, \theta_{\max}(\gamma, \eta))$ such that $\varepsilon_{min}(\gamma, \eta, \theta) < 1 - 2\eta$, and $\varepsilon \in (\varepsilon_{\min}(\gamma, \eta, \theta), 1 - 2\eta]$. Let $D \in \mathcal{U}^{\gamma,\eta,\theta}$. Let $M(n, \gamma, \eta, \theta, \varepsilon, \delta) = m_1\left(m_0\left(10(n+1), 2, \frac{3\delta}{4}, \frac{\varepsilon\gamma}{16}\right), \gamma, \theta, \varepsilon, \frac{\delta}{4}\right)$ and $\mathbf{w}$ be a unit vector.* If *$\mathcal{S}$ is a sample of size $m > M(n, \gamma, \eta, \theta, \varepsilon, \delta)$ drawn from D* then*, with probability $1-\delta$, $\forall R \in [-1, 1]$:*

$$\left| \hat{E}[(\mathbf{w}\cdot\mathbf{x})\boldsymbol{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})] - E[(\mathbf{w}\cdot\mathbf{x})\boldsymbol{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})] \right| \leq \frac{\varepsilon\gamma}{8}$$

where $\boldsymbol{1}_{\leq R}(\alpha) = 1$ if $\alpha \leq R$ and 0 otherwise.

Proof. By Lemma 1, we know that $|\{\mathbf{x} \in \mathcal{S} | \theta(\mathbf{x}) = 0\}| > m_0\left(10(n+1), 2, \frac{3\delta}{4}, \frac{\varepsilon\gamma}{16}\right)$ with probability $1 - \frac{3\delta}{4}$. So, by Theorem 1, with probability $1 - \frac{3\delta}{4} - \frac{\delta}{4}$, $\forall R \in [-1, 1]$

$$\left| \hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x}) | \theta(\mathbf{x}) = 0] - E[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x}) | \theta(\mathbf{x}) = 0] \right| \leq \frac{\varepsilon\gamma}{16} \qquad (1)$$

In addition, we have

$$
\begin{aligned}
&\left|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})] - E[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})]\right| \\
&= \Big|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0]\,(Pr_{\mathbf{x}\in S}[\theta(\mathbf{x})=0] - Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x})=0]) \\
&\quad + \Big(\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0] - E[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0]\Big)\, Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x})=0] \\
&\quad + \hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1]\,(Pr_{\mathbf{x}\in S}[\theta(\mathbf{x})=1] - Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x})=1]) \\
&\quad + \Big(\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1] - E[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1]\Big)\, Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x})=1]\Big| \\
&\leq \Big|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0]\Big|\,|Pr_{\mathbf{x}\in S}[\theta(\mathbf{x})=0] - Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x})=0]| \\
&\qquad\qquad (\leq \tfrac{\varepsilon\gamma}{64} \text{ by lemma 1}) \\
&\quad + \Big|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0] - E[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0]\Big|\, Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x})=0] \\
&\qquad\qquad (\leq \tfrac{\varepsilon\gamma}{16} \text{ by equation 1}) \\
&\quad + \Big|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1]\Big|\,|Pr_{\mathbf{x}\in S}[\theta(\mathbf{x})=1] - Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x})=1]| \\
&\qquad\qquad (\leq \tfrac{\varepsilon\gamma}{64} \text{ by lemma 1}) \\
&\quad + \Big|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1] - E[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1]\Big|\, Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x})=1] \\
&\leq 1\times\frac{\varepsilon\gamma}{64} + \frac{\varepsilon}{16}(1-\theta) + 1\times\frac{\varepsilon\gamma}{64} + 2\theta \qquad \text{(with probability } 1-\delta) \\
&\leq \frac{6\varepsilon}{64} + 2\theta \leq 2\varepsilon \qquad \text{(according to the values of } \varepsilon_{\min} \text{ and } \theta_{\max})
\end{aligned}
$$

□

The following lemma shows that a CN-consistent vector $\mathbf{w}^*$ allows for a positive expectation of $\mathbf{w}^*\cdot\mathbf{x}$ over a Mixed-Noise distribution.

Lemma 3. *Let $\gamma > 0, \eta \in [0, 0.5), \theta \in [0, \theta_{\max}(\gamma,\eta))$. Suppose that $D \in \mathcal{U}^{\gamma,\eta,\theta}$.* If *$\mathbf{w}^*$ is CN-consistent on the CN-part of D, then $E[\mathbf{w}^*\cdot\mathbf{x}] \geq (1-2\eta)(1-\theta)\gamma - \theta$.*

Proof.

$$
\begin{aligned}
E[\mathbf{w}^*\cdot\mathbf{x}] &= E[\mathbf{w}^*\cdot\mathbf{x}|\theta(\mathbf{x})=0]\,Pr(\theta(\mathbf{x})=0) + E[\mathbf{w}^*\cdot\mathbf{x}|\theta(\mathbf{x})=1]\,Pr(\theta(\mathbf{x})=1) \\
&= E[\mathbf{w}^*\cdot\mathbf{x}|\theta(\mathbf{x})=0](1-\theta) + E[\mathbf{w}^*\cdot\mathbf{x}|\theta(\mathbf{x})=1]\,\theta \\
&\geq E[\mathbf{w}^*\cdot\mathbf{x}|\theta(\mathbf{x})=0](1-\theta) - \theta \geq (1-2\eta)(1-\theta)\gamma - \theta
\end{aligned}
$$

It is easy to check that the lower bound is strictly positive. □

We will make use of the following lemma due to Bylander and extend it to the case of Mixed-noise distributions.

Lemma 4 ([8])
Let $\gamma > 0, \eta \in [0, 0.5)$, $\varepsilon \in (0, 1-2\eta]$. Let $D \in \mathcal{U}^{\gamma,\eta}$. Let $\mathbf{w}$ be an arbitrary weight vector. If $\mathbf{w}^*$ is CN-consistent on D, and if $\mathbf{w}$ has accuracy $1-\eta-\varepsilon$, then:

$$(1-2\eta)\,E[(\mathbf{w}^*\cdot\mathbf{x})\boldsymbol{I}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})] + \eta E[\mathbf{w}^*\cdot\mathbf{x}] \geq \varepsilon\gamma \tag{2}$$

$$(1-2\eta)\,E[(\mathbf{w}\cdot\mathbf{x})\boldsymbol{I}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})] + \eta E[\mathbf{w}\cdot\mathbf{x}] \leq 0 \tag{3}$$

Lemma 5. *Let $\gamma > 0, \eta \in [0, 0.5)$ and $\delta \in (0, 1]$. Let $\theta \in [0, \theta_{\max}(\gamma,\eta))$ such that $\varepsilon_{min}(\gamma,\eta,\theta) < \frac{4(1-2\eta)}{3}$, and $\varepsilon \in (\varepsilon_{\min}(\gamma,\eta,\theta), \frac{4(1-2\eta)}{3}]$. Let $D \in \mathcal{U}^{\gamma,\eta,\theta}$. Let $\mathbf{w}$ be*

an arbitrary weight vector and $D \in \mathcal{U}^{\gamma,\eta,\theta}$. *If* $\mathbf{w}^*$ *is CN-consistent on the CN part of* D, *and if* $\mathbf{w}$ *has accuracy* $1-\eta-\frac{3\varepsilon}{4}$ *on the CN part of* D, *then the following holds:*

$$(1-2\eta)\,E\left[(\mathbf{w}^*\cdot\mathbf{x})\boldsymbol{I}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})\right]+\eta E\left[\mathbf{w}^*\cdot\mathbf{x}\right]\geq\frac{5\varepsilon\gamma}{8} \tag{4}$$

$$(1-2\eta)\,E\left[(\mathbf{w}\cdot\mathbf{x})\boldsymbol{I}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})\right]+\eta E\left[\mathbf{w}\cdot\mathbf{x}\right]\ \leq\eta\theta \tag{5}$$

Proof. For the first inequality, we have:

$$\begin{aligned}
&(1-2\eta)\,E\left[(\mathbf{w}^*\cdot\mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})\right]+\eta E\left[\mathbf{w}^*\cdot\mathbf{x}\right]\\
&\quad=(1-2\eta)\,E\left[(\mathbf{w}^*\cdot\mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1\right]Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad+\eta E\left[\mathbf{w}^*\cdot\mathbf{x}|\theta(\mathbf{x})=1\right]Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad+(1-2\eta)\,E\left[(\mathbf{w}^*\cdot\mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0\right]Pr\left[\theta(\mathbf{x})=0\right]\\
&\qquad+\eta E\left[\mathbf{w}^*\cdot\mathbf{x}|\theta(\mathbf{x})=0\right]Pr\left[\theta(\mathbf{x})=0\right]\\
&\quad\geq(1-\theta)\,\frac{3}{4}\varepsilon\gamma \qquad\text{(by lemma 4 eq. 2)}\\
&\qquad+(1-2\eta)\,E\left[(\mathbf{w}^*\cdot\mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1\right]Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad+\eta E\left[\mathbf{w}^*\cdot\mathbf{x}|\,\theta(\mathbf{x})=1\right]Pr\left[\theta(\mathbf{x})=1\right]\\
&\quad\geq(1-\theta)\,\frac{3}{4}\varepsilon\gamma-(1-2\eta)\,\theta-\eta\theta\\
&\quad\geq(1-\theta)\,\frac{3}{4}\varepsilon\gamma-(1-\eta)\,\theta\geq\frac{5\varepsilon\gamma}{8} \qquad\text{(by definition of }\varepsilon)
\end{aligned}$$

For the second inequality, we have:

$$\begin{aligned}
&(1-2\eta)\,E\left[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})\right]+\eta E\left[\mathbf{w}\cdot\mathbf{x}\right]\\
&\quad=(1-2\eta)\,E\left[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1\right]Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad+\eta E\left[\mathbf{w}\cdot\mathbf{x}|\theta(\mathbf{x})=1\right]Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad+(1-2\eta)\,E\left[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0\right]Pr\left[\theta(\mathbf{x})=0\right]\\
&\qquad+\eta E\left[\mathbf{w}\cdot\mathbf{x}|\theta(\mathbf{x})=0\right]Pr\left[\theta(\mathbf{x})=0\right]\\
&\quad\leq 0 \qquad\text{(by lemma 4 eq.3)}\\
&\qquad+(1-2\eta)\,E\left[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1\right]Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad+\eta E\left[\mathbf{w}\cdot\mathbf{x}|\,\theta(\mathbf{x})=1\right]Pr\left[\theta(\mathbf{x})=1\right]\leq 0+\eta\theta
\end{aligned}$$

□

We now state our core lemma. It says that, with high probability, Algorithm 2 outputs a vector that can be used as an update vector in the Perceptron algorithm (cf. Fig. 1), that is a vector erroneously classified by the current classifier but correctly classified by the target hyperplane (i.e. the vector is noise free). Calling Algorithm 2 iteratively makes it possible to learn a separating hyperplane from a mixed-noise distribution.

Lemma 6. *Let* $\gamma>0, \eta\in[0,0.5)$ *and* $\delta\in(0,1)$. *Let* $\theta\in[0,\theta_{\max}(\gamma,\eta))$ *such that* $\varepsilon_{\min}(\gamma,\eta,\theta)<\frac{4}{3}(1-\eta)$. *Let* $D\in\mathcal{U}^{\gamma,\eta,\theta}$ *and* $\mathbf{w}^*$ *be the target hyperplane (CN-consistent on the CN-part of* D*).* $\forall\varepsilon\in\left[\varepsilon_{\min}(\gamma,\eta,\theta),\frac{4}{3}(1-\eta)\right)$, *for all input samples* $\mathcal{S}$ *of size* $M(n,\gamma,\eta,\theta,\delta,\varepsilon)$, *with probability at least* $1-\delta$, $\forall\mathbf{w}\in\mathcal{X}$ if $\mathbf{w}$ *has accuracy at most* $1-\eta-\frac{3\varepsilon}{4}$ *on the CN-part of* D then *Cnoise-update (Algorithm 2), when given inputs* $\mathcal{S}$, $\mathbf{w}$, $\frac{\varepsilon\gamma}{4}$, *outputs a vector* $\mathbf{z}$ *such that* $\mathbf{w}\cdot\mathbf{z}\leq 0$ *and* $\mathbf{w}^*\cdot\mathbf{z}\geq\frac{\varepsilon\gamma}{4}$.

Proof. The projection step guarantees that $\mathbf{w}\cdot\mathbf{z} \leq 0$. We focus on the second inequality.

Case 1. Suppose that $\mathbf{w}\cdot\boldsymbol{\mu} < \|\mathbf{w}\|\frac{\varepsilon\gamma}{4}$: $\mathbf{z}$ is set to $\boldsymbol{\mu}$ by the algorithm, and, if needed, is projected on the $\mathbf{w}$ hyperplane.

Every linear threshold function has accuracy at least η on the CN-part of D, so an overall accuracy at least $(1-\theta)\eta$. $\mathbf{w}$ has accuracy on the CN-part of D of, at most, $1-\eta-\frac{3\varepsilon}{4}$ and so an overall accuracy at most of $1-(1-\theta)\left(\eta+\frac{3\varepsilon}{4}\right)+\theta$.

It is easy to check that

$$1-(1-\theta)\left(\frac{3\varepsilon}{4}+\eta\right)+\theta \geq (1-\theta)\eta \Leftrightarrow (1-2\eta)(1-\theta)\gamma - \theta \geq (1-\theta)\frac{3\varepsilon}{4}\gamma - (2\gamma+1)\theta,$$

and thus, from Lemma 3, $E[\mathbf{w}^*\cdot\mathbf{x}] \geq (1-\theta)\frac{3\varepsilon}{4}\gamma - (2\gamma+1)\theta$. Because $\theta < \theta_{\max}(\gamma,\eta)$ and $\varepsilon > \varepsilon_{\min}(\gamma,\eta,\theta)$, we have $E[\mathbf{w}^*\cdot\mathbf{x}] \geq \frac{5\varepsilon\gamma}{8}$. Because of Lemma 2 and because $|\mathcal{S}| \geq M(n,\gamma,\eta,\theta,\delta,\varepsilon)$, we know that $\mathbf{w}^*\cdot\mathbf{z}$ is, with probability $1-\delta$, within $\frac{\varepsilon\gamma}{8}$ of its expected value on the entire sample; hence we can conclude that $\mathbf{w}^*\cdot\boldsymbol{\mu} \geq \frac{\varepsilon\gamma}{2}$.

If $\mathbf{w}\cdot\boldsymbol{\mu} < 0$, then the lemma follows directly.

If $0 < \mathbf{w}\cdot\boldsymbol{\mu} < \|\mathbf{w}\|\frac{\varepsilon\gamma}{4}$, then $\mathbf{z}$ is set to $\boldsymbol{\mu}$ and, if needed, projected to $\mathbf{w}$. Let $\mathbf{z}_{\|} = \boldsymbol{\mu} - \mathbf{z}$ ($\mathbf{z}_{\|}$ is parallel to $\mathbf{w}$). It follows that

$$\begin{aligned}\mathbf{w}^*\cdot\boldsymbol{\mu} \geq \frac{\varepsilon\gamma}{2} &\Leftrightarrow \mathbf{w}^*\cdot\mathbf{z} + \mathbf{w}^*\cdot\mathbf{z}_{\|} \geq \frac{\varepsilon\gamma}{2} \Rightarrow \mathbf{w}^*\cdot\mathbf{z} \geq \frac{\varepsilon\gamma}{2} - \|\mathbf{z}_{\|}\| \Rightarrow \mathbf{w}^*\cdot\mathbf{z} \geq \frac{\varepsilon\gamma}{2} - \|\boldsymbol{\mu}\| \\ &\Rightarrow \mathbf{w}^*\cdot\mathbf{z} \geq \frac{\varepsilon\gamma}{4}.\end{aligned}$$

And the lemma again follows.

Case 2. Suppose instead that $\mathbf{w}\cdot\boldsymbol{\mu} \geq \|\mathbf{w}\|\frac{\varepsilon\gamma}{4}$. Let $a \geq 0$ and $b \geq 0$ be chosen so that $a\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\boldsymbol{\mu}' + b\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\boldsymbol{\mu} = \frac{\varepsilon\gamma}{4}$ and $a+b=1$. $\mathbf{w}\cdot\boldsymbol{\mu}'$ is negative and $\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\boldsymbol{\mu} \geq \frac{\varepsilon\gamma}{4}$ in this case, so such an a and b can always be chosen. Note that in this case, Cnoise-update sets $\mathbf{z}$ to $a\boldsymbol{\mu}' + b\boldsymbol{\mu}$ and then projects $\mathbf{z}$ to the $\mathbf{w}$ hyperplane. Because $\mathbf{w}\cdot\mathbf{z} = \|\mathbf{w}\|\frac{\varepsilon\gamma}{4}$ before $\mathbf{z}$ is projected to the $\mathbf{w}$ hyperplane, then the projection will decrease $\mathbf{w}^*\cdot\mathbf{z}$ by at most $\frac{\varepsilon\gamma}{4}$ (recall that $\mathbf{w}^*$ is a unit vector).

Note that $a\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\boldsymbol{\mu}' + b\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\boldsymbol{\mu} = a\hat{E}\left[\left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\mathbf{x}\right)\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})\right] + b\hat{E}\left[\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\mathbf{x}\right]$. Because, by lemma 2, sample averages are, with probability $1-\delta$, within $\frac{\varepsilon\gamma}{8}$ of their expected values, it follows that

$$aE\left[\left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\mathbf{x}\right)\mathbf{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})\right] + bE\left[\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\mathbf{x}\right] \geq \frac{\varepsilon\gamma}{8}.$$

Lemma 5 implies that $a' = \frac{\eta}{1-\eta}$ and $b' = \frac{1-2\eta}{1-\eta}$ results in $a'E\left[\left(\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\mathbf{x}\right)\mathbb{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})\right] + b'E[\frac{\mathbf{w}}{\|\mathbf{w}\|}\cdot\mathbf{x}] \leq \frac{\eta\theta}{1-\eta}$ and so less than $\frac{\varepsilon\gamma}{8}$. So, it must be the case when $a \leq \frac{\eta}{1-\eta}$ because a larger a would result in an expected value less than $\frac{\varepsilon\gamma}{8}$ and a sample average less than $\frac{\varepsilon\gamma}{4}$.

Lemma 5 also implies that choosing $a' = \frac{\eta}{1-\eta}$ and $b' = \frac{1-2\eta}{1-\eta}$ results in $a'E[(\mathbf{w}^*\cdot\mathbf{x})\mathbb{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})] + b'E[\mathbf{w}^*\cdot\mathbf{x}] \geq \frac{5\varepsilon\gamma}{8}$

Because $a' \geq a$ and $b' \leq b$, and because Lemma 3 implies $E[\mathbf{w}^* \cdot \mathbf{x}] \geq \frac{5\varepsilon\gamma}{8}$, it follows that $aE[(\mathbf{w}^* \cdot \mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w} \cdot \mathbf{x})] + bE[\mathbf{w}^* \cdot \mathbf{x}] \geq \frac{5\varepsilon\gamma}{8}$ and $a\mathbf{w}^* \cdot \boldsymbol{\mu}' + b\mathbf{w}^* \cdot \boldsymbol{\mu} \geq \frac{\varepsilon\gamma}{2}$.

Thus, when $\mathbf{z}$ is projected onto hyperplane $\mathbf{w}$, $\mathbf{w}^* \cdot \mathbf{z} \geq \frac{\varepsilon\gamma}{4}$ and $\mathbf{w} \cdot \mathbf{z} = 0$. Consequently a total of m examples, implies , with probability $1 - \delta$, that $\mathbf{w}^* \cdot \mathbf{z} \geq \frac{\varepsilon\gamma}{4}$ and $\mathbf{w} \cdot \mathbf{z} \leq 0$ for the $\mathbf{z}$ computes by Cnoise-update. This proves the Lemma. □

We finally have the Theorem 4 for Mixed-Noise learnability using Cnoise-update.

Theorem 4. *Let $\gamma > 0, \eta \in [0, 0.5)$ and $\delta \in (0, 1)$. Let $\theta \in [0, \theta_{\max}(\gamma, \eta))$ such that $\varepsilon_{\min}(\gamma, \eta, \theta) < 1-2\eta$. Let $D \in \mathcal{U}^{\gamma,\eta,\theta}$ and $\mathbf{w}^*$ be the target hyperplane (CN-consistent on the CN-part of D). $\forall \varepsilon \in (\varepsilon_{\min}(\gamma, \eta, \theta), 1-2\eta], \forall \mathbf{w} \in \mathcal{X}$, when given inputs $\mathcal{S}$ of size at least $M(n, \gamma, \eta, \theta, \delta, \varepsilon)$,* if *the Perceptron algorithm uses update vectors from CNoise update for more than $\frac{16}{\varepsilon^2\gamma^2}$ updates, then the $\mathbf{w}_i$ with the highest sample accuracy on the CN-part has accuracy on the CN-part of D at least $1 - \eta - \varepsilon$ with probability $1 - \delta$.*

Proof. By lemma 6, with probability $1-\delta$, whenever $\mathbf{w}_i$ has accuracy at most $1-\eta-\frac{3\varepsilon}{4}$ on the CN-part of S then Cnoise-update$(X, \mathbf{w}_i, \frac{\varepsilon\gamma}{16})$ will return an update vector $\mathbf{z}_i$ such that $\mathbf{w}^* \cdot \mathbf{z}_i \geq \frac{\varepsilon\gamma}{4}$ and $\mathbf{w}_i \cdot \mathbf{z}_i \leq 0$. The length of a sequence $(\mathbf{z}_1, \ldots, \mathbf{z}_l)$ where each $\mathbf{z}_i$ has $\frac{\varepsilon\gamma}{4}$ separation, is at most $\frac{16}{(\varepsilon\gamma)^2}$ [15,16]. Thus, if more than $\frac{16}{(\varepsilon\gamma)^2}$ update vectors are obtained, then at least one update vector must have less than $\frac{\varepsilon\gamma}{4}$ separation, which implies at least one $\mathbf{w}$ has more than $1 - \eta - \frac{3\varepsilon\gamma}{4}$ accuracy on CN-part.

The sample accuracy of $\mathbf{w}_i$ corresponds to the sample average of an indicator function. By Theorem 1, the indicator functions are covered with probability $1 - \delta$. So, assuming that the situation is in the $1-\delta$ region, the sample accuracy of each $\mathbf{w}_i$ on the CN-part of the distribution will be within $\frac{\varepsilon\gamma}{16}$ of its expected value. Since at least one $\mathbf{w}_i$ will have $1 - \eta - \frac{3\varepsilon}{4}$ accuracy on the CN-part, this implies that its sample accuracy on the CN-part is at least $1 - \eta - \frac{13\varepsilon}{16}$. The accuracy on the distribution is more than $1 - (1 - \theta)\left(\eta - \frac{13\varepsilon}{16}\right) - \theta < 1 - (1 - \theta)\left(\eta - \frac{13\varepsilon}{16}\right) - \frac{\varepsilon}{32}$. Any other w_i with a better sample accuracy will have accuracy of at least $1 - (1 - \theta)\left(\eta - \frac{13\varepsilon}{16}\right) - \frac{5\varepsilon}{32}$ and so an accuracy on the CN-part of at least $1 - \eta - \varepsilon$. □

Remark 4. An interpretation of the latter result is that distributions from $\mathcal{D}^{\gamma,\varepsilon}$, for $\varepsilon > 0$ can also be learned if corrupted by classification noise. The extent to which the learning can take place of course depends on the value of ε (which would play the role of θ in the derivation made above).

In the next section, we show how random projections can help us reduce a problem of learning from a possibly infinite dimensional CN distribution to a problem of finite Mixed-Noise distribution where the parameters of the Mixed-Noise distribution can be controlled. This will directly give a proof to Theorem 2.

3.2 Random Projections and Separable Distributions

Here, we do not make the assumption that $\mathcal{X}$ is finite-dimensional.

Theorem 5 ([13]). *Let $D \in \mathcal{D}^{\gamma,0}$. For a random sample $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ from D, let $\pi(\mathcal{S}) : \mathcal{X} \to span\langle\mathcal{S}\rangle$ be the orthogonal projection on the space spanned by $\mathbf{x}_1, \ldots, \mathbf{x}_n$.*

If *a sample $\mathcal{S}$ of size $n \geq \frac{8}{\theta}[\frac{1}{\gamma^2} + \ln \frac{1}{\delta}]$ is drawn according to D* then *with probability at least $1-\delta$, the mapping $\pi = \pi(\mathcal{S})$ is such that $\exists \mathbf{w}\ Pr_{\mathbf{x}\sim D}[\mathbf{w} \cdot \pi(\mathbf{x}) > \gamma/2] < \theta$ on span$\langle \mathcal{S} \rangle \subseteq \mathcal{X}$.*

This theorem says a random projection can transform a linearly separable distribution into an *almost linearly separable* one defined in a finite dimensional space. We can therefore consider that such a transformation incurs a *projection noise*; this noise should possess some exploitable regularities for learning, but we leave the characterization of these regularities for a future work and apprehend in the sequel this projection noise as malicious. In RP-classifier, the vectors used to define π are selected randomly within the training set.

Corollary 1 (of Theorem 2). *Let $\gamma > 0, \eta \in [0, 0.5)$ and $D \in \mathcal{U}^{\gamma,\eta}$. $\forall \varepsilon \in (0, 1-2\eta], \forall \delta \in (0,1]$,* if *a sample $\mathcal{S}$ of $m > M(\frac{K}{\varepsilon\gamma(1-2\eta)}\left[\frac{1}{\gamma^2} + \ln \frac{2}{\delta}\right], \frac{\gamma}{2}, \eta, \frac{\delta}{2}, \frac{\varepsilon}{2})$ examples drawn from D is input to RP-classifier,* then *with probability $1-\delta$ RP-classifier outputs a classifier with accuracy at least $1-\eta-\varepsilon$. ($K > 0$ is a universal constant.)*

Proof. Fix γ, η, $D \in \mathcal{U}^{\gamma,\eta}$ and ε. Fix $\theta = \frac{\gamma\varepsilon(1-2\eta)}{2080}$.

First, it is straightforward to check that $\theta \leq \theta_{\max}(\gamma, \eta)$, $\varepsilon_{min} \leq \min(\frac{\varepsilon}{2}, 1-2\eta)$ and, since $\theta \leq \varepsilon_{\min}(\gamma, \eta, \theta)$, $\theta \leq \frac{\varepsilon}{2}$. (The assumptions of Theorem 4 hold true.)

By Theorem 5, choosing $n = \frac{8}{\theta}[\frac{1}{\gamma^2} + \ln \frac{2}{\delta}]$ guarantees with probability $1 - \frac{\delta}{2}$, that the projection D' of D onto a random subspace of dimension n is a distribution having a CN part of weight $1-\theta$ and another part of weight θ corrupted by projection noise. D' can therefore be considered as an element of $\mathcal{U}^{\frac{\gamma}{2},\eta,\theta}$[4].

By Theorem 4, using m examples (with m set as in the Theorem) allows with probability $1 - \frac{\delta}{2}$ the learning algorithm that iteratively calls Cnoise-update to return in polynomial time a classifier with accuracy at least $\frac{\varepsilon}{2}$ on the CN-part of the distribution.

Therefore, the accuracy of the classifier on the examples drawn from D is, with probability $1 - \frac{\delta}{2} - \frac{\delta}{2} = 1 - \delta$, at least $1 - (1-\theta)\frac{\varepsilon}{2} - \theta \geq 1 - \frac{\varepsilon}{2} - \frac{\delta}{2} = 1 - \delta$. Theorem 2 now follows. □

Remark 5. We could also learn with an initial malicious noise θ_{init} less than $\theta_{\max}$. In this case, the maximum amount of noise added by random projections must obviously be less than $\theta_{\max} - \theta_{\text{init}}$.

Remark 6. Random projections based on the Johnson-Lindenstrauss lemma could be directly combined with a CN-noise tolerant perceptron to achieve the same kind of learnability results. It however requires numerous data resamplings and the resulting sample and time complexities are very high.

4 Related Work

Learning from a noisy sample of data implies that the linear problem at hand might not necessarily be consistent, that is, some linear constraints might contradict others. In that

[4] The choices of θ and n give $K = 2080 \times 8$.

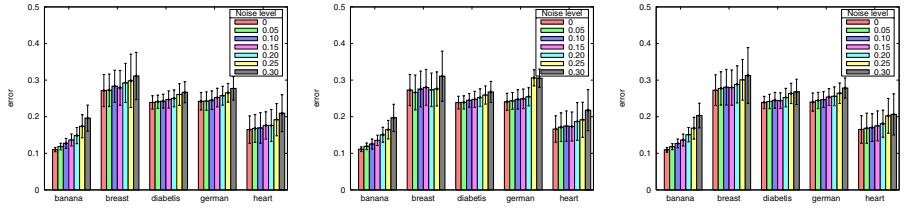

Fig. 2. Error rates on UCI datasets with random projections, KPCA and KGS projection with different amount of classification noise; 1-standard deviation error bars are shown

case, as stated before, the problem at hand boils down to that of finding an approximate solution to a linear program such that a minimal number of constraints are violated, which is know as an NP-hard problem (see, e.g., [17]).

In order to cope with this problem, and leverage the classical perceptron learning rule to render it tolerant to noise classification, one line of approaches has mainly been exploited. It relies on exploiting the statistical regularities in the studied distribution by computing various sample averages as it is presented here; this makes it possible to 'erase' the classification noise. As for Bylander's algorithms [5,8], whose analysis we have just extended, the other notable contributions are those of [6] and [7]. However, they tackle a different aspect of the problem of learning noisy distributions and are more focused on showing that, in finite dimensional spaces, the running time of their algorithms can be lowered to something that depends on $\log {}^{1}/_{\gamma}$ instead of ${}^{1}/_{\gamma}$.

Regarding the use of kernel projections to tackle classification problems, the *Kernel Projection Machine* of [18] has to be mentioned. It is based on the use of Kernel PCA as a feature extraction step. The main points of this interesting work are a proof on the regularizing properties of KPCA and the fact that it gives a practical model selection procedure. However, the question of learning noisy distributions is not addressed.

Freund and Schapire [19] provide data-dependent bounds for the voted kernel perceptron that support some robustness against outliers. However, as for SVM, it is not clear whether this algorithm is tolerant to 'systematic' uniform classification noise.

Cesa-Bianchi and al. in [20] propose bounds for online perceptron on non-separable data. However, the authors specifiy that their algorithms tolerate only a low rate of non-linearly separable examples and thus are not valid for uniform classification noise.

Finally, the empirical study of [21] provides some insights on how random projections might be useful for classification. No sample and running time complexity results are given and the question of learning with noise is not addressed.

5 Numerical Simulations

UCI Datasets. We carried out numerical simulations on benchmark datasets from the UCI repository preprocessed and made available by Gunnar Rätsch[5]. For each problem (Banana, Breast Cancer, Diabetes, German, Heart), we have 100 training and 100 test

[5] http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm

samples. All these problems contain a few hundred training examples, which is far from what the theoretical results require to get interesting accuracy and confidence.

We have tested three projection procedures: random, Kernel PCA (KPCA), Kernel Gram-Schmidt (KGS) [22]. This latter projection is sometimes referred to as a 'sparse version of Kernel PCA' (note that KPCA and KGS are deterministic projections and that RP-classifier is not a random-projection learning algorithm anymore). In order to cope with the non separability of the problems, we have used Gaussian kernels, and thus infinite-dimensional spaces, whose widths have been set to the best value for SVM classification as reported on Gunnar Rätsch's website.

In our protocol, we have corrupted the data with classification noises of rates 0.0, 0,05, 0.10, 0.15, 0.20, 0.25, 0.30. Instead of carrying out a cumbersome cross-validation procedure, we provide the algorithm RP-classifier with the actual value of η.

To determine the right projection size we resort to the same cross-validation procedure as in [23], trying subspace sizes of 2 to 200. The results obtained are summarized on Figure 2. We observe that classifiers produced on a dataset with no extra noise have an accuracy a little lower than that of the classifiers tested by Gunnar Rätsch, with a very reasonable variance. We additionally note that, when the classification noise amount artificially grows, the achieved accuracy decreases very weakly and the variance grows rather slowly. It is particularly striking since again, the sample complexities used are far from meeting the theoretical requirements; moreover, it is interesting to see that the results are good even if no separation margin exists. We can also note that when the actual values of the accuracies (not reported here for sake of space) are compared, KGS and KPCA roughly achieve the same accuracies and both are a little (not significantly though) better than random projection. Eventually, the main conclusion from the numerical simulations is that RP-classifier has a very satisfactory behavior on real data.

Toy Problems. We have carried out additional simulations on five 2-dimensional toy problems. Due to space limitations however, we only discuss and show the learning results for three of them[6] (cf. Figure 3). Here, we have used the KGS projection since due to the uniform distribution of points on $[-10; 10] \times [-10; 10]$, random projections provide exactly the same results. For each problem, we have produced 50 train sets and 50 test sets of 2000 examples each. Note that we do not impose any separation margin.

We have altered the data with 5 different amounts of noise (from 0.0 to 0.40), 12 Gaussian kernel width (from 10.0 to 0.25) and 12 projection dimensions (from 5 to 200) have been tested and for each problem and for each noise rate, we have selected the couple which minimizes the error rate of the produced classifier (proceeding as above). Figure 3 depicts the learning results obtained with a noise rate of 0.20.

The essential point showed by these simulations is that, again, RP-classifier is very effective in learning from noisy nonlinear distributions. Numerically (the numerical results are not reported here due to space limitations), we have observed that our algorithm can tolerate noise levels as high as 0.4 and still provide small error rates (around 10%). Finally, our simulations show that the algorithm is tolerant to classification noise and thus illustrate our theoretical results, while extending already existing experiments to this particular framework of learning.

[6] Full results are available at http://hal.archives-ouvertes.fr/hal-00137941

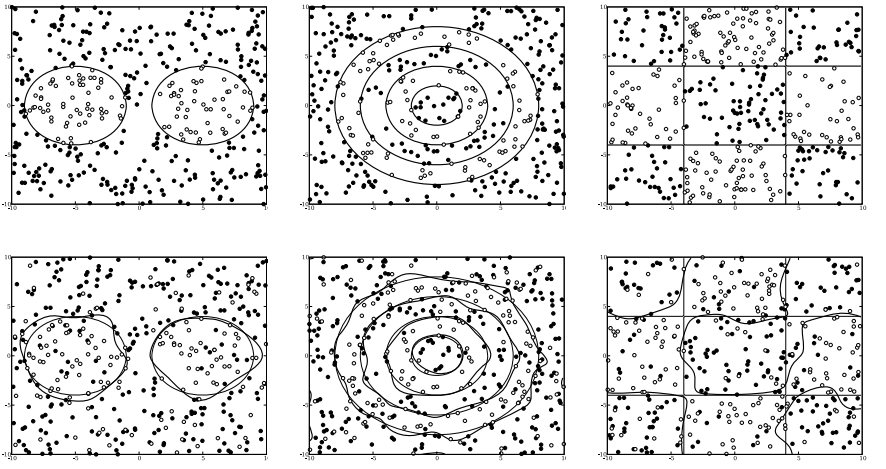

Fig. 3. Toy problems: first row show the clean concepts with black disks being of class +1 and white ones of class -1. Second row shows the concept learned by RP-classifier with a uniform classification noise rate of 0.20 and KGS projection.

6 Conclusion and Outlook

In this paper, we have given theoretical results on the learnability of kernel perceptrons when faced to classification noise. The keypoint is that this result is independent of the dimension of the kernel feature space. In fact, it is the use of finite-dimensional projections having good generalization that allows us to transform a possibly infinite dimensional problem into a finite dimension one that, in turn, we tackle with Bylander's noise tolerant perceptron algorithm. This algorithm is shown to be robust to some additional 'projection noise' provided the sample complexity are adjusted in a suitable way. A better characterization of the projection noise, more intelligent than 'malicious', could, in a future work, allow us to use projection dimensions appreciably smaller. Several simulation results support the soundness of our approach. Note that the random projection procedure using Johnson-Lindenstrauss lemma, described in [13], could be associated with RP-learn and would lead to lower sample and time complexities for the perceptron learning step.

Several questions are raised by this work. Among them, the question about the generalization properties of the Kernel Gram-Schmidt projector: we think tight generalization bounds can be exhibited in the framework of PAC Bayesian bounds, by exploiting, in particular, the sparseness of this projection. Resorting again to the PAC Bayesian framework it might be interesting to work on generalization bound on noisy projection classifiers, which would potentially provide a way to automatically estimate a reasonable projection dimension *and* noise level. Finally, we have been recently working on the harder problem of learning optimal separating hyperplane from noisy distributions.

References

1. Schölkopf, B., Smola, A.J.: Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond. MIT University Press, Cambridge (2002)
2. Vapnik, V.: The nature of statistical learning theory. Springer, New York (1995)
3. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
4. Angluin, D., Laird, P.: Learning from Noisy Examples. Machine Learning 2 (1988)
5. Bylander, T.: Learning Linear Threshold Functions in the Presence of Classification Noise. In: Proc. of 7^{th} Ann. Work. on Computat. Learning Theory., pp. 340–347 (1994)
6. Blum, A., Frieze, A.M., Kannan, R., Vempala, S.: A Polynomial-Time Algorithm for Learning Noisy Linear Threshold Functions. In: Proc. of 37th IEEE Symposium on Foundations of Computer Science, pp. 330–338. IEEE Computer Society Press, Los Alamitos (1996)
7. Cohen, E.: Learning Noisy Perceptrons by a Perceptron in Polynomial Time. In: Proc. of 38th IEEE Symposium on Foundations of Computer Science, pp. 514–523. IEEE Computer Society Press, Los Alamitos (1997)
8. Bylander, T.: Learning Noisy Linear Threshold Functions (1998) (submitted to journal)
9. Rosenblatt, F.: The Perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65, 386–407 (1958)
10. Graepel, T., Herbrich, R., Williamson, R.C.: From Margin to Sparsity. In: Adv. in Neural Information Processing Systems, vol. 13, pp. 210–216 (2001)
11. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons, inc., West Sussex, England (1998)
12. Ralaivola, L., Denis, F., Magnan, C.N.: CN=CPCN. In: Proc. of the 23rd Int. Conf. on Machine Learning (2006)
13. Balcan, M.F., Blum, A., Vempala, S.: Kernels as Features: on Kernels, Margins, and Low-dimensional Mappings. In: Ben-David, S., Case, J., Maruoka, A. (eds.) ALT 2004. LNCS (LNAI), vol. 3244, Springer, Heidelberg (2004)
14. Kearns, M., Li, M.: Learning in the presence of malicious errors. SIAM Journal on Computing 22(4), 807–837 (1993)
15. Block, H.D.: The perceptron: A model for brain functioning. Reviews of Modern Physics 34, 123–135 (1962)
16. Novikoff, A.B.J.: On convergence proofs on perceptrons. In: Proc. of the Symp. on the Mathematical Theory of Automata, pp. 615–622 (1962)
17. Amaldi, E., Kann, V.: On the approximability of some NP-hard minimization problems for linear systems. Electronic Colloquium on Computational Complexity (ECCC) 3(015) (1996)
18. Zwald, L., Vert, R., Blanchard, G., Massart, P.: Kernel projection machine: a new tool for pattern recognition. In: Adv. in Neural Information Processing Systems, vol. 17 (2004)
19. Freund, Y., Schapire, R.E.: Large Margin Classification Using the Perceptron Algorithm. Machine Learning 37(3), 277–296 (1999)
20. Cesa-Bianchi, N., Conconi, A., Gentile, C.: On the generalization ability of online learning algorithms. IEEE Transactions on Information Theory 50(9), 2050–2057 (2004)
21. Fradkin, D., Madigan, D.: Experiments with random projections for machine learning. In: Proc. of the 9th ACM SIGKDD int. conf. on Knowledge discovery and data mining, ACM Press, New York (2003)
22. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
23. Rätsch, G., Onoda, T., Müller, K.R.: Soft Margins for AdaBoost. Machine Learning 42, 287–320 (2001)