Learning Kernel Perceptrons on Noisy Data and Random Projections

Guillaume Stempfel, Liva Ralaivola

Learning Kernel Perceptrons on Noisy Data and Random Projections

Guillaume Stempfel, 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.
Keywords: Kernel Classifier, Random Projections, Classification Noise, Perceptron

1 Introduction

For a couple of years, it has been known that kernel methods (Schölkopf & Smola, 2002) provide a set of efficient techniques and associated models for, among others, classification. In addition, strong theoretical results (see, e.g. (Vapnik, 1995; Cristianini & Shawe-Taylor, 2000)), mainly based on *margin* criteria and the fact they constitute a generalization of the well-studied class of linear separators, support the relevance of their use.

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 (Angluin & Laird, 1988), 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 (Bylander, 1994; Blum *et al.*, 1996; Cohen, 1997; Bylander, 1998).

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 will be 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 classification noise model assumed. 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 synthetical 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 perceptron)**.** *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 (Rosenblatt, 1958) (cf. Fig. 1) is a well-studied greedy strategy to derive a linear classifier from a sample $\mathcal{S} = \{(\mathbf{x}_1, y_1) \dots (\mathbf{x}_m, y_m)\}$ of m labeled pairs $(\mathbf{x}_i, y_i)$ from $\mathcal{X} \times \mathcal{Y}$. which are 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 an hyperplane $\mathbf{w}$ that correctly separates $\mathcal{S}$ and might with high probability exhibit good generalization properties (Graepel *et al.*, 2001).

Input: $\mathcal{S} = \{(\mathbf{x}_1, y_1) \dots (\mathbf{x}_m, y_m)\}$
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 $\mathbf{w}$

Figure 1: Perceptron algorithm.

We are interested in the possibility of learning linearly separable distributions on which a random *uniform classification noise* process, denoted as CN (Angluin & Laird, 1988), has been applied, that is, distributions where correct labels are flipped with some given probability η. In order to solve this problem, Bylander (1994) has proposed

[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 $Pr(y = 1|\mathbf{x}) = 0$, but 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, $\mathbf{w}$

a simple algorithmic strategy later exploited by Blum *et al.* (1996): 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 $1 - \delta$ (for $\delta \in (0, 1)$) 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 $1/(1 - 2\eta)$.

Theorem 1 (Vapnik (1998))**.** 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*

$$M \geq m_0(h, R, \delta, \varepsilon) = \frac{8R^2 \left(2h \ln \frac{4R}{\varepsilon} + \ln \frac{9}{\delta}\right)}{\varepsilon^2}$$

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

In light of what we have just seen, 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 instanciation 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 know 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 seek 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 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 $\mathcal{U}^{\gamma,\eta}$ (Angluin & Laird, 1988))**.** *For* $\eta \in [0, 0.5)$*, let the random transformation* U^η *which maps an example* $\mathbf{x}$ *to* $-\mathbf{x}$ *with probability* η *and leaves it unchanged with probability* $1 - \eta$*.*

The set of distributions $\mathcal{U}^{\gamma,\eta}$ *is defined as* $\mathcal{U}^{\gamma,\eta} := U^\eta(\mathcal{D}^{\gamma,0})$*.*

We can now state our main result:

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

$\forall \varepsilon \in (0,1), \forall \delta \in (0,1), \forall \gamma > 0, \forall \eta \in [0, 0.5), \forall D \in \mathcal{D}^{\gamma,0}$*,* 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 Learning 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 Bylander (1998) and on properties of simple random projections proved by Balcan *et al.* (2004) is capable of efficiently learning CN distributions (Dee definition 2) independently of the dimension of the input space.

The proof works in two steps. First, in section 3.1, we show that Cnoise-update (see Algorithm 2) in finite dimension can tolerate a small amount of *malicious noise* and still return relevant update vectors. Then, in section 3.2, thanks to properties of random projections (see (Balcan *et al.*, 2004)) we show that random projections can be efficiently used to transform a CN noisy problem into one that meets the requirements of Cnoise-update (and Theorem 4 below).

3.1 Perceptron Learning with Mixed Noise

As said earlier, we suppose in this subsection that $\mathcal{X}$ if of finite dimension n. We will make use of the following definitions.

Algorithm 2 Cnoise-Update (Bylander, 1998)

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

Definition 3 (Sample and population accuracies)**.** *Let* $\mathbf{w}$ *a unit vector,* D *a distribution on* $\mathcal{X}$ *and* $\mathcal{S}$ *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 weight vector* $\mathbf{w}^*$ *is CN-consistent on* $D \in \mathcal{U}^{\gamma,\eta}$ if $Pr_{\mathbf{x}\sim D}\left[\mathbf{w}^*\cdot\mathbf{x} < \gamma\right] = \eta$*. This means that* $\mathbf{w}$ *makes* no error *on the noise free version of* D*.*

We recall that according to the following theorem (Bylander, 1998), Cnoise-updaate, depicted in Algorithm 2, when used in a perceptron-like iterative procedure, renders the learning of CN-distribution possible in finite dimension.

Theorem 3 (Bylander (1998))**.** *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 that 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* (Kearns & Li, 1993), i.e. a noise process whose statistical properties

[2]Here, and for the remaining of the paper, ε is not the usual error parameter ε' used in PAC, but $\varepsilon'(1-2\eta)$.

cannot be exploited in learning (this is an 'uncompressible' 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 Bylander (1998) 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 which leaves an input $\mathbf{x}$ unchanged with probability $1-\theta$ and changes 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 $\mathcal{U}^{\gamma,\eta,\theta} := U^\theta\left(U^\eta(\mathcal{D}^{\gamma,0})\right)$.

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 $E\left[f(\mathbf{x})\right]$ for $E_{\mathbf{x}\sim D}\left[f(\mathbf{x})\right]$ and $\hat{E}\left[f(\mathbf{x})\right]$ for $E_{\mathbf{x}\in S}\left[f(\mathbf{x})\right]$.

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

Proof. Simple Chernoff bounds arguments prove the inequalities. (It suffices to observe that $\frac{1}{m}\sum_{x\in\mathcal{S}}\theta(\mathbf{x}) = \hat{E}\left[\theta(\mathbf{x})\right]$ 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 $D \in \mathcal{U}^{\gamma,\eta,\theta}$. A hyperplane $\mathbf{w}^*$ is CN-consistent* if $Pr_{\mathbf{x}\sim D}\left[\mathbf{w}^*\cdot\mathbf{x} \leq \gamma|\theta(\mathbf{x})=0\right] = \eta$

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

[3]Slightly larger amount of noise and smaller error rate could be theoretically targeted. But the choices we have made suffice to our purpose.

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 $\mathbf{w}$ 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 $1 - \delta$, $\forall R \in [-1, 1]$:*

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

where $\mathbf{1}_{\leq R}(\alpha) = 1$ 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 $1 - \frac{3\delta}{4} - \frac{\delta}{4}$, $\forall R \in [-1, 1]$

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

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] - E[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x}) | \theta(\mathbf{x}) = 0] 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] - E[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x}) | \theta(\mathbf{x}) = 1] Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x}) = 1]\Big| \\
&= \Big|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x}) | \theta(\mathbf{x}) = 0] \left(Pr_{\mathbf{x}\in S}[\theta(\mathbf{x}) = 0] - Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x}) = 0]\right) \\
&\quad + \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) 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] \left(Pr_{\mathbf{x}\in S}[\theta(\mathbf{x}) = 1] - Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x}) = 1]\right) \\
&\quad + \left(\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]\right) Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x}) = 1]\Big| \\
&= \left|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x}) | \theta(\mathbf{x}) = 0]\right| \left|Pr_{\mathbf{x}\in S}[\theta(\mathbf{x}) = 0] - Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x}) = 0]\right| \\
&\qquad\qquad (\leq \tfrac{\varepsilon\gamma}{64} \text{ by lemma 1}) \\
&\quad + \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| Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x}) = 0] \\
&\qquad\qquad (\leq \tfrac{\varepsilon\gamma}{16} \text{ by equation 1}) \\
&\quad + \left|\hat{E}[(\mathbf{w}\cdot\mathbf{x})\mathbf{1}_{\leq R}(\mathbf{w}\cdot\mathbf{x}) | \theta(\mathbf{x}) = 1]\right| \left|Pr_{\mathbf{x}\in S}[\theta(\mathbf{x}) = 1] - Pr_{\mathbf{x}\sim D}[\theta(\mathbf{x}) = 1]\right| \\
&\qquad\qquad (\leq \tfrac{\varepsilon\gamma}{64} \text{ by lemma 1}) \\
&\quad + \left|\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]\right| 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 $E[\mathbf{w}^* \cdot \mathbf{x}] \geq (1 - 2\eta)(1 - \theta)\gamma - \theta > 0$.*

Proof.

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

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

We extend the 2 inequalities of Lemma 6 (cf. Appendix) to the case of a Mixed-Noise distribution.

Lemma 4. *Let $\gamma > 0, \eta \in [0, 0.5)$ and $\delta \in (0, 1]$. Let $\theta \in [0, \theta_{\max}(\gamma, \eta))$ such that $\varepsilon_{min}(\gamma, \eta, \theta) < \frac{4(1-2\eta)}{3}$, 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 inequalities hold:*

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

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

Proof. For the first inequality, we have:

$$\begin{aligned} &(1 - 2\eta)\, E\left[(\mathbf{w}^* \cdot \mathbf{x}) \mathbf{1}_{\leq 0}(\mathbf{w} \cdot \mathbf{x})\right] + \eta E\left[\mathbf{w}^* \cdot \mathbf{x}\right] \\ &\quad = (1 - 2\eta)\, E\left[(\mathbf{w}^* \cdot \mathbf{x}) \mathbf{1}_{\leq 0}(\mathbf{w} \cdot \mathbf{x}) | \theta(\mathbf{x}) = 1\right] Pr\left[\theta(\mathbf{x}) = 1\right] \\ &\qquad + \eta E\left[\mathbf{w}^* \cdot \mathbf{x} | \theta(\mathbf{x}) = 1\right] Pr\left[\theta(\mathbf{x}) = 1\right] \\ &\qquad + (1 - 2\eta)\, E\left[(\mathbf{w}^* \cdot \mathbf{x}) \mathbf{1}_{\leq 0}(\mathbf{w} \cdot \mathbf{x}) | \theta(\mathbf{x}) = 0\right] Pr\left[\theta(\mathbf{x}) = 0\right] \\ &\qquad + \eta E\left[\mathbf{w}^* \cdot \mathbf{x} | \theta(\mathbf{x}) = 0\right] Pr\left[\theta(\mathbf{x}) = 0\right] \\ &\quad \geq (1 - \theta)\, \frac{3}{4}\varepsilon\gamma \qquad \text{(by lemma 6 eq. 4)} \\ &\qquad + (1 - 2\eta)\, E\left[(\mathbf{w}^* \cdot \mathbf{x}) \mathbf{1}_{\leq 0}(\mathbf{w} \cdot \mathbf{x}) | \theta(\mathbf{x}) = 1\right] Pr\left[\theta(\mathbf{x}) = 1\right] \\ &\qquad + \eta E\left[\mathbf{w}^* \cdot \mathbf{x} |\, \theta(\mathbf{x}) = 1\right] Pr\left[\theta(\mathbf{x}) = 1\right] \\ &\quad \geq (1 - \theta)\, \frac{3}{4}\varepsilon\gamma - (1 - 2\eta)\, \theta - \eta\theta \\ &\quad \geq (1 - \theta)\, \frac{3}{4}\varepsilon\gamma - (1 - \eta)\, \theta \\ &\quad \geq \frac{5\varepsilon\gamma}{8} \qquad \text{(by definition of } \varepsilon) \end{aligned}$$

Now, for the second inequality, we have:

$$
\begin{aligned}
&(1-2\eta)\, E\left[(\mathbf{w}\cdot\mathbf{x})\mathbb{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})\right] + \eta E\left[\mathbf{w}\cdot\mathbf{x}\right]\\
&\quad= (1-2\eta)\, E\left[(\mathbf{w}\cdot\mathbf{x})\mathbb{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1\right] Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad + \eta E\left[\mathbf{w}\cdot\mathbf{x}|\theta(\mathbf{x})=1\right] Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad + (1-2\eta)\, E\left[(\mathbf{w}\cdot\mathbf{x})\mathbb{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=0\right] Pr\left[\theta(\mathbf{x})=0\right]\\
&\qquad + \eta E\left[\mathbf{w}\cdot\mathbf{x}|\theta(\mathbf{x})=0\right] Pr\left[\theta(\mathbf{x})=0\right]\\
&\quad\leq 0 && \text{(by lemma 6 eq.5)}\\
&\qquad + (1-2\eta)\, E\left[(\mathbf{w}\cdot\mathbf{x})\mathbb{1}_{\leq 0}(\mathbf{w}\cdot\mathbf{x})|\theta(\mathbf{x})=1\right] Pr\left[\theta(\mathbf{x})=1\right]\\
&\qquad + \eta E\left[\mathbf{w}\cdot\mathbf{x}|\, \theta(\mathbf{x})=1\right] Pr\left[\theta(\mathbf{x})=1\right]\\
&\quad\leq 0 + \eta\theta
\end{aligned}
$$

□

Now, we will show the core lemma. It states that Algorithm 2 outputs with high probability a vector that can be used as an update vector in the Perceptron algorithm (cf. Figure 1), that is a vector that is erroneously classified by the current classifier but that is 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 5. *Let $\gamma > 0, \eta \in [0, 0.5)$ and $\delta \in (0, 1)$. Let $\theta \in [0, \theta_{\max}(\gamma, \eta))$ such that $\varepsilon_{\min}(\gamma, \eta, \theta) < \frac{4}{3}(1-\eta)$. Let $D \in \mathcal{U}^{\gamma,\eta,\theta}$ and $\mathbf{w}^*$ 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 therefore 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, $E\left[\mathbf{w}^*\cdot\mathbf{x}\right] \geq (1-\theta)\frac{3\varepsilon}{4}\gamma-(2\gamma+1)\theta$. Because $\theta < \theta_{\max}(\gamma, \eta)$ and $\varepsilon > \varepsilon_{\min}(\gamma, \eta, \theta)$, we have $E\left[\mathbf{w}^*\cdot\mathbf{x}\right] \geq \frac{5\varepsilon\gamma}{8}$. 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

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

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 4 implies that $a' = \frac{\eta}{1-\eta}$ and $b' = \frac{1-2\eta}{1-\eta}$ results in $a'E\left[\left(\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}\right)\mathbf{1}_{\leq 0}(\mathbf{w} \cdot \mathbf{x})\right] + b'E[\frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}] \leq \frac{\eta\theta}{1-\eta}$ 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 4 also implies that choosing $a' = \frac{\eta}{1-\eta}$ and $b' = \frac{1-2\eta}{1-\eta}$ results in $a'E[(\mathbf{w}^* \cdot \mathbf{x})\mathbf{1}_{\leq 0}(\mathbf{w} \cdot \mathbf{x})] + b'E[\mathbf{w}^* \cdot \mathbf{x}] \geq \frac{5\varepsilon\gamma}{8}$

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 to the $\mathbf{w}$ hyperplane the $\mathbf{w}^* \cdot \mathbf{z} \geq \frac{\varepsilon\gamma}{4}$ 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 the CNoise update algorithm. This proves the Lemma.

□

We finally have the following theorem 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 $\mathbf{w}^*$ 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 5, 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 $\frac{16}{(\varepsilon\gamma)^2}$ (Block, 1962; Novikoff, 1962). 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 depends of course 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 of Theorem 2.

3.2 Random Projections and Separable Distributions

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

Theorem 5 (Balcan *et al.* (2004))**.** *Let $D \in \mathcal{D}^{\gamma,0}$. For a random sample $\mathcal{S} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ from D, let $\pi(\mathcal{S}) : \mathcal{X} \to span\langle\mathcal{S}\rangle$ the orthonormal 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 $\pi(D)$ is a $\gamma/2$-separable with error θ on $span\langle\mathcal{S}\rangle \subseteq \mathcal{X}$.*

This theorem says that a random projection can transform a linearly separable distribution in an almost linearly separable one defined on 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 will be 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 $1-\eta-\varepsilon$.*

Here, $K > 0$ 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 $\theta \leq \varepsilon_{\min}(\gamma,\eta,\theta)$, $\theta \leq \frac{\varepsilon}{2}$. (We are in agreement with the assumptions of Theorem 4.)

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 distribtion having a CN part of weight $1-\theta$ and 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, we know that 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 $1-(1-\theta)\frac{\varepsilon}{2} - \theta \geq 1 - \frac{\varepsilon}{2} - \frac{\delta}{2} = 1-\delta$. □

Remark 5. *Note that 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 $\theta_{\max} - \theta_{init}$.*

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 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 a NP-hard problem (see, e.g., Amaldi & Kann (1996)).

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 Bylander (1994, 1998), whose analysis we have just extended, the other notable contributions are those of (Blum *et al.*, 1996) and (Cohen, 1997). 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 (Zwald *et al.*, 2004) has to be mentioned. It is based on the use of Kernel PCA as a feature extraction step. The main points of this very interesting work are a proof on the regularizing properties of Kernel PCA and the fact that it gives a practical model selection procedure. However, the question of learning noisy distributions is not addressed.

Finally, the empirical study of (Fradkin & Madigan, 2003) provides some insights on how random projections might be useful for classification. No sample and run-

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

ning time complexity results are given and the question of learning with noise is not addressed.

5 Numerical Simulations

5.1 UCI Datasets

We have 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, Diabetis, German, Heart), we have 100 training samples and 100 test samples. All these probems only contain a few hundreds training examples, which is far frow what the theoretical bounds showed above would require.

We have tested three projection procedures: random, Kernel PCA (KPCA), Kernel Gram-Schmidt (KGS) (Shawe-Taylor & Cristianini, 2004). 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 algorihtm anymore).

Note that, to perform random projections, we chose randomly our projection vectors *among* the learning set. Since the learning set is drawn from the distribution, selecting examples among it returns to same than drawing directly projection vectors from distribution. Thus, our process meets the process described by (Balcan *et al.*, 2004).

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.

In order to determine the right projection size, we resort to a cross-validation procedure which works as follows. Considering only the first five training (noisy) samples of each problem, we learn on one of the samples and measure the accuracy on the other four, and we try subspace sizes of 2, 5, 10,. . . , 100, 125, 150, 200. The subspace dimension giving the smallest error is the one that is picked for the estimation of the generalization accuracy. For the KPCA method, we have chosen the last dimension for which the reconstruction error is rather widely larger compared to the test with higher dimension.

The results obtained are summarized on Figure 2 and on Tables 1 and 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 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

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

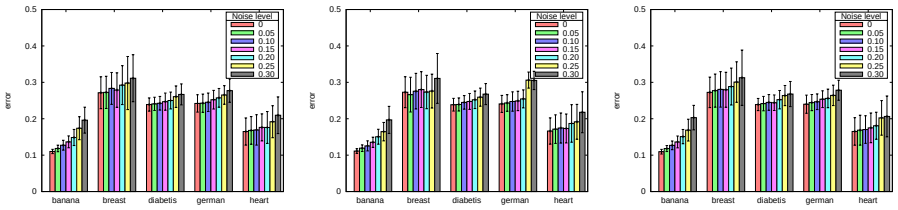

Figure 2: Error rates on UCI datasets with random projections, KPCA and KGS projection with different amount of classification noise; 1-standard deviation error bars are shown.

theoretical requirements. We can also note that when the actual values of the accuracies are compared, KGS and KPCA roughly achieve the same accuracies than random projection. This supports and, because KGS is the faster projection to compute, our objective to study its properties more thoroughly in a near future.

The main point of the set of numerical simulations conducted here is that RP-classifier has a very satisfactory behavior on real data, with really convicing classification noise tolerance.

Another parameter (Table 1) that we point out is the selected projection dimension for each projection process. KPCA (almost) always requires a smaller dimension of projection than KGS and random projection. That is not really surprising, due to the totally deterministic aspect of this process, and so to the fact that it is optimal, from the point of view of the reconstruction error. The behaviours of random and KGS projection dimension selections seem to be harder to analyze. The two processes seem to be extremely unstable from the point of view of selected dimension (sometimes near from KPCA dimension, sometimes 10 times larger), probably because of their *random aspects* (selection of first vector for KGS, totally random for random projections). However, they are a lot faster than KPCA, and the accuracy results are comparable, so this instability does not constitute a real drawback.

5.2 Toy Problems

We have carried out additional simulations on five toy 2-dimensional toy problems (cf. Figure 3). Here, we have used the KGS projection since due to the uniform distribution of points on the rectangle $[-10; 10] \times [-10; 10]$, 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 (0.0, 0.10, 0.20, 0.30, 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 and 0.30.

Additional results concerning the accuracy of the produced classifiers, the dimension and kernel width selection are provided in Tables 3 and 4.

These experiments confirm the conclusions made on UCI datasets, about accuracy results and dimension selection. Note that the results remain good, even if the number of examples is a lot less than theoritically needed, even if the classification noise (0.30 or 0.40) is important and even if no margin has been defined .

Last remark, note that, not surprisingly, the selected gaussian kernel width seems to be not affected by the increase of noise level.

The essential point showed by these simulations is that, again, RP-classifier is very effective in learning from noisy nonlinear distributions. In the numerical results, we have observed that our algorithm can tolerate noise levels as high as 0.3 and still provide reasonnable error rates (typically around 10%).

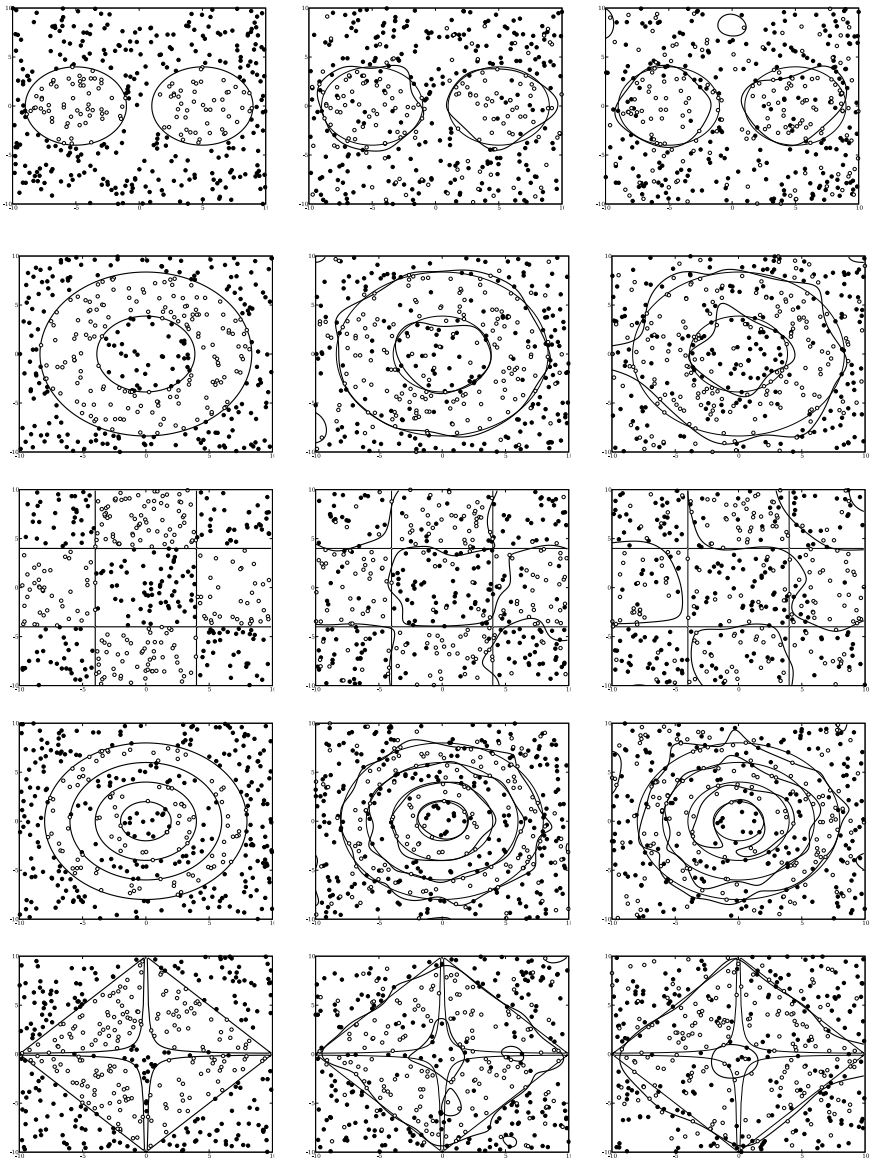

Figure 3: Toy problems: first column show the clean concepts with black disks being of class +1 and white ones of class -1. Second and third columns show the concepts learned by RP-classifier with KGS projection and respectively a uniform classificaton noise rate of 0.20 and of 0.30.

Noise	Projection	Banana	Breast Cancer	Diabetis	German	Heart
0.00	KPCA	11.13 ± 0.65	27.29 ± 4.25	23.86 ± 1.74	24.08 ± 2.34	16.63 ± 3.62
	KGS	10.95 ± 0.64	27.25 ± 4.19	23.9 ± 1.68	24.0 ± 2.49	16.51 ± 3.78
	Random	11.01 ± 0.59	27.14 ± 4.39	23.9 ± 1.83	24.21 ± 2.38	16.49 ± 3.73
0.05	KPCA	11.92 ± 0.92	26.62 ± 4.77	23.92 ± 1.81	24.33 ± 2.29	17.15 ± 3.92
	KGS	11.81 ± 0.82	27.73 ± 4.56	24.16 ± 1.98	24.44 ± 2.22	16.86 ± 4.12
	Random	11.84 ± 0.91	27.25 ± 4.42	24.09 ± 1.82	24.27 ± 2.51	16.79 ± 3.8
0.10	KPCA	12.55 ± 1.39	27.57 ± 4.9	24.49 ± 1.87	24.73 ± 2.66	17.45 ± 4.15
	KGS	12.69 ± 1.2	28.06 ± 4.87	24.48 ± 2.17	24.67 ± 2.16	17.03 ± 3.78
	Random	12.73 ± 1.36	28.34 ± 4.38	24.25 ± 1.89	24.53 ± 2.54	16.93 ± 4.17
0.15	KPCA	13.54 ± 1.41	28.01 ± 4.92	24.75 ± 2.19	24.89 ± 2.67	17.33 ± 3.98
	KGS	13.63 ± 1.63	27.96 ± 4.81	24.4 ± 2.14	25.37 ± 2.3	17.5 ± 4.09
	Random	13.65 ± 1.63	27.88 ± 4.74	24.7 ± 2.36	25.23 ± 2.56	17.62 ± 3.73
0.20	KPCA	15.06 ± 2.05	27.34 ± 4.59	25.22 ± 2.42	25.46 ± 2.47	18.71 ± 5.16
	KGS	15.09 ± 1.97	28.84 ± 5.07	25.23 ± 2.53	25.59 ± 2.56	18.08 ± 3.71
	Random Projection	14.85 ± 2.22	29.23 ± 5.35	25.01 ± 2.32	25.74 ± 2.58	17.6 ± 4.37
0.25	KPCA	16.45 ± 2.53	27.6 ± 4.67	25.93 ± 2.49	30.62 ± 2.21	19.16 ± 4.79
	KGS	16.87 ± 3.0	30.08 ± 5.56	26.36 ± 2.77	26.42 ± 2.81	20.24 ± 4.75
	Random	17.4 ± 3.16	29.81 ± 7.3	26.07 ± 2.95	26.53 ± 2.56	19.18 ± 4.4
0.30	KPCA	19.69 ± 3.72	31.08 ± 6.85	26.77 ± 2.89	30.53 ± 2.52	21.78 ± 5.62
	KGS	20.31 ± 3.37	31.27 ± 7.61	26.78 ± 3.46	27.84 ± 2.75	20.64 ± 5.62
	Random	19.61 ± 3.57	31.13 ± 6.46	26.69 ± 2.88	27.73 ± 3.21	20.97 ± 5.04

Table 1: Mean and standard deviation for each UCI problem, each classification noise rate and each projection strategy

Noise	Projection	Banana	Breast Cancer	Diabetis	German	Heart
0.00	KPCA	20	10	15	20	15
	KGS	125	45	15	125	50
	Random	30	15	125	20	40
0.05	KPCA	20	2	10	20	15
	KGS	150	150	50	15	45
	Random	40	50	125	100	40
0.10	KPCA	15	10	10	15	10
	KGS	15	50	125	100	50
	Random	30	40	30	125	75
0.15	KPCA	15	10	15	20	15
	KGS	25	35	15	40	100
	Random	75	30	25	20	50
0.20	KPCA	25	2	20	20	10
	KGS	75	45	35	75	150
	Random	150	125	45	100	50
0.25	KPCA	20	2	10	2	10
	KGS	30	5	100	75	45
	Random	125	10	20	30	15
0.30	KPCA	15	5	10	2	2
	KGS	75	50	15	25	30
	Random	20	150	15	25	125

Table 2: Projection dimension chosen for each UCI problem, each noise rate and each projection strategy

Noise	Double Ellipse	Ring	Chess Board	Dart Board	Hyper
0.00	0.49 ± 0.16	0.59 ± 0.2	0.74 ± 0.21	1.74 ± 0.42	2.99 ± 0.44
0.10	2.1 ± 0.51	2.85 ± 0.65	3.61 ± 0.71	4.72 ± 0.88	6.26 ± 0.78
0.20	3.38 ± 0.9	4.67 ± 0.87	5.88 ± 1.52	7.88 ± 1.12	8.12 ± 1.16
0.30	6.3 ± 2.03	7.75 ± 1.74	10.87 ± 2.6	13.42 ± 2.24	11.77 ± 1.72
0.40	13.51 ± 4.86	18.48 ± 4.42	16.19 ± 4.54	28.47 ± 4.85	18.14 ± 5.39

Table 3: Mean and standard deviation for each toy problem and each classification noise rate

Noise	Parameter	Double Ellipse	Ring	Chess Board	Dart Board	Hyper
0.00	Projection Dimension	50	30	50	200	200
	Kernel Width	3.0	4.0	2.5	2.5	1.5
0.10	Projection Dimension	100	150	150	200	200
	Kernel Width	1.5	4.0	2.0	1.5	1.5
0.20	Projection Dimension	75	50	100	150	200
	Kernel Width	2.0	2.5	1.5	1.5	2.5
0.30	Projection Dimension	75	40	75	150	25
	Kernel Width	3.0	3.0	2.0	2.0	2.5
0.40	Projection Dimension	25	15	5	100	5
	Kernel Width	2.5	4.0	4.0	1.5	4.0

Table 4: Projection dimension (with KGS strategy) and gaussian kernel width chosen for each toy problem and each noise rate

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 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 algorihtm. This algorithm is shown to be robust to some additional 'projection noise' provided the sample complexity are adjusted in a suitable way. Several simulation results support the soundness of our approach. Note that it exists another projection, based on the Jonsson-Lindenstrauss lemma and described in (Balcan *et al.*, 2004), that allows us to reduce the time and the sample complexity of the learning step.

Several questions are raised by the present work. Among them, there is the question about the generalization properties of the Kernel Gram-Schmidt projector. We think that tight generalization bounds can be exhibited rather easily in the framework of PAC Bayesian bound, by exploiting, in particular, the sparseness of this projector. 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 wonder whether there is a way to learn optimal separating hyperplane from noisy distributions.

Appendix

Lemma 6 (Bylander (1998))**.** *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 the following inequalities hold:*

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

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

References

AMALDI E. & KANN V. (1996). On the approximability of some NP-hard minimization problems for linear systems. *Electronic Colloquium on Computational Complexity (ECCC)*, **3**(015).

ANGLUIN D. & LAIRD P. (1988). Learning from Noisy Examples. *Machine Learning*, **2**.

BALCAN M.-F., BLUM A. & VEMPALA S. (2004). Kernels as Features: on Kernels, Margins, and Low-dimensional Mappings. In *Proc. of the 15th Conf. on Algorithmic Learning Theory*.

BLOCK H. D. (1962). The perceptron: A model for brain functioning. *Reviews of Modern Physics*, **34**, 123–135.

BLUM A., FRIEZE A. M., KANNAN R. & VEMPALA S. (1996). A Polynomial-Time Algorithm for Learning Noisy Linear Threshold Functions. In *Proc. of 37th IEEE Symposium on Foundations of Computer Science*, p. 330–338.

BYLANDER T. (1994). Learning Linear Threshold Functions in the Presence of Classification Noise. In *Proc. of 7th Annual Workshop on Computational Learning Theory*, p. 340–347: ACM Press, New York, NY, 1994.

BYLANDER T. (1998). Learning Noisy Linear Threshold Functions. Submitted for journal publication.

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

CRISTIANINI N. & SHAWE-TAYLOR J. (2000). *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods.* Cambridge University Press.

FRADKIN D. & MADIGAN D. (2003). Experiments with random projections for machine learning. In *Proc. of the 9th ACM SIGKDD int. conf. on Knowledge discovery and data mining*.

GRAEPEL T., HERBRICH R. & WILLIAMSON R. C. (2001). From Margin to Sparsity. In *Adv. in Neural Information Processing Systems*, volume 13, p. 210–216.

KEARNS M. & LI M. (1993). Learning in the presence of malicious errors. *SIAM Journal on Computing*, **22**(4), 807–837.

NOVIKOFF A. B. J. (1962). On convergence proofs on perceptrons. In *Proc. of the Symp. on the Mathematical Theory of Automata*, p. 615–622.

ROSENBLATT F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, **65**, 386–407.

SCHÖLKOPF B. & SMOLA A. J. (2002). *Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond*. MIT University Press.

SHAWE-TAYLOR J. & CRISTIANINI N. (2004). *Kernel Methods for Pattern Analysis.* Cambridge University Press.

VAPNIK V. (1995). *The nature of statistical learning theory*. Springer, New York.

VAPNIK V. (1998). *Statistical Learning Theory*. John Wiley and Sons, inc.

ZWALD L., VERT R., BLANCHARD G. & MASSART P. (2004). Kernel projection machine: a new tool for pattern recognition. In *Adv. in Neural Information Processing Systems*, volume 17.