

---

# Margins, Kernels and Non-linear Smoothed Perceptrons

---

**Aaditya Ramdas**

Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

ARAMDAS@CS.CMU.EDU

**Javier Peña**

Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

JFP@ANDREW.CMU.EDU

## Abstract

We focus on the problem of finding a non-linear classification function that lies in a Reproducing Kernel Hilbert Space (RKHS) both from the primal point of view (finding a perfect separator when one exists) and the dual point of view (giving a certificate of non-existence), with special focus on generalizations of two classical schemes - the Perceptron (primal) and Von-Neumann (dual) algorithms.

We cast our problem as one of maximizing the regularized normalized hard-margin ( $\rho$ ) in an RKHS and rephrase it in terms of a Mahalanobis dot-product/semi-norm associated with the kernel's (normalized and signed) Gram matrix. We derive an accelerated smoothed algorithm with a convergence rate of  $\frac{\sqrt{\log n}}{\rho}$  given  $n$  separable points, which is strikingly similar to the classical kernelized Perceptron algorithm whose rate is  $\frac{1}{\rho^2}$ . When no such classifier exists, we prove a version of Gordan's separation theorem for RKHSs, and give a reinterpretation of negative margins. This allows us to give guarantees for a primal-dual algorithm that halts in  $\min\{\frac{\sqrt{n}}{|\rho|}, \frac{\sqrt{n}}{\epsilon}\}$  iterations with a perfect separator in the RKHS if the primal is feasible or a dual  $\epsilon$ -certificate of near-infeasibility.

## 1. Introduction

We are interested in the problem of finding a non-linear separator for a given set of  $n$  points  $x_1, \dots, x_n \in \mathbb{R}^d$  with labels  $y_1, \dots, y_n \in \{\pm 1\}$ . Finding a linear separator can be stated as the problem of finding a unit vector  $w \in \mathbb{R}^d$  (if

one exists) such that for all  $i$

$$y_i(w^\top x_i) \geq 0 \quad \text{i.e.} \quad \text{sign}(w^\top x_i) = y_i. \quad (1)$$

This is called the primal problem. In the more interesting non-linear setting, we will be searching for functions  $f$  in a Reproducing Kernel Hilbert Space (RKHS)  $\mathcal{F}_K$  associated with kernel  $K$  (to be defined later) such that for all  $i$

$$y_i f(x_i) \geq 0. \quad (2)$$

We say that problems (1), (2) have an unnormalized margin  $\rho > 0$ , if there exists a unit vector  $w$ , such that for all  $i$ ,

$$y_i(w^\top x_i) \geq \rho \quad \text{or} \quad y_i f(x_i) \geq \rho.$$

True to the paper's title, margins of non-linear separators in an RKHS will be a central concept, and we will derive interesting *smoothed accelerated* variants of the Perceptron algorithm that have convergence rates (for the aforementioned primal and a dual problem introduced later) that are inversely proportional to the RKHS-margin as opposed to inverse squared margin for the Perceptron.

The linear setting is well known by the name of linear feasibility problems - we are asking if there exists any vector  $w$  which makes an acute angle with all the vectors  $y_i x_i$ , i.e.

$$(XY)^\top w > \mathbf{0}_n, \quad (3)$$

where  $Y := \text{diag}(y)$ ,  $X := [x_1, \dots, x_n]$ . This can be seen as finding a vector  $w$  inside the dual cone of  $\text{cone}\{y_i x_i\}$ .

When normalized, as we will see in the next section, the margin is a well-studied notion of conditioning for these problems. It can be thought of as the width of the feasibility cone as in (Freund & Vera, 1999), a radius of well-posedness as in (Cheung & Cucker, 2001), and its inverse can be seen as a special case of a condition number defined by (Renegar, 1995) for these systems.

### 1.1. Related Work

In this paper we focus on the famous Perceptron algorithm (Rosenblatt, 1958) and the less-famous Von-Neumann algorithm (Dantzig, 1992) that we introduce in later sections.

Our work builds on (Soheili & Peña, 2012; 2013a) from the field of optimization - we generalize the setting to learning functions in RKHSs, extend the algorithms, simplify proofs, and simultaneously bring new perspectives to it. There is extensive literature around the Perceptron algorithm in the learning community; we restrict ourselves to discussing only a few directly related papers, in order to point out the several differences from existing work.

We provide a general unified proof in the Appendix which borrows ideas from accelerated smoothing methods developed by Nesterov (Nesterov, 2005) - while this algorithm and others by (Nemirovski, 2004), (Saha et al., 2011) can achieve similar rates for the same problem, those algorithms do not possess the simplicity of the Perceptron or Von-Neumann algorithms and our variants, and also don't look at the infeasible setting or primal-dual algorithms.

Accelerated smoothing techniques have also been seen in the learning literature like in (Tseng, 2008) and many others. However, most of these deal with convex-concave problems where both sets involved are the probability simplex (as in game theory, boosting, etc), while we deal with hard margins where one of the sets is a unit  $\ell_2$  ball. Hence, their algorithms/results are not extendable to ours trivially. This work is also connected to the idea of  $\epsilon$ -coresets (Clarkson, 2010), though we will not explore that angle.

A related algorithm is called the Winnow (Littlestone, 1991) - this works on the  $\ell_1$  margin and is a saddle point problem over two simplices. One can ask whether such accelerated smoothed versions exist for the Winnow. The answer is in the affirmative - however such algorithms look completely different from the Winnow, while in our setting the new algorithms retain the simplicity of the Perceptron.

## 1.2. Paper Outline

Sec.2 will introduce the Perceptron and Normalized Perceptron algorithm and their convergence guarantees for linear separability, with specific emphasis on the unnormalized and normalized margins. Sec.3 will then introduce RKHSs and the Normalized Kernel Perceptron algorithm, which we interpret as a subgradient algorithm for a regularized normalized hard-margin loss function.

Sec.4 describes the Smoothed Normalized Kernel Perceptron algorithm that works with a smooth approximation to the original loss function, and outlines the argument for its faster convergence rate. Sec.5 discusses the non-separable case and the Von-Neumann algorithm, and we prove a version of Gordan's theorem in RKHSs.

We finally give an algorithm in Sec.6 which terminates with a separator if one exists, and with a dual certificate of near-infeasibility otherwise, in time inversely proportional to the margin. Sec.7 has a discussion and some open problems.

## 2. Linear Feasibility Problems

### 2.1. Perceptron

The classical perceptron algorithm can be stated in many ways, one is in the following form

---

#### Algorithm 1 Perceptron

---

```

Initialize  $w_0 = 0$ 
for  $k = 0, 1, 2, 3, \dots$  do
    if  $\text{sign}(w_k^\top x_i) \neq y_i$  for some  $i$  then
         $w_{k+1} := w_k + y_i x_i$ 
    else
        Halt: Return  $w_k$  as solution
    end if
end for
    
```

---

It comes with the following classic guarantee as proved by (Block, 1962) and (Novikoff, 1962): *If there exists a unit vector  $u \in \mathbb{R}^d$  such that  $YX^\top u \geq \rho > 0$ , then a perfect separator will be found in  $\frac{\max_i \|x_i\|_2^2}{\rho^2}$  iterations/mistakes.*

The algorithm works when updated with any arbitrary point  $(x_i, y_i)$  that is misclassified; it has the same guarantees when  $w$  is updated with the point that is misclassified by the largest amount,  $\arg \min_i y_i w^\top x_i$ . Alternately, one can define the probability distribution over examples

$$p(w) = \arg \min_{p \in \Delta_n} \langle YX^\top w, p \rangle, \quad (4)$$

where  $\Delta_n$  is the  $n$ -dimensional probability simplex.

Intuitively,  $p$  picks the examples that have the lowest margin when classified by  $w$ . One can also normalize the updates so that we can maintain a probability distribution over examples used for updates from the start, as seen below:

---

#### Algorithm 2 Normalized Perceptron

---

```

Initialize  $w_0 = 0, p_0 = 0$ 
for  $k = 0, 1, 2, 3, \dots$  do
    if  $YX^\top w_k > 0$  then
        Exit, with  $w_k$  as solution
    else
         $\theta_k := \frac{1}{k+1}$ 
         $w_{k+1} := (1 - \theta_k)w_k + \theta_k XY p(w_k)$ 
    end if
end for
    
```

---

**Remark.** Normalized Perceptron has the same guarantees as perceptron - the Perceptron can perform its update *online* on any misclassified point, while the Normalized Perceptron performs updates on the *most* misclassified point(s), and yet there does not seem to be any change in performance. However, we will soon see that the ability to see all the examples at once gives us much more power.

## 2.2. Normalized Margins

If we normalize the data points by the  $\ell_2$  norm, the resulting mistake bound of the perceptron algorithm is slightly different. Let  $X_2$  represent the matrix with columns  $x_i/\|x_i\|_2$ . Define the unnormalized and normalized margins as

$$\begin{aligned}\rho &:= \sup_{\|w\|_2=1} \inf_{p \in \Delta_n} \langle YX^\top w, p \rangle, \\ \rho_2 &:= \sup_{\|w\|_2=1} \inf_{p \in \Delta_n} \langle YX_2^\top w, p \rangle.\end{aligned}$$

**Remark.** Note that we have  $\sup_{\|w\|_2=1}$  in the definition, this is equivalent to  $\sup_{\|w\|_2 \leq 1}$  iff  $\rho_2 > 0$ .

Normalized Perceptron has the following guarantee on  $X_2$ : *If  $\rho_2 > 0$ , then it finds a perfect separator in  $\frac{1}{\rho_2^2}$  iterations.*

**Remark.** Consider the max-margin separator  $u^*$  for  $X$  (which is also a valid perfect separator for  $X_2$ ). Then

$$\begin{aligned}\frac{\rho}{\max_i \|x_i\|_2} &= \min_i \left( \frac{y_i x_i^\top u^*}{\max_i \|x_i\|_2} \right) \leq \min_i \left( \frac{y_i x_i^\top u^*}{\|x_i\|_2} \right) \\ &\leq \sup_{\|u\|_2=1} \min_i \left( \frac{y_i x_i^\top u}{\|x_i\|_2} \right) = \rho_2.\end{aligned}$$

Hence, it is always better to normalize the data as pointed out in (Graepel et al., 2001). This idea extends to RKHSs, motivating the normalized Gram matrix considered later.

**Example** Consider a simple example in  $\mathbb{R}_+^2$ . Assume that  $+$  points are located along the line  $6x_2 = 8x_1$ , and the  $-$  points along  $8x_2 = 6x_1$ , for  $1/r \leq \|x\|_2 \leq r$ , where  $r > 1$ . The max-margin linear separator will be  $x_1 = x_2$ . If all the data were normalized to have unit Euclidean norm, then all the  $+$  points would all be at  $(0.6, 0.8)$  and all the  $-$  points at  $(0.8, 0.6)$ , giving us a normalized margin of  $\rho_2 \approx 0.14$ . Unnormalized, the margin is  $\rho \approx 0.14/r$  and  $\max_i \|x_i\|_2 = r$ . Hence, in terms of bounds, we get a discrepancy of  $r^4$ , which can be arbitrarily large.

**Winnov** The question arises as to which norm we should normalize by. There is a now classic algorithm in machine learning, called Winnov (Littlestone, 1991) or Multiply Weights. It works on a slight transformation of the problem where we only need to search for  $u \in \mathbb{R}_+^d$ . It comes with some very well-known guarantees - *If there exists a  $u \in \mathbb{R}_+^d$  such that  $YX^\top u \geq \rho > 0$ , then feasibility is guaranteed in  $\|u\|_1^2 \max_i \|a_i\|_\infty^2 \log n / \rho^2$  iterations.* The appropriate notion of normalized margin here is

$$\rho_1 := \max_{w \in \Delta_d} \min_{p \in \Delta_n} \langle YX_\infty^\top w, p \rangle,$$

where  $X_\infty$  is a matrix with columns  $x_i/\|x_i\|_\infty$ . Then, the appropriate iteration bound is  $\log n / \rho_1^2$ . We will return to this  $\ell_1$ -margin in the discussion section. In the next section, we will normalize by using the kernel appropriately.

## 3. Kernels and RKHSs

The theory of Reproducing Kernel Hilbert Spaces (RKHSs) has a rich history, and for a detailed introduction, refer to (Schölkopf & Smola, 2002). Let  $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  be a symmetric positive definite kernel, giving rise to a Reproducing Kernel Hilbert Space  $\mathcal{F}_K$  with an associated feature mapping at each point  $x \in \mathbb{R}^d$  called  $\phi_x : \mathbb{R}^d \rightarrow \mathcal{F}_K$  where  $\phi_x(\cdot) = K(x, \cdot)$  i.e.  $\phi_x(y) = K(x, y)$ .  $\mathcal{F}_K$  has an associated inner product  $\langle \phi_u, \phi_v \rangle_K = K(u, v)$ . For any  $f \in \mathcal{F}_K$ , we have  $f(x) = \langle f, \phi_x \rangle_K$ .

Define the normalized feature map

$$\tilde{\phi}_x = \frac{\phi_x}{\sqrt{K(x, x)}} \in \mathcal{F}_K \quad \text{and} \quad \tilde{\phi}_X := [\tilde{\phi}_{x_i}]_1^n.$$

For any function  $f \in \mathcal{F}_K$ , we use the following notation

$$Y\tilde{f}(X) := \langle f, Y\tilde{\phi}_X \rangle_K = [y_i \langle f, \tilde{\phi}_{x_i} \rangle_K]_1^n = \left[ \frac{y_i f(x_i)}{\sqrt{K(x_i, x_i)}} \right]_1^n.$$

We analogously define the normalized margin here to be

$$\rho_K := \sup_{\|f\|_K=1} \inf_{p \in \Delta_n} \langle Y\tilde{f}(X), p \rangle. \quad (5)$$

Consider the following regularized empirical loss function

$$L(f) = \left\{ \sup_{p \in \Delta_n} \langle -Y\tilde{f}(X), p \rangle \right\} + \frac{1}{2} \|f\|_K^2. \quad (6)$$

Denoting  $t := \|f\|_K > 0$  and writing  $f = t \left( \frac{f}{\|f\|_K} \right) = t\bar{f}$ , let us calculate the minimum value of this function

$$\begin{aligned}\inf_{f \in \mathcal{F}_K} L(f) &= \inf_{t>0} \inf_{\|f\|_K=1} \sup_{p \in \Delta_n} \langle -\langle t\bar{f}, Y\tilde{\phi}_X \rangle_K, p \rangle + \frac{t^2}{2} \\ &= \inf_{t>0} \left\{ -t\rho_K + \frac{1}{2}t^2 \right\} \\ &= -\frac{1}{2}\rho_K^2 \quad \text{when } t = \rho_K > 0.\end{aligned} \quad (7)$$

Since  $\max_{p \in \Delta_n} \langle -Y\tilde{f}(X), p \rangle$  is some empirical loss function on the data and  $\frac{1}{2}\|f\|_K^2$  is an increasing function of  $\|f\|_K$ , the Representer Theorem (Schölkopf et al., 2001) implies that the minimizer of the above function lies in the span of  $\phi_{x_i}$ s (also the span of the  $y_i \tilde{\phi}_{x_i}$ s). Explicitly,

$$\arg \min_{f \in \mathcal{F}_K} L(f) = \sum_{i=1}^n \alpha_i y_i \tilde{\phi}_{x_i} = \langle Y\tilde{\phi}_X, \alpha \rangle. \quad (8)$$

Substituting this back into Eq.(6), we can define

$$L(\alpha) := \left\{ \sup_{p \in \Delta_n} \langle -\alpha, p \rangle \right\} + \frac{1}{2} \|\alpha\|_G^2, \quad (9)$$

where  $G$  is a normalized signed Gram matrix with  $G_{ii} = 1$ ,

$$G_{ji} = G_{ij} := \frac{y_i y_j K(x_i, x_j)}{\sqrt{K(x_i, x_i) K(x_j, x_j)}} = \langle y_i \tilde{\phi}_{x_i}, y_j \tilde{\phi}_{x_j} \rangle_K,$$

and  $\langle p, \alpha \rangle_G := p^\top G \alpha$ ,  $\|\alpha\|_G := \sqrt{\alpha^\top G \alpha}$ . One can verify that  $G$  is a PSD matrix and the G-norm  $\|\cdot\|_G$  is a semi-norm, whose properties are of great importance to us.

### 3.1. Some Interesting and Useful Lemmas

The first lemma justifies our algorithms' exit condition.

**Lemma 1.**  $L(\alpha) < 0$  implies  $G\alpha > 0$  and there exists a perfect classifier iff  $G\alpha > 0$ .

*Proof.*  $L(\alpha) < 0 \Rightarrow \sup_{p \in \Delta_n} \langle -G\alpha, p \rangle < 0 \Leftrightarrow G\alpha > 0$ .  
 $G\alpha > 0 \Rightarrow f_\alpha := \langle \alpha, Y\tilde{\phi}_X \rangle$  is perfect since

$$\begin{aligned} \frac{y_j f_\alpha(x_j)}{\sqrt{K(x_j, x_j)}} &= \sum_{i=1}^n \alpha_i \frac{y_i y_j K(x_i, x_j)}{\sqrt{K(x_i, x_i) K(x_j, x_j)}} \\ &= G_j \alpha > 0. \end{aligned}$$

If a perfect classifier exists, then  $\rho_K > 0$  by definition and

$$L(f^*) = L(\alpha^*) = -\frac{1}{2} \rho_K^2 < 0 \Rightarrow G\alpha > 0,$$

where  $f^*, \alpha^*$  are the optimizers of  $L(f), L(\alpha)$ .  $\square$

The second lemma bounds the G-norm of vectors.

**Lemma 2.** For any  $\alpha \in \mathbb{R}^n$ ,  $\|\alpha\|_G \leq \|\alpha\|_1 \leq \sqrt{n} \|\alpha\|_2$ .

*Proof.* Using the triangle inequality of norms, we get

$$\begin{aligned} \sqrt{\alpha^\top G \alpha} &= \sqrt{\langle \langle \alpha, Y\tilde{\phi}_X \rangle, \langle \alpha, Y\tilde{\phi}_X \rangle \rangle_K} \\ &= \left\| \sum_i \alpha_i y_i \tilde{\phi}_{x_i} \right\|_K \leq \sum_i \|\alpha_i y_i \tilde{\phi}_{x_i}\|_K \\ &\leq \sum_i |\alpha_i| \left\| y_i \frac{\phi_{x_i}}{\sqrt{K(x_i, x_i)}} \right\|_K = \sum_i |\alpha_i|, \end{aligned}$$

where we used  $\langle \phi_{x_i}, \phi_{x_i} \rangle_K = K(x_i, x_i)$ .  $\square$

The third lemma gives a new perspective on the margin.

**Lemma 3.** When  $\rho_K > 0$ ,  $f$  maximizes the margin iff  $\rho_K f$  optimizes  $L(f)$ . Hence, the margin is equivalently

$$\rho_K = \sup_{\|\alpha\|_G=1} \inf_{p \in \Delta_n} \langle \alpha, p \rangle_G \leq \|p\|_G \text{ for all } p \in \Delta_n.$$

*Proof.* Let  $f_\rho$  be any function with  $\|f_\rho\|_K = 1$  that achieves the max-margin  $\rho_K > 0$ . Then, it is easy to plug  $\rho_K f_\rho$  into Eq. (6) and verify that  $L(\rho_K f_\rho) = -\frac{1}{2} \rho_K^2$  and hence  $\rho_K f_\rho$  minimizes  $L(f)$ .

Similarly, let  $f_L$  be any function that minimizes  $L(f)$ , i.e. achieves the value  $L(f_L) = -\frac{1}{2} \rho_K^2$ . Defining  $t := \|f_L\|_K$ , and examining Eq. (7), we see that  $L(f_L)$  cannot achieve the value  $-\frac{1}{2} \rho_K^2$  unless  $t = \rho_K$  and  $\sup_{p \in \Delta_n} \langle -Y\tilde{f}_L(X), p \rangle = -\rho_K^2$  which means that  $f_L/\rho_K$  must achieve the max-margin.

Hence considering only  $f = \sum_i \alpha_i y_i \tilde{\phi}_{x_i}$  is acceptable for both. Plugging this into Eq. (5) gives the equality and

$$\begin{aligned} \rho_K &= \inf_{p \in \Delta_n} \sup_{\|\alpha\|_G=1} \langle \alpha, p \rangle_G \leq \sup_{\|\alpha\|_G=1} \langle \alpha, p \rangle_G \\ &\leq \|p\|_G \text{ by applying Cauchy-Schwartz} \end{aligned}$$

(can also be seen by going back to function space).  $\square$

## 4. Smoothed Normalized Kernel Perceptron

Define the distribution over the worst-classified points

$$\begin{aligned} p(f) &:= \arg \min_{p \in \Delta_n} \langle Y\tilde{f}(X), p \rangle \\ \text{or } p(\alpha) &:= \arg \min_{p \in \Delta_n} \langle \alpha, p \rangle_G. \end{aligned} \quad (10)$$

---

### Algorithm 3 Normalized Kernel Perceptron (NKP)

---

```

Set  $\alpha_0 := 0$ 
for  $k = 0, 1, 2, 3, \dots$  do
  if  $G\alpha_k > 0_n$  then
    Exit, with  $\alpha_k$  as solution
  else
     $\theta_k := \frac{1}{k+1}$ 
     $\alpha_{k+1} := (1 - \theta_k)\alpha_k + \theta_k p(\alpha_k)$ 
  end if
end for

```

---

$$\begin{aligned} \text{Implicitly } f_{k+1} &= (1 - \theta_k) f_k + \theta_k \langle Y\tilde{\phi}_X, p(f_k) \rangle \\ &= f_k - \theta_k \left( f_k - \langle Y\tilde{\phi}_X, p(f_k) \rangle \right) \\ &= f_k - \theta_k \partial L(f_k) \end{aligned}$$

and hence the Normalized Kernel Perceptron (NKP) is a *subgradient algorithm* to minimize  $L(f)$  from Eq. (6).

**Remark.** Lemma 3 yields deep insights. Since NKP can get arbitrarily close to the minimizer of strongly convex  $L(f)$ , it also gets arbitrarily close to a margin maximizer. It is known that it finds a perfect classifier in  $1/\rho_K^2$  iterations - we now additionally infer that it will continue to improve to find an approximate max-margin classifier. While both classical and normalized Perceptrons find perfect classifiers in the same time, the latter is guaranteed to improve.

**Remark.**  $\alpha_{k+1}$  is always a probability distribution. Curiously, a guarantee that the solution will lie in  $\Delta_n$  is *not* made by the Representer Theorem in Eq. (8) - any  $\alpha \in \mathbb{R}^n$  could satisfy Lemma 1. However, since NKP is a subgradient method for minimizing Eq. (6), we know that we will approach the optimum while only choosing  $\alpha \in \Delta_n$ .

Define the smooth minimizer analogous to Eq. (10) as

$$\begin{aligned} p_\mu(\alpha) &:= \arg \min_{p \in \Delta_n} \left\{ \langle \alpha, p \rangle_G + \mu d(p) \right\} \quad (11) \\ &= \frac{e^{-G\alpha/\mu}}{\|e^{-G\alpha/\mu}\|_1}, \end{aligned}$$

$$\text{where } d(p) := \sum_i p_i \log p_i + \log n \quad (12)$$

is 1-strongly convex with respect to the  $\ell_1$ -norm (Nesterov, 2005). Define a smoothened loss function as in Eq. (9)

$$L_\mu(\alpha) = \sup_{p \in \Delta_n} \left\{ -\langle \alpha, p \rangle_G - \mu d(p) \right\} + \frac{1}{2} \|\alpha\|_G^2.$$

Note that the maximizer above is precisely  $p_\mu(\alpha)$ .

**Algorithm 4** Smoothed Normalized Kernel Perceptron

---

Set  $\alpha_0 = \mathbf{1}_n/n$ ,  $\mu_0 := 2$ ,  $p_0 := p_{\mu_0}(\alpha_0)$   
**for**  $k = 0, 1, 2, 3, \dots$  **do**  
   **if**  $G\alpha_k > \mathbf{0}_n$  **then**  
     Halt:  $\alpha_k$  is solution to Eq. (8)  
   **else**  
      $\theta_k := \frac{2}{k+3}$   
      $\alpha_{k+1} := (1 - \theta_k)(\alpha_k + \theta_k p_k) + \theta_k^2 p_{\mu_k}(\alpha_k)$   
      $\mu_{k+1} = (1 - \theta_k)\mu_k$   
      $p_{k+1} := (1 - \theta_k)p_k + \theta_k p_{\mu_{k+1}}(\alpha_{k+1})$   
   **end if**  
**end for**

---

**Lemma 4** (Lower Bound). *At any step  $k$ , we have*

$$L_{\mu_k}(\alpha_k) \geq L(\alpha_k) - \mu_k \log n.$$

*Proof.* First note that  $\sup_{p \in \Delta_n} d(p) = \log n$ . Also,

$$\begin{aligned} & \sup_{p \in \Delta_n} \{ -\langle \alpha, p \rangle_G - \mu d(p) \} \\ & \geq \sup_{p \in \Delta_n} \{ -\langle \alpha, p \rangle_G \} - \sup_{p \in \Delta_n} \{ \mu d(p) \}. \end{aligned}$$

Combining these two facts gives us the result.  $\square$

**Lemma 5** (Upper Bound). *In any round  $k$ , SNKP satisfies*

$$L_{\mu_k}(\alpha_k) \leq -\frac{1}{2} \|p_k\|_G^2.$$

*Proof.* We provide a concise, self-contained and unified proof by induction in the Appendix for Lemma 5 and Lemma 8, borrowing ideas from Nesterov's excessive gap technique (Nesterov, 2005) for smooth minimization of structured non-smooth functions.  $\square$

Finally, we combine the above lemmas to get the following theorem about the performance of SNKP.

**Theorem 1.** *The SNKP algorithm finds a perfect classifier  $f \in \mathcal{F}_K$  when one exists in  $O\left(\frac{\sqrt{\log n}}{\rho_K}\right)$  iterations.*

*Proof.* Lemma 4 gives us for any round  $k$ ,

$$L_{\mu_k}(\alpha_k) \geq L(\alpha_k) - \mu_k \log n.$$

From Lemmas 3, 5 we get

$$L_{\mu_k}(\alpha_k) \leq -\frac{1}{2} p_k^\top G p_k \leq -\frac{1}{2} \rho_K^2.$$

Combining the two equations, we get that

$$L(\alpha_k) \leq \mu_k \log n - \frac{1}{2} \rho_K^2.$$

Noting that  $\mu_k = \frac{4}{(k+1)(k+2)} < \frac{4}{(k+1)^2}$ , we see that  $L(\alpha_k) < 0$  (and hence we solve the problem by Lemma 1) after at most  $k = 2\sqrt{2 \log n} / \rho_K$  steps.  $\square$

## 5. Infeasible Problems

What happens when the points are not separable by any function  $f \in \mathcal{F}_K$ ? We would like an algorithm that terminates with a solution when there is one, and terminates with a certificate of non-separability if there isn't one. The idea is based on theorems of the alternative like Farkas' Lemma, specifically a version of Gordan's theorem (Chvatal, 1983):

**Lemma 6** (Gordan's Thm). *Exactly one of the following two statements can be true*

1. *Either there exists a  $w \in \mathbb{R}^d$  such that for all  $i$ ,*

$$y_i(w^\top x_i) > 0,$$

2. *Or, there exists a  $p \in \Delta_n$  such that*

$$\|XYp\|_2 = 0, \quad (13)$$

*or equivalently  $\sum_i p_i y_i x_i = 0$ .*

As mentioned in the introduction, the primal problem can be interpreted as finding a vector in the interior of the dual cone of  $\text{cone}\{y_i x_i\}$ , which is infeasible the dual cone is flat i.e. if  $\text{cone}\{y_i x_i\}$  is not pointed, which happens when the origin is in the convex combination of  $y_i x_i$ s.

We will generalize the following algorithm for linear feasibility problems, that can be dated back to Von-Neumann, who mentioned it in a private communication with Dantzig, who later studied it himself (Dantzig, 1992).

---

**Algorithm 5** Normalized Von-Neumann (NVN)

---

Initialize  $p_0 = \mathbf{1}_n/n$ ,  $w_0 = XYp_0$   
**for**  $k = 0, 1, 2, 3, \dots$  **do**  
   **if**  $\|XYp_k\|_2 \leq \epsilon$  **then**  
     Exit and return  $p_k$  as an  $\epsilon$ -solution to (13)  
   **else**  
      $j := \arg \min_i y_i x_i^\top w_k$   
      $\theta_k := \arg \min_{\lambda \in [0,1]} \|(1 - \lambda)w_k + \lambda y_j x_j\|_2$   
      $p_{k+1} := (1 - \theta_k)p_k + \theta_k e_j$   
      $w_{k+1} := XYp_{k+1} = (1 - \theta_k)w_k + \theta_k y_j x_j$   
   **end if**  
**end for**

---

This algorithm comes with a guarantee: *If the problem (3) is infeasible, then the above algorithm will terminate with an  $\epsilon$ -approximate solution to (13) in  $1/\epsilon^2$  iterations.*

(Epelman & Freund, 2000) proved an incomparable bound - Normalized Von-Neumann (NVN) can compute an  $\epsilon$ -solution to (13) in  $O\left(\frac{1}{\rho^2} \log\left(\frac{1}{\epsilon}\right)\right)$  and can also find a solution to the primal (using  $w_k$ ) in  $O\left(\frac{1}{\rho^2}\right)$  when it is feasible.

We derive a smoothed variant of NVN in the next section, after we prove some crucial lemmas in RKHSs.



### 5.1. A Separation Theorem for RKHSs

While finite dimensional Euclidean spaces come with strong separation guarantees that come under various names like the separating hyperplane theorem, Gordan's theorem, Farkas' lemma, etc, the story isn't always the same for infinite dimensional function spaces which can often be tricky to deal with. We will prove an appropriate version of such a theorem that will be useful in our setting.

What follows is an interesting version of the Hahn-Banach separation theorem, which looks a lot like Gordan's theorem in finite dimensional spaces. The conditions to note here are that either  $G\alpha > 0$  or  $\|p\|_G = 0$ .

**Theorem 2.** *Exactly one of the following has a solution:*

1. Either  $\exists f \in \mathcal{F}_K$  such that for all  $i$ ,

$$\frac{y_i f(x_i)}{\sqrt{K(x_i, x_i)}} = \langle f, y_i \tilde{\phi}_{x_i} \rangle_K > 0 \quad \text{i.e.} \quad G\alpha > 0,$$

2. Or  $\exists p \in \Delta_n$  such that

$$\sum_i p_i y_i \tilde{\phi}_{x_i} = 0 \in \mathcal{F}_K \quad \text{i.e.} \quad \|p\|_G = 0. \quad (14)$$

*Proof.* Consider the following set

$$\begin{aligned} Q &= \left\{ (f, t) = \left( \sum_i p_i y_i \tilde{\phi}_{x_i}, \sum_i p_i \right) : p \in \Delta_n \right\} \\ &= \text{conv} \left[ (y_1 \tilde{\phi}_{x_1}, 1), \dots, (y_n \tilde{\phi}_{x_n}, 1) \right] \\ &\subseteq \mathcal{F}_K \times \mathbb{R}. \end{aligned}$$

If (2) does not hold, then it implies that  $(0, 1) \notin Q$ . Since  $Q$  is closed and convex, we can find a separating hyperplane between  $Q$  and  $(0, 1)$ , or in other words there exists  $(f, t) \in \mathcal{F}_K \times \mathbb{R}$  such that

$$\begin{aligned} \langle (f, t), (g, s) \rangle &\geq 0 \quad \forall (g, s) \in Q \\ \text{and } \langle (f, t), (0, 1) \rangle &< 0. \end{aligned}$$

The second condition immediately yields  $t < 0$ . The first condition, when applied to  $(g, s) = (y_i \tilde{\phi}_{x_i}, 1) \in Q$  yields

$$\begin{aligned} \langle f, y_i \tilde{\phi}_{x_i} \rangle_K + t &\geq 0 \\ \Leftrightarrow \frac{y_i f(x_i)}{\sqrt{K(x_i, x_i)}} &> 0 \end{aligned}$$

since  $t < 0$ , which shows that (1) holds.

It is also immediate that if (2) holds, then (1) cannot.  $\square$

Note that  $G$  is positive semi-definite - infeasibility requires both that it is not positive definite, and also that the witness to  $p^\top G p = 0$  must be a probability vector. Similarly, while it suffices that  $G\alpha > 0$  for some  $\alpha \in \mathbb{R}^n$ , but coincidentally in our case  $\alpha$  will also lie in the probability simplex.

### 5.2. The infeasible margin $\rho_K$

Note that constraining  $\|f\|_K = 1$  (or  $\|\alpha\|_G = 1$ ) in Eq. (5) and Lemma 3 allows  $\rho_K$  to be negative in the infeasible case. If it was  $\leq$ , then  $\rho_K$  would have been non-negative because  $f = 0$  (ie  $\alpha = 0$ ) is always allowed.

So what is  $\rho_K$  when the problem is infeasible? Let

$$\text{conv}(Y\tilde{\phi}_X) := \left\{ \sum_i p_i y_i \tilde{\phi}_{x_i} \mid p \in \Delta_n \right\} \subset \mathcal{F}_K$$

be the convex hull of the  $y_i \tilde{\phi}_{x_i}$ s.

**Theorem 3.** *When the primal is infeasible, the margin<sup>1</sup> is*  
 $|\rho_K| = \delta_{\max} := \sup \left\{ \delta \mid \|f\|_K \leq \delta \Rightarrow f \in \text{conv}(Y\tilde{\phi}_X) \right\}$

*Proof. (1) For inequality  $\geq$ .* Choose any  $\delta$  such that  $f \in \text{conv}(Y\tilde{\phi}_X)$  for any  $\|f\|_K \leq \delta$ . Given an arbitrary  $f' \in \mathcal{F}_K$  with  $\|f'\|_K = 1$ , put  $\tilde{f} := -\delta f'$ .

By our assumption on  $\delta$ , we have  $\tilde{f} \in \text{conv}(Y\tilde{\phi}_X)$  implying there exists a  $\tilde{p} \in \Delta_n$  such that  $\tilde{f} = \langle Y\tilde{\phi}_X, \tilde{p} \rangle$ . Also

$$\begin{aligned} \langle f', \langle Y\tilde{\phi}_X, \tilde{p} \rangle \rangle_K &= \langle f', \tilde{f} \rangle_K \\ &= -\delta \|f'\|_K^2 = -\delta. \end{aligned}$$

Since this holds for a particular  $\tilde{p}$ , we can infer

$$\inf_{p \in \Delta_n} \langle f', \langle Y\tilde{\phi}_X, \tilde{p} \rangle \rangle_K \leq -\delta.$$

Since this holds for any  $f'$  with  $\|f'\|_G = 1$ , we have

$$\sup_{\|f\|_K=1} \inf_{p \in \Delta_n} \langle f', \langle Y\tilde{\phi}_X, \tilde{p} \rangle \rangle_K \leq -\delta \quad \text{i.e.} \quad |\rho_K| \geq \delta.$$

**(2) For inequality  $\leq$ .** It suffices to show  $\|f\|_K \leq |\rho_K| \Rightarrow f \in \text{conv}(Y\tilde{\phi}_X)$ . We will prove the contrapositive  $f \notin \text{conv}(Y\tilde{\phi}_X) \Rightarrow \|f\|_K > |\rho_K|$ .

Since  $\Delta_n$  is compact and convex,  $\text{conv}(Y\tilde{\phi}_X) \subset \mathcal{F}_K$  is closed and convex. Therefore if  $f \notin \text{conv}(Y\tilde{\phi}_X)$ , then there exists  $g \in \mathcal{F}_K$  with  $\|g\|_K = 1$  that separates  $f$  and  $\text{conv}(Y\tilde{\phi}_X)$ , i.e. for all  $p \in \Delta_n$ ,

$$\begin{aligned} \langle g, f \rangle_K &< 0 \quad \text{and} \quad \langle g, \langle Y\tilde{\phi}_X, p \rangle \rangle_K \geq 0 \\ \text{i.e. } \langle g, f \rangle_K &< \inf_{p \in \Delta_n} \langle g, \langle Y\tilde{\phi}_X, p \rangle \rangle_K \\ &\leq \sup_{\|f\|_K=1} \inf_{p \in \Delta_n} \langle f, \langle Y\tilde{\phi}_X, p \rangle \rangle_K = \rho_K. \end{aligned}$$

$$\begin{aligned} \text{Since } \rho_K < 0 \quad |\rho_K| &< |\langle f, g \rangle_K| \\ &\leq \|f\|_K \|g\|_K = \|f\|_K. \end{aligned}$$

$\square$

<sup>1</sup>We thank a reviewer for pointing out that by this definition,  $\rho_K$  might always be 0 for infinite dimensional RKHSs because there are always directions perpendicular to the finite-dimensional hull - we conjecture the definition can be altered to restrict attention to the relative interior of the hull, making it non-zero.

## 6. Kernelized Primal-Dual Algorithms

The preceding theorems allow us to write a variant of the Normalized VonNeumann algorithm from the previous section that is smoothed and works for RKHSs. Define

$$W := \left\{ p \in \Delta_n \mid \sum_i p_i y_i \tilde{\phi}_{x_i} = 0 \right\} = \left\{ p \in \Delta_n \mid \|p\|_G = 0 \right\}$$

as the set of witnesses to the infeasibility of the primal. The following lemma bounds the distance of any point in the simplex from the witness set by its  $\|\cdot\|_G$  norm.

**Lemma 7.** *For all  $q \in \Delta_n$ , the distance to the witness set*

$$\text{dist}(q, W) := \min_{w \in W} \|q - w\|_2 \leq \min \left\{ \sqrt{2}, \frac{\sqrt{2}\|q\|_G}{|\rho_K|} \right\}.$$

As a consequence,  $\|p\|_G = 0$  iff  $p \in W$ .

*Proof.* This is trivial for  $p \in W$ . For arbitrary  $p \in \Delta_n \setminus W$ , let  $\tilde{p} := -\frac{|\rho_K|p}{\|p\|_G}$  so that  $\|Y\tilde{\phi}_X, \tilde{p}\|_K = \|\tilde{p}\|_G \leq |\rho_K|$ .

Hence by Theorem 3, there exists  $\alpha \in \Delta_n$  such that

$$\langle Y\tilde{\phi}_X, \alpha \rangle = \langle Y\tilde{\phi}_X, \tilde{p} \rangle.$$

Let  $\beta = \lambda\alpha + (1 - \lambda)p$  where  $\lambda = \frac{\|p\|_G}{\|p\|_G + |\rho_K|}$ . Then

$$\begin{aligned} \langle Y\tilde{\phi}_X, \beta \rangle &= \frac{1}{\|p\|_G + |\rho_K|} \langle Y\tilde{\phi}_X, \|p\|_G\alpha + |\rho_K|p \rangle \\ &= \frac{1}{\|p\|_G + |\rho_K|} \langle Y\tilde{\phi}_X, \|p\|_G\tilde{p} + |\rho_K|p \rangle \\ &= 0, \end{aligned}$$

so  $\beta \in W$  (by definition of what it means to be in  $W$ ) and

$$\|p - \beta\|_2 = \lambda\|p - \alpha\|_2 \leq \lambda\sqrt{2} \leq \min \left\{ \sqrt{2}, \frac{\sqrt{2}\|q\|_G}{|\rho_K|} \right\}.$$

We take min with  $\sqrt{2}$  because  $\rho_K$  might be 0.  $\square$

Hence for the primal or dual problem, points with small G-norm are revealing - either Lemma 3 shows that the margin  $\rho_K \leq \|p\|_G$  will be small, or if it is infeasible then the above lemma shows that it is close to the witness set.

We need a small alteration to the smoothing entropy prox-function that we used earlier. We will now use

$$d_q(p) = \frac{1}{2}\|p - q\|_2^2$$

for some given  $q \in \Delta_n$ , which is strongly convex with respect to the  $\ell_2$  norm. This allows us to define

$$p_\mu^q(\alpha) = \arg \min_{p \in \Delta_n} \langle G\alpha, p \rangle + \frac{\mu}{2}\|p - q\|_2^2,$$

$$L_\mu^q(\alpha) = \sup_{p \in \Delta_n} \left\{ -\langle \alpha, p \rangle_G - \mu d_q(p) \right\} + \frac{1}{2}\|\alpha\|_G^2,$$

which can easily be found by sorting the entries of  $q - \frac{G\alpha}{\mu}$ .

**Algorithm 6** Smoothed Normalized Kernel Perceptron-VonNeumann (SNKPVN( $q, \delta$ ))

**input**  $q \in \Delta_n$ , accuracy  $\delta > 0$

Set  $\alpha_0 = q$ ,  $\mu_0 := 2n$ ,  $p_0 := p_{\mu_0}^q(\alpha_0)$

**for**  $k = 0, 1, 2, 3, \dots$  **do**

**if**  $G\alpha_k > \mathbf{0}_n$  **then**

Halt:  $\alpha_k$  is solution to Eq. (8)

**else if**  $\|p_k\|_G < \delta$  **then**

Return  $p_k$

**else**

$$\theta_k := \frac{2}{k+3}$$

$$\alpha_{k+1} := (1 - \theta_k)(\alpha_k + \theta_k p_k) + \theta_k^2 p_{\mu_k}^q(\alpha_k)$$

$$\mu_{k+1} = (1 - \theta_k)\mu_k$$

$$p_{k+1} := (1 - \theta_k)p_k + \theta_k p_{\mu_{k+1}}^q(\alpha_{k+1})$$

**end if**

**end for**

When the primal is feasible, SNKPVN is similar to SNKP.

**Lemma 8** (When  $\rho_K > 0$  and  $\delta < \rho_K$ ). *For any  $q \in \Delta_n$ ,*

$$-\frac{1}{2}\|p_k\|_G^2 \geq L_{\mu_k}^q(\alpha_k) \geq L(\alpha_k) - \mu_k.$$

Hence SNKPVN finds a separator  $f$  in  $O\left(\frac{\sqrt{n}}{\rho_K}\right)$  iterations.

*Proof.* We give a unified proof for the first inequality and Lemma 5 in the Appendix. The second inequality mimics Lemma 4. The final statement mimics Theorem 1.  $\square$

The following lemma captures the near-infeasible case.

**Lemma 9** (When  $\rho_K < 0$  or  $\delta > \rho_K$ ). *For any  $q \in \Delta_n$ ,*

$$-\frac{1}{2}\|p_k\|_G^2 \geq L_{\mu_k}^q(\alpha_k) \geq -\frac{1}{2}\mu_k \text{dist}(q, W)^2.$$

Hence SNKPVN finds a  $\delta$ -solution in at most  $O\left(\min\left\{\frac{\sqrt{n}}{\delta}, \frac{\sqrt{n}\|q\|_G}{\delta|\rho_K|}\right\}\right)$  iterations.

*Proof.* The first inequality is the same as in the above Lemma 8, and is proved in the Appendix.

$$\begin{aligned} L_{\mu_k}^q(\alpha_k) &= \sup_{p \in \Delta_n} \left\{ -\langle \alpha, p \rangle_G - \mu_k d_q(p) \right\} + \frac{1}{2}\|\alpha\|_G^2 \\ &\geq \sup_{p \in W} \left\{ -\langle \alpha, p \rangle_G - \mu_k d_q(p) \right\} \\ &= \sup_{p \in W} \left\{ -\frac{1}{2}\mu_k \|p - q\|_2^2 \right\} \\ &= -\frac{1}{2}\mu_k \text{dist}(q, W)^2 \\ &\geq -\mu_k \min \left\{ 2, \frac{\|q\|_G^2}{|\rho_K|^2} \right\} \quad \text{using Lemma 7.} \end{aligned}$$

Since  $\mu_k = \frac{4n}{(k+1)(k+2)} \leq \frac{4n}{(k+1)^2}$  we get

$$\|p_k\|_G \leq \frac{2\sqrt{n}}{k+1} \min \left\{ \sqrt{2}, \frac{\|q\|_G}{\rho_K} \right\}.$$

Hence  $\|p\|_G \leq \delta$  after  $\frac{2\sqrt{n}}{\delta} \min \left\{ \sqrt{2}, \frac{\|q\|_G}{\rho_K} \right\}$  steps.  $\square$

Using SNKPVN as a subroutine gives our final algorithm.

**Algorithm 7** Iterated Smoothed Normalized Kernel Perceptron-VonNeumann (*ISNKPVN*( $\gamma, \epsilon$ ))

---

**input** Constant  $\gamma > 1$ , accuracy  $\epsilon > 0$   
 Set  $q_0 := \mathbf{1}_n/n$   
**for**  $t = 0, 1, 2, 3, \dots$  **do**  
      $\delta_t := \|q_t\|_G/\gamma$   
      $q_{t+1} := \text{SNKPVN}(q_t, \delta_t)$   
     **if**  $\delta_t < \epsilon$  **then**  
         Halt;  $q_{t+1}$  is a solution to Eq. (14)  
     **end if**  
**end for**

---

**Theorem 4.** *Algorithm ISNKPVN satisfies*

1. *If the primal (2) is feasible and  $\epsilon < \rho_K$ , then each call to SNKPVN halts in at most  $\frac{2\sqrt{2n}}{\rho_K}$  iterations. Algorithm ISNKPVN finds a solution in at most  $\frac{\log(1/\rho_K)}{\log(\gamma)}$  outer loops, bounding the total iterations by*

$$O\left(\frac{\sqrt{n}}{\rho_K} \log\left(\frac{1}{\rho_K}\right)\right).$$

2. *If the dual (14) is feasible or  $\epsilon > \rho_K$ , then each call to SNKPVN halts in at most  $O\left(\min\left\{\frac{\sqrt{n}}{\epsilon}, \frac{\sqrt{n}}{|\rho_K|}\right\}\right)$  steps. Algorithm ISNKPVN finds an  $\epsilon$ -solution in at most  $\frac{\log(1/\epsilon)}{\log(\gamma)}$  outer loops, bounding the total iterations by*

$$O\left(\min\left\{\frac{\sqrt{n}}{\epsilon}, \frac{\sqrt{n}}{|\rho_K|}\right\} \log\left(\frac{1}{\epsilon}\right)\right).$$

*Proof.* First note that if ISNKPVN has not halted, then we know that after  $t$  outer iterations,  $q_{t+1}$  has small G-norm:

$$\|q_{t+1}\|_G \leq \delta_t \leq \frac{\|q_0\|_G}{\gamma^{t+1}}. \quad (15)$$

The first inequality holds because of the inner loop return condition, the second because of the update for  $\delta_t$ .

1. Lemma 3 shows that for all  $p$  we have  $\rho_K \leq \|p\|_G$ , so the inner loop will halt with a solution to the primal as soon as  $\delta_t \leq \rho_K$  (so that  $\|p\|_G < \delta_t \leq \rho_K$  cannot be satisfied for the inner loop to return). From Eq. (15), this will definitely happen when  $\frac{\|q_0\|_G}{\gamma^{t+1}} \leq \rho_K$ , ie within  $T = \frac{\log(\|q_0\|_G/\rho_K)}{\log(\gamma)}$  iterations. By Lemma 8, each iteration runs for at most  $\frac{2\sqrt{2n}}{\rho_K}$  steps.
2. We halt with an  $\epsilon$ -solution when  $\delta_t < \epsilon$ , which definitely happens when  $\frac{\|q_0\|_G}{\gamma^{t+1}} < \epsilon$ , ie within  $T = \frac{\log(\|q_0\|_G/\epsilon)}{\log(\gamma)}$  iterations. Since  $\frac{\|q_t\|_G}{\delta_t} = \gamma$ , by Lemma 9, each iteration runs for at most  $O\left(\min\left\{\frac{\sqrt{n}}{\epsilon}, \frac{\sqrt{n}}{|\rho_K|}\right\}\right)$  steps.  $\square$

## 7. Discussion

The SNK-Perceptron algorithm presented in this paper has a convergence rate of  $\frac{\sqrt{\log n}}{\rho_K}$  and the Iterated SNK-Perceptron-Von-Neumann algorithm has a  $\min\left\{\frac{\sqrt{n}}{\epsilon}, \frac{\sqrt{n}}{|\rho_K|}\right\}$  dependence on the number of points. Note that both of these are independent of the underlying dimensionality of the problem. We conjecture that it is possible to reduce this dependence to  $\sqrt{\log n}$  for the primal-dual algorithm also, without paying a price in terms of the dependence on margin  $1/\rho$  (or the dependence on  $\epsilon$ ).

It is possible that tighter dependence on  $n$  is possible if we try other smoothing functions instead of the  $\ell_2$  norm used in the last section. Specifically, it might be tempting to smooth with the  $\|\cdot\|_G$  semi-norm and define:

$$p^q(\alpha) = \arg \min_{p \in \Delta_n} \langle \alpha, p \rangle_G + \frac{\mu}{2} \|p - q\|_G^2$$

One can actually see that the proofs in the Appendix go through with no dimension dependence on  $n$  at all! However, it is not possible to solve this in closed form - taking  $\alpha = q$  and  $\mu = 1$  reduces the problem to asking

$$p^q(q) = \arg \min_{p \in \Delta_n} \frac{1}{2} \|p\|_G^2$$

which is an oracle for our problem as seen by equation (14) - the solution's G-norm is 0 iff the problem is infeasible.

In the bigger picture, there are several interesting open questions. The ellipsoid algorithm for solving linear feasibility problems has a logarithmic dependence on  $1/\epsilon$ , and a polynomial dependence on dimension. Recent algorithms involving repeated rescaling of the space like (Dunagan & Vempala, 2008) have logarithmic dependence on  $1/\rho$  and polynomial in dimension. While both these algorithms are poly-time under the real number model of computation of (Blum et al., 1998), it is unknown whether there is any algorithm that can achieve a polylogarithmic dependence on the margin/accuracy, and a polylogarithmic dependence on dimension. This is strongly related to the open question of whether it is possible to learn a decision list polynomially in its binary description length.

One can nevertheless ask whether rescaled smoothed perceptron methods like (Dunagan & Vempala, 2008) can be lifted to RKHSs, and whether using an iterated smoothed kernel perceptron would yield faster rates. The recent work (Soheili & Peña, 2013b) is a challenge to generalize - the proofs relying on geometry involve arguing about volumes of balls of functions in an RKHS - we conjecture that it is possible to do, but we leave it for a later work.

### Acknowledgements

We thank Negar Soheili, Avrim Blum for discussions and the excellent reviewers for references and Footnote 1.



## References

- Block, HD. The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, 34(1):123, 1962.
- Blum, Lenore, Cucker, Felipe, Shub, Michael, and Smale, Steve. *Complexity and real computation*. Springer, 1998.
- Cheung, Dennis and Cucker, Felipe. A new condition number for linear programming. *Mathematical programming*, 91(1):163–174, 2001.
- Chvatal, Vasek. *Linear programming*. Macmillan, 1983.
- Clarkson, Kenneth L. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010.
- Dantzig, George B. An  $\epsilon$ -precise feasible solution to a linear program with a convexity constraint in  $1/\epsilon^2$  iterations independent of problem size. Technical report, Technical Report, Stanford University, 1992.
- Dunagan, John and Vempala, Santosh. A simple polynomial-time rescaling algorithm for solving linear programs. *Mathematical Programming*, 114(1):101–114, 2008.
- Epelman, Marina and Freund, Robert M. Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Mathematical Programming*, 88(3):451–485, 2000.
- Freund, Robert M and Vera, Jorge R. Condition-based complexity of convex optimization in conic linear form via the ellipsoid algorithm. *SIAM Journal on Optimization*, 10(1):155–176, 1999.
- Graepel, Thore, Herbrich, Ralf, and Williamson, Robert C. From margin to sparsity. *Advances in neural information processing systems*, pp. 210–216, 2001.
- Littlestone, Nicholas. Redundant noisy attributes, attribute errors, and linear-threshold learning using winnow. In *Proceedings of the fourth annual workshop on Computational learning theory*, pp. 147–156. Morgan Kaufmann Publishers Inc., 1991.
- Nemirovski, Arkadi. Prox-method with rate of convergence  $O(1/t)$  for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.
- Nesterov, Yu. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1):235–249, 2005.
- Novikoff, Albert BJ. On convergence proofs for perceptrons. Technical report, 1962.
- Renegar, James. Incorporating condition measures into the complexity theory of linear programming. *SIAM Journal on Optimization*, 5(3):506–524, 1995.
- Rosenblatt, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- Saha, Ankan, Vishwanathan, SVN, and Zhang, Xinhua. New approximation algorithms for minimum enclosing convex shapes. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1146–1160. SIAM, 2011.
- Schölkopf, Bernhard and Smola, Alexander J. *Learning with kernels*. The MIT Press, 2002.
- Schölkopf, Bernhard, Herbrich, Ralf, and Smola, Alex J. A generalized representer theorem. In *Computational learning theory*, pp. 416–426. Springer, 2001.
- Soheili, Negar and Peña, Javier. A smooth perceptron algorithm. *SIAM Journal on Optimization*, 22(2):728–737, 2012.
- Soheili, Negar and Peña, Javier. A primal–dual smooth perceptron–von Neumann algorithm. In *Discrete Geometry and Optimization*, pp. 303–320. Springer, 2013a.
- Soheili, Negar and Peña, Javier. A deterministic rescaled perceptron algorithm. 2013b.
- Tseng, Paul. On accelerated proximal gradient methods for convex-concave optimization. *SIAM Journal on Optimization*, 2008.