# Unsupervised SVMs: On the Complexity of the Furthest Hyperplane Problem

**Zohar Karnin** ZKARNIN@YAHOO-INC.COM
**Edo Liberty** EDO@YAHOO-INC.COM
*Yahoo! Research, Haifa*

**Shachar Lovett** SLOVETT@MATH.IAS.EDU
*IAS, Institute for Advanced Studies*

**Roy Schwartz** SCHWARTZ@CS.TECHNION.AC.IL
*Technion and Yahoo! Research*

**Omri Weinstein** OWEINSTE@CS.PRINCETON.EDU
*Princeton and Yahoo! Research*

**Editor:** Shie Mannor, Nathan Srebro, Robert C. Williamson

## Abstract

This paper introduces the Furthest Hyperplane Problem (FHP), which is an unsupervised counterpart of Support Vector Machines. Given a set of $n$ points in $\mathbb{R}^d$, the objective is to produce the hyperplane (passing through the origin) which maximizes the separation margin, that is, the minimal distance between the hyperplane and any input point.

To the best of our knowledge, this is the first paper achieving provable results regarding FHP. We provide both lower and upper bounds to this NP-hard problem. First, we give a simple randomized algorithm whose running time is $n^{O(1/\theta^2)}$ where $\theta$ is the optimal separation margin. We show that its exponential dependency on $1/\theta^2$ is tight, up to sub-polynomial factors, assuming SAT cannot be solved in sub-exponential time. Next, we give an efficient approximation algorithm. For any $\alpha \in [0, 1]$, the algorithm produces a hyperplane whose distance from at least $1 - 3\alpha$ fraction of the points is at least $\alpha$ times the optimal separation margin. Finally, we show that FHP does not admit a PTAS by presenting a gap preserving reduction from a particular version of the PCP theorem.

## 1. Introduction

One of the most well known and studied objective functions in machine learning for obtaining linear classifiers is the Support Vector Machines (SVM) objective. SVM's are extremely well studied, both in theory and in practice. We refer the reader to Vapnik and Lerner (1963); Mangasarian (1965) and to Burges (1998) for a thorough survey and references therein. The simplest possible setup is the separable case. Given a set of $n$ points $\{x^{(i)}\}_{i=1}^n$ in $\mathbb{R}^d$ and labels $y_1, \ldots y_n \in \{1, -1\}$ find hyperplane parameters $w \in \mathbb{S}^{d-1}$ (the unit sphere in $\ell_2$ in dimension $d$) and $b \in \mathbb{R}$ which maximize $\theta'$ subject to $(\langle w, x^{(i)}\rangle + b)y_i \geq \theta'$. The intuition is that different concepts will be "well separated" from each other and that the best decision boundary is the one that maximizes the separation. This intuition is supported by extensive research which is beyond the scope of this paper. Algorithmically, the optimal solution for this problem can be obtained using Quadratic Programing or the Ellipsoid Method in polynomial time. In cases where the problem has no feasible solution the constraints must be made “soft” and the optimization problem becomes significantly harder. This discussion, however, also goes beyond the scope of this paper.

As a whole, SVM’s fall under the category of supervised learning, although semi-supervised and unsupervised versions have also been considered (see references below). We note that to the best of our knowledge the papers dealing with the unsupervised scenario were purely experimental and did not contain any rigorous proofs. In this model, the objective remains unchanged but some (or possibly all) of the point labels are unknown. The maximization, thus, ranges not only over the parameters $w$ and $b$ but also over the possible labels for the unlabeled points $y_i \in \{1, -1\}$. The integer constraints on the values of $y_i$ make this problem significantly harder than SVM’s.

The name Maximal Margin Clustering (MMC) was coined by Xu et al. (2005) for the case where none of the labels are known. Indeed, in this setting the learning procedure behaves very much like clustering. The objective is to assign the points to two groups (indicated by $y_i$) such that solving the labeled SVM problem according to this assignment produces the maximal margin.[1] Bennett and Demiriz (1998) propose to solve the resulting mixed integer quadratic program directly using general solvers and give some encouraging experimental results. Bie and Cristianini (2003) and Xu et al. (2005) suggest an SDP relaxation approach and show that it works well in practice. Joachims (1999) suggests a local search approach which iteratively improves on a current best solution. While the above algorithms produce good results in practice, their analysis does not guaranty the optimality of the solution. Moreover, the authors of these papers state their belief that the non convexity of this problem makes it hard, but to the best of our knowledge no proof of this was given. In a recent work Peng et al. (2011) suggests an efficient approach to MMC based on gradual feature selection, but is mainly supported by numerical experiments.

FHP is very similar to unsupervised SVM or Maximum Margin Clustering. The only difference is that the solution hyperplane is constrained to pass through the origin. Formally, given $n$ points $\{x^{(i)}\}_{i=1}^{n}$ in a $d$-dimensional Euclidean space, FHP is defined as follows:

$$\begin{array}{rrcl} \text{Maximize} & \theta' & & \\ & \text{s.t} \quad \|w\|^2 & = & 1 \\ \forall\, 1 \le i \le n & |\langle w \cdot x^{(i)} \rangle| & \ge & \theta' \end{array} \tag{1}$$

The labels in this formulation are given by $y_i = sign(\langle w \cdot x^{(i)} \rangle)$ which can be viewed as the “side” of the hyperplane to which $x^{(i)}$ belongs. At first glance, MMC appears to be harder than FHP since it optimizes over a larger set of possible solutions. Namely, those for which $b$ (the hyperplane offset) is not necessarily zero. We claim however that any MMC problem can be solved using at most $\binom{n}{2}$ invocations of FHP. The simple observation is that any optimal solution for MMC must have two equally distant points in opposite sides of the hyperplane. Therefore, there always are at least two points $i$ and $j$ such that $(\langle w, x^{(i)} \rangle + b) = -(\langle w, x^{(j)} \rangle + b)$. This means that the optimal hyperplane obtained by MMC must pass through the point $(x^{(i)} + x^{(j)})/2$. Thus, solving FHP centered at $(x^{(i)} + x^{(j)})/2$ will yield the same hyperplane as MMC. Iterating over all pairs of points concludes the observation. From this point on we explore FHP exclusively but the reader should keep in mind that any algorithmic claim made for FHP holds also for MMC due to the above.

[1] The assignment is required to label at least one point to each cluster to avoid a trivial unbounded margin.

### 1.1. Results and techniques

In Section 2 we begin by describing three exact (yet exponential) algorithms for FHP. These algorithms are somewhat naïve and their proofs use standard techniques. However, we choose to present them for two reasons. First, they are the natural directions to consider and give the reader a reacher and fuller understanding of the problem (so we hope). Second, they turn out to be preferable to one another for different problem parameters. These parameter are: the dimension $d$, the number of points $n$, and the optimal margin $\theta$ which is not known apriori.

The first algorithm is a brute force search through all feasible labelings which runs in time $n^{O(d)}$. The second looks for a solution by enumerating over an $\varepsilon$-net of the $d$-dimensional unit sphere and requires $(1/\theta)^{O(d)}$ operations. The last generates solutions created by random unit vectors and can be shown to find the right solution after $n^{O(1/\theta^2)}$ tries (w.h.p.). While algorithmically the random hyperplane algorithm is the simplest, its analysis is the most complex. Assuming a large *constant* margin, which is not unrealistic in machine learning applications, this algorithm provides the first polynomial time solution to FHP. Unfortunately, due to the hardness result below, its exponential dependency on $\theta$ cannot be improved.

In section 3 we show that if one is allowed to discard a small fraction of the points then much better results can be obtained. We note that in the perspective of machine learning, a hyperplane that separates almost all of the points still provides a meaningful result (see the discussion at the end of section 3) . We give an efficient algorithm which finds a hyperplane whose distance from at least $1 - 3\alpha$ fraction of the points is at least $\alpha\theta$ , where $\alpha \in [0, 1]$ is any constant and $\theta$ is the optimal margin of the original problem. The main idea is to first find a small set of solutions which perform well 'on average'. These solutions are the singular vectors of row reweighed versions of a matrix containing the input points. We then randomly combine those to a single solution.

In section 4 we prove that FHP is NP-hard to approximate to within a small multiplicative constant factor, ruling out a PTAS. We present a two-step gap preserving reduction from MAX-3SAT using a particular version of the PCP theorem, see Arora (1994). It shows that the problem is hard even when the number of points is linear in the dimension and when all the points have approximately the same norm. As a corollary of the hardness result we get that the running time of our exact solution algorithm is, in a sense, optimal. There cannot be an algorithm solving FHP in time $n^{O(1/\theta^{2-\varepsilon})}$ for any constant $\varepsilon > 0$, unless SAT admits a sub-exponential time algorithm.

### 1.2. Preliminaries and notations

The set $\{x^{(i)}\}_{i=1}^{n}$ of input points for FHP is assumed to lie in a Euclidean space $\mathbb{R}^d$, endowed with the standard inner product denoted by $\langle \cdot, \cdot \rangle$. Unless stated otherwise, we denote by $\| \cdot \|$ the $\ell_2$ norm. Throughout the paper we let $\theta$ denote the solution of the optimization problem defined in Equation (1). The parameter $\theta$ is also referred to as "the margin of $\{x^{(i)}\}_{i=1}^{n}$", or simply "the margin" when it is obvious to which set of points it refers to. Unless stated otherwise, we consider only hyperplanes which pass through the origin. They are defined by their normal vector $w$ and include all points $x$ for which $\langle w, x \rangle = 0$. By a slight abuse of notation, we usually refer to a hyperplane by its defining normal vector $w$. Due to the scaling invariance of this problem we assume w.l.o.g. that $\|x^{(i)}\| \leq 1$. One convenient consequence of this assumption is that $\theta \leq 1$. We denote by $\mathcal{N}(\mu, \sigma)$ the standard Gaussian distribution with mean $\mu$ and standard deviation $\sigma$.

**Definition 1 (Labeling, feasible labeling)** *We refer to any assignment of $y_1, \ldots, y_n \in \{1, -1\}$ as a labeling. We say that a labeling is feasible if there exists $w \in \mathbb{S}^{d-1}$ such that $\forall i :\ y_i \left\langle w, x^{(i)} \right\rangle > 0$. For any hyperplane $w \in \mathbb{S}^{d-1}$ we define its labeling as $y_i = sign(\left\langle w, x^{(i)} \right\rangle)$.*

**Definition 2 (Labeling margin)** *The margin of a feasible labeling is the margin obtained by solving* SVM *on $\{x^{(i)}\}_{i=1}^n$ using the corresponding labels but constraining the hyperplane to pass through the origin. This problem is polynomial time solvable by Quadratic Programing or by the Ellipsoid Method Kozlov et al. (1979). We say a feasible labeling is optimal if it obtains the maximal margin.*

## 2. Exact algorithms

### 2.1. Enumeration of feasible labelings

The most straightforward algorithm for this problem enumerates over all feasible labelings of the points and outputs the one maximizing the margin. Note that there are at most $n^{d+1}$ different feasible labelings to consider. This is due to Sauer's Lemma Sauer (1972) and the fact that the VC dimension of hyperplanes in $\mathbb{R}^d$ is $d + 1$.[2] This enumeration can be achieved by a Breadth First Search (BFS) on the graph $G(Y, E)$ of feasible labelings. Every node in the graph $G$ is a feasible labeling ($|Y| \leq n^{d+1}$) and two nodes are connected by an edge iff their corresponding labelings differ by at most one point label. Thus, the maximal degree in the graph is $n$ and the number of edges in this graph is at most $|E| \leq |Y|n \leq n^{d+2}$. Moreover, computing for each node its neighbors list can be done efficiently since we only need to check the feasibility (linear separability) of at most $n$ labelings. Performing BFS thus requires at most $O(|Y|\text{poly}(n, d) + |E| \log(|E|)) = n^{d+O(1)}$. The only non trivial observation is that the graph $G$ is connected. To see this, consider the path from a labeling $y$ to a labeling $y'$. This path exists since it is achieved by rotating a hyperplane corresponding to $y$ to one corresponding to $y'$. By an infinitesimal perturbation on the point set (which does not effect any feasible labeling) we get that this rotation encounters only one point at a time and constitutes a path in $G$. To conclude, there is a simple enumeration procedure for all $n^{d+1}$ linearly separable labelings which runs in time $n^{d+O(1)}$.

### 2.2. An $\varepsilon$-net algorithm

The second approach is to search through a large enough set of hyperplanes and measure the margins produced by the labelings they induce. Note that it is enough to find one hyperplane which obtains the same labels as the optimal margin does. This is because having the labels suffices for solving the labeled problem and obtaining the optimal hyperplane. We observe that the correct labeling is obtained by any hyperplane $w$ whose distance from the optimal one is $\|w - w^*\| < \theta$. To see this, let $y^*$ denote the correct optimal labeling $y_i^* \langle w, x^{(i)} \rangle = \langle w^*, y_i^* x^{(i)} \rangle + \langle w - w^*, y_i^* x^{(i)} \rangle \geq \theta - \|w - w^*\| \cdot \|x^{(i)}\| > 0$. Hence, it is enough to consider hyperplane normals $w$ which belong to an $\varepsilon$-net on the sphere $\mathbb{S}^{d-1}$ with $\varepsilon < \theta$. Deterministic constructions of such nets exist with size $(1/\theta)^{O(d)}$ Lorentz et al. (1996). Enumerating all the points on the net produces an algorithm which runs in time $O((1/\theta)^{O(d)}\text{poly}(n, d))$.[3]

[2] Sauer's Lemma Sauer (1972) states that the number of possible feasible labelings of $n$ data points by a classifier with VC dimension $d_{VC}$ is bounded by $n^{d_{VC}}$.

[3] This procedure assumes the margin $\theta$ is known. This assumption can be removed by a standard doubling argument.

### 2.3. Random Hyperplane Algorithm

Both algorithms above are exponential in the dimension, even when the margin $\theta$ is large. A first attempt at taking advantage of the large margin uses dimension reduction. An easy corollary of the well known Johnson-Lindenstrauss lemma yields that randomly projecting the data points into dimension $O(\log(n)/\theta^2)$ preserves the margin up to a constant. Then, applying the $\varepsilon$-net algorithm on the reduced space requires only $n^{O(\log(1/\theta)/\theta^2)}$ operations. Similar ideas were introduced in Arriaga and Vempala (1999) and subsequently used by Klivans and Servedio (2004); Har-peled et al. (2006) and florina Balcan et al. (2004). However, a simpler approach improves on this: pick $n^{O(1/\theta^2)}$ unit vectors $w$ uniformly at random from the unit sphere. Output the labeling induced by one of those vectors which maximizes the margin. To establish the correctness of this algorithm it suffices to show that a random hyperplane induces the optimal labeling with a large enough probability.

**Lemma 3** *Let $w^*$ and $y^*$ denote the optimal solution of margin of $\theta$ and the labeling it induces. Let $y$ be the labeling induced by a random hyperplane $w$. The probability that $y = y*$ is at least $n^{-O(1/\theta^2)}$.*

The proof of the lemma is somewhat technical and is deferred to Appendix A. The assertion of the lemma may seem surprising at first. The measure of the spherical cap of vectors $w$ whose distance from $w^*$ is at most $\theta$ is only $\approx \theta^d$. Thus, the probability that a random $w$ falls in this spherical cap is very small. However, we show that it suffices for $w$ to merely have a weak correlation with $w^*$ in order to guarantee that (with large enough probability) it induces the optimal labeling.

Given Lemma 3, the Random Hyperplane Algorithm is straightforward: randomly sample $n^{O(1/\theta^2)}$ hyperplanes, compute their induced labelings, and output the labeling (or hyperplane) which admits the largest margin. If the margin $\theta$ is not known, we use a standard doubling argument to enumerate it. The algorithm solves FHP w.h.p. in time $n^{O(1/\theta^2)}$.

**Tightness of Our Result** A corollary of our hardness result (Theorem 12) is that, unless SAT has sub-exponential time algorithms, there exists no algorithm for FHP whose running time is $n^{O(\theta^{1/(2-\zeta)})}$ for any $\zeta > 0$. Thus, the exponential dependency of the Random Hyperplane Algorithm on $\theta$ is optimal. This is since the hard FHP instance produced by the reduction in Theorem 12 from SAT has $n$ points in $\mathbb{R}^d$ with $d = O(n)$ where the optimal margin is $\theta = \Omega(1/\sqrt{d})$. Thus, if there exists an algorithm which solves FHP in time $n^{O(\theta^{1/(2-\zeta)})}$, it can be used to solve SAT in time $2^{O(n^{1-\zeta/2}\log(n))} = 2^{o(n)}$.

## 3. Approximation algorithm

In this section we present a simple and efficient algorithm which approximates the optimal margin if one is allowed to discard a small fraction of the points. For any $\alpha > 0$ it finds a hyperplane whose distance from $(1 - O(\alpha))$-fraction of the points is at least $\alpha$ times the optimal margin $\theta$ of the original problem.

Consider first the easier problem of finding the hyperplane whose *average* margin is larger than $\theta$. The optimal hyperplane $w$ is simply the top right singular vector of a matrix $A$ whose $i$'th row contains $x^{(i)}$. To see this, assume the problem has a separating hyperplane $w^*$ with margin $\theta$ and let $\mathbb{E}_i$ denote the expectation over choosing $i$ uniformly at random from $[n]$. Then, $\mathbb{E}_i \left\langle w, x^{(i)} \right\rangle^2 = 1/n \sum_i \left\langle w, x^{(i)} \right\rangle^2 \geq 1/n \sum_i \left\langle w^*, x^{(i)} \right\rangle^2 = \mathbb{E}_i \left\langle w^*, x^{(i)} \right\rangle^2 \geq \theta^2$. This is simply because $w$ maximizes the expresion $\sum_i \left\langle w, x^{(i)} \right\rangle^2$. However, there is no guarantee that this singular vector obtains a high margin value $|\langle w, x^{(i)}\rangle|$ for *all* the points $x^{(i)}$. It is possible, for example, that $|\langle w, x^{(i)}\rangle| = 1$ for $\theta^2 n$ points and 0 for all the rest. Our first goal is to produce a set of weak solution hyperplanes $w^{(1)}, \dots, w^{(t)}$ which are good on average for *every* point. Namely, $\forall\, i:\ \mathbb{E}_j \langle w^{(j)}, x^{(i)}\rangle^2 = \Omega(\theta^2)$. To achieve this, we adaptively re-weight points according to their distance to previous weak solutions. Points which exhibit a large margin to current weak solutions, are weighted down so their influence is reduced. We then combine the weak solutions using random Gaussian weights to obtain a single random hyperplane which is good for any individual point w.p.

```
Algorithm 1: Approximate FHP Algorithm
Input: Set of points $\{x^{(i)}\}_{i=1}^n \in \mathbb{R}^d$
Output: $w \in \mathbb{S}^{d-1}$
$\tau_1(i) \leftarrow 1$ for all $i \in [n]$
$j \leftarrow 1$
while $\sum_{i=1}^n \tau_j(i) \geq 1/n$ do
    $A_j \leftarrow n \times d$ matrix whose $i$'th row is $\sqrt{\tau_j(i)} \cdot x^{(i)}$
    $w^{(j)} \leftarrow$ top right singular vector of $A_j$
    $\sigma_j(i) \leftarrow |\langle x^{(i)}, w^{(j)}\rangle|$
    $\tau_{j+1}(i) \leftarrow \tau_j(i) \cdot c^{-\sigma_j^2(i)}$
    $j \leftarrow j + 1$
end while
$w' \leftarrow \sum_{j=1}^t g_j \cdot w^{(j)}$ for $g_j \sim \mathcal{N}(0,1)$
return: $w \leftarrow w'/\|w'\|$
```

We note that our technique resembles the regret minimization framework. However, due to the different nature of our objective, a straight forward implementation of this approach does not work.[4] Additionally, note that the last step of combining the solution does not use averaging, as in the regret minimization framework, but rather a random Gaussian combination, as the former fails. The constant $c$ will be determined later.

**Claim 4** *Algorithm 1 terminates after at most $t \leq 2\ln(n)/\left(\theta^2(1-1/c)\right)$ iterations.*

**Proof** Fix some $j$. Define $\tau_j \triangleq \sum_{i=1}^n \tau_j(i)$. We know that for some unit vector $w^*$ (the optimal solution to the FHP) it holds that $|\langle x^{(i)}, w^*\rangle| \geq \theta$ for all $i$. Also since $w^{(j)}$ maximizes the expression $\|A_j w\|^2$ we have:

$$\sum_{i=1}^n \sigma_j^2(i)\tau_j(i) = \|A_j w^{(j)}\|^2 \geq \|A_j w^*\|^2 = \sum_{i=1}^n \tau_j(i) \cdot \left\langle x^{(i)}, w^*\right\rangle^2 \geq \tau_j \cdot \theta^2.$$

It follows that (and by using the fact that $c^{-x} \leq 1 - (1-1/c)x$ whenever $0 \leq x \leq 1$):

$$\tau_{j+1} = \sum_{i=1}^n \tau_j(i) \cdot c^{-\sigma_j^2(i)} \leq \sum_{i=1}^n \tau_j(i) \cdot \left(1 - \left(1 - \frac{1}{c}\right)\sigma_j^2(i)\right) \leq \tau_j \cdot \left(1 - \theta^2\left(1 - \frac{1}{c}\right)\right),$$

[4] A more involved, and somewhat less intuitive, use of the regret minimization framework can be applied. We defer the details to a full version of this paper, and include for completeness a full proof using continuous weights.

and the claim follows since $\tau_1 = n$ and $\ln\left(\frac{1}{1-x}\right) \geq x$ whenever $0 \leq x < 1$. ■

**Claim 5** *Let $\sigma_i \triangleq \sqrt{\sum_{j=1}^{t} \sigma_j^2(i)}$. When Algorithm 1 terminates, for each $i$ it holds*

$$\sigma_i^2 \geq \ln(n)/\ln(c).$$

**Proof** Fix $i \in [n]$. When the process ends, $\tau_t(i) \leq \tau_t < 1/n$. As $\tau_1(i) = 1$ we get that:

$$1/n \geq \tau_t(i) = \tau_1(i) \cdot \prod_{j=1}^{t} c^{-\sigma_2^2(i)} = c^{-\sum_{j=1}^{t} \sigma_j^2(i)}.$$

By taking logarithms from both sides, we get that $\sum_{j=1}^{t} \sigma_j^2(i) \geq \log(n)/\ln(c)$ as claimed. ■

The following lemma states the approximation guarantee of Algorithm 1.[5]

**Lemma 6** *Let $0 < \alpha < 1$. Algorithm 1 outputs a random $w \in \mathbb{S}^{d-1}$ such that with probability at least $1/147$ at most a $3\alpha$ fraction of the points are such that $\left|\left\langle x^{(i)}, w\right\rangle\right| \leq \alpha\theta$.*

**Proof** First, by Markov's inequality and the fact that $\mathbb{E}[\|w'\|^2] = t$ we have that $\|w'\| \leq 7/4 \cdot \sqrt{t}$ w.p. at least $33/49$. We assume this to be the case from this point on. Note that we do not condition on this event happening. Rather, we accept a $16/49$ failure probability which we include in a union bound later in the proof. Now we bound the probability that the algorithm 'fails' for point $i$.

$$\begin{aligned}
\Pr\left[\left|\left\langle w, x^{(i)}\right\rangle\right| \leq \alpha\theta\right] &\leq \Pr\left[\left|\left\langle w', x^{(i)}\right\rangle\right| \leq \frac{7}{4}\sqrt{t}\alpha\theta\right] \\
&\leq \Pr_{Z\sim\mathcal{N}(0,\sqrt{\ln(n)/\ln(c)})}\left[|Z| \leq \frac{7}{4}\sqrt{t}\alpha\theta\right] \\
&= \Pr_{Z\sim\mathcal{N}(0,1)}\left[|Z| \leq \frac{7}{4}\frac{\sqrt{\ln(c)}\sqrt{t}\alpha\theta}{\sqrt{\ln(n)}}\right] \\
&\leq \frac{7}{2\sqrt{2\pi}}\frac{\sqrt{\ln(c)}\sqrt{t}\alpha\theta}{\sqrt{\ln(n)}} \leq \frac{7\sqrt{\ln(c)}\alpha}{\sqrt{4\pi\left(1-\frac{1}{c}\right)}}
\end{aligned}$$

The second inequality is derived by using Lemma 5 and the last inequality is derived by using Lemma 4. Since the expected fraction of failed points is less than $7\sqrt{\ln(c)}\alpha/\sqrt{4\pi\left(1-\frac{1}{c}\right)}$ we have, using Markov's inequality again, that the probability that the number of failed points is more than $3/2 \cdot 7\sqrt{\ln(c)}\alpha/\sqrt{4\pi\left(1-\frac{1}{c}\right)} \cdot n$ is at most $2/3$. We also might fail with probability at most $16/49$ in the case that $\|w'\| > 7/4 \cdot \sqrt{t}$. Using the union bound on the two failure probabilities and choosing $c = 1.02$ completes the proof. ■

[5] We note that we did not try to optimize the constants since the application at hand might provide different restrictions on the several parameters of the algorithm, such as its success probability, its running time or the fraction of "bad" points in its output, i.e., points $x^{(i)}$ such that $\left|\left\langle x^{(i)}, w\right\rangle\right| \leq \alpha\theta$.

**Discussion** We note that the problem of finding a hyperplane that separates all but a small fraction of the points is the non-supervised analog of the well studied *soft margin* SVM problem. The motivation behind the problem, from the perspective of machine learning, is that a hyperplane that separates most of the data points is still likely to correctly label future points. Hence, if a hyperplane that separates all of the points cannot be obtained, it suffices to find one that separates most (e.g. $1-\alpha$ fraction) of the data points. The more common setting in which this problem is presented is when a separating hyperplane does not necessarily exist. In our case, although a separating hyperplane is guaranteed to exist, it is (provably) computationally hard to obtain it, as we show in the next section.

## 4. Hardness of approximation

The main result of this section is that FHP does not admit a PTAS unless P=NP. That is, obtaining a $(1-\varepsilon)$-approximation for FHP is NP-hard for some universal constant $\varepsilon$. The main idea is straightforward: Reduce from MAX-3SAT for which such a guarantee is well known, mapping each clause to a vector. We show that producing a "far" hyperplane from this set of vectors encodes a good solution for the satisfiability problem. However, FHP is inherently a symmetric problem (negating a solution does not change its quality) while MAX-3SAT does not share this property. Thus, we carry out our reduction in two steps: in the first step we reduce MAX-3SAT to a symmetric satisfaction problem. In the second step we reduce this symmetric satisfaction problem to FHP. It turns out that in order to show that such a symmetric problem can be geometrically embedded as a FHP instance, we need the extra condition that each variable appears in at most a constant number of clauses, and that the number of variables and clauses is comparable to each other. The reduction process is slightly more involved in order to guarantee this. In the rest of this section we consider the following satisfaction problem.

**Definition 7 (SYM formulas)** *A SYM formula is a CNF formula where each clause has either 2 or 4 literals. Moreover, clauses appear in pairs, where the two clauses in each pair have negated literals. For example, a pair with 4 literals has the form*

$$(x_1 \vee x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4).$$

*We denote by SYM$(t)$ the class of SYM formulas in which each variable occurs in at most $t$ clauses.*

We note that SYM formulas are invariant to negations: if an assignment $x$ satisfies $m$ clauses in a SYM formula than its negation $\neg x$ will satisfy the same number of clauses.

The following definition will play a central role in the reduction we next describe.

**Definition 8 (Expander Graphs)** *An undirected graph $G=(V,E)$ is called an $(n,d,\tau)$-expander if $|V|=n$, the degree of each node is $d$, and its edge expansion $h(G)=\min_{|S|<n/2}(|E(S,S^c)|)/|S|$ is at least $\tau$. By Cheeger's inequality Alon and Milman (1985), $h(G)\geq(d-\lambda)/2$, where $\lambda$ is the second largest eigenvalue, in absolute value, of the adjacency matrix of $G$. For every $d=p+1\geq 14$, where $p$ is a prime congruent to 1 modulo 4, there are explicit constructions of $(n,d,\tau)$-expanders with $\tau>d/5$ for infinitely many $n$. This is due to the fact that these graphs exhibit $\lambda\leq 2\sqrt{d-1}$ (see Lubotzky et al. (1988)), and hence by the above $h(G)\geq(d-$*

*$2\sqrt{d-1})/2 > d/5$ (say) for $d \geq 14$. Expander graphs will play a central role in the construction of our hardness result in section 4.*

The first step is to reduce $\mathrm{MAX\text{-}3SAT}$ to $\mathrm{SYM}$ with the additional property that each variable appears in a constant number of clauses. We denote by $\mathrm{MAX\text{-}3SAT}(t)$ the class of $\mathrm{MAX\text{-}3SAT}$ formulas where each variable appears in at most $t$ clauses. Theorem 9 is the starting point of our reduction. It asserts that $\mathrm{MAX\text{-}3SAT}(13)$ is hard to approximate.

**Theorem 9 (Arora (1994), Hardness of approximating $\mathrm{MAX\text{-}3SAT}(13)$)** *Let $\varphi$ be a 3-CNF boolean formula on $n$ variables and $m$ clauses, where no variable appears in more than 13 clauses. Then there exists a constant $\gamma > 0$ such that it is NP- hard to distinguish between the following cases:*

1. *$\varphi$ is satisfiable.*
2. *No assignment satisfies more than a $(1-\gamma)$-fraction of the clauses of $\varphi$.*

### 4.1. Reduction from $\mathrm{MAX\text{-}3SAT}(13)$ to $\mathrm{SYM}(30)$

The main idea behind the reduction is to add a new global variable to each $\mathrm{MAX\text{-}3SAT}(13)$ clause which will determine whether the assignment should be negated or not, and then to add all negations of clauses. The resulting formula is clearly a $\mathrm{SYM}$ formula. However, such a global variable will appear in too many clauses. We thus "break" it into many local variables (one per clause), and impose equality constraints between them. To achieve that the number of clauses remains linear in the number of variables, we only impose equality constraints based on the edges of a constant degree expander graph. The strong connectivity property of expanders ensures that a maximally satisfying assignment to such a formula would assign the same value to all these local variables, achieving the same effect of one global variable.

We now show how to reduce $\mathrm{MAX\text{-}3SAT}$ to $\mathrm{SYM}$, while maintaining the property that each variable occurs in at most a constant number of clauses.

**Theorem 10** *It is NP-hard to distinguish whether a $\mathrm{SYM}(30)$ formula can be satisfied, or whether all assignments satisfy at most $1-\delta$ fraction of the clauses, where $\delta = \gamma/16$ and $\gamma$ is the constant in Theorem 9.*

**Remark 11** *We note that Theorem 10 is qualitatively implied by a more general result by Jonsson et al. (2009), who use expanders together with more powerful algebraic techniques to show that for any natural [6] constraint satisfaction problem, it is NP hard to distinguish between the case that all constraints are satisfiable, or only $1-\varepsilon'$ fraction of them are satisfiable, even when any variable appears only in constantly many constraints (See Theorem 3.6 in Jonsson et al. (2009)). Here we provide an elementary proof of this result for the special case of symmetric CNF formulas, which is shorter, uses only combinatorial arguments (and not algebraic) and produces a better gap.*

**Proof** (of Theorem 10) We describe a gap-preserving reduction from $\mathrm{MAX\text{-}3SAT}(13)$ to $\mathrm{SYM}(30)$. Given an instance of $\mathrm{MAX\text{-}3SAT}(13)$ $\varphi$ with $n$ variables $y_1, \ldots, y_n$ and $m$ clauses, construct a $\mathrm{SYM}$ formula $\psi$ as follows: each clause $C_i \in \varphi$ is mapped to a pair of clauses $A_i = (C_i \vee \neg z_i)$

[6] Any CSP which is NP-hard under the Algebraic Dichotomy Conjecture, see Jonsson et al. (2009) for details.

and $A'_i = (C'_i \vee z_i)$ where $C'_i$ is the same as $C_i$ with all literals negated and $z_i$ is a new variable associated only with the $i$-th clause. For example:

$$(y_1 \vee \neg y_2 \vee y_3) \longrightarrow (y_1 \vee \neg y_2 \vee y_3 \vee \neg z_i) \wedge (\neg y_1 \vee y_2 \vee \neg y_3 \vee z_i).$$

We denote the resulting set of clauses by $\mathcal{A}$. We also add a set of "equality constraints", denoted $\mathcal{B}$, between the variables $z_i$ and $z_j$ as follows. Let $G$ be an $(m, d, \tau)$ explicit expander with $d = 14$ and $\tau \geq d/5$ (the existence of such constructions is established in definition 8). For each edge $(i, j)$ of the expander $\mathcal{B}$ includes two clauses: $(z_i \vee \neg z_j)$ and $(\neg z_i \vee z_j)$. Let $\psi$ denote the conjunction of the clauses in $\mathcal{A}$ and $\mathcal{B}$.

We first note that the above reduction is polynomial time computable; that $\psi$ contains $M = (d+2)m = 16m$ clauses; and that every variable of $\psi$ appears in at most $t := max\{26, 2d+2\} = 30$ clauses. Therefore, $\psi$ is indeed an instance of SYM(30). To prove the theorem we must show:

- Completeness: If $\varphi$ is satisfiable then so is $\psi$.
- Soundness: If an assignment satisfies $1 - \delta$ fraction of $\psi$'s clauses then there is an assignment that satisfies $1 - \gamma$ of $\varphi$'s clauses.

The completeness is straight-forward: given an assignment $y_1, \ldots, y_n$ that satisfies $\varphi$, we can simply set $z_1, \ldots, z_m$ to $true$ to satisfy $\psi$. For the soundness, suppose that there exists an assignment which satisfies $1-\delta$ fraction of $\psi$'s clauses, and let $v = y_1, \ldots, y_n, z_1, \ldots, z_m$ be a maximally satisfying assignment.[7] Clearly, $v$ satisfies at least $1 - \delta$ fraction of $\psi$'s clauses. We can assume that at least half of $z_1, \ldots, z_m$ are set to $true$ since otherwise we can negate the solution while maintaining the number of satisfied clauses.

We first claim that, in fact, *all* the $z_i$'s must be set to true in $v$. Indeed, let $S = \{i : z_i = false\}$ and denote $k := |S|$ (recall that $k \leq m/2$). Suppose $k > 0$ and let $G$ be the expander graph used in the reduction. If we change the assignment of all the variables in $S$ to $true$, we violate at most $k$ clauses from $\mathcal{A}$ (as each variable $z_i$ appears in exactly 2 clauses, but one of them is always satisfied). On the other hand, by definition of $G$, the edge boundary of the set $S$ in $G$ is at least $\tau k = kd/5$, and every such edge corresponds to a previously violated clause from $\mathcal{B}$. Therefore, flipping the assignment of the variables in $S$ contributes at least $kd/5 - k = \frac{14}{5}k - k > k$ to the number of satisfied clauses, contradicting the maximality of $v$. Now, since all the $z'_is$ are set to true, a clause $C_i \in \varphi$ is satisfied iff the clause $A_i \in \psi$ is satisfied. As the number of unsatisfied clauses among $A_1, \ldots, A_m$ is at most $\delta M = \delta(d+2)m$ we get that the number of unsatisfied clauses in $\varphi$ is at most $\delta(d+2)m = \frac{\gamma}{16} \cdot 16m = \gamma m$, as required. ■

### 4.2. Reduction from SYM to FHP

We proceed by describing a gap preserving reduction from SYM($t$) to FHP.

**Theorem 12** *Given $\{x^{(i)}\}_{i=1}^n \in \mathbb{R}^d$, it is NP-hard to distinguish whether the furthest hyperplane has margin $\frac{1}{\sqrt{d}}$ from all points or at most a margin of $(1 - \varepsilon)\frac{1}{\sqrt{d}}$ for $\varepsilon = \Omega(\delta)$, where $\delta$ is the constant in Theorem 10.*

[7] An assignment which satisfies the maximum possible number of clauses from $\psi$.

**Remark 13** *For convenience and ease of notation we use vectors whose norm is more than* $1$ *but at most* $\sqrt{12}$*. The reader should keep in mind that the entire construction should be shrunk by this factor to facilitate* $\|x^{(i)}\|_2 \le 1$*. Note that the construction constitutes hardness even for the special case where* $n = O(d)$ *and for all points* $1/\sqrt{12} \le \|x^{(i)}\|_2 \le 1$*.*

**Proof** Let $\psi$ be a $\mathrm{SYM}(t)$ formula with $d$ variables $y_1, ..., y_d$ and $m$ clauses $C_1, \ldots, C_m$. We map each clause $C_i$ to a point $x^{(i)}$ in $\mathbb{R}^d$. Consider first clauses with two variables of the form $(y_{j_1} \vee y_{j_2})$ with $j_1 < j_2$. Let $s_{j_1}, s_{j_2} \in \{-1, 1\}$ denote whether the variables are negated in the clause, where $1$ means not negated and $-1$ means negated. Then define the point $x^{(i)}$ as follows: $x^{(i)}_{j_1} = s_{j_1}$; $x^{(i)}_{j_2} = -s_{j_2}$; and $x^{(i)}_j = 0$ for $j \notin \{j_1, j_2\}$. For example:

$$(y_2 \vee y_3) \longrightarrow (0, 1, -1, 0, \ldots, 0).$$

For clauses with four variables $y_{j_1}, \ldots, y_{j_4}$ with $j_1 < \ldots < j_4$ let $s_{j_1}, \ldots, s_{j_4} \in \{-1, 1\}$ denote whether each variable is negated. Define the point $x^{(i)}$ as follows: $x^{(i)}_{j_1} = 3s_{j_1}$; $x^{(i)}_{j_r} = -s_{j_r}$ for $r = 2, 3, 4$; and $x^{(i)}_j = 0$ for $j \notin \{j_1, \ldots, j_4\}$. For example:

$$(\neg y_1 \vee y_3 \vee y_4 \vee \neg y_6) \longrightarrow (-3, 0, -1, -1, 0, 1, 0, \ldots, 0).$$

Finally, we also add the $d$ unit vectors $e_1, \ldots, e_d$ to the set of points (the importance of these "artificially" added points will become clear later). We thus have a set of $n = m + d$ points. To constitute the correctness of the reduction we must argue the following:

- Completeness: If $\psi$ is satisfiable there exists a unit vector $w$ whose margin is at least $1/\sqrt{d}$.
- Soundness: If there exists a unit vector $w$ whose margin is at least $(1 - \varepsilon)/\sqrt{d}$ then there exists an assignment to variables which satisfies $1 - \delta$ fraction of $\psi$'s clauses.

We first show completeness. let $y_1, \ldots, y_d$ be an assignment that satisfies $\psi$. Define $w_i = 1/\sqrt{d}$ if $y_i$ is set to $true$, and $w_i = -1/\sqrt{d}$ if $y_i$ is set to $false$. This satisfies $\|w\|_2 = 1$. Since the coordinates of all points $x^{(1)}, \ldots, x^{(n)}$ are integers, to show that the margin of $w$ is at least $1/\sqrt{d}$ it suffices to show that $\langle w, x^{(i)} \rangle \neq 0$ for all points. This is definitely true for the unit vectors $e_1, \ldots, e_d$. Consider now a point $x^{(i)}$ which corresponds to a clause $C_i$. We claim that if $\langle w, x^{(i)} \rangle = 0$ then $y$ cannot satisfy both $C_i$ and its negation $C'_i$, which also appears in $\psi$ since it is a symmetric formula. If $C_i$ has two variables, say $C_i = (y_1 \vee y_2)$, then $x^{(i)} = (1, -1, 0, \ldots, 0)$, and so if $\langle w, x^{(i)} \rangle = 0$ we must have $w_1 = w_2$ and hence $y_1 = y_2$. This does not satisfy either $C_i = y_1 \vee y_2$ or $C'_i = \neg y_1 \vee \neg y_2$. If $C_i$ has four variables, say $C_i = y_1 \vee y_2 \vee y_3 \vee y_4$, then $x^{(i)} = (3, -1, -1, -1, 0, \ldots, 0)$, and so if $\langle w, x^{(i)} \rangle = 0$ then either $w = (1/\sqrt{d})(1, 1, 1, 1, \ldots)$ or $w = (1/\sqrt{d})(-1, -1, -1, -1, \ldots)$. That is, $y_1 = y_2 = y_3 = y_4$, which does not satisfy either $C_i$ or $C'_i$. The same applies if some variables are negated.

We now turn to prove soundness. Assume there exists a unit vector $w \in \mathbb{R}^d$ such that $| \langle w, x^{(i)} \rangle | \geq (1 - \varepsilon)\frac{1}{\sqrt{d}}$. Define an assignment $y_1, \ldots, y_d$ as follows: if $w_i \geq 0$ set $y_i = true$, otherwise set $y_i = false$. If we had that all $|w_i| \approx 1/\sqrt{d}$ then this assignment would have satisfied all clauses of $\psi$. This does not have to be the case, but we will show that it is so for most $w_i$. Call $w_i$ whose absolute value is close to $1/\sqrt{d}$ "good", and ones which deviate from $1/\sqrt{d}$ "bad". We will show that each clause which contains only good variables must be satisfied. Since each bad variable appears only in a constant number of clauses, showing that there are not many bad variables would imply that most clauses of $\psi$ are satisfied.

**Claim 14** *Let $B = \{i : |w_i - 1/\sqrt{d}| \geq 0.1/\sqrt{d}\}$ be the set of "bad" variables. Then $|B| \leq 10\varepsilon d$.*

**Proof** For all $i$ we have $|w_i| \geq (1-\varepsilon)/\sqrt{d}$ since the unit vectors $e_1, \ldots, e_d$ are included in the point set. Thus if $i \in B$ then $|w_i| \geq 1.1/\sqrt{d}$. Since $w$ is a unit vector we have

$$1 = \sum w_i^2 = \sum_{i \in B} w_i^2 + \sum_{i \notin B} w_i^2 \geq |B|\frac{1.1^2}{d} + (d - |B|)\frac{(1-\varepsilon)^2}{d},$$

which after rearranging gives $|B| \leq d\frac{1-(1-\varepsilon)^2}{1.1^2-(1-\varepsilon)^2} \leq 10\varepsilon d$. ■

**Claim 15** *Let $C_i$ be a clause which does not contain any variable from $B$. Then the assignment $y_1, \ldots, y_d$ satisfies $C$.*

**Proof** Assume by contradiction that $C_i$ is not satisfied. Let $x^{(i)}$ be the point corresponding to $C_i$. We show that $\langle w, x^{(i)} \rangle < (1-\varepsilon)/\sqrt{d}$, contradicting our assumption on $w$.

Consider first the case that $C_i$ contains two variables, say $C_i = (y_1 \vee y_2)$, which gives $x^{(i)} = (1, -1, 0, \ldots, 0)$. Since $C_i$ is not satisfied we have $y_1 = y_2 = false$, hence $w_1, w_2 \in (-1/\sqrt{d} \pm \eta)$ where $\eta < 0.1/\sqrt{d}$ which implies that $| \langle w, x^{(i)} \rangle | \leq 0.2/\sqrt{d} < (1-\varepsilon)/\sqrt{d}$. Similarly, suppose $C_i$ contains four variables, say $C_i = (y_1 \vee y_2 \vee y_3 \vee y_4)$, which gives $x^{(i)} = (3, -1, -1, -1, 0, \ldots, 0)$. Since $C_i$ is not satisfied we have $y_1 = y_2 = y_3 = y_4 = false$, hence $w_1, w_2, w_3, w_4 \in (-1/\sqrt{d} \pm \eta)$ where $\eta < 0.1/\sqrt{d}$ which implies that $| \langle w, x^{(i)} \rangle | \leq 0.6/\sqrt{d} < (1-\varepsilon)/\sqrt{d}$. The other cases where some variables are negated are proved in the same manner. ■

We now conclude the proof of Theorem 12. We have $|B| \leq 10\varepsilon d$. Since any variable occurs in at most $t$ clauses, there are at most $10\varepsilon dt$ clauses containing a "bad" variable. As all other clauses are satisfied, the fraction of clauses that the assignment to $y_1, \ldots, y_d$ does not satisfy is at most $10\varepsilon dt/m \leq 10\varepsilon t < \delta$ for $\varepsilon = 0.1(\delta/t) = \Omega(\delta)$ since $t = 30$ in Theorem 10. ■

## 5. Discussion

A question which is not resolved in this paper is whether there exists an efficient constant factor approximation algorithm for the margin of FHP but for all points in the input. The authors have considered several techniques to try to rule out an $O(1)$ approximation for the problem. For example, trying to amplify the gap of the reduction in section 4. This, however, did not succeed. Even so, the resemblance of FHP to some hard algebraic problems admitting no constant factor approximation leads the authors to believe that the problem is indeed inapproximable to within a constant factor.

### Acknowledgments

We thank Koby Crammer and Nir Ailon for very helpful discussions and references, as well as several anonymous reviewers.

## References

Noga Alon and V. D. Milman. lambda1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory*, 38:73–88, 1985.

Sanjeev Arora. Probabilistic checking of proofs and hardness of approximation problems. *Revised version of a dissertation submitted at CS Division, U C Berkeley*, CS-TR-476-94, August 1994.

Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *IEEE Symposium on Foundations of Computer Science*, pages 616–623, 1999.

Kristin P. Bennett and Ayhan Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems*, pages 368–374. MIT Press, 1998.

Tijl De Bie and Nello Cristianini. Convex methods for transduction. In *Advances in Neural Information Processing Systems 16*, pages 73–80. MIT Press, 2003.

Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

Maria florina Balcan, Avrim Blum, and Santosh Vempala. On kernels, margins, and low-dimensional mappings. In *Algorithmic Learning Theory*, pages 194–205, 2004.

Sariel Har-peled, Dan Roth, and Dav Zimak. Maximum margin coresets for active and noise tolerant learning. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI*, 2006.

Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-612-2. URL http://portal.acm.org/citation.cfm?id=645528.657646.

Peter Jonsson, Andrei A. Krokhin, and Fredrik Kuivinen. Hard constraint satisfaction problems have hard gaps at location 1. *Theor. Comput. Sci.*, 410(38-40):3856–3874, 2009.

Adam R. Klivans and Rocco A. Servedio. Learning intersections of halfspaces with a margin. In *in proceedings of the 17th annual conference on learning theory*, pages 348–362, 2004.

M. K. Kozlov, S. P. Tarasov, and L. G. Khachiyan. Polynomial solvability of convex quadratic programming. *Doklady Akademiia Nauk SSSR*, 248, 1979.

Rafal Latala. Some estimates of norms of random matrices. *Proceedings of the American Mathematical Society*, 133(5):1273–1282, May 2005.

G.G. Lorentz, M. Golitschek, and Y. Makovoz. *Constructive approximation: advanced problems*. Grundlehren der mathematischen Wissenschaften. Springer, 1996. ISBN 9783540570288. URL http://books.google.com/books?id=pl0_AQAAIAAJ.

Alexander Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

O. L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Oper. Res.*, 13:444–452, 1965.

Jiming Peng, Jiming Peng, Lopamudra Mukherjee, Vikas Singh, Dale Schuurmans, and Linli Xu. An efficient algorithm for maximal margin clustering. *To appear in Journal of Global Optimization*, 2011.

N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1): 145 – 147, 1972. ISSN 0097-3165. doi: DOI:10.1016/0097-3165(72)90019-2. URL http://www.sciencedirect.com/science/article/pii/0097316572900192.

V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Autom. Remote Control*, 24:774–780, 1963.

Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans. Maximum margin clustering. In *Advances in Neural Information Processing Systems 17*, pages 1537–1544. MIT Press, 2005.

## Appendix A. Proof of Lemma 3

**Lemma* 1** *Let $w^*$ and $y^*$ denote the optimal solution of margin of $\theta$ and the labeling it induces. Let $y$ be the labeling induced by a random hyperplane $w$. The probability that $y = y*$ is at least $n^{-O(\theta^{-2})}$.*

**Proof** Let $c_1, c_2$ be some sufficiently large constants whose exact values will be determined later. For technical reasons, assume w.l.o.g. that[8] $d > c_1 \log(n)\theta^{-2}$. Denote by $E$ the event that

$$\langle w^*, w\rangle > \sqrt{c_2 \log(n)}\theta^{-1} \cdot \sqrt{\frac{1}{d}}.$$

The following lemma gives an estimate for the probability of $E$. Although its proof is quite standard, we give it for completeness.

**Lemma 16** *Let $w$ be a uniformly random unit vector in $\mathbb{R}^d$. There exists some universal constant $c_3$ such that for any $1 \le h \le c_3\sqrt{d}$ and any fixed unit vector $w^*$ it holds that*

$$\Pr[\langle w, w^*\rangle > h/\sqrt{d}] = 2^{-\Theta(h^2)}.$$

*As an immediate corollary we get that by setting appropriate values for $c_1, c_2, c_3$ we guarantee that $\Pr[E] \ge n^{-O(\theta^{-2})}$.*

**Proof** Notice that $\Pr[\langle w, w^*\rangle > h/\sqrt{d}]$ is exactly the ratio between the surface area of a spherical cap defined by the direction $w^*$ and height (i.e., distance from the origin) $h/\sqrt{d}$ and the surface area of the entire spherical cap. To estimate the probability we give a lower bound for the mentioned ratio.

Define $S_d, C_{d,h}$ as the surface areas of the $d$ dimensional unit sphere and $d$ dimensional spherical cap of hight $h/\sqrt{d}$ correspondingly. Denote by $S_{d-1,r}$ be the surface area of a $d-1$ dimensional sphere with radius $r$. Then,

$$C_{d,h}/S_d = \int_{H=h/\sqrt{d}}^{1} \frac{S_{d-1,\sqrt{1-H^2}}}{S_d} dH$$

We compute the ratio $\frac{S_{d-1,\sqrt{1-H^2}}}{S_d}$ with the well know formula for the surface area of a sphere of radius $r$ and dimension $d$ of $2\pi^{d/2}r^{d-1}/\Gamma(d/2)$ where $\Gamma$ is the Gamma function, for which $\Gamma(d/2) = (\frac{d-2}{2})!$ when $d$ is even and $\Gamma(d/2) = \frac{(d-2)(d-4)\cdots 1}{2^{(d-1)/2}}$ when $d$ is odd. We get that for any $H < 1/2$,

$$\frac{S_{d-1,\sqrt{1-H^2}}}{S_d} = \Omega(\sqrt{d} \cdot (1-H^2)^{(d-2)/2}) = \Omega(\sqrt{d} \cdot e^{-dH^2/2})$$

and that for any $H < 1$,

$$\frac{S_{d-1,\sqrt{1-H^2}}}{S_d} = O(\sqrt{d} \cdot (1-H^2)^{(d-2)/2}) = O(\sqrt{d} \cdot e^{-dH^2/2}).$$

The lower bound is given in the following equation.

$$\Pr\left[\langle w, w^*\rangle > h/\sqrt{d}\right] = C_{d,h}/S_d = \int_{H=h/\sqrt{d}}^{1} \frac{S_{d-1,\sqrt{1-H^2}}}{S_d} dH \ge \int_{H=h/\sqrt{d}}^{2h/\sqrt{d}} \frac{S_{d-1,\sqrt{1-H^2}}}{S_d} dH \overset{(*)}{=}$$

[8] If that is not the case to begin with, we can simply embed the vectors in a space of higher dimension.

$$\Omega\left(\int_{H=h/\sqrt{d}}^{2h/\sqrt{d}} \sqrt{d}\cdot e^{-dH^2/2}dH\right) = \Omega\left(\int_{h'=h}^{2h} e^{-h'^2/2}dh'\right) = \Omega\left(h\cdot e^{-2h^2}\right) = e^{-O(h^2)}$$

Equation $(*)$ holds since $2h/\sqrt{d} < 1/2$. The upper bound is due to the following.

$$\Pr\left[\langle w, w^*\rangle > h/\sqrt{d}\right] = \int_{H=h/\sqrt{d}}^{1} \frac{S_{d-1,\sqrt{1-H^2}}}{S_d}dH = O\left(\int_{H=h/\sqrt{d}}^{1} \sqrt{d}\cdot e^{-dH^2/2}dH\right) =$$

$$O\left(\int_{h'=h}^{\infty} e^{-h'^2/2}dh'\right) \overset{(**)}{=} O\left(\int_{h'=h}^{\infty} e^{-h^2/2-hh'}dh'\right) = e^{-\Omega(h^2)}$$

In equation $(**)$ we used the fact that $h^2/2 + hh' \leq h'^2/2$ for all $h' \geq h$. The last equation holds since $h \geq 1$. ■

We continue with the proof of Lemma 3. We now analyze the success probability given the event $E$ has occurred. For the analysis, we rotate the vector space so that $w^* = (1, 0, 0, \ldots, 0)$. A vector $x$ can now be viewed as $x = (x_1, \tilde{x})$ where $x_1 = \langle w^*, x\rangle$ and $\tilde{x}$ is the $d-1$ dimensional vector corresponding to the projection of $x$ onto the hyperplane orthogonal to $w^*$. Since $w$ is chosen as a random unit vector, we know that given the mentioned event $E$, it can be viewed as $w = (w_1, \tilde{w})$ where $\tilde{w}$ is a uniformly chosen vector from the $d-1$ dimensional sphere of radius $\sqrt{1-w_1^2}$ and $w_1 \geq \sqrt{c\log(n)}\theta^{-1}\cdot\sqrt{\frac{1}{d}}$.

Consider a vector $x \in \mathbb{R}^d$ where $\|x\| \leq 1$ such that $\langle w^*, x\rangle \geq \theta$. As before we write $x = (x_1, \tilde{x})$ where $\|\tilde{x}\| \leq \sqrt{1-x_1^2}$. Then

$$\langle x, w\rangle = x_1 w_1 + \langle \tilde{x}, \tilde{w}\rangle \geq \sqrt{\frac{c\log n}{d}} + \langle \tilde{x}, \tilde{w}\rangle$$

Notice that both $\tilde{x}, \tilde{w}$ are vectors whose norms are at most 1 and the direction of $\tilde{w}$ is chosen uniformly at random, and is independent of $E$. Hence, according to Lemma 16,

$$\Pr_w\left[|\langle \tilde{x}, \tilde{w}\rangle| \geq \sqrt{c\log n}/\sqrt{d}\right] \leq n^{-\Omega(c)}.$$

It follows that the sign of $\langle w, x\rangle$ is positive with probability $1 - n^{-\Omega(c)}$. By symmetry we get an analogous result for a vector $x$ s.t. $\langle w^*, x\rangle \leq -\theta$. By union bound we get that for sufficiently large $c$, with probability $1/2$ we get that for all $i \in [n]$, $sign\langle w, x^{(i)}\rangle = sign\langle w^*, x^{(i)}\rangle$ (given the event $E$ has occurred) as required. To conclude

$$\Pr_{w\in\mathbb{S}^{d-1}}[y = y^*] \geq \Pr_{w\in\mathbb{S}^{d-1}}[E]\cdot \Pr_{w\in\mathbb{S}^{d-1}}[y = y^*|E] \geq n^{-O(\theta^{-2})}.$$

■

## Appendix B. A note on average case complexity of FHP

Given the hardness results above, a natural question is whether random instances of FHP are easier to solve. As our algorithmic results suggest, the answer to this question highly depends on the maximal separation margin of such instances. We consider a natural model in which the points $\{x^{(i)}\}_{i=1}^n$ are drawn isotropically and independently at random close to the unit sphere $S^{d-1}$. More formally, each coordinate of each point is drawn independently at random from a Normal distribution with standard deviation $1/\sqrt{d}$: $x_j^{(i)} \sim \mathcal{N}(0, 1/\sqrt{d})$.

Let us denote by $\theta_{rand}$ the maximal separation margin of the set of points $\{x^{(i)}\}_{i=1}^n$. While computing the exact value of $\theta_{rand}$ is beyond the reach of this paper [9] , we prove the following simple bounds on it:

**Theorem 17** *With probability at least* $2/3$

$$\Omega\Big(\frac{1}{n\sqrt{d}}\Big) \quad = \quad \theta_{rand} \quad = \quad O\Big(\frac{1}{\sqrt{d}}\Big).$$

**Proof** For the upper bound, let $w$ be the normal vector of the furthest hyperplane achieving margin $\theta_{rand}$, and let $y_i \in \{\pm 1\}$ be the sides of the hyperplane to which the points $x^{(i)}$ belong, i.e, for all $1 \le i \le n$ we have $y_i \left\langle w, x^{(i)} \right\rangle \ge \theta_{rand}$. Summing both sides over all $i$ and using linearity of inner products we get

$$\left\langle w, \sum_{i=1}^{n} y_i \cdot x^{(i)} \right\rangle \ge \theta_{rand} \cdot n \tag{2}$$

By Cauchy-Schwartz and the fact that $\|w\| = 1$ we have that the LHS of (2) is at most $\| \sum_{i=1}^n y_i \cdot x^{(i)} \| = \|Xy\|$. Here $X$ denotes the $d \times n$ matrix whose $i$'th column is $x^{(i)}$, and by $y$ the $\{\pm 1\}^n$ vector whose $i$'th entry is $y_i$.

$$\theta_{rand} \cdot n \le \|Xy\| \le \|y\| \cdot \|X\| \le \sqrt{n} \cdot O\Big(\frac{\sqrt{n} + \sqrt{d}}{\sqrt{d}}\Big) = O\Big(\frac{n}{\sqrt{d}}\Big) \tag{3}$$

where the second inequality follows again from Cauchy-Schwartz, and the third inequality follows from the facts that the spectral norm of a $d \times n$ matrix whose entries are $\mathcal{N}(0, 1)$ distributed is $O(\sqrt{n} + \sqrt{d})$ w.h.p. (see Latala (2005)) and the fact that $\|y\| = \sqrt{n}$. Rearranging (3) yields the desired upper bound.

For the lower bound, consider a random hyperplane defined by the normal vector $w'/||w'||$ where the entries of $w'$ distribute i.i.d. $\frac{1}{\sqrt{d}}\mathcal{N}(0, 1)$. From the rotational invariance of the Gaussian distribution we have that $\left\langle w', x^{(i)} \right\rangle$ also distributes $\frac{1}{\sqrt{d}}\mathcal{N}(0, 1)$. Using the fact that w.h.p $||w'|| \le 2$ we have for any $c > 1$:

$$\Pr\Big[|\left\langle w, x^{(i)} \right\rangle| \le \frac{1}{c \cdot n\sqrt{d}}\Big] \le \Pr\Big[|\left\langle w', x^{(i)} \right\rangle| \le \frac{2}{c \cdot n\sqrt{d}}\Big] = \Pr_{Z \sim \mathcal{N}(0,1)}\Big[|Z| \le \frac{2}{c \cdot n}\Big] = O\Big(\frac{1}{c \cdot n}\Big). \tag{4}$$

For a sufficiently large constant $c$, a simple union bound implies that the probability that there exists a point $x^{(i)}$ which is closer than $1/(c \cdot n\sqrt{d})$ to the hyperplane defined by $w$ is at most $1/3$. Note that the analysis of the lower bound does not change even if the points are arbitrarily spread on the unit sphere (since the normal distribution is spherically symmetric). Therefore, choosing a random hyperplane also provides a trivial $O(n\sqrt{d})$ worst case approximation for FHP. ■

[9] The underlying probabilistic question to be answered is: what is the probability that $n$ random points on $\mathbb{S}^{d-1}$ all fall into a cone of measure $\theta$ ?