Random Separating Hyperplane Theorem and Learning Polytopes

Chiranjib Bhattacharyya
Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India

Ravindran Kannan
Department of Operations Research, Carnegie Mellon University, Pittsburgh, USA

Amit Kumar
Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India

Abstract

The Separating Hyperplane theorem is a fundamental result in Convex Geometry with myriad applications. The theorem asserts that for a point a not in a closed convex set K, there is a hyperplane with K on one side and a strictly on the other side. Our first result, Random Separating Hyperplane Theorem (RSH), is a strengthening of this for polytopes. RSH asserts that if the distance between a and a polytope K with k vertices and unit diameter in $\Re^d$ is at least δ, where δ is a fixed constant in $(0,1)$, then a randomly chosen hyperplane separates a and K with probability at least $1/\operatorname{poly}(k)$ and margin at least $\Omega\left(\delta/\sqrt{d}\right)$.

RSH has algorithmic applications in learning polytopes. We consider a fundamental problem, denoted the "Hausdorff problem", of learning a unit diameter polytope K within Hausdorff distance δ, given an optimization oracle for K. Using RSH, we show that with polynomially many random queries to the optimization oracle, K can be approximated within error $O(\delta)$. To our knowledge, this is the first provable algorithm for the Hausdorff Problem in this setting. Building on this result, we show that if the vertices of K are well-separated, then an optimization oracle can be used to generate a list of points, each within distance $O(\delta)$ of K, with the property that the list contains a point close to each vertex of K. Further, we show how to prune this list to generate a (unique) approximation to each vertex of the polytope. We prove that in many latent variable settings, e.g., topic modeling, LDA, optimization oracles do exist provided we project to a suitable SVD subspace. Thus, our work yields the first efficient algorithm for finding approximations to the vertices of the latent polytope under the well-separatedness assumption. This assumption states that each vertex of K is far from the convex hull of the remaining vertices of K, and is much weaker than other assumptions behind algorithms in the literature which find vertices of the latent polytope.

2012 ACM Subject Classification Theory of computation → Unsupervised learning and clustering

Keywords and phrases Separating Hyperplane Theorem, Learning Polytopes, Optimization Oracles

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.25

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://doi.org/10.48550/arXiv.2307.11371` [9]

1 Introduction

The Separating Hyperplane theorem is a fundamental result in Convex Geometry with myriad applications (see e.g. [6]). The theorem asserts that for a point a not in a closed convex set K, there is a hyperplane with K on one side and a strictly on the other side.

This paper makes two main contributions. Our theoretical contribution, which is an extension of the classical Separating Hyperplane Theorem, is what we call the Random Separating Hyperplane Theorem (RSH). Our main algorithmic contribution is to use RSH to prove that a natural algorithm, which we call the k-OLP algorithm, can learn (vertices of) latent polytopes arising in a number of problems in Latent Variable Models including

Clustering, Mixture Learning , LDA (linear discriminant analysis), Topic Models. The algorithmic result is shown by reducing the problem of learning latent polytopes in a variety of settings to that of constructing approximate optimization oracles for the corresponding polytopes. Bulk of our algorithmic contribution is in proving this reduction and the existence of such oracles.

1.1 Random Separating Hyperplane Theorem (RSH)

RSH draws its motivation mainly from the Separating Hyperplane Theorem of Convex Geometry. It also has connections to the Johnson-Lindenstrauss Random Projection theorem [15]. The Separating Hyperplane Theorem formally states that given a closed convex set K and a point $a \notin K$, there exists a vector u such that

$$u \cdot a \; > \; \mathrm{Max}_{y \in K}\, u \cdot y.$$

The following question arises: Does a randomly picked u separate a from K? Taking into account some necessary conditions for a positive answer, we can ask if the following inequality holds for a randomly chosen u: (here Δ is the diameter of K, a is at distance at least $\delta\Delta$ from K, where $\delta \in (0,1)$):

$$\mathbf{Pr}\left(u \cdot a \geq \mathrm{Max}_{y \in K} u \cdot y + |u|\alpha\delta\Delta\right) \; \geq \; 1/\mathrm{poly}_{\delta},^{1} \tag{1}$$

with α being as high as possible.

The question (1) is also motivated from the Johnson-Lindenstrauss Random Projection theorem (JL theorem) [15] which states that if a, b are points in $\mathbf{R}^d$ and U is a random subspace of dimension s, then with probability bounded away from 0, the distance between the projection of a and b on U is at least $\Omega(|a-b|\sqrt{s}/\sqrt{d})$. The following natural generalization of this is interesting already for $s = 1$:

Instead of b being a point, if it is now a polytope K, does a similar lower bound on the distance of a to K in the projection onto a random line hold?

It is easy to see that in spirit, this is the same question as whether (1) holds. It is also easy (see below) to see that the projection shrinks the distance between a and K by a factor of $\Omega^*(\sqrt{d})$. The RSH theorem proves that the shrinkage is $O(\sqrt{d})$ thus making this parameter nearly (within log factors) tight. We now state the RSH theorem:

▶ **Theorem 1** (Random Separating Hyperplane Theorem(RSH): Informal version)**.** *Suppose K is a k vertex polytope with diameter $\Delta(K)$ and a is a point at distance at least $\delta\Delta(K)$, $\delta \in (0,1)$, from K. Let V be an $m-$dimensional subspace containing $K \cup \{a\}$. For a random Gaussian vector $u \in V$, the following event happens with probability at least $1/\mathrm{poly}_\delta(k)$:*

$$u \cdot a \geq Max_{y \in K} u \cdot y + \frac{\delta\Delta(K)|u|}{10\sqrt{m}}. \tag{2}$$

We provide a simple example where K is a line segment (see Appendix A) to show that the factor $\sqrt{m}$ cannot be improved. It is also interesting to note that the success probability of the event in (2) needs to depend on k. (see Appendix A). In particular, RSH does not hold for general convex sets (where k is not necessarily finite.)

---

[1] $\mathrm{poly}_\delta(z)$ denotes $z^{\mathrm{poly}(1/\delta)}$.

1.2 Algorithmic application of RSH

We now discuss the second main contribution of our work, i.e., applications of RSH to learning vertices of a latent polytope. We begin with the definition of an approximate optimization oracle.

▶ **Definition 2.** *For a non-empty convex set $K \subseteq \Re^d$ and $\varepsilon \in (0,1)$, an Optimization oracle for K with error ε, denoted $\mathsf{OptOr}_\varepsilon(K)$ oracle, takes as input any $u \in \Re^d, |u| = 1$, and returns a point $x(u)$ satisfying* both *these conditions :*

- $x(u) \in K + \varepsilon\Delta(K)B_d$, *where B_d is the unit ball, i.e., $\{x \in \Re^d : |x| \leq 1\}$, and*
- $u \cdot x(u) \geq Max_{y \in K} u \cdot y - \varepsilon\Delta(K)$

Problem Formulation. Several latent variable problems including Clustering, LDA, MMBM can be reduced (See Section 6 for details) to a problem that we call k-OLP: given an ε-optimization oracle for a k vertex polytope $K \subseteq \mathbf{R}^d$, learn the vertices of K (approximately). We define two simpler (than k-OLP) problems – ListLearn and Hausdorff, that are related to k-OLP, and then we define k-OLP.

The first problem, Hausdorff, seeks to find an approximation to a polytope K when we are given an approximate optimization oracle for the polytope.

▶ **Definition 3** ((ε, δ)-Hausdorff-Problem)**.** *Given an $\mathsf{OptOr}_\varepsilon(K)$ oracle for a polytope K in $\mathbf{R}^d$ with k vertices, find a set P of $m = \text{poly}_\delta(dk)$ points such that $\mathsf{Haus}(\mathsf{CH}(P), K) \leq \delta\Delta(K)$, where,* Haus *denotes Hausdorff distance (see Definition 20 for a formal definition), and $\mathsf{CH}(P)$ is the convex hull of P.*

In the problem ListLearn, we also wish to find a small list of points, such that each vertex of K is close to at least one point in this list.

▶ **Definition 4** ((ε, δ)-ListLearn Problem)**.** *Given an $\mathsf{OptOr}_\varepsilon(K)$ oracle for a polytope K in $\Re^d$ with k vertices, each separated from the convex hull of the other $k-1$ vertices by at least $\delta\Delta(K)$, find a list $P \subseteq K + \delta\Delta(K)B_d$ of $m = \text{poly}_\delta(dk)$ points such that for every vertex v of K, there is some $v' \in P$ with $|v - v'| \leq \delta\Delta(K)/10$.*

When the parameters ε, δ will be clear from the context, we shall abbreviate the above two problems as Hausdorff and ListLearn problems respectively. It is not difficult to see that any solution P to ListLearn is also a solution to Hausdorff, but, the converse need not hold: CH(P) may nearly contain K without P having any point close to some vertex of K. Our technical results (see below for the informal versions) show that if $\varepsilon \in O_\delta(1/\sqrt{d})$ [2], then we can solve the above-mentioned problems efficiently and indeed then, the following simple algorithm gives the desired answers (the proof crucially uses RSH):

> **Random Probes Algorithm**
> - Pick uniformly at random unit vectors $u_1, u_2, \ldots u_m$, where $m = \text{poly}_\delta(dk)$.
> - Return P, which is the set of m answers of the $\mathsf{OptOr}_\varepsilon(K)$ oracle to the queries $u_1, u_2, \ldots u_m$.

The first result (see Theorem 21 for a formal statement) states that the convex hull of answers to polynomially many random queries to the approximate optimization oracle approximates K well.

[2] $O_\delta(x)$ stands for $f(\delta)x$ for some function f.

▶ **Theorem 5** (Hausdorff Approximation from oracle (Informal)). *Consider an instance of the* (ε, δ)-Hausdorff *problem for a polytope* $K \subseteq \Re^d$. *Assume that* $\varepsilon \in O_\delta(1/\sqrt{d})$, *and let* P *be the set of points returned by the* **Random Probes Algorithm** *above. Then with high probability,*

$$\mathsf{Haus}(CH(P), K) \leq \delta\Delta(K).$$

The second result (see Theorem 28 for a formal statement) shows that as long as each vertex of K is *well-separated* from the convex hull of the other vertices of K, the set P constructed by the **Random Probes Algorithm** contains an approximation to each of the vertices. Thus, answers to polynomially many random queries list-learns the polytope.

▶ **Theorem 6** (List-Learning from Oracle (Informal)). *Consider an instance of the* ListLearn *problem for a polytope* $K \subseteq \Re^d$, *and assume that* $\varepsilon \in O_\delta(1/\sqrt{d})$. *Then, with high probability, the set* P *output by the* **Random Probes Algorithm** *above has the following property: for every vertex* a *of* K, *there is a point* $a' \in P$ *with*

$$|a' - a| \leq \delta\Delta(K)/10.$$

The $\sqrt{d}$ factor in both the above theorems is near-optimal (within a $\log d$ factor). Indeed, we shall prove:

▶ **Theorem 7** (Oracle Lower Bound). *The problem where, one is required to output a point which is within* $\Delta(K)/10$ *of some vertex of* K, *given only by an* $\mathsf{OptOr}_\varepsilon(K)$ *oracle, cannot be solved in deterministic polynomial time when* $\varepsilon \geq 8 \ln d/\sqrt{d}$.

We now define the k-OLP problem (the parameters ε, δ in the definition will often be clear from the context and may not be mentioned explicitly). The problem statement is similar to that of ListLearn, but we want to output a list of exactly k points.

▶ **Definition 8** ((ε, δ)-k-OLP Problem). *Under the same hypothesis as for the* ListLearn *problem, find a set of points* P, $|P| = k$, *satisfying the following condition: for each vertex* v *of* K, *there is a (unique) point* $v' \in P$ *such that* $|v - v'| \leq \delta\Delta(K)/10$.

Our next result gives a strengthening of Theorem 6.

▶ **Theorem 9.** *Consider an instance of the* k-OLP *problem on a polytope* $K \subseteq \Re^d$, *and assume* $\varepsilon \in O_\delta(1/\sqrt{d})$ *in a* k-OLP *problem. Let* P *be the set of points returned by the* **Random Probes Algorithm**. *Then, in polynomial time, we can find a* $Q \subseteq P, |Q| = k$ *satisfying the following condition: for each vertex* v *of* K, *there is a (unique)* $v' \in P, |v - v'| \leq \delta\Delta(K)/10$.

The algorithm for finding Q from P is likely of independent interest. We call this problem the "Soft Convex Hull" problem and it is described in Section 6.1.

Do Approximate Optimization Oracles exist?

The answer to this question is a qualified Yes. They exist, but unfortunately, as we point out below, for many latent variable problems including the simple mixture of two Gaussians with means separated by $\Omega(1)$ standard deviations, we do not get $\varepsilon \in O_\delta(1/\sqrt{d})$, when, $k < d$. Thus, we do not satisfy the hypothesis of the results mentioned in Theorem 5,Theorem 6, and Theorem 9. But we are able to tackle this hurdle by projecting to the k-SVD subspace (of the input data points which satisfy conditions discussed below) where, we do get the necessary $\varepsilon \in O_\delta(1/\sqrt{k})$.

First we observe that approximate optimization oracles arise in a natural setting – that of latent variable models. [7] show that these models can be reduced to a geometric problem called LkP described below. [We will not reproduce the reduction here.] LkP is the following problem: Let K be a k vertex polytope in $\Re^d$. Let $M_{\cdot,1}, \ldots, M_{\cdot,k}$ denote the vertices of K. Assume that there are latent (hidden) points $P_{\cdot,j}, j = 1, 2, \ldots, n$, in K. The observed data points $A_{\cdot,j}, j = 1, 2, \ldots, n$ are generated (not necessarily under any stochastic assumptions) by adding *displacements* $A_{\cdot,j} - P_{\cdot,j}$ respectively to $P_{\cdot,j}$. Let[3]

$$\sigma_0 := \frac{||\mathbf{P} - \mathbf{A}||}{\sqrt{n}}.$$

We assume that there is a certain w_0 fraction of latent points close to every vertex of K, i.e., for all $\ell \in [k]$,

$$C_\ell := \{j : |P_{\cdot,j} - M_{\cdot,\ell}| \leq \frac{\sigma_0}{\sqrt{w_0}}\} \text{ satisfies } |C_\ell| \geq w_0 n.$$

▶ **Theorem 10** (From Data to Oracles). *Using the above notation, the following "Subset Smoothing algorithm" gives us a polynomial time* $\mathsf{OptOr}_{\frac{4\sigma_0}{\Delta\sqrt{w_0}}}(K)$ *oracle..*

> **Subset Smoothing Algorithm**
> Given query u, let S be the set of the $w_0 n$ j's with the highest $u \cdot A_{\cdot,j}$ values.
> Return $A_{\cdot,S} := \frac{1}{w_0 n} \sum_{j \in S} A_{\cdot,j}$.

The Subset Smoothing algorithm was used in [7]. It is also reminiscent of Super-quantiles [18], though our use here is not directly related to them. While this theorem helps us get optimization oracles, the error guarantee of $O(\sigma_0/\Delta\sqrt{w_0})$ is not good enough in many applications. An elementary example illustrates this issue:

Consider a mixture of two equal weight standard Gaussians centered at $-v$ and v, where, v is a vector of length 10. [This fits the paradigm "means separated by $\Omega(1)$ standard deviations".] Then, data generated by the mixture model fits our data generation process with $K = \{\lambda v, \lambda \in [-1, 1]\}$, and each $P_{\cdot,j}$ is either v or $-v$ depending on the Gaussian from which the point has been sampled. Here $A_{\cdot,j}$ denotes the actual sampled point from the mixture. Now, $\Delta = 20$ and it can be seen from Random Matrix Theorems (see e.g., [19]) that $\sigma_0 = O(1)$ with high probability. So, $\sigma_0/\Delta\sqrt{w_0} \in O(1)$ with high probability, and hence, the Theorem above yields an $\mathsf{OptOr}_\varepsilon(K)$ oracle with $\varepsilon \in \Omega(1)$. But d can be arbitrarily large and so we do not have the required $\mathsf{OptOr}_{O(1/\sqrt{d})}(K)$ oracle.

This elementary example can be tackled in several ways. Our algorithm which we call the "k-OLP algorithm" is simply stated and works in general settings (including on this toy example) for several Latent Variable problems (see Section 6 for details). The main idea is to first project the input points on a suitable SVD subspace and then use the approximate optimization oracle in the projection.

SVD and the k-OLP Algorithm

We now state the result (see Theorem 39 for a formal version) for k-OLP in the setting of LkP. As mentioned above, this uses SVD followed by subset smoothing.

[3] By the standard definition of spectral norm, it is easy to see that σ_0^2 is the maximum mean squared displacement in any direction.

▶ **Theorem 11.** *Recall the notation and assumptions of Theorem 10. In addition, we assume that each vertex of K is $\delta\Delta(K)$ far from the convex hull of other vertices of K, where, δ satisfies:*

$$\sigma_0 \leq c\delta^2\Delta\sqrt{w_0}/\sqrt{k}.$$

Then, the set of points P returned by the following k-OLP algorithm list learns the vertices of K. Further, we can find a subset Q of P with $|Q| = k$ and for each v, vertex of K, Q contains a v' with $|v - v'| \leq \delta\Delta/10$:

Algorithm k-OLP

1. Project to the k-dim SVD subspace V corresponding to the points $A_{\cdot,j}$.
2. Pick $m = \text{poly}_\delta(k)$ random vectors $u_1, u_2, \ldots, u_m$ in V.
3. For each u_i, take the mean of the $A_{\cdot,j}$ with the $w_0 n/2$ highest values of $u_i \cdot A_{\cdot,j}$.
4. Let P be the set of m means computed in the step above.
5. Output a subset Q of P, $|Q| = k$, using Theorem 9.

We sketch here the steps in the proof (the details are contained in the proof of Theorem 39.)

Sketch. Let $\widehat{K}$ denote projection of K onto V. By Theorem 10 , Step 3 of Algorithm k-OLP is an $\mathsf{OptOr}_\varepsilon(\widehat{K})$ oracle, where $\varepsilon = O\left(\frac{\sigma_0}{\Delta\sqrt{w_0}}\right)$. Also, each $\widehat{M}_{\cdot,\ell}$, which is the projection of $\widehat{M}_{\cdot,\ell}$ on V, is $O(\delta\Delta(K))$ far from the convex hull of the other vertices of $\widehat{K}$. Now, Theorem 9 applied to $\widehat{K}$ in the subspace V implies the desired result. ◀

It is worth noting that data obtained from several generative models are known to satisfy the LkP condition stated in Theorem 11, e.g., Stochastic Mixture models with k components, Topic Models, Mixed membership community models.

From List Learning to k-OLP

As outlined above, the k-OLP algorithm works in two stages: (i) Project the data points on the SVD subspace V of dimension k, and (ii) make polynomially calls to the $\mathsf{OptOr}_\varepsilon(K)$ oracle, where each query is given by a randomly chosen unit vector in the subspace V (as in the statement of Theorem 5) – let P be the set of points returned by the oracle. The first statement in Theorem 11 shows that the convex hull of P is close to K.

Obtaining approximations to the vertices of K from P requires addressing a new problem: given a set of points W, find a small subset T of W, such that their convex hulls are close. We call this the *soft convex hull* problem. A similar problem was addressed by [12]; however they gave a bi-criteria approximation algorithm for this problem. Under stronger assumptions, where we assume that there in the optimal solution $T^\star$, each point of $T^\star$ is well-separated from the convex hull of the rest of the points of $T^\star$, we show that one can recover approximations to each of the points in $T^\star$. Applying this result to the set of points P returned by the optimization oracle, we get a set of k points Q, each of which approximates a unique vertex of the polytope K.

The algorithm for obtaining soft convex hull proceeds as follows. We first prune points $w \in W$ which have the following property: consider the subset X of points in W which are sufficiently far from w. Then w is close the convex hull of X. After pruning such points from W, we pick a subset of points which are sufficiently far-apart from each other. The main technical result shows that this procedure outputs the desired set T.

1.3 Related Work

The well known result of [15] shows that given a set of n points in $\Re^d$, projection to a random subspace of dimension $O(\log n/\varepsilon^2)$ preserves all pair-wise distances up to $(1+\varepsilon)$-factor with high probability. Further, this bound on the dimension on which the points are projected is known to be tight [2, 16]. Note that in our setting, there are $k+1$ points "of interest", namely, the k vertices of K and a point $a \notin K$ and by the above, a random projection to $O^*(\log k)$ dimensional space preserves all pairwise distances among them. But this is not sufficient for our problems. We need separation of a not just from the vertices of K, but from all of K in the projection. We achieve this by projecting to a set of random 1-dimensional subspaces, and show that the distance between a point and a polytope does not scale down by more than $O(\sqrt{d})$ factor for at least one of them with high probability (this is an immediate corollary of the RSH Theorem).

The problem of learning vertices of a polytope arises in many settings where data is assumed to be generated by a stochastic process parameterized by a model. Examples include topic models [10], stochastic block models [1], latent Dirichlet allocation [11]. A variety of techniques have been developed for these specific problems (see e.g. [3, 4, 14]). [7] (see also [5]) proposed the latent k-polytope (LkP) model which seeks to unify all of these latent variable models. In this model, there is *latent* polytope with k vertices, and data is generated in a two step process: first we pick *latent* points from this polytope, and then the observed points are obtained by perturbing these latent points in an adversarial manner. They showed that under suitable assumptions on this deterministic setting, one can capture the above-mentioned latent variable problems. Assuming strong separability conditions on the vertices of the polytope (i.e., each vertex of K is far from the *affine* hull of other vertices of K), they showed that one can efficiently recover good approximations to the vertices of the polytope from the input data points. In comparison, our assumption on K is that each vertex of K is far from the convex hull of the remaining vertices of K. This is a much milder condition, e.g., it allows a polytope with more than 2 vertices in a plane. [8] showed how to infer the parameter k from data in the LkP setting (under the strong separation condition).

[12] addressed a problem similar to the Hausdorff problem: instead of an ε-optimization oracle for a polytope K, we are given an explicit set P of points, whose convex hull is within Hausdorff distance at most $\delta\Delta(K)$ from K. They are able to get better dependencies on the parameters k, δ, ε than Theorem 21 under these stronger assumptions.

2 Preliminaries

For two points $x, y \in \Re^d$, $|x-y|$ denotes the Euclidean distance between the points. Given a point $x \in \Re^d$ and a subset $X \subseteq \Re^d$, define $\mathsf{dist}(x, X)$ as the minimum distance between x and a point in X, i.e., $\inf_{y\in X}|x-y|$. For a set of points X, $\Delta(X)$ denotes the diameter of X, i.e., $\sup_{x,y\in X}|x-y|$. We denote the convex hull of X by $\mathsf{CH}(X)$. For two subsets A, B of $\Re^d$, define their Minkowski sum $A+B$ as $\{x+y : x \in A, y \in B\}$. Similarly, define λA, where $\lambda \in \Re$, as $\{\lambda x : x \in A\}$. For an $m \times n$ matrix B, we use $B_{\cdot,j}$ to denote the j^{th} column of B. For a subset $S \subseteq [n]$ of columns of B, $B_{\cdot,S}$ denotes $\frac{1}{|S|}\sum_{j\in S} B_{\cdot,j}$. Often, we represent the vertices of a polytope K in $\Re^d$ by a $d \times k$ matrix M, and so the columns $M_{\cdot,1}, \ldots, M_{\cdot,k}$ would represent the vertices of K. We shall use the notation $\mathrm{poly}_\delta(z)$ to denote a quantity which is $z^{\mathrm{poly}(1/\delta)}$. Further the notation $O_\delta(z)$ shall denote a quantity which is $f(\delta)z$, where $f(\delta)$ is a function depending on δ only (and hence, is constant if δ is constant).

We now give an outline of rest of the paper. In Section 3, we prove the Random Separating Hyperplane theorem. In Section 4, we prove Theorem 5 by showing that an $\mathsf{OptOr}_\varepsilon(K)$ oracle leads to efficient constructions of approximation to K. In Section 5 we give an algorithm for the ListLearn problem under the stronger assumption that the vertices of K are well separated. We also prove the lower bound result Theorem 7 in this section. In Section 6, we extend the algorithm for ListLearn to the k-OLP problem. This requires the concept of soft convex hulls. The algorithm for constructing soft convex hulls in given in Section 6.1. Finally, in Section 7, we apply the k-OLP algorithm for the latent polytope problem. As note earlier, in the setting of latent polytopes K, we can only guarantee $\mathsf{OptOr}_\varepsilon(K)$ oracles with ε being $O(1/\sqrt{k})$, whereas our algorithm for k-OLP requires ε to be $O(1/\sqrt{d})$. We handle this issue by projecting to a suitable SVD subspace and executing the k-OLP algorithm in this subspace. We conclude with some open problems in Section 8.

3 Random Separating Hyperplane (RSH) Theorem

In this section, we prove RSH for polytopes: If a point p point is at a distance from a polytope K, then, the Gaussian measure of the set of *well-separating* hyperplanes has a positive lower bound depending on the number of vertices in K. More specifically, we show:

▶ **Theorem 12.** *Suppose K is a polytope in $\Re^d$ with k vertices and diameter $\Delta(K)$. Suppose a is a point in $\mathbf{R}^d$ and $\delta \in (0,1]$ with*

$$\min_{y\in K} |a-y| \geq \delta\Delta(K). \tag{3}$$

Let V be an m-dimensional subspace containing $\mathsf{Span}(K \cup \{a\})$ and let u be a random vector drawn from the normal distribution $N(0, I_m)$ in V. Then,

$$\mathbf{Pr}_u\left[(u\cdot a - \max_{y\in K} u\cdot y) \geq |u|\cdot\delta\Delta(K)\cdot\frac{\sqrt{\log k}}{3\sqrt{\log k}+4\delta\sqrt{m}}\right] \geq \frac{1}{40}k^{-10/\delta^2}.$$

Proof Overview

We give an informal overview of the proof. A naive, though incorrect, idea would be the following: let ζ be the closest vertex of K to a. We can argue that with reasonable probability (i.e., $\Omega(k^{-1/\delta^2})$) that $|(\zeta-a)\cdot u|$ is at least $\frac{2\sqrt{\log k}\cdot|\zeta-a|}{\delta\sqrt{m}} \geq \frac{2\sqrt{\log k}\cdot\Delta(K)}{\sqrt{m}}$. Call this event $\mathcal{E}_1$. Now consider a line segment joining two distinct vertices, say ζ_i, ζ_j, of K. The length of such a line segment is at most $\Delta(K)$, and hence there is high probability that $|(\zeta_i-\zeta_j)\cdot u| \leq \frac{1.5\sqrt{\log k}\cdot\Delta(K)}{\sqrt{m}}$. In fact, we can apply union bound, and show that this property holds for all pairs of vertices of K – call this event $\mathcal{E}_2$. If both $\mathcal{E}_1$ and $\mathcal{E}_2$ occur, then it is easy to see that along the direction u, a is separated from all the vertices of K. However, the events $\mathcal{E}_1$ and $\mathcal{E}_2$ are not independent, and hence, we cannot argue that both events will occur with non-trivial probability. Here one could use the union bound, but a calculation shows that for this we need $\delta > 1$. But $\delta < 1$ in our applications, so we cannot use this approach.

Instead, we rely on the following more nuanced idea. Let w be the unit direction along the line joining a and the closest point b of K. As argued above, we can show that $|(b-a)\cdot u|$ is at least $\frac{\Omega(\sqrt{\log k}\cdot|b-a|}{\delta\sqrt{m}} \geq \frac{\Omega(\sqrt{\log k})\cdot\Delta(K)}{\sqrt{m}}$ – this only requires arguing about the component of u along w; let $\mathcal{E}_1'$ denote this event. Now we show that the following event, say $\mathcal{E}_2'$, occurs with high probability: along the component of u orthogonal to w, the projections of the line segments joining any two vertices of K is at most $\frac{O(\sqrt{\log k})\Delta(K)}{\sqrt{m}}$. Since $\mathcal{E}_1'$ and $\mathcal{E}_2'$ are concerned with orthogonal directions, they are independent using properties of the Gaussian. Thus we can show that both of these events happen with non-trivial probability. This suffices to show the desired result. We now give the formal proof of this result. First, we state the following well-known concentration bound for Gaussian random variables (see e.g. [19]):

▶ **Theorem 13** (Gaussian Concentration Bounds). *Let X be an $N(0,1)$ Gaussian random variable, i.e., with mean 0 and variance 1. Then, for any $t > 0$,*

$$e^{-t^2/4}/2 \leq \Pr[|X| \geq t] \leq 2e^{-t^2}.$$

We shall also need the following result on the sum of squares of i.i.d. normal random variables.

▶ **Theorem 14** ([17]). *Let $X_1, \dots, X_n$ be n i.i.d. $N(0,1)$ Gaussian random variables. Then,*

$$\Pr[\sum_{i=1}^{n} X_i^2 > 4n] \leq e^{-\sqrt{n}}.$$

We now proceed with the proof of Theorem 12.

Proof of Theorem 12. Let $\zeta_1, \zeta_2, \dots, \zeta_k$ be the k vertices of K. Let b be the closest point in K to a, and define

$$w = \frac{a-b}{|a-b|}.$$

Then by standard results on separation between a point and a closed convex set (see e.g. [13]), we know that for all points $y \in K$:

$$w \cdot y \leq w \cdot b. \tag{4}$$

Now, we extend w to an orthonormal basis $w_1 := w, w_2, \dots, w_m$ of the subspace V. The random vector u can be generated as follows: sample m i.i.d. $N(0,1)$ random variables $\lambda_1, \dots, \lambda_m$, and define $u := \sum_{i=1}^{m} \lambda_i w_i$. We first observe that it is enough to argue that the projection of a along u is well separated from that for each vertex ζ_ℓ of K. Indeed, we can express $(u \cdot a - \max_{y \in K} u \cdot y)$ as

$$u \cdot (a-b) - \max_{y \in K} u \cdot (y-b) = u \cdot (a-b) - \max_{\ell=1,\dots,k} u \cdot (\zeta_\ell - b) \tag{5}$$

It suffices to show that with reasonably high probability, there is a lower bound on $u \cdot (a-b)$ and an upper bound on $u \cdot (\zeta_\ell - b)$ for all ℓ. We proceed with the latter goal first. Observe that along the direction w, K lies *below* b. Thus, it suffices to argue that the projection of $(\zeta_\ell - b)$ along the direction orthogonal to w is not large for any vertex ζ_ℓ of K. Let z denote $\lambda_2 w_2 + \ldots + \lambda_m w_m$. Then u can be expressed as $\lambda_1 w + z$. We now define events upper bounding $z \cdot (\zeta_\ell - b)$ for all vertices ζ_ℓ of K.

▶ **Definition 15.** *For $\ell = 1, 2, \dots, k$, define the event $\mathcal{E}_\ell$ as $[|z \cdot (\zeta_\ell - b)| \leq 2\sqrt{\ln k}\Delta(K)]$.*

▶ **Proposition 16.** *Suppose the events $\mathcal{E}_\ell$ occur for all $\ell = 1, \dots, k$. Then, for any $\ell \in [k]$,*

$$u \cdot (\zeta_\ell - b) \leq 2\sqrt{\ln k}\Delta(K).$$

Proof. Observe that

$$u \cdot (\zeta_\ell - b) = \lambda_1 w \cdot (\zeta_\ell - b) + z \cdot (\zeta_\ell - b) \overset{(4)}{\leq} z \cdot (\zeta_\ell - b).$$

The desired result now follows from the definition of the event $\mathcal{E}_\ell$. ◀

We now define events to lower bound $u \cdot (a - b)$. Observe that $(a - b)$ is along w, and hence, is orthogonal to z. Therefore $u \cdot (a - b) = \lambda_1 w(a - b) = \lambda_1 |a - b|$. Therefore, we define an event that lower bounds λ_1. We also define an event that upper bounds $|z|$.

▶ **Definition 17.** *Define $\mathcal{E}_0$ as the event $[|z| \leq 4\sqrt{m}]$. Also define $\mathcal{E}_{k+1}$ as the event $[\lambda \geq 3\sqrt{\ln k}/\delta]$.*

▶ **Proposition 18.** *Suppose the events $\mathcal{E}_0, \ldots, \mathcal{E}_{k+1}$ happen. Then,*

$$u \cdot a - \max_{y \in K} u \cdot y \geq \frac{\lambda \delta \Delta(K)|u|}{3(\lambda + 4\sqrt{m})} \geq \frac{\delta \sqrt{\ln k}\Delta(K)|u|}{3\sqrt{\ln k} + 4\delta\sqrt{m}}.$$

Proof. As observed above, $u \cdot (a - b) = \lambda_1 |a - b|$. Now observe that for any vertex ζ_ℓ of K, Proposition 16 shows that

$$u \cdot (\zeta_\ell - b) \leq 2\sqrt{\ln k}\Delta(K) \overset{(3)}{\leq} 2\sqrt{\ln k}|a - b|/\delta \leq 2\lambda_1 |a - b|/3,$$

where the last inequality follows from the definition of of event $\mathcal{E}_{k+1}$. Thus, we see from the above two inequalities and (5) that

$$u \cdot a - \max_{y \in K} u \cdot y \geq \lambda_1 |a - b|/3 \overset{(3)}{\geq} \lambda_1 \delta |\Delta(K)|/3. \tag{6}$$

It remains to upper bound $|u|$. Recall that $u = \lambda_1 w + z$, and hence, $|u| \leq |\lambda_1| + |z| \leq \lambda_1 + 4\sqrt{m}$, where we have used the definition of the event $\mathcal{E}_{k+1}$ and the fact that $\lambda_1 > 0$ because of event $\mathcal{E}_0$. Using this in (6) yields the first inequality in the desired claim. The second desired inequality follows from the fact that $\lambda_1 \delta \geq 3\sqrt{\ln k}$ (definition of event $\mathcal{E}_{k+1}$). ◀

It remains to show that the events $\mathcal{E}_0, \ldots, \mathcal{E}_{k+1}$ occur with reasonable probability.

▶ **Proposition 19.** $\mathbf{Pr}\left[\wedge_{\ell=0}^{k+1} \mathcal{E}_\ell\right] \geq \frac{1}{40} k^{-10/\delta^2}$.

Proof. Consider any $\ell \in [k]$. Observe that $z \cdot (\zeta_\ell - b)$ is a 1-dimensional Gaussian random variable with zero mean and with variance $|\zeta_\ell - b|^2 \leq \Delta(K)^2$. Thus, $\frac{z \cdot (\zeta_\ell - b)}{\Delta(K)}$ is an $N(0, 1)$ normal random variable. Therefore, Theorem 13 shows that

$$\Pr[\neg \mathcal{E}_\ell] = \Pr\left[\frac{z \cdot (\zeta_\ell - b)}{\Delta(K)} \geq 2\sqrt{\ln k}\right] \leq \frac{1}{2k^2}.$$

By definition of z, $|z|^2 = \lambda_2 + \ldots + \lambda_m^2$. By Theorem 14,

$$\Pr[\neg \mathcal{E}_0] = \Pr[|z|^2 > 16m] \leq e^{-\sqrt{m}} \leq 1/20,$$

for large enough m. Applying union bound to the above two inequalities, we get

$$\mathbf{Pr}\left(\wedge_{\ell=0}^{k} \mathcal{E}_\ell\right) \geq 1 - \sum_{\ell=0}^{k} \mathbf{Pr}(\neg \mathcal{E}_\ell) \geq 1/4 \tag{7}$$

Finally we consider event $\mathcal{E}_{k+1}$. Observe that λ_1 is $N(0, 1)$ Gaussian random variable. Applying Theorem 13 to λ_1, we see that

$$\Pr[\mathcal{E}_{k+1}] = \Pr\left[\lambda_1 \geq 3\sqrt{\ln k}/\delta\right] \geq \frac{1}{10} k^{-10/\delta^2}. \tag{8}$$

Observe that the event $\wedge_{\ell=0}^{k} \mathcal{E}_\ell$ depends on the random variables $\lambda_2, \ldots, \lambda_k$, and hence, is independent of the event $\mathcal{E}_{k+1}$. The desired result now follows from the (7) and (8). ◀

The theorem now follows from Proposition 18 and Proposition 19. ◀

4 From $\mathsf{OptOr}_\varepsilon(K)$ oracles to the Hausdorff Problem

We prove Theorem 5 in this section. We begin by defining Hausdorff distance formally.

▶ **Definition 20.** *The* Hausdorff-distance*,* $\mathsf{Haus}(K, K')$*, between two polytopes* K *and* K' *is the infimum over all values* α *such that the following condition is satisfied: for every point* $x \in K$*, there is a point* $y \in K'$ *such that* $|x - y| \leq \alpha$*, and vice versa.*

We now give the formal version of Theorem 5:

▶ **Theorem 21.** *Suppose* ε, δ *are reals in* $[0, 1]$ *with*

$$\delta > c\varepsilon\sqrt{d}, \delta > c/\sqrt{d}, \tag{9}$$

where c *is a large enough constant. Let* K *be a* k*-vertex polytope with* $\Delta(K)$ *denoting the diameter of* K*. Suppose we are also given an* $\mathsf{OptOr}_\varepsilon(K)$ *oracle* $\mathcal{O}$*. Let* P *be the set of answers from the oracle on* $m := k^{10+c\delta^{-2}}$ *independent random queries. Then,* $\mathsf{Haus}(K, \mathsf{CH}(P)) \leq \delta \cdot \Delta(K)$*.*

Before giving the proof, we highlight the main ideas here. Let the random queries correspond to unit directions $u^1, \ldots, u^m$, and for each of these queries u^j, the oracle $\mathcal{O}$ returns a near-optimal point $x(u^j)$. Let P^j denote the subset $\{x(u^1), \ldots, x(u^j)\}$. Let v be a vertex of K. The non-triviality in the proof lies in showing that there is a point in $\mathsf{CH}(P)$ close to v. Suppose there is no such point in $\mathsf{CH}(P^j)$, and let b the closest point in $\mathsf{CH}(P^j)$ to v. Then we show that with constant probability the following events happen: (i) the unit vector u^{j+1} has a non-trivial component along the direction joining b and v, and (ii) its component along the orthogonal direction is not too large. If both of these events happen, we show that $\mathsf{dist}(v, \mathsf{CH}(P^{j+1})) \leq (1 - O(\delta^2))\mathsf{dist}(v, \mathsf{CH}(P^j))$. We now give details of the proof.

Proof. We describe the algorithm for obtaining the desired set S (referred as **Random Probes Algorithm** in Section 1.2) in Algorithm 1. The set P is constructed as follows: we pick a set of m random unit vectors. For each such unit vector u, we add the corresponding point $x(u)$ returned by the oracle $\mathcal{O}$ to the set P.

Algorithm 1 Algorithm for finding the set P such that $\mathsf{CH}(P)$ approximates K.

Input: An $\mathsf{OptOr}_\varepsilon(K)$ oracle $\mathcal{O}$.
Initialize a set P to $\varnothing$
Repeat m times:
  Let u be a random unit vector in $\Re^d$.
  Call $\mathcal{O}$ on u to get a vector $x(u)$.
  Add $x(u)$ to P.
Output P.

One side of the desired result is easy to show:

▷ Claim 22. For each $x \in \mathsf{CH}(P)$, there is a $y \in K$ such that $|x - y| \leq \delta\Delta(K)$.

Proof. For a point $x(u) \in P$, we know by the definition of $\mathsf{OptOr}_\varepsilon(K)$ oracle that there is a point $y \in K$ such that $|x(u) - y| \leq \varepsilon\Delta(K) \leq \delta\Delta(K)$, where $\varepsilon \leq \delta$ follows from (9). The desired result now follows from the convexity of K. ◁

It remains to show that for any vertex v of K, there is a point $y \in \mathsf{CH}(P)$ such that $|v-y| \leq \delta\Delta(K)$. Fix such a vertex v of K for rest of the discussion. Let the random unit vectors considered in Algorithm 1 (in the order they get generated) be $u^1, \ldots, u^m$. Let P^j denote the subset $\{x(u^1), \ldots, x(u^j)\}$ of P. Define an event $\mathcal{E}_j$ as follows:

$$\mathsf{dist}(v, \mathsf{CH}(P^j)) \leq \delta \cdot \Delta(K) \quad \text{or} \quad \mathsf{dist}(v, \mathsf{CH}(P^{j+1})) \leq \left(1 - \frac{\delta^2}{c'}\right) \mathsf{dist}(v, \mathsf{CH}(P^j)),$$

where c' is a large enough constant. Our main technical result is to show that conditioned on *any* choice of $u^1, \ldots, u^j$ the event $\mathcal{E}_j$ happens with reasonably high probability (where the probability is over the choice of u^{j+1}):

▶ **Lemma 23.** *For any index $j \in [m-1]$,*

$$\Pr_{u^{j+1}}\left[\mathcal{E}_j | u^1, \ldots, u^j\right] \geq \frac{1}{100}.$$

Proof. Fix the vectors $u^1, \ldots, u^j$. If $\mathsf{dist}(v, \mathsf{CH}(P^j)) \leq \delta \cdot \Delta(K)$, then we are done. So assume this is not the case. Let b be the closest point in $\mathsf{CH}(P^j)$ to v. Thus,

$$|v-b| \geq \delta \cdot \Delta(K). \tag{10}$$

Define w as

$$w := \frac{v-b}{|v-b|}.$$

We can now express the vector u^{j+1} as $\lambda w + z$, where $\langle z, w\rangle = 0$. We first show the following useful properties of these vectors.

▷ Claim 24. With probability at least $\frac{1}{100}$, the following three events happen:

$$|z| \leq 4\sqrt{d} \tag{11}$$

$$\max_{y \in K} |z \cdot (v-y)| \leq 2\sqrt{\ln k}\,\Delta(K) \tag{12}$$

$$\lambda \geq \frac{100}{\delta}\sqrt{\ln k} \tag{13}$$

Proof. The proofs of these three inequalities are same as the arguments in Proposition 19 (in order to prove (12), it suffices to show it for points y which are vertices of K). ◁

The following fact is also easy to show:

▶ **Proposition 25.**

$$|v - x(u^{j+1})| \leq 2\Delta(K).$$

Proof. By the definition of $\mathsf{OptOr}_\varepsilon(K)$ oracle, there is a point $p \in K$ such that $|p - x(u^{j+1})| \leq \delta \cdot \Delta(K) \leq \Delta(K)$. The desired result now follows by triangle inequality. ◀

Let δ_1 denote $\frac{\delta^2}{100}$. Let b_1 denote the vector

$$\delta_1 x(u^{j+1}) + (1-\delta_1) b.$$

Since $b_1 \in \mathsf{CH}(P^{j+1})$, the desired result will follow if we prove the following:

$$|v-b_1|^2 \leq \left(1 - \frac{\delta^2}{100}\right)|v-b|^2. \tag{14}$$

Now,

$$\begin{aligned}|v-b_1|^2 &= \delta_1^2|v-x(u^{j+1})|^2+(1-\delta_1)^2|v-b|^2+2\delta_1(1-\delta_1)(v-x(u^{j+1}))\cdot(v-b)\\ &\overset{\text{Prop. 25}}{\leq} 4\delta_1^2\Delta(K)^2+(1-2\delta_1)|v-b|^2+\delta_1^2\Delta(K)^2+2\delta_1(1-\delta_1)(v-x(u^{j+1}))\cdot(v-b)\\ &\overset{(10)}{\leq}\left(1-\frac{3\delta_1}{2}\right)|v-b|^2++2\delta_1(1-\delta_1)|v-b|\cdot(v-x(u^{j+1}))\cdot w\\ &=\left(1-\frac{3\delta_1}{2}\right)|v-b|^2++2\delta_1(1-\delta_1)\left(\underbrace{\frac{|v-b|}{\lambda}\cdot(v-x(u^{j+1}))\cdot u^{j+1}}_{:=A}\right.\\ &\qquad\left.\underbrace{-\frac{|v-b|}{\lambda}\cdot(v-x(u^{j+1}))\cdot z}_{:=B}\right)\end{aligned}\tag{15}$$

We now bound each of the terms A and B above. Now,

$$A\leq\frac{|v-b|}{\lambda}\varepsilon\Delta(K)\overset{(9)}{\leq}\frac{|v-b|\delta\Delta(K)}{c\lambda}\overset{(10)}{\leq}\frac{|v-b|^2}{c\lambda}\overset{(13)}{\leq}\frac{|v-b|^2\delta}{c},$$

where the first inequality follows from the definition of $\mathsf{OptOr}_\varepsilon(K)$ oracle. We now bound the quantity B. Let y be the point in K closest to $x(u^{j+1})$. We know that $|y-x(u^{j+1})|\leq\varepsilon\Delta(K)$. Therefore,

$$\begin{aligned}B&\leq\frac{|v-b|}{\lambda}\left(|(v-y)\cdot z|+|z|\varepsilon\Delta(K)\right)\\ &\overset{(11),(12)}{\leq}\frac{|v-b|}{\lambda}\left(2\sqrt{\ln k}\Delta(K)+4\varepsilon\sqrt{d}\Delta(K)\right)\\ &\overset{(9),(13)}{\leq}\frac{\delta|v-b|\Delta(K)}{10}\overset{(10)}{\leq}\frac{|v-b|^2}{10}.\end{aligned}$$

Substituting the above bound on A and B in (15) yields the desired result. ◀

We are now almost done. As the following result shows, it suffices to argue that enough number of events $\mathcal{E}_j$ happen:

▷ Claim 26. If at least $\frac{c'}{\delta^2}\ln(2/\varepsilon)$ of the events $\mathcal{E}_j, j\in[m-1]$ happen, then $\mathsf{dist}(v,\mathsf{CH}(P))\leq\delta\Delta(K)$.

Proof. Assume, for the sake of contradiction, that $\mathsf{dist}(v,\mathsf{CH}(P))>\varepsilon\Delta(K)$. Assume that events $\mathcal{E}_{j_1},\ldots,\mathcal{E}_{j_h}$ happen, where $h:=\frac{c'}{\delta^2}\ln(2/\varepsilon)$. Now, for any index $i\in[h-1]$, the definition of $\mathcal{E}_{j_{i+1}}$ implies that

$$\mathsf{dist}(v,\mathsf{CH}(P^{j_{i+1}}))\leq\left(1-\frac{\delta^2}{c'}\right)\mathsf{dist}(v,\mathsf{CH}(P^{j_{i+1}-1})\leq\mathsf{dist}(v,\mathsf{CH}(P^{j_i})).$$

Therefore,

$$\mathsf{dist}(v,\mathsf{CH}(P^{j_h}))\leq\left(1-\frac{\delta^2}{c'}\right)^h\mathsf{dist}(v,P^1)\overset{\text{Proposition 25}}{\leq}\left(1-\frac{\delta^2}{c'}\right)^h 2\Delta(K)\leq\delta\Delta(K).\ \triangleleft$$

It remains to show that with high probability at least $h := \frac{c'}{\delta^2}\ln(2/\varepsilon)$ of the events happen. In order to prove this, we divide the sequence $[m]$ into $[h]$ subsequences, each of length m/h. Call these subsequences $C_1, \dots, C_h$. It follows from Lemma 23 that for any $i \in [h]$,

$$\Pr\left[\wedge_{j \in C_i} \neg \mathcal{E}_j\right] \leq 0.99^{m/h} \leq \frac{1}{h^2}.$$

A simple union bound now shows that with probability at least $1 - 1/h$, at least one event $\mathcal{E}_j$ happens during each of the subsequences $C_1, \dots, C_h$. Claim 26 now proves the theorem. ◀

5 From $\mathsf{OptOr}_\varepsilon(K)$ oracles to ListLearn

We first define the notion of well-separatedness.

▶ **Definition 27.** *We say that a polytope K with vertex set V is* δ-well-separated *if for every vertex $v \in V$, we have* $\mathsf{dist}(v, \mathsf{CH}(V \setminus \{v\})) \geq \delta \cdot \Delta(K)$.

Recall that Theorem 7 states that given access to only an $\mathsf{OptOr}_\varepsilon(K)$ oracle, ε must be $\Omega(\ln d/\sqrt{d})$ for any deterministic algorithm to output a point within $\Delta(K)/10$ of K. We show this result in the full version [9]. We now prove Theorem 6, which gives such an algorithm for the ListLearn problem.

▶ **Theorem 28.** *Suppose ε, δ are reals in $[0,1]$ with $\delta^2 \geq c\varepsilon\sqrt{d}, \delta^3 \geq c\,\varepsilon$, where c is a large enough constant. Let K be a δ-well-separated k-vertex polytope. Suppose we are also given* $\mathsf{OptOr}_\varepsilon(K)$ *oracle $\mathcal{O}$. Let W be the set of answers of the oracle to $m = poly(d) \cdot k^{\Omega(1/\delta^2)}$ independent random queries. Then for each vertex v of K, there is a point $v' \in W$ such that $|v - v'| \leq O(\delta^2 \Delta(K)/c)$.*

Proof. The algorithm chooses a set U of $k^{\Omega(1/\delta^2)}$ unit length i.i.d. Gaussian vectors. For each $u \in U$, it calls the oracle $\mathcal{O}$ to find a vector $x(u)$. Let W denote the set $\{x(u) : u \in U\}$. We first show that for every vertex of K, there is a direction u in U along which the projection of this vertex is higher than the projection of the remaining vertices by a large enough margin. Let $M_{\cdot,1}, \dots, M_{\cdot,k}$ be the vertices of K. The proof of the following result is deferred to the full version [9].

▷ Claim 29. With high probability, the following event happens: for each $\ell \in [k]$, there is a vector $u^{(\ell)} \in U$ such that for all $\ell' \in [k], \ell' \neq \ell$, we have

$$u^{(\ell)} \cdot M_{\cdot,\ell} > u^{(\ell)} \cdot M_{\cdot,\ell'} + \frac{c\,\varepsilon \cdot \Delta(K)}{8\delta^2} \tag{16}$$

For rest of the proof, assume that the statement in Claim 29 holds true, i.e., there are directions $u^{(1)}, \dots, u^{(k)} \in U$ satisfying (16). We now show that for every vertex of $M_{\cdot,\ell}$ of K, the corresponding point $x(u^\ell)$ is close to $M_{\cdot,\ell}$.

▷ Claim 30. For every $\ell \in [k]$, $|x(u^{(\ell)}) - M_{\cdot,\ell}| \leq 17\delta^2\Delta(K)/c$.

Proof. By the definition of $\mathcal{O}$, we know that $x(u^{(\ell)})$ can be written as $y(u^\ell) + z(u^{(\ell)})$, where $y(u^{(\ell)}) \in K$ and $|z(u^{(\ell)})| \leq \varepsilon\Delta(K)$. Thus, there is a convex combination $\lambda_{\ell'}, \ell' \in [k]$, of the vertices $M_{\cdot,\ell'}$ of K such that $x(u^{(\ell)}) = \sum_{\ell' \in [k]} \lambda_{\ell'} M_{\cdot,\ell'} + z(u^{(\ell)})$.

By the definition of $\mathcal{O}$, $x(u^{(\ell)}) \cdot u^{(\ell)} \geq M_{\cdot,\ell} \cdot u^{(\ell)} - \varepsilon\Delta(K)$ and $|z(u^{(\ell)}) \cdot u^{(\ell)}| \leq \varepsilon\Delta(K)$. So, we get

$$M_{\cdot,\ell} \cdot u^{(\ell)} - \varepsilon\Delta(K) \leq \sum_{\ell' \in [k]} \lambda_{\ell'} M_{\cdot,\ell'} \cdot u^{(\ell')} + \varepsilon\Delta(K),$$

which implies (after subtracting $\lambda_\ell M_{\cdot,\ell} \cdot u^{(\ell)}$ from both sides):

$$(1-\lambda_\ell)M_{\cdot,\ell}u^{(\ell)} \leq \sum_{\ell' \neq \ell} \lambda_{\ell'} M_{\cdot,\ell'} \cdot u^{(\ell')} + 2\varepsilon\Delta(K)$$

which, using Claim 29, yields:

$$(1-\lambda_\ell)M_{\cdot,\ell}u^{(\ell)} \leq (1-\lambda_\ell)M_{\cdot,\ell}u^{(\ell)} - (1-\lambda_\ell)\frac{c\varepsilon\Delta(K)}{8\delta^2} + 2\varepsilon\Delta(K).$$

It follows from the above inequality that $1-\lambda_\ell \leq \frac{16\delta^2}{c}$. Therefore,

$$|x(u^{(\ell)}) - M_{\cdot,\ell}| = \left|\sum_{\ell' \neq \ell} \lambda_{\ell'}(M_{\cdot,\ell} - M_{\cdot,\ell'})\right| + \varepsilon\Delta(K) \tag{17}$$

$$\leq \sum_{\ell' \neq \ell} \lambda_{\ell'}\Delta(K) + \varepsilon\Delta(K) \leq \frac{17\delta^2\Delta(K)}{c}. \qquad \triangleleft$$

This completes proof of the Theorem. ◀

6 From ListLearn to the k-OLP Problem

In this section, we show that for well-separated polytopes, a solution for the ListLearn problem can be used to solve the k-OLP problem as well. This algorithm uses the notion of soft convex hulls. We first describe the algorithm for constructing soft convex hulls, and then use it to solve the k-OLP problem.

6.1 Soft Convex Hulls

Let W be a finite set of points in $\Re^d$, and T be the vertices of $\mathsf{CH}(W)$. The subset T of W is the unique subset of W with the following properties:

(P1) $W \subseteq \mathsf{CH}(T)$

(P2) $\forall w \in W$, if $w \notin \mathsf{CH}(W \setminus \{w\})$, then $w \in T$.

We now define a natural notion of *soft* convex hull.

▶ **Definition 31.** *For an $\varepsilon \geq 0$, and $S \subseteq W$, define the ε-convex hull of S, ε-$\mathsf{CH}(S)$, as $\mathsf{CH}(S) + \varepsilon\Delta(W)B$, where B is the unit ball of the Euclidean norm.*

The intuition behind the above definition is that $\mathsf{CH}(W)$ can have many vertices, but there may be a small set of points whose soft convex hull contains W. This is defined more formally as follows:

▶ **Definition 32.** *We call a subset $T \subseteq W$ an ε-envelope of W, written ε-$\mathsf{ENV}(W)$, if $W \subseteq \varepsilon$-$\mathsf{CH}(T)$.*

Remarks. The following observations about the set ε-$\mathsf{ENV}(W)$ are easy to see:
(a) There are several distinct sets T which could qualify as ε-$\mathsf{ENV}(W)$. For example, let W consist of the following set of points in $\Re^2$: a set of points W_1 close to $(0,0)$ and a set of points W_2 close to $(1,0)$. Let T be a pair of points $\{x,y\}$ with $x \in W_1, y \in W_2$. Then it is easy to check that T is an ε-$\mathsf{ENV}(W)$; and (b) Let T be ε-$\mathsf{ENV}(W)$. Unlike property (P2) above, it is not necessary that if $w \in W$ is such that $w \notin \varepsilon$-$\mathsf{CH}(W \setminus \{w\})$, then $w \in T$.

Since the set ε-$\mathsf{ENV}(W)$ is not uniquely determined, we will impose one more condition on it to make it unique (if it exists) and polynomial time computable. This condition requires the points of T to be "far apart" from each other. More precisely:

▶ **Definition 33.** *For $\varepsilon, \delta \in [0,1]$, a set T is called a (ε, δ)-$\mathsf{ENV}(W)$ if it is an ε-$\mathsf{ENV}(W)$ and*

$$\forall w \in T, \mathsf{dist}(w, \mathsf{CH}(T \setminus \{w\})) > \delta\Delta(W) \tag{18}$$

The proof of the following result follows from standard arguments and is deferred to the appendix.

Given a subset T of W, we can check in polynomial time whether T is an (ε, δ)-$\mathsf{ENV}(W)$.

For rest of the section, we address the following question: given the set W, and parameters ε, δ, is there a (ε, δ)-$\mathsf{ENV}(W)$, and if so, can we find in polynomial time an approximation to this set? In the full version [9], we argue that several "natural" greedy strategies do not work. If $\varepsilon = 0$, the above question is easy to answer in polynomial time. The answer is yes iff the set T of vertices of $\mathsf{CH}(W)$ satisfies (18). Also, if $\delta = 1$, T has to be a singleton to satisfy (18).

In rest of this section, we consider the following problenm: For what pairs of values of ε, δ, can we prove that there is *essentially* at most one (ε, δ)-$\mathsf{ENV}(W)$, and if so, can we determine this set efficiently? We do not know the exact answer to this, but our main result here (which suffices for the applications) is (verbally stated) an affirmative answer to the question if the following condition is satisfied: $\delta \in \Omega(\sqrt{\varepsilon})$. This will follow as a corollary of our main result:

▶ **Theorem 34.** *Let $\delta, \varepsilon, \varepsilon_3$ be reals in $(0, 1/8)$ satisfying*

$$\delta > \max\left(\frac{2\varepsilon}{\varepsilon_3 - \varepsilon}, 4\varepsilon_3\right) \tag{19}$$

Let W be a finite set of points in $\Re^d$. We can determine in polynomial time whether exists a set T in (ε, δ)-$\mathsf{ENV}(W)$, and if so, we can efficiently find a subset Q of W such that

$$|Q| = |T| \tag{20}$$

$$\forall w \in T, \exists x \in Q : |w - x| \leq 2\varepsilon_3\Delta(W) \tag{21}$$

▶ **Corollary 35.** *Let δ, ε be reals in $(0, 1/8)$ satisfying $\delta > 16\sqrt{\varepsilon}$ Let W be a finite set of points in $\Re^d$. We can determine in polynomial time whether exists a set T forming a (ε, δ)-$\mathsf{ENV}(W)$, and if so, we can efficiently find a subset Q of W such that*

$$|Q| = |T| \tag{22}$$

$$\forall w \in T, \exists x \in Q : |w - x| \leq 8\sqrt{\varepsilon}\Delta(W) \tag{23}$$

Proof. The Corollary follows from Theorem (34) by taking $\varepsilon_3 = 4\sqrt{\varepsilon}$ ◀

We defer the proof of Theorem 34 to the full version [9]. The procedure for computing the set Q is as follows: We first compute a subset Q'' of W consisting of points $w \in W$ which do not lie in the soft convex hull of the points in W which are "far" from w – this can be done in polynomial time by using arguments similar to those in the proof of Section 6.1. Then Q is defined as a maximal subset of points in Q'' such that the pair-wise distance between the points in it is large.

6.2 Algorithm for k-OLP

We now show how soft convex hulls can be used to generate a solution for the k-OLP problem. The following result, which formalizes Theorem 9, uses the same setting as that in Theorem 28:

▶ **Theorem 36.** *Suppose ε, δ are reals in $[0,1]$ with $\delta^2 \geq c\varepsilon\sqrt{d}, \delta^3 \geq c\,\varepsilon$, where c is a large enough constant. Let K be a δ-well-separated k-vertex polytope. Suppose we are also given an $\mathsf{OptOr}_\varepsilon(K)$ oracle $\mathcal{O}$. Let W be the set of answers of the oracle $\mathcal{O}$ to $m = poly(d) \cdot k^{\Omega(1/\delta^2)}$ independent random queries. We can find $Q \subseteq W, |Q| = k$ in randomized $poly(d) \cdot k^{\Omega(1/\delta^2)}$-time which satisfies the following condition w.h.p.: for every vertex v of K, there is a point v' in Q with $|v - v'| \leq \delta\Delta(K)/10$.*

Proof. The proof of Theorem 28 shows that for every vertex $M_{\cdot,\ell}$ of K, there is a point $x(u^\ell) \in W$ such that

$$|x(u^{(\ell)}) - M_{\cdot,\ell}| \leq 17\delta^2\Delta(K)/c. \tag{24}$$

Let T denote $\{x(u^{(\ell)}) : \ell \in [k]\}$. Our first claim is that the points of T are also well-separated.

▷ Claim 37. For any $\ell \in [k]$, $\mathsf{dist}(x(u^{(\ell)}), \mathsf{CH}(T \setminus \{x(u^{(\ell)})\})) \geq \delta\Delta(K)/2$. Further, the diameter of $\mathsf{CH}(T)$ is at most $2\Delta(K)$.

Proof. Fix an index $\ell \in [k]$ and a point $y \in \mathsf{CH}(T \setminus \{x(u^{(\ell)})\})$. We can express y as a convex combination of points in $T \setminus \{x(u^{(\ell)})\}$, i.e., $y = \sum_{\ell' \neq \ell} \lambda_{\ell'} \cdot x(u^{(\ell')})$, where $\sum_{\ell' \neq \ell} \lambda_{\ell'} = 1$. Now,

$$\begin{aligned}
|x(u^{(\ell)}) - y| &\geq \left| M_{\cdot,\ell} - \sum_{\ell' \neq \ell} \lambda_{\ell'} \cdot M_{\cdot,\ell'} \right| - |x(u^{(\ell)}) - M_{\cdot,\ell}| - \sum_{\ell' \neq \ell} \lambda_{\ell'} \cdot |x(u^{(\ell')}) - M_{\cdot,\ell'}| \\
&\geq \left(\delta - \frac{34\delta^2}{c}\right)\Delta(K) \geq \delta\Delta(K)/2,
\end{aligned}$$

where the second last inequality follows from (24) and the fact that K is δ-well-separated. Since $\mathsf{dist}(x(u^{(\ell)}), K) \leq \varepsilon\Delta(K)$, it follows that $\Delta(\mathsf{CH}(T)) \leq 2\Delta(K)$. ◁

Recall that W denotes the set $\{x(u) : u \in U\}$. We now show that $\mathsf{CH}(T)$ closely approximates $\mathsf{CH}(W)$.

▷ Claim 38. $W \subseteq \varepsilon'\text{-}\mathsf{CH}(T)$, where $\varepsilon' = \frac{32\delta^2}{c}$.

Proof. Fix a point $x(u) \in W$. We know that $x(u)$ can be written as $x(u) = y(u) + z(u)$, $y(u) \in K, |z(u)| \leq \varepsilon$. Let $y(u) = \sum_{\ell \in [k]} \lambda_\ell \cdot M_{\cdot,\ell}$, where the coefficients λ_ℓ form a convex combination. Then

$$\left| x(u) - \sum_{\ell \in [k]} \lambda_\ell x(u^{(\ell)}) \right| \leq \sum_{\ell \in [k]} \lambda_\ell \cdot |x(u^{(\ell)}) - M_{\cdot,\ell}| + |z(u)| \leq \frac{17\delta^2\Delta(K)}{c} + \varepsilon\Delta(K) \leq \frac{32\delta^2\Delta(K)}{c},$$

where the second last inequality follows from (24) and the last inequality by the assumption in. ◁

Claim 37 and Claim 38 imply that T is (ε', δ')-$\mathsf{ENV}(W)$ with $\delta' = \delta/4, \varepsilon' = \frac{32\delta^2}{c}$. We can now apply Corollary 35 to get approximations to $x(u^{(\ell)})$ within distance $17\sqrt{\varepsilon'}\Delta(K)$. Claim 38 now implies that we can get approximations to $M_{\cdot,\ell}$ within distance

$$17\sqrt{\varepsilon'}\Delta(K) + \frac{16\delta^2\Delta(K)}{c} \leq \frac{\delta\Delta(K)}{10}.$$

This proves the desired result. ◀

7 k-OLP algorithm for Latent Polytopes using Singular Value Decomposition

Theorem 7 showed that a solution to k-OLP problem requires the error parameter ε to be $O^*(1/\sqrt{d})$. Theorem 28 gives an algorithm achieving this bound. However, for many polytopes with k vertices, we can solve the k-OLP problem with ε being $O^*(1/\sqrt{k})$. However, if $k < d$, this error is too high. To tackle this, we find a good approximation to the subspace spanned by the vertices of K, then we project to this subspace and use the result in Theorem 28. One such example is the "Latent $k-$ Polytope" (abbreviated LkP) problem which we now describe.

The LkP problem has been studied in [7]. Certain assumptions were made on the model, namely, the hidden polytope K as well as on the (hidden) process for generating observed data from latent points in K. These assumptions are (a) shown to hold in several important Latent Variable models and (b) are sufficient to enable one to get polynomial time learning algorithms.

Here, we formulate assumptions which are similar, but, weaker in one important aspect. Whereas [7] assumed that each vertex of K has a separation from the **affine hull** of the other vertices (thus, in particular, each vertex is affinely independent of other vertices), we assume here that each vertex is separated only from the convex hull of the others. Under this weaker assumption, the algorithm of [7] does not work. We give a different algorithm which we prove works. It is also simpler to state and carry out and its proof is based on a new general tool we introduce here - the Random Separating Hyperplane theorem (Theorem 12).

Assumptions on data in the LkP problem. Let $M_{\cdot,1}, \ldots, M_{\cdot,k}$ denote the vertices of K and $\mathbf{M}$ be the $d \times k$ matrix with columns representing the vertices of K. We assume there are latent (hidden) points $P_{\cdot,j}, j = 1, 2, \ldots, n$ in K and observed data points $A_{\cdot,j}, j = 1, 2, \ldots, n$ are generated (not necessarily under any stochastic assumptions) by adding *displacements* $A_{\cdot,j} - P_{\cdot,j}$ respectively to $P_{\cdot,j}$. Clearly if the displacements are arbitrary, it is not possible to learn K given only the observed data. So we need some bound on the displacements.

Secondly, if all (or almost all) latent points lie in (or close to) the convex hull of a subset of $k-1$ or fewer vertices of K, the missing vertex cannot be learnt. To avoid this, we will assume that there is a certain w_0 fraction of latent points close to every vertex of K.

Let [4] $\sigma_0 := \frac{||\mathbf{P}-\mathbf{A}||}{\sqrt{n}}$. We now show that the k-OLP-**Algorithm** mentioned in Section 1.2 has the desired properties:

▶ **Theorem 39.** *Suppose K is a latent poytope with k vertices $M_{\cdot,1}, M_{\cdot,2}, \ldots, M_{\cdot,k}$ and $\mathbf{P}, \mathbf{A}$ are latent points (all in K) and observed data respectively. Assume*

$$\text{For all } \ell \in [k]\ ,\ C_\ell := \{j : |P_{\cdot,j} - M_{\cdot,\ell}| \leq \frac{\sigma_0}{\sqrt{w_0}}\} \text{ satisfies } |C_\ell| \geq w_0 n. \tag{25}$$

Suppose $(\sqrt{\log k}/\sqrt{c_0 k}) \leq \delta \leq 1$ and c_0 is a large constant satisfying

$$\sigma_0 \leq \frac{\delta^2 \Delta(K)}{100 c_0} \frac{\sqrt{w_0}}{\sqrt{k}}. \tag{26}$$

Let V be the k-dimensional SVD subspace of $\mathbf{A}$, and $\widehat{K}$ denote the projection of K on V.

[4] By the standard definition of spectral norm, it is easy to see that σ_0^2 is the maximum mean squared displacement in any direction.

- *There is an* $\mathsf{OptOr}_{\frac{10\sigma_0}{\sqrt{w_0}\Delta}}(\widehat{K})$ *oracle* $\mathcal{O}$.
- *The algorithm* k-OLP **Algorithm** *in Section 1.2 outputs a set* Q *of* k *points such that the following condition is satisfied w.h.p.: for every vertex* v *of* K*, there is a point* v' *in* Q *with* $|v - v'| \leq \delta\Delta(K)/5$.

In the Theorem above, (26) implies an upper bound on σ_0 of $\Delta(K)\sqrt{w_0}/(c_0\sqrt{k})$ and so the oracle $\mathcal{O}$ used by the Theorem lies in $\mathsf{opt}_\varepsilon(\widehat{K})$ for $\varepsilon \leq 1/(c_0\sqrt{k})$. Thus, we get around the lower bound result Theorem 7 by working in the k-dimensional SVD subspace of $\mathbf{A}$. The proof of this result is deferred to the full version [9].

8 Open Problems

We now mention some problems that remain open in our work:

(i) In the statement of RSH, the success probability of the desired event is $O\left(1/k^{O(1/\delta^2)}\right)$. Can we improve the exponential dependence of the success probability on $1/\delta$?

(ii) Theorem 21 on the Haus problem returns exponentially many points whose convex hull approximates K. Can this be improved, either via an improvement mentioned in the first open problem above or by feeding the exponentially many points to the algorithm of [12]?

(iii) RSH asserts that if a point is sufficiently far from a k vertex polytope, then, a random hyperplane separates them, where, "sufficienlty" is measured with respect to the diameter of the polytope. An interesting question is whether there is a "dual" statement. The dual of points are facets, so, intuitively, a dual statement might assert that if for a polytope with k facets, there is a sufficiently deep cut, then, a random cut cuts reasonably deep into the polytope. But it is not obvious how to measure "deep" (and reasonably deep) . We leave a dual statement of RSH as an open question.

References

1 Edoardo M. Airoldi, David M. Blei, Elena A. Erosheva, and Stephen E. Fienberg. Introduction to mixed membership models and methods. In Edoardo M. Airoldi, David M. Blei, Elena A. Erosheva, and Stephen E. Fienberg, editors, *Handbook of Mixed Membership Models and Their Applications*, pages 3–13. Chapman and Hall/CRC, 2014. doi:10.1201/b17520-3.

2 Noga Alon. Problems and results in extremal combinatorics—i. *Discrete Mathematics*, 273(1-3):31–53, 2003.

3 Anima Anandkumar, Dean P. Foster, Daniel J. Hsu, Sham M. Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. *Algorithmica*, 72(1):193–214, 2015. doi:10.1007/s00453-014-9909-1.

4 Sanjeev Arora, Rong Ge, Yoni Halpern, David M. Mimno, Ankur Moitra, David A. Sontag, Yichen Wu, and Michael Zhu. Learning topic models - provably and efficiently. *Commun. ACM*, 61(4):85–93, 2018. doi:10.1145/3186262.

5 Ainesh Bakshi, Chiranjib Bhattacharyya, Ravi Kannan, David P. Woodruff, and Samson Zhou. Learning a latent simplex in input sparsity time. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: https://openreview.net/forum?id=04LZCAxMSco.

6 D. Bertsekas. *Convex Optimization Theory*. Athena Scientific optimization and computation series. Athena Scientific, 2009. URL: https://books.google.co.in/books?id=0H1iQwAACAAJ.

7 Chiranjib Bhattacharyya and Ravindran Kannan. Finding a latent k-simplex in O^* ($k \cdot$ nnz(data)) time via subset smoothing. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 122–140. SIAM, 2020. doi:10.1137/1.9781611975994.8.

8 Chiranjib Bhattacharyya, Ravindran Kannan, and Amit Kumar. Finding k in latent k-polytope. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 894–903. PMLR, 2021. URL: http://proceedings.mlr.press/v139/bhattacharyya21a.html.
9 Chiranjib Bhattacharyya, Ravindran Kannan, and Amit Kumar. Random separating hyperplane theorem and learning polytopes, 2023. arXiv:2307.11371.
10 David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, 2012. doi:10.1145/2133806.2133826.
11 David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. URL: http://jmlr.org/papers/v3/blei03a.html.
12 Avrim Blum, Sariel Har-Peled, and Benjamin Raichel. Sparse approximation via generating point sets. *ACM Trans. Algorithms*, 15(3):32:1–32:16, 2019. doi:10.1145/3302249.
13 Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014. doi:10.1017/CBO9780511804441.
14 Samuel B. Hopkins and David Steurer. Efficient bayesian estimation from few samples: Community detection and related problems. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 379–390. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.42.
15 William Johnson and Joram Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, January 1984. doi:10.1090/conm/026/737400.
16 Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 633–638. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.64.
17 B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000. doi:10.1214/aos/1015957395.
18 R Tyrrell Rockafellar and Johannes O Royset. Random variables, monotone relations, and convex analysis. *Mathematical Programming*, 148:297–331, 2014.
19 Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint*, 2010. arXiv:1011.3027.

A Some Examples

The first example shows that the factor $1/\sqrt{m}$ in the statement of RSH (Theorem 1) cannot be improved, even for a simple polytope K consisting of just one line segment.

▶ **Example 1.** Suppose K is a polytope in $\Re^d$ with just two vertices Δ distance apart and $V = \Re^d, \delta = 1$. It is easy to verify from standard results on random projection on a line that the additive term in inequality (2) is tight up to constant factors.

The next example shows that the success probability of the desired event (2) in Theorem 1 needs to depend on k, the number of vertices of the polytope K.

▶ **Example 2.** Consider the Euclidean space $\Re^d$ and the $d-1$ dimensional sphere $K := \{x \in \Re^d : |x| \leq 1, x_1 = 0\}$. Observe that the diameter Δ of K is 1. Let a denote the point $(1, 0, 0, \ldots, 0)$. It is at distance 1 from K, and so the parameter δ in the statement of Theorem 1 can be set to 1. But, with high probability, a random unit length vector u has $u \cdot a \approx \frac{c}{\sqrt{d}}$ for some constant c, whereas the maximum of $v \cdot x$ over all points x in the sphere K is about 1. Therefore the probability of the event (2) is exponentially small in d.