Hypercube LSH for Approximate near Neighbors

Thijs Laarhoven[*]

IBM Research, Rüschlikon, Switzerland
mail@thijs.com

Abstract

A celebrated technique for finding near neighbors for the angular distance involves using a set of *random* hyperplanes to partition the space into hash regions [Charikar, STOC 2002]. Experiments later showed that using a set of *orthogonal* hyperplanes, thereby partitioning the space into the Voronoi regions induced by a hypercube, leads to even better results [Terasawa and Tanaka, WADS 2007]. However, no theoretical explanation for this improvement was ever given, and it remained unclear how the resulting hypercube hash method scales in high dimensions.

In this work, we provide explicit asymptotics for the collision probabilities when using hypercubes to partition the space. For instance, two near-orthogonal vectors are expected to collide with probability $(\frac{1}{\pi})^{d+o(d)}$ in dimension d, compared to $(\frac{1}{2})^d$ when using random hyperplanes. Vectors at angle $\frac{\pi}{3}$ collide with probability $(\frac{\sqrt{3}}{\pi})^{d+o(d)}$, compared to $(\frac{2}{3})^d$ for random hyperplanes, and near-parallel vectors collide with similar asymptotic probabilities in both cases.

For c-approximate nearest neighbor searching, this translates to a decrease in the exponent ρ of locality-sensitive hashing (LSH) methods of a factor up to $\log_2(\pi) \approx 1.652$ compared to hyperplane LSH. For $c = 2$, we obtain $\rho \approx 0.302 + o(1)$ for hypercube LSH, improving upon the $\rho \approx 0.377$ for hyperplane LSH. We further describe how to use hypercube LSH in practice, and we consider an example application in the area of lattice algorithms.

1998 ACM Subject Classification F.2 Analysis of algorithms and problem complexity, G.3 Probability and statistics, H.3 Information storage and retrieval

Keywords and phrases (approximate) near neighbors, locality-sensitive hashing, large deviations, dimensionality reduction, lattice algorithms

Digital Object Identifier 10.4230/LIPIcs.MFCS.2017.7

1 Introduction

Finding (approximate) near neighbors. A key computational problem in various research areas, including machine learning, pattern recognition, data compression, coding theory, and cryptanalysis [34, 11, 15, 16, 29, 23], is finding near neighbors: given a data set $\mathcal{D} \subset \mathbb{R}^d$ of cardinality n, design a data structure and preprocess $\mathcal{D}$ in a way that, when given a query vector $\boldsymbol{q} \in \mathbb{R}^d$, one can efficiently find a near point to $\boldsymbol{q}$ in $\mathcal{D}$. Due to the "curse of dimensionality" [18] this problem is known to be hard to solve exactly (in the worst case) in high dimensions d, so a common relaxation of this problem is the (c, r)*-approximate near neighbor problem* ((c, r)-ANN): given that the nearest neighbor lies at distance at most r from $\boldsymbol{q}$, design an algorithm that finds an element $\boldsymbol{p} \in \mathcal{D}$ at distance at most $c \cdot r$ from $\boldsymbol{q}$.

Locality-sensitive hashing (LSH) and filtering (LSF). A prominent class of algorithms for finding near neighbors in high dimensions is formed by locality-sensitive hashing (LSH) [18] and locality-sensitive filtering (LSF) [9]. These solutions are based on partitioning the space

[*] The author is supported by the SNSF ERC Transfer Grant CRETP2-166734 FELICITY.

42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017).
Editors: Kim G. Larsen, Hans L. Bodlaender, and Jean-Francois Raskin; Article No. 7; pp. 7:1–7:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

into regions, in a way that nearby vectors have a higher probability of ending up in the same hash region than distant vectors. By carefully tuning (i) the number of hash regions per hash table, and (ii) the number of randomized hash tables, one can then guarantee that with high probability (a) nearby vectors will *collide* in at least one of the hash tables, and (b) distant vectors will not collide in any of the hash tables. For LSH, a simple lookup in all of $\boldsymbol{q}$'s hash buckets then provides a fast way of finding near neighbors to $\boldsymbol{q}$, while for LSF the lookups are slightly more involved. For various metrics, LSH and LSF currently provide the best performance in high dimensions [8, 9, 7, 13].

Near neighbors on the sphere. In this work we will focus on the near neighbor problem under the *angular distance*, where two vectors $\boldsymbol{x}, \boldsymbol{y}$ are considered nearby iff their common angle θ is small [12, 35, 33, 5]. This equivalently corresponds to near neighbor searching for the ℓ_2-norm, where the entire data set is assumed to lie on a sphere. A special case of (c, r)-ANN on the sphere, often considered in the literature, is the *random* case $r = \frac{1}{c}\sqrt{2}$ and $c \cdot r = \sqrt{2}$, in part due to a reduction from near neighbor under the Euclidean metric for general data sets to (c, r)-ANN on the sphere with these parameters [8].

1.1 Related work

Upper bounds. Perhaps the most well-known and widely used solution for ANN for the angular distance is Charikar's hyperplane LSH [12], where a set of *random* hyperplanes is used to partition the space into regions. Due to its low computational complexity and the simple form of the collision probabilities (with no hidden order terms in d), this method is easy to instantiate in practice and commonly achieves the best performance out of all LSH methods when d is not too large. For large d, both spherical cap LSH [6, 8] and cross-polytope LSH [36, 17, 5, 21] are known to perform better than hyperplane LSH. Experiments from [36, 37] showed that using *orthogonal* hyperplanes, partitioning the space into Voronoi regions induced by the vertices of a hypercube, also leads to superior results compared to hyperplane LSH; however, no theoretical guarantees for the resulting hypercube LSH method were given, and it remained unclear whether the improvement persists in high dimensions.

Lower bounds. For the case of *random* data sets, lower bounds have also been found, matching the performance of spherical cap and cross-polytope LSH for large c [30, 32, 5]. These lower bounds are commonly in a model where it is assumed that collision probabilities are "not too small", and in particular not exponentially small in d. Therefore it is not clear whether one can further improve upon cross-polytope LSH when the number of hash regions is exponentially large, which would for instance be the case for hypercube LSH. Together with the experimental results from [36, 37], this naturally begs the question: how efficient is hypercube LSH? Is it better than hyperplane LSH and/or cross-polytope LSH? And how does hypercube LSH compare to other methods in practice?

1.2 Contributions

Hypercube LSH. By carefully analyzing the collision probabilities for hypercube LSH using results from large deviations theory, we show that hypercube LSH is indeed different from, and superior to hyperplane LSH for large d. The following main theorem states the asymptotic form of the collision probabilities when using hypercube LSH, which are also visualized in Figure 1 in comparison with hyperplane LSH.

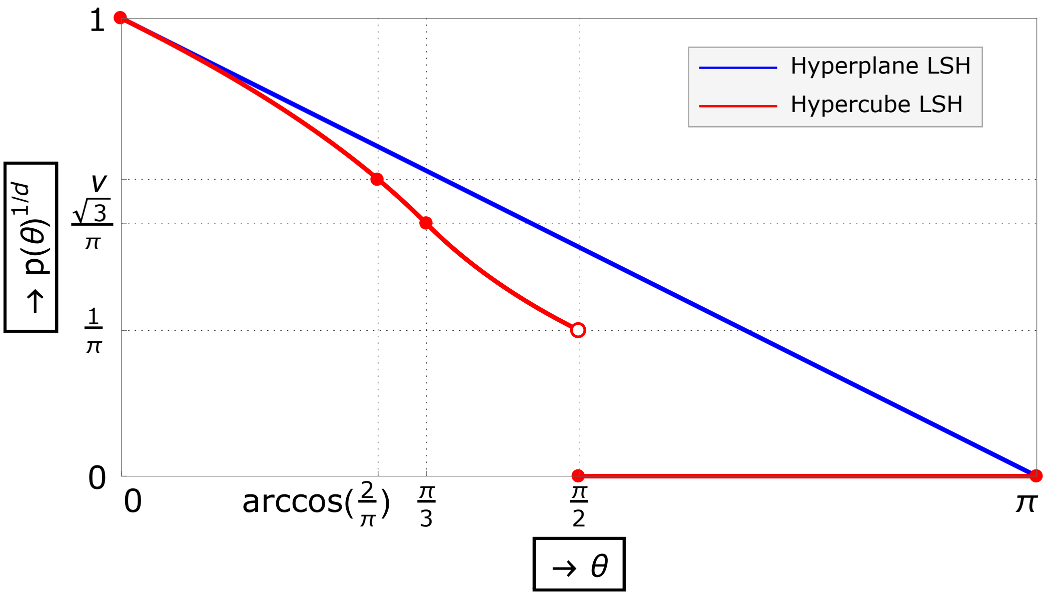

Figure 1 Asymptotics of collision probabilities for hypercube LSH, compared to hyperplane LSH. Here $\nu = \pi/(2\sqrt{\pi^2 - 4})$, and the dashed vertical lines correspond to boundary points of the piecewise parts of Theorem 1. The blue line indicates hyperplane LSH with d random hyperplanes.

▶ **Theorem 1** (Collision probabilities for hypercube LSH). *Let $\boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}(0,1)^d$, let $\theta \in [0, \pi]$ denote the angle between $\boldsymbol{X}$ and $\boldsymbol{Y}$, and let $p(\theta)$ denote the probability that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are mapped to the same hypercube hash region. For $\theta \in (0, \arccos \frac{2}{\pi})$ (respectively $\theta \in (\arccos \frac{2}{\pi}, \frac{\pi}{3})$), let $\beta_0 \in (1, \infty)$ (resp. $\beta_1 \in (1, \infty)$) be the unique solution to:*

$$\arccos\left(\frac{-1}{\beta_0}\right) = \frac{(\beta_0 - \cos\theta)\sqrt{\beta_0^2 - 1}}{\beta_0(\beta_0 \cos\theta - 1)}, \qquad \arccos\left(\frac{1}{\beta_1}\right) = \frac{(\beta_1 + \cos\theta)\sqrt{\beta_1^2 - 1}}{\beta_1(\beta_1 \cos\theta + 1)}. \qquad (1)$$

Then, as d tends to infinity, $p(\theta)$ satisfies:

$$p(\theta) = \begin{cases} \left(\dfrac{(\beta_0 - \cos\theta)^2}{\pi\beta_0(\beta_0\cos\theta - 1)\sin\theta}\right)^{d+o(d)}, & \text{if } \theta \in [0, \arccos\frac{2}{\pi}]; \\ \left(\dfrac{(\beta_1 + \cos\theta)^2}{\pi\beta_1(\beta_1\cos\theta + 1)\sin\theta}\right)^{d+o(d)}, & \text{if } \theta \in [\arccos\frac{2}{\pi}, \frac{\pi}{3}]; \\ \left(\dfrac{1 + \cos\theta}{\pi\sin\theta}\right)^{d+o(d)}, & \text{if } \theta \in [\frac{\pi}{3}, \frac{\pi}{2}); \\ 0, & \text{if } \theta \in [\frac{\pi}{2}, \pi]. \end{cases} \qquad (2)$$

Denoting the query complexity of LSH methods by $n^{\rho + o(1)}$, the parameter ρ for hypercube LSH is up to $\log_2(\pi) \approx 1.65$ times smaller than for hyperplane LSH. For large d, hypercube LSH is dominated by cross-polytope LSH (unless $c \cdot r > \sqrt{2}$), but as the convergence to the limit is rather slow, in practice either method might be better, depending on the exact parameter setting. For the random setting, Figure 2 shows limiting values for ρ for hyperplane, hypercube and cross-polytope LSH. We again remark that these are asymptotics for $d \to \infty$, and may not accurately reflect the performance of these methods for moderate d. We further briefly discuss how the hashing for hypercube LSH can be made efficient.

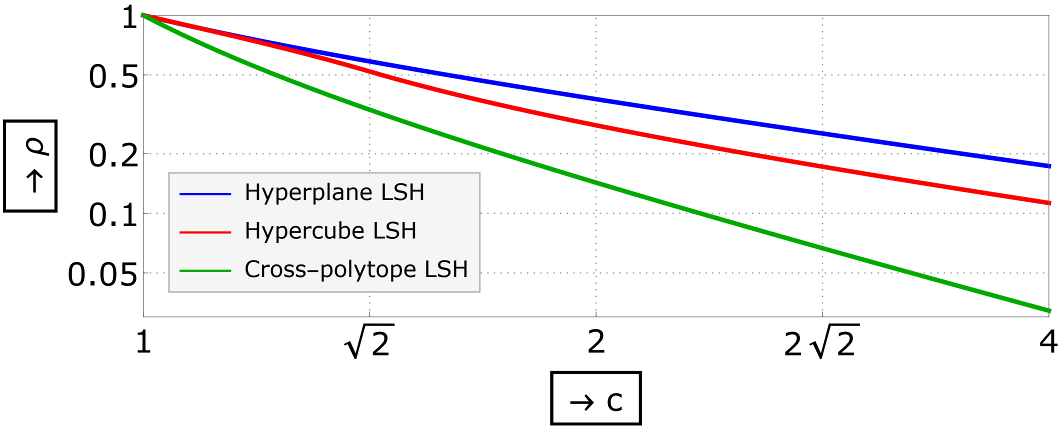

Figure 2 Asymptotics for the LSH exponent ρ when using hyperplane LSH, hypercube LSH, and cross-polytope LSH, for (c, r)-ANN with $c \cdot r = \sqrt{2}$. The curve for hyperplane LSH is exact for arbitrary d, while for the other two curves, order terms vanishing as $d \to \infty$ have been omitted.

Partial hypercube LSH. As the number of hash regions of a full-dimensional hypercube is often prohibitively large, we also consider *partial* hypercube LSH, where a d'-dimensional hypercube is used to partition a data set in dimension d. Building upon a result of Jiang [19], we characterize when hypercube and hyperplane LSH are asymptotically equivalent in terms of the relation between d' and d, and we empirically illustrate the convergence towards either hyperplane or hypercube LSH for larger d'. An important open problem remains to identify how large the ratio d'/d must be for the asymptotics of partial hypercube LSH to be equivalent to those of full-dimensional hypercube LSH.

Application to lattice sieving. Finally, we consider a specific use case of different LSH methods, in the context of lattice cryptanalysis. We show that the heuristic complexity of lattice sieving with hypercube LSH is expected to be slightly better than when using hyperplane LSH, and we discuss how experiments have previously indicated that in this application, hypercube LSH is superior to other dimensions up to dimensions $d \approx 80$.

2 Preliminaries

Notation. We denote probabilities with $\mathbb{P}(\cdot)$ and expectations with $\mathbb{E}(\cdot)$. Capital letters commonly denote random variables, and boldface letters denote vectors. We informally write $\mathbb{P}(X = x)$ for continuous X to denote the density of X at x. For probability distributions $\mathcal{D}$, we write $X \sim \mathcal{D}$ to denote that X is distributed according to $\mathcal{D}$. For sets S, with abuse of notation we further write $X \sim S$ to denote X is drawn uniformly at random from S. We write $\mathcal{N}(\mu, \sigma^2)$ for the normal distribution with mean μ and variance σ^2, and $\mathcal{H}(\mu, \sigma^2)$ for the distribution of $|X|$ when $X \sim \mathcal{N}(\mu, \sigma^2)$. For $\mu = 0$ the latter corresponds to the *half-normal* distribution. We write $\boldsymbol{X} \sim \mathcal{D}^d$ to denote a d-dimensional vector where each entry is independently distributed according to $\mathcal{D}$. In what follows, $\|\boldsymbol{x}\| = \sqrt{\sum_i x_i^2}$ denotes the Euclidean norm, and $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_i x_i y_i$ denotes the standard inner product. We denote the angle between two vectors by $\phi(\boldsymbol{x}, \boldsymbol{y}) = \arccos\langle \boldsymbol{x}/\|\boldsymbol{x}\|, \boldsymbol{y}/\|\boldsymbol{y}\|\rangle$.

▶ **Lemma 2** (Distribution of angles between random vectors [9, Lemma 2]). *Let $\boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}(0, 1)^d$ be two independent standard normal vectors. Then $\mathbb{P}(\phi(\boldsymbol{X}, \boldsymbol{Y}) = \theta) = (\sin \theta)^{d+o(d)}$.*

Locality-sensitive hashing. Locality-sensitive hash functions [18] are functions h mapping a d-dimensional vector $\boldsymbol{x}$ to a low-dimensional *sketch* $h(\boldsymbol{x})$, such that vectors which are nearby in $\mathbb{R}^d$ are more likely to be mapped to the same sketch than distant vectors. For the angular distance[1] $\phi(\boldsymbol{x}, \boldsymbol{y})$, we quantify a set of hash functions $\mathcal{H}$ as follows (see [18]):

▶ **Definition 3.** A hash family $\mathcal{H}$ is called $(\theta_1, \theta_2, p_1, p_2)$-sensitive if for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ we have:

- If $\phi(\boldsymbol{x}, \boldsymbol{y}) \leq \theta_1$ then $\mathbb{P}_{h \sim \mathcal{H}}(h(\boldsymbol{x}) = h(\boldsymbol{y})) \geq p_1$;
- If $\phi(\boldsymbol{x}, \boldsymbol{y}) \geq \theta_2$ then $\mathbb{P}_{h \sim \mathcal{H}}(h(\boldsymbol{x}) = h(\boldsymbol{y})) \leq p_2$.

The existence of locality-sensitive hash families implies the existence of fast algorithms for (approximate) near neighbors, as the following lemma describes[2]. For more details on the general principles of LSH, we refer the reader to e.g. [18, 4].

▶ **Lemma 4** (Locality-sensitive hashing [18]). *Suppose there exists a $(\theta_1, \theta_2, p_1, p_2)$-sensitive family $\mathcal{H}$. Let $\rho = \frac{\log(p_1)}{\log(p_2)}$. Then w.h.p. we can either find an element $\boldsymbol{p} \in L$ at angle at most θ_2 from $\boldsymbol{q}$, or conclude that no elements $\boldsymbol{p} \in L$ at angle at most θ_1 from $\boldsymbol{q}$ exist, in time $n^{\rho + o(1)}$ with space and preprocessing costs $n^{1+\rho+o(1)}$.*

Hyperplane LSH. For the angular distance, Charikar [12] introduced the hash family $\mathcal{H} = \{h_{\boldsymbol{a}} : \boldsymbol{a} \sim \mathcal{D}\}$ where $\mathcal{D}$ is any spherically symmetric distribution on $\mathbb{R}^d$, and $h_{\boldsymbol{a}}$ satisfies:

$$h_{\boldsymbol{a}}(\boldsymbol{x}) = \begin{cases} +1, & \text{if } \langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq 0; \\ -1, & \text{if } \langle \boldsymbol{a}, \boldsymbol{x} \rangle < 0. \end{cases} \tag{3}$$

The vector $\boldsymbol{a}$ can be interpreted as the normal vector of a random hyperplane, and the hash value depends on which side of the hyperplane $\boldsymbol{x}$ lies on. For this hash function, the probability of a collision is directly proportional to the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\mathbb{P}_{h \sim \mathcal{H}}\big(h(\boldsymbol{x}) = h(\boldsymbol{y})\big) = 1 - \frac{\phi(\boldsymbol{x}, \boldsymbol{y})}{\pi} . \tag{4}$$

For any two angles $\theta_1 < \theta_2$, the above family $\mathcal{H}$ is $(\theta_1, \theta_2, 1 - \frac{\theta_1}{\pi}, 1 - \frac{\theta_2}{\pi})$-sensitive.

Large deviations theory. Let $\{\boldsymbol{Z}_d\}_{d \in \mathbb{N}} \subset \mathbb{R}^k$ be a sequence of random vectors corresponding to an empirical mean, i.e. $\boldsymbol{Z}_d = \frac{1}{d} \sum_{i=1}^d \boldsymbol{U}_i$ with $\boldsymbol{U}_i$ i.i.d. We define the logarithmic moment generating function Λ of $\boldsymbol{Z}_d$ as:

$$\Lambda(\boldsymbol{\lambda}) = \ln \mathbb{E}_{\boldsymbol{U}_1} \left[\exp \langle \boldsymbol{\lambda}, \boldsymbol{U}_1 \rangle\right] . \tag{5}$$

Define $\mathcal{D}_\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^k : \Lambda(\boldsymbol{\lambda}) < \infty\}$. The *Fenchel-Legendre* transform of Λ is defined as:

$$\Lambda^*(\boldsymbol{z}) = \sup_{\boldsymbol{\lambda} \in \mathbb{R}^k} \left\{ \langle \boldsymbol{\lambda}, \boldsymbol{z} \rangle - \Lambda(\boldsymbol{\lambda}) \right\} . \tag{6}$$

The following result describes that under certain conditions on $\{\boldsymbol{Z}'_d\}$, the asymptotics of the probability measure on a set F are related to the function Λ^*.

▶ **Lemma 5** (Gärtner-Ellis theorem [14, Theorem 2.3.6 and Corollary 6.1.6]). *Let $\mathbf{0}$ be contained in the interior of $\mathcal{D}_\Lambda$, and let $\boldsymbol{Z}_d$ be an empirical mean. Then for arbitrary sets F,*

$$\lim_{d \to \infty} \tfrac{1}{d} \ln \mathbb{P}(\boldsymbol{z} \in F) = - \inf_{\boldsymbol{z} \in F} \Lambda^*(\boldsymbol{z}). \tag{7}$$

The latter statement can be read as $\mathbb{P}(\boldsymbol{z} \in F) = \exp(-d \inf_{\boldsymbol{z} \in F} \Lambda^*(\boldsymbol{z}) + o(d))$, and thus tells us exactly how $\mathbb{P}(\boldsymbol{z} \in F)$ scales as d tends to infinity, up to order terms.

[1] Formally speaking, the angular distance is only a similarity measure, and not a metric.

[2] Various conditions and order terms (which are commonly $n^{o(1)}$) are omitted here for brevity.

3 Hypercube LSH

In this section, we will analyze full-dimensional hypercube hashing, with hash family $\mathcal{H} = \{h_A : A \in SO(d)\}$ where $SO(d) \subset \mathbb{R}^{d \times d}$ denotes the rotation group, and h_A satisfies:

$$h_A(\boldsymbol{x}) = (h_1(A\boldsymbol{x}), \dots, h_d(A\boldsymbol{x})), \qquad h_i(\boldsymbol{x}) = \begin{cases} +1, & \text{if } x_i \geq 0; \\ -1, & \text{if } x_i < 0. \end{cases} \tag{8}$$

In other words, a hypercube hash function first applies a uniformly random rotation, and then maps the resulting vector to the orthant it lies in. This equivalently corresponds to a concatenation of d hyperplane hash functions, where all hyperplanes are orthogonal. Collision probabilities for prescribed angles θ between $\boldsymbol{x}$ and $\boldsymbol{y}$ are denoted by:

$$p(\theta) = \mathbb{P}(h_A(\boldsymbol{x}) = h_A(\boldsymbol{y}) \mid \phi(\boldsymbol{x}, \boldsymbol{y}) = \theta). \tag{9}$$

Above, the randomness is over $h_A \sim \mathcal{H}$, with $\boldsymbol{x}$ and $\boldsymbol{y}$ arbitrary vectors at angle θ (e.g. $\boldsymbol{x} = \boldsymbol{e}_1$ and $\boldsymbol{y} = \boldsymbol{e}_1 \cos\theta + \boldsymbol{e}_2 \sin\theta$). Alternatively, the random rotation A inside h_A may be omitted, and the probability can be computed over $\boldsymbol{X}, \boldsymbol{Y}$ drawn uniformly at random from a spherically symmetric distribution, *conditioned* on their common angle being θ.

3.1 Outline of the proof of Theorem 1

Although Theorem 1 is a key result, due to space restrictions we have decided to defer the full proof (approximately 5.5 pages) to the appendix. The approach of the proof can be summarized by the following four steps:

- Rewrite the collision probabilities in terms of (normalized) half-normal vectors $\boldsymbol{X}, \boldsymbol{Y}$;
- Introduce dummy variables x, y for the norms of these half-normal vectors, so that the probability can be rewritten in terms of unnormalized half-normal vectors;
- Apply the Gärtner-Ellis theorem (Lemma 5) to the three-dimensional vector $\boldsymbol{Z} = \frac{1}{d}(\sum_i X_i Y_i, \sum_i X_i^2, \sum_i Y_i^2)$ to compute the resulting probabilities for arbitrary x, y;
- Maximize the resulting expressions over $x, y > 0$ to get the final result.

The majority of the technical part of the proof lies in computing $\Lambda^*(\boldsymbol{z})$, which involves a somewhat tedious optimization of a multivariate function through a case-by-case analysis.

A note on Gaussian approximations. From the (above outline of the) proof, and the observation that the final optimization over x, y yields $x = y = 1$ as the optimum, one might wonder whether a simpler analysis might be possible by assuming (half-)normal vectors are already normalized. Such a computation however would only lead to an approximate solution, which is perhaps easiest to see by computing collision probabilities for $\theta = 0$. In the exact computation, where vectors are normalized, $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = 1$ implies $\boldsymbol{X} = \boldsymbol{Y}$. If however we do not take into account the norms of $\boldsymbol{X}$ and $\boldsymbol{Y}$, and do not condition on the norms being equal to 1, then $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = 1$ could also mean that $\boldsymbol{X}, \boldsymbol{Y}$ are slightly longer than 1 and have a small, non-zero angle. In fact, such a computation would indeed yield $p(\theta)^{1/d} \not\to 0$ as $\theta \to 0$.

3.2 Consequences of Theorem 1

From Theorem 1, we can draw several conclusions. Substituting values for θ, we can find asymptotics for $p(\theta)$, such as $p(\frac{\pi}{3})^{1/d} = \frac{\sqrt{3}}{\pi} + o(1)$ and $p(\frac{\pi}{2})^{1/d} = \frac{1}{\pi} + o(1)$. We observe that the limiting function of Theorem 1 (without the order terms) is continuous everywhere except at $\theta = \frac{\pi}{2}$. To understand the boundary $\theta = \arccos \frac{2}{\pi}$ of the piece-wise limit function, note that two (normalized) half-normal vectors $\boldsymbol{X}, \boldsymbol{Y}$ have expected inner product $\mathbb{E}\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \frac{2}{\pi}$.

LSH exponents ρ for random settings. Using Theorem 1, we can explicitly compute LSH exponents ρ for given angles θ_1 and θ_2 for large d. As an example, consider the random setting[3] with $c = \sqrt{2}$, corresponding to $\theta_2 = \frac{\pi}{2}$ and $\theta_1 = \frac{\pi}{3}$. Substituting the collision probabilities from Theorem 1, we get $\rho \to 1 - \frac{1}{2}\log_\pi(3) \approx 0.520$ as $d \to \infty$. To compare, if we had used random hyperplanes, we would have gotten a limiting value $\rho \to \log_2(\frac{3}{2}) \approx 0.585$. For the random case, Figure 2 compares limiting values ρ using random and orthogonal hyperplanes, and using the asymptotically superior cross-polytope LSH.

Scaling at $\theta \to 0$ and asymptotics of ρ for large c. For θ close to 0, by Theorem 1 we are in the regime defined by β_0. For $\cos\theta = 1 - \varepsilon$ with $\varepsilon > 0$ small, observe that $\beta_0 \approx 1$ satisfies $\beta_0 > 1/\cos\theta$. Computing a Taylor expansion around $\varepsilon = 0$, we eventually find $\beta_0 = 1 + \varepsilon + \frac{2\sqrt{2}}{\pi}\varepsilon^{3/2} + O(\varepsilon^2)$. Substituting this value β_0 into $p(\theta)$ with $\cos\theta = 1 - \varepsilon$, we find:

$$p(\theta) = \left(1 - \frac{\sqrt{2}}{\pi}\sqrt{\varepsilon} + O(\varepsilon)\right)^{d+o(d)}. \tag{10}$$

To compare this with hyperplane LSH, recall that the collision probability for d random hyperplanes is equal to $(1 - \frac{\theta}{\pi})^d$. Since $\cos\theta = 1 - \varepsilon$ translates to $\theta = \sqrt{2\varepsilon}(1 + O(\varepsilon))$, the collision probabilities for hyperplane hashing in this regime are also $(1 - \frac{\sqrt{2}}{\pi}\sqrt{\varepsilon} + O(\varepsilon))^d$. In other words, for angles $\theta \to 0$, the collision probabilities for hyperplane hashing and hypercube hashing are similar. This can also be observed in Figure 1. Based on this result, we further deduce that in random settings with large c, for hypercube LSH we have:

$$\rho \to \frac{\ln\left(1 - \frac{\sqrt{2}}{\pi c} + O\left(\frac{1}{c^2}\right)\right)}{\ln(1/\pi)} = \frac{\sqrt{2}}{\pi c \ln \pi} + O\left(\frac{1}{c^2}\right) \approx \frac{0.393}{c} + O\left(\frac{1}{c^2}\right). \tag{11}$$

For hyperplane LSH, the numerator is the same, while the denominator is $\ln(\frac{1}{2})$ instead of $\ln(\frac{1}{\pi})$, leading to values ρ which are a factor $\log_2 \pi + o(1) \approx 1.652 + o(1)$ larger. Both methods are inferior to cross-polytope LSH for large d, as there $\rho = O(1/c^2)$ for large c [5].

3.3 Convergence to the limit

To get an idea how hypercube LSH compares to other methods when d is not too large, we start by giving explicit collision probabilities for the first non-trivial case, namely $d = 2$.

▶ **Proposition 6** (Square LSH). *For $d = 2$, $p(\theta) = 1 - \frac{2\theta}{\pi}$ for $\theta \leq \frac{\pi}{2}$ and $p(\theta) = 0$ otherwise.*

Proof. In two dimensions, two randomly rotated vectors $\boldsymbol{X}, \boldsymbol{Y}$ at angle θ can be modeled as $\boldsymbol{X} = (\cos\psi, \sin\psi)$ and $\boldsymbol{Y} = (\cos(\psi+\theta), \sin(\psi+\theta))$ for $\psi \sim [0, 2\pi)$. The conditions $\boldsymbol{X}, \boldsymbol{Y} > 0$ are then equivalent to $\psi \in (0, \frac{\pi}{2}) \cap (-\theta, \frac{\pi}{2} - \theta)$, which for $\theta < \frac{\pi}{2}$ occurs with probability $\frac{\pi/2-\theta}{2\pi}$ over the randomness of ψ. As a collision can occur in any of the four quadrants, we finally multiply this probability by 4 to obtain the stated result. ◀

Figure 3 depicts $p(\theta)^{1/2}$ in green, along with hyperplane LSH (blue) and the asymptotics for hypercube LSH (red). For larger d, computing $p(\theta)$ exactly becomes more complicated,

[3] Here we assume that $c \cdot r \to (\sqrt{2})^-$, i.e. $c \cdot r$ approaches $\sqrt{2}$ from below. Alternatively, one might interpret this as that if distant points lie at distance $\sqrt{2} \pm o(1)$, then we might expect approximately half of them to lie at distance less than $\sqrt{2}$, with query complexity $O(n/2)^{\rho+o(1)} = n^{\rho+o(1)}$. If however $c \cdot r \geq \sqrt{2}$ then clearly $\rho = 0$, regardless of d and c.

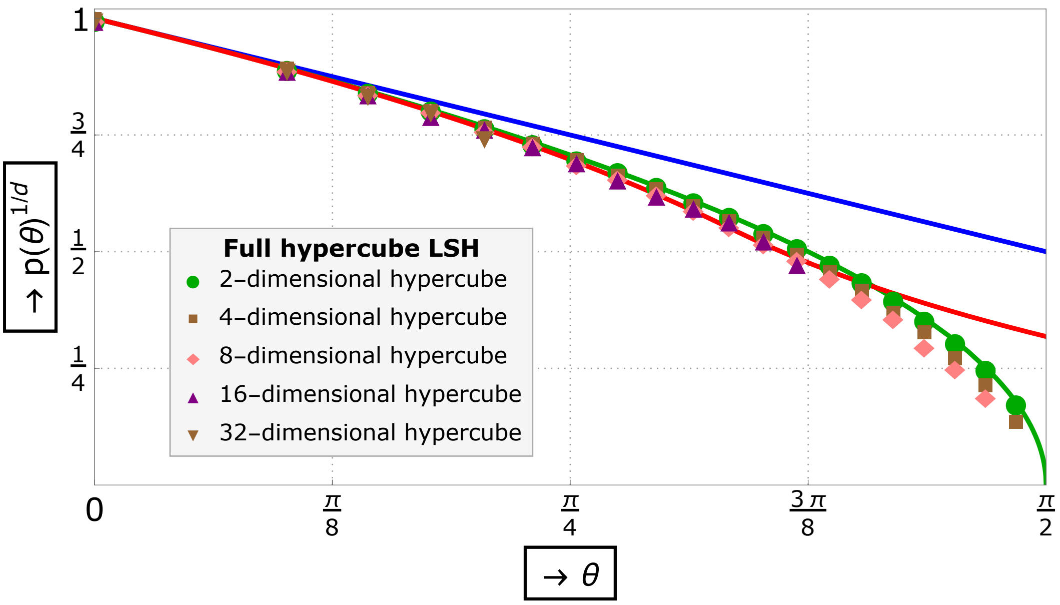

Figure 3 Empirical collision probabilities for hypercube LSH for small d. The green curve denotes the exact collision probabilities for $d = 2$ from Proposition 6.

and so instead we performed experiments to empirically obtain estimates for $p(\theta)$ as d increases. These estimates are also shown in Figure 3, and are based on 10^5 trials for each θ and d. Observe that as $\theta \to \frac{\pi}{2}$ and/or d grows larger, $p(\theta)$ decreases and the empirical estimates become less reliable. Points are omitted for cases where no successes occurred.

Based on these estimates and our intuition, we conjecture that (1) for $\theta \approx 0$, the scaling of $p(\theta)^{1/d}$ is similar for all d, and similar to the asymptotic behavior of Theorem 1; (2) the normalized collision probabilities for $\theta \approx \frac{\pi}{2}$ approach their limiting value from below; and (3) $p(\theta)$ is likely to be continuous for arbitrary d, implying that for $\theta \to \frac{\pi}{2}$, the collision probabilities tend to 0 for each d. These together suggest that values for ρ are actually *smaller* when d is small than when d is large, and the asymptotic estimate from Figure 2 might be pessimistic in practice. For the random setting, this would suggest that $\rho \approx 0$ regardless of c, as $p(\theta) \to 0$ as $\theta \to \frac{\pi}{2}$ for arbitrary d.

Comparison with hyperplane/cross-polytope LSH. Finally, [36, Figures 1 and 2] previously illustrated that among several LSH methods, the smallest values ρ (for their parameter sets) are obtained with hypercube LSH with $d = 16$, achieving smaller values ρ than e.g. cross-polytope LSH with $d = 256$. An explanation for this can be found in:

- The (conjectured) convergence of ρ to its limit from *below*, for hypercube LSH;
- The slow convergence of ρ to its limit (from above) for cross-polytope LSH[4].

This suggests that the actual values ρ for moderate dimensions d may well be smaller for hypercube LSH (and hyperplane LSH) than for cross-polytope LSH. Based on the limiting cases $d = 2$ and $d \to \infty$, we further conjecture that compared to hyperplane LSH, hypercube LSH achieves smaller values ρ for arbitrary d.

[4] [5, Theorem 1] shows that the leading term in the asymptotics for ρ scales as $\Theta(\ln d)$, with a first order term scaling as $O(\ln \ln d)$, i.e. a relative order term of the order $O(\ln \ln d / \ln d)$.

3.4 Fast hashing in practice

To further assess the practicality of hypercube LSH, recall that hashing is done as follows:

- Apply a uniformly random rotation A to $\boldsymbol{x}$;
- Look at the signs of $(A\boldsymbol{x})_i$.

Theoretically, a uniformly random rotation will be rather expensive to compute, with A being a real, dense matrix. As previously discussed in e.g. [3], it may suffice to only consider a sparse subset of all rotation matrices with a large enough amount of randomness, and as described in [5, 21] pseudo-random rotations may also be help speed up the computations in practice. As described in [21], this can even be made provable, to obtain a reduced $O(d \log d)$ computational complexity for applying a random rotation.

Finally, to compare this with cross-polytope LSH, note that cross-polytope LSH in dimension d partitions the space in $2d$ regions, as opposed to 2^d for hypercube hashing. To obtain a similar fine-grained partition of the space with cross-polytopes, one would have to concatenate $\Theta(d/\log d)$ random cross-polytope hashes, which corresponds to computing $\Theta(d/\log d)$ (pseudo-)random rotations, compared to only one rotation for hypercube LSH. We therefore expect hashing to be up to a factor $\Theta(d/\log d)$ less costly.

4 Partial hypercube LSH

Since a high-dimensional hypercube partitions the space in a large number of regions, for various applications one may only want to use hypercubes in a lower dimension $d' < d$. In those cases, one would first apply a random rotation to the data set, and then compute the hash based on the signs of the first d' coordinates of the rotated data set. This corresponds to the hash family $\mathcal{H} = \{h_{A,d'} : A \in SO(d)\}$, with $h_{A,d'}$ satisfying:

$$h_{A,d'}(\boldsymbol{x}) = (h_1(A\boldsymbol{x}), \dots, h_{d'}(A\boldsymbol{x})), \qquad h_i(\boldsymbol{x}) = \begin{cases} +1, & \text{if } x_i \geq 0; \\ -1, & \text{if } x_i < 0. \end{cases} \tag{12}$$

When “projecting” down onto the first d' coordinates, observe that distances and angles are distorted: the angle between the vectors formed by the first d' coordinates of $\boldsymbol{x}$ and $\boldsymbol{y}$ may not be the same as $\phi(\boldsymbol{x}, \boldsymbol{y})$. The amount of distortion depends on the relation between d' and d. Below, we will investigate how the collision probabilities $p_{d',d}(\theta)$ for partial hypercube LSH scale with d' and d, where $p_{d',d}(\theta) = \mathbb{P}(h(\boldsymbol{x}) = h(\boldsymbol{y}) \mid \phi(\boldsymbol{x}, \boldsymbol{y}) = \theta)$.

4.1 Convergence to hyperplane LSH

First, observe that for $d' = 1$, partial hypercube LSH is *equal* to hyperplane LSH, i.e. $p_{1,d}(\theta) = 1 - \frac{\theta}{\pi}$. For $1 < d' \ll d$, we first observe that both (partial) hypercube LSH and hyperplane LSH can be modeled by a projection onto d' dimensions:

- Hyperplane LSH: $\boldsymbol{x} \mapsto A\boldsymbol{x}$ with $A \sim \mathcal{N}(0,1)^{d' \times d}$;
- Hypercube LSH: $\boldsymbol{x} \mapsto (A^*)\boldsymbol{x}$ with $A \sim \mathcal{N}(0,1)^{d' \times d}$.

Here A^* denotes the matrix obtained from A after applying Gram-Schmidt orthogonalization to the rows of A. In both cases, hashing is done after the projection by looking at the signs of the projected vector. Therefore, the only difference lies in the projection, and one could ask: for which d', as a function of d, are these projections equivalent? When is a set of random hyperplanes already (almost) orthogonal?

This question was answered in [19]: if $d' = o(d/\log d)$, then $\max_{i,j} |A_{i,j} - A^*_{i,j}| \to 0$ in probability as $d \to \infty$ (implying $A^* = (1 + o(1))A$), while for $d' = \Omega(d/\log d)$ this maximum does not converge to 0 in probability. In other words, for large d a set of d' random hyperplanes in d dimensions is (approximately) orthogonal iff $d' = o(d/\log d)$.

▶ **Proposition 7** (Convergence to hyperplane LSH). *Let $p_{d',d}(\theta)$ denote the collision probabilities for partial hypercube LSH, and let $d' = o(d/\log d)$. Then $p_{d',d}(\theta)^{1/d'} \to 1 - \frac{\theta}{\pi}$.*

As $d' = \Omega(d/\log d)$ random vectors in d dimensions are asymptotically *not* orthogonal, in that case one might expect either convergence to full-dimensional hypercube LSH, or to something in between hyperplane and hypercube LSH.

4.2 Convergence to hypercube LSH

To characterize when partial hypercube LSH is equivalent to full hypercube LSH, we first observe that if d' is large compared to $\ln n$, then convergence to the hypercube LSH asymptotics follows from the Johnson-Lindenstrauss lemma.

▶ **Proposition 8** (Sparse data sets). *Let $d' = \omega(\ln n)$. Then the same asymptotics for the collision probabilities as those of full-dimensional hypercube LSH apply.*

Proof. Let $\theta \in (0, \frac{\pi}{2})$. By the Johnson-Lindenstrauss lemma [20], we can construct a projection $\boldsymbol{x} \mapsto A\boldsymbol{x}$ from d onto d' dimensions, preserving all pairwise distances up to a factor $1 \pm \varepsilon$ for $\varepsilon = \Theta((\ln n)/d') = o(1)$. For fixed $\theta \in (0, \frac{\pi}{2})$, this implies the angle ϕ between $A\boldsymbol{x}$ and $A\boldsymbol{y}$ will be in the interval $\theta \pm o(1)$, and so the collision probability lies in the interval $p(\theta \pm o(1))$. For large d, this means that the asymptotics of $p(\theta)$ are the same. ◀

To analyze collision probabilities for partial hypercube LSH when neither of the previous two propositions applies, note that through a series of transformations similar to those for full-dimensional hypercube LSH, it is possible to eventually end up with the following probability to compute, where $d_1 = d'$ and $d_2 = d - d'$:

$$\max_{x,y,u,v,\phi} \mathbb{P}\left(\frac{1}{d_1}\sum_{i=1}^{d_1} X_i Y_i = xy\cos\phi, \quad \frac{1}{d_1}\sum_{i=1}^{d_1} X_i^2 = x^2, \quad \frac{1}{d_1}\sum_{i=1}^{d_1} Y_i^2 = y^2, \right. \tag{13}$$

$$\left. \frac{1}{d_2}\sum_{i=1}^{d_2} U_i V_i = uvf(\phi,\theta), \quad \frac{1}{d_2}\sum_{i=1}^{d_2} U_i^2 = u^2, \quad \frac{1}{d_2}\sum_{i=1}^{d_2} V_i^2 = v^2 \right). \tag{14}$$

Here f is some function of ϕ and θ. The approach is comparable to how we ended up with a similar probability to compute in the proof of Theorem 1, except that we split the summation indices $I = [d]$ into two sets $I_1 = \{1, \ldots, d'\}$ of size d_1 and $I_2 = \{d'+1, \ldots, d\}$ of size d_2. We then substitute $U_i = X_{d'+i}$ and $V_i = Y_{d'+i}$, and add dummy variables x, y, u, v for the norms of the four partial vectors, and a dummy angle ϕ for the angle between the d_1-dimensional vectors, given the angle θ between the d-dimensional vectors.

Although the vector $\boldsymbol{Z}$ formed by the six random variables in (14) is not an empirical mean over a fixed number d of random vectors (the first three are over d_1 terms, the last three over d_2 terms), one may expect a similar large deviations result such as Lemma 5 to apply here. In that case, the function $\Lambda^*(\boldsymbol{z}) = \Lambda^*(z_1, \ldots, z_6)$ would be a function of six variables, which we would like to evaluate at $(xy\cos\phi, x^2, y^2, uvf(\phi,\theta), u^2, v^2)$. The function Λ^* itself involves an optimization (finding a supremum) over another six variables $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_6)$, so to compute collision probabilities for given d, d', θ exactly, using large deviations theory, one would have to compute an expression of the following form:

$$\min_{x,y,u,v,\phi} \left\{ \sup_{\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_5,\lambda_6} F_{d,d',\theta}(x,y,u,v,\phi,\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_5,\lambda_6) \right\}. \tag{15}$$

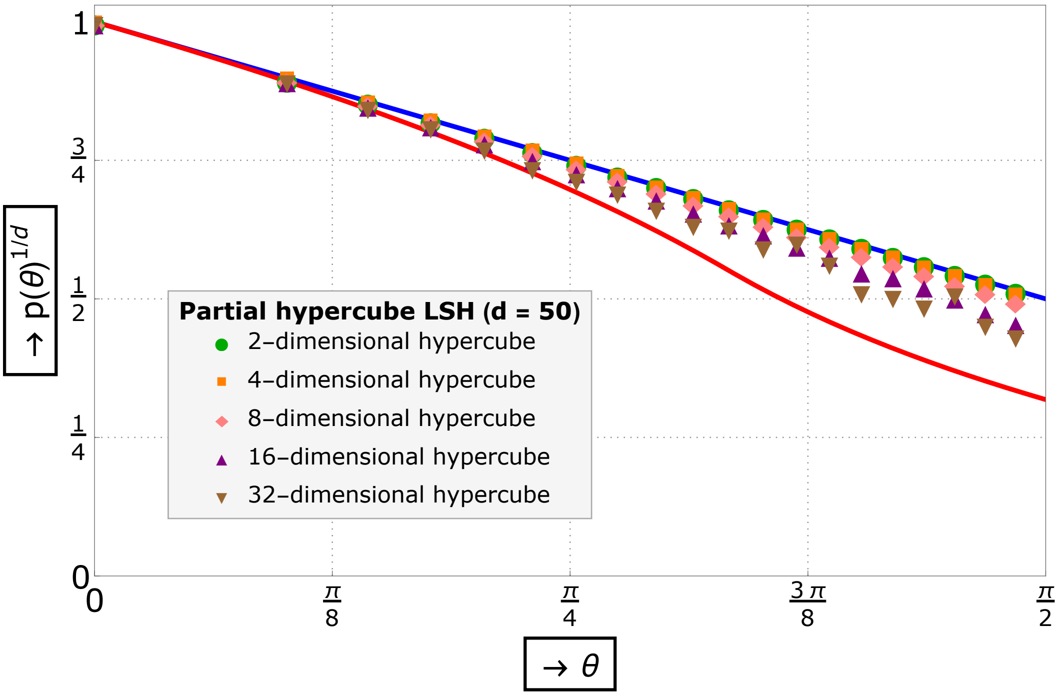

Figure 4 Experimental values of $p_{d',50}(\theta)^{1/d'}$, for different values d', compared with the asymptotics for hypercube LSH (red) and hyperplane LSH (blue).

As this is a very complex task, and the optimization will depend heavily on the parameters d, d', θ defined by the problem setting, we leave this optimization as an open problem. We only mention that intuitively, from the limiting cases of small and large d' we expect that depending on how d' scales with d (or n), we obtain a curve somewhere in between the two curves depicted in Figure 1.

4.3 Empirical collision probabilities

To get an idea of how $p_{d',d}(\theta)$ scales with d' in practice, we empirically computed several values for fixed $d = 50$. For fixed θ we then applied a least-squares fit of the form $e^{c_1 d + c_2}$ to the resulting data, and plotted e^{c_1} in Figure 4. These data points are again based on at least 10^5 experiments for each d' and θ. We expect that as d' increases, the collision probabilities slowly move from hyperplane hashing towards hypercube hashing, this can also be seen in the graph – for $d' = 2$, the least-squares fit is almost equal to the curve for hyperplane LSH, while as d' increases the curve slowly moves down towards the asymptotics for full hypercube LSH. Again, we stress that as d' becomes larger, the empirical estimates become less reliable, and so we did not consider even larger values for d'.

Compared to full hypercube LSH and Figure 3, we observe that we now approach the limit from above (although the fitted collision probabilities never seem to be smaller than those of hyperplane LSH), and therefore the values ρ for partial hypercube LSH are likely to lie in between those of hyperplane and (the asymptotics of) hypercube LSH.

5 Application: Lattice sieving for the shortest vector problem

We finally consider an explicit application for hypercube LSH, namely lattice sieving algorithms for the shortest vector problem. Given a basis $\boldsymbol{B} = \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_d\} \subset \mathbb{R}^d$ of a lattice $\mathcal{L}(\boldsymbol{B}) = \{\sum_i \lambda_i \boldsymbol{b}_i : \lambda_i \in \mathbb{Z}\}$, the shortest vector problem (SVP) asks to find a shortest non-zero vector in this lattice. Various different methods for solving SVP in high dimensions are known, and currently the algorithm with the best heuristic time complexity in high dimensions is based on lattice sieving, combined with nearest neighbor searching [9].

In short, lattice sieving works by generating a long list L of pairwise reduced lattice vectors, where $\boldsymbol{x}, \boldsymbol{y}$ are reduced iff $\|\boldsymbol{x} - \boldsymbol{y}\| \geq \min\{\|\boldsymbol{x}\|, \|\boldsymbol{y}\|\}$. The previous condition is equivalent to $\phi(\boldsymbol{x}, \boldsymbol{y}) \leq \frac{\pi}{3}$, and so the length of L can be bounded by the kissing constant in dimension d, which is conjectured to scale as $(4/3)^{d/2+o(d)}$. Therefore, if we have a list of size $n = (4/3)^{d/2+o(d)}$, any newly sampled lattice vector can be reduced against the list many times to obtain a very short lattice vector. The time complexity of this method is dominated by doing $\text{poly}(d) \cdot n$ reductions (searches for nearby vectors) with a list of size n. A linear search trivially leads to a heuristic complexity of $n^{2+o(1)} = (4/3)^{d+o(d)}$ (with space $n^{1+o(1)}$), while nearest neighbor techniques can reduce the time complexity to $n^{1+\rho+o(1)}$ for $\rho < 1$ (increasing the space to $n^{1+\rho+o(1)}$). For more details, see e.g. [31, 23, 9].

Based on the collision probabilities for hypercube LSH, and assuming the asymptotics for partial hypercube LSH (with $d' = O(d)$) are similar to those of full-dimensional hypercube LSH, we obtain the following result. An outline of the proof is given in the appendix.

▶ **Proposition 9** (Complexity of lattice sieving with hypercube LSH). *Suppose the asymptotics for full hypercube LSH also hold for partial hypercube LSH with $d' \approx 0.1335d$. Then lattice sieving with hypercube LSH heuristically solves SVP in time and space $2^{0.3222d+o(d)}$.*

As expected, the conjectured asymptotic performance of (sieving with) hypercube LSH lies in between those of hyperplane LSH and cross-polytope LSH.

Linear search [31]:	$2^{0.4150d+o(d)}$.
Hyperplane LSH [23]:	$2^{0.3366d+o(d)}$.
Hypercube LSH:	$2^{0.3222d+o(d)}$.
Spherical cap LSH [24]:	$2^{0.2972d+o(d)}$.
Cross-polytope LSH [10]:	$2^{0.2972d+o(d)}$.
Spherical LSF [9]:	$2^{0.2925d+o(d)}$.

In practice however, the picture is almost entirely reversed [1]. The lattice sieving method used to solve SVP in the highest dimension to date ($d = 116$) used a very optimized linear search [22]. The furthest that any nearest neighbor-based sieve has been able to go to date is $d = 107$, using hypercube LSH [27, 26][5]. Experiments further indicated that spherical LSF only becomes competitive with hypercube LSH as $d \gtrsim 80$ [9, 28], while sieving with cross-polytope LSH turned out to be rather slow compared to other methods [10, 25]. Although it remains unclear which nearest neighbor method is the "most practical" in the application of lattice sieving, hypercube LSH is one of the main contenders.

Acknowledgments. The author is indebted to Ofer Zeitouni for his suggestion to use results from large deviations theory, and for his many helpful comments regarding this application. The author further thanks Brendan McKay and Carlo Beenakker for their comments.

[5] Although phrased as hyperplane LSH, the implementations from [23, 27, 26] are using hypercube LSH.

References

1 SVP challenge, 2015. URL: http://latticechallenge.org/svp-challenge/.

2 Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Formulas*. Dover Publications, 1972. URL: http://people.math.sfu.ca/~cbm/aands/toc.htm.

3 Dimitris Achlioptas. Database-friendly random projections. In *PODS*, pages 274–281, 2001. doi:10.1145/375551.375608.

4 Alexandr Andoni. *Nearest Neighbor Search: the Old, the New, and the Impossible*. PhD thesis, Massachusetts Institute of Technology, 2009. URL: http://hdl.handle.net/1721.1/55090.

5 Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal LSH for angular distance. In *NIPS*, pages 1225–1233, 2015. URL: https://papers.nips.cc/paper/5893-practical-and-optimal-lsh-for-angular-distance.

6 Alexandr Andoni, Piotr Indyk, Huy Lê Nguyên, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *SODA*, pages 1018–1028, 2014. doi:10.1137/1.9781611973402.76.

7 Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *SODA*, pages 47–66, 2017. doi:10.1137/1.9781611974782.4.

8 Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *STOC*, pages 793–801, 2015. doi:10.1145/2746539.2746553.

9 Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA*, pages 10–24, 2016. doi:10.1137/1.9781611974331.ch2.

10 Anja Becker and Thijs Laarhoven. Efficient (ideal) lattice sieving using cross-polytope LSH. In *AFRICACRYPT*, pages 3–23, 2016. doi:10.1007/978-3-319-31517-1_1.

11 Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

12 Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388, 2002. doi:10.1145/509907.509965.

13 Tobias Christiani. A framework for similarity search with space-time tradeoffs using locality-sensitive filtering. In *SODA*, pages 31–46, 2017. doi:10.1137/1.9781611974782.3.

14 Amir Dembo and Ofer Zeitouni. *Large deviations techniques and applications (2nd edition)*. Springer, 2010. doi:10.1007/978-3-642-03311-7.

15 Moshe Dubiner. Bucketing coding and information theory for the statistical high-dimensional nearest-neighbor problem. *IEEE Transactions on Information Theory*, 56(8):4166–4179, Aug 2010. doi:10.1109/TIT.2010.2050814.

16 Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley, 2000.

17 Kave Eshghi and Shyamsundar Rajaram. Locality sensitive hash functions based on concomitant rank order statistics. In *KDD*, pages 221–229, 2008. doi:10.1145/1401890.1401921.

18 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613, 1998. doi:10.1145/276698.276876.

19 Tiefeng Jiang. How many entries of a typical orthogonal matrix can be approximated by independent normals? *The Annals of Probability*, 34(4):1497–1529, 2006. doi:10.1214/009117906000000205.

20 William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(1):189–206, 1984. doi:10.1090/conm/026/737400.

21 Christopher Kennedy and Rachel Ward. Fast cross-polytope locality-sensitive hashing. In *ITCS*, 2017. URL: `https://arxiv.org/abs/1602.06922`.

22 Thorsten Kleinjung. Private communication, 2014.

23 Thijs Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. In *CRYPTO*, pages 3–22, 2015. `doi:10.1007/978-3-662-47989-6_1`.

24 Thijs Laarhoven and Benne de Weger. Faster sieving for shortest lattice vectors using spherical locality-sensitive hashing. In *LATINCRYPT*, pages 101–118, 2015. `doi:10.1007/978-3-319-22174-8_6`.

25 Artur Mariano. Private communication., 2016.

26 Artur Mariano and Christian Bischof. Enhancing the scalability and memory usage of HashSieve on multi-core CPUs. In *PDP*, pages 545–552, 2016. `doi:10.1109/PDP.2016.31`.

27 Artur Mariano, Thijs Laarhoven, and Christian Bischof. Parallel (probable) lock-free HashSieve: a practical sieving algorithm for the SVP. In *ICPP*, pages 590–599, 2015. URL: `https://eprint.iacr.org/2015/041`.

28 Artur Mariano, Thijs Laarhoven, and Christian Bischof. A parallel variant of LDSieve for the SVP on lattices. *PDP*, 2017.

29 Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *EUROCRYPT*, pages 203–228, 2015. `doi:10.1007/978-3-662-46800-5_9`.

30 Rajeev Motwani, Assaf Naor, and Rina Panigrahy. Lower bounds on locality sensitive hashing. *SIAM Journal of Discrete Mathematics*, 21(4):930–935, 2007. `doi:10.1137/050646858`.

31 Phong Q. Nguyên and Thomas Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2(2):181–207, 2008. `doi:10.1515/JMC.2008.009`.

32 Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality sensitive hashing (except when q is tiny). In *ICS*, pages 276–283, 2011. URL: `http://conference.itcs.tsinghua.edu.cn/ICS2011/content/papers/2.html`.

33 Ludwig Schmidt, Matthew Sharifi, and Ignacio Lopez-Moreno. Large-scale speaker identification. In *ICASSP*, pages 1650–1654, 2014. `doi:10.1109/ICASSP.2014.6853878`.

34 Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice.* MIT Press, 2005. URL: `http://ttic.uchicago.edu/~gregory/annbook/book.html`.

35 Narayanan Sundaram, Aizana Turmukhametova, Nadathur Satish, Todd Mostak, Piotr Indyk, Samuel Madden, and Pradeep Dubey. Streaming similarity search over one billion tweets using parallel locality-sensitive hashing. *VLDB*, 6(14):1930–1941, 2013. `doi:10.14778/2556549.2556574`.

36 Kengo Terasawa and Yuzuru Tanaka. Spherical LSH for approximate nearest neighbor search on unit hypersphere. In *WADS*, pages 27–38, 2007. `doi:10.1007/978-3-540-73951-7_4`.

37 Kengo Terasawa and Yuzuru Tanaka. Approximate nearest neighbor search for a dataset of normalized vectors. In *IEICE Transactions on Information and Systems*, volume 92, pages 1609–1619, 2009. URL: `http://search.ieice.org/bin/summary.php?id=e92-d_9_1609`.

A Proof of Theorem 1

Theorem 1 will be proved through a series of lemmas, each making partial progress towards a final solution. Reading only the claims made in the lemmas may give the reader an idea how the proof is built up. Before starting the proof, we begin with a useful lemma regarding integrals of (exponentials of) quadratic forms.

▶ **Lemma 10** (Integrating an exponential of a quadratic form in the positive quadrant). *Let $a, b, c \in \mathbb{R}$ with $a, c < 0$ and $D = b^2 - 4ac < 0$. Then:*

$$\int_0^\infty \int_0^\infty \exp(ax^2 + bxy + cy^2)\, dx\, dy = \frac{\pi + 2\arctan\left(\frac{b}{\sqrt{-D}}\right)}{2\sqrt{-D}}\,. \tag{16}$$

Proof. The proof below is based on substituting $y = xs$ (and $dy = x\,ds$) before computing the integral over x. An integral over $1/(a + bs + cs^2)$ then remains, which leads to the arctangent solution in case $b^2 < 4ac$.

$$I = \int_{y=0}^\infty \int_0^\infty \exp(ax^2 + bxy + cy^2)\, dx\, dy \tag{17}$$

$$= \int_{s=0}^\infty \left(\int_0^\infty x \exp\left((a + bs + cs^2)x^2\right)\, dx \right) ds \tag{18}$$

$$= \int_0^\infty \left[\frac{\exp\left((a + bs + cs^2)x^2\right)}{2(a + bs + cs^2)} \right]_{x=0}^\infty ds \tag{19}$$

$$= \int_0^\infty \left[0 - \frac{1}{2(a + bs + cs^2)} \right] ds \tag{20}$$

$$= \frac{-1}{2} \int_0^\infty \frac{1}{a + bs + cs^2}\, ds. \tag{21}$$

The last equality used the assumptions $a, c < 0$ and $b^2 < 4ac$ so that $a + bs + cs^2 < 0$ for all $s > 0$. We then solve the last remaining integral (see e.g. [2, Equation (3.3.16)]) to obtain:

$$I = \frac{-1}{2} \left[\frac{2}{\sqrt{4ac - b^2}} \arctan\left(\frac{b + 2cs}{\sqrt{4ac - b^2}} \right) \right]_{s=0}^\infty \tag{22}$$

$$= \frac{-1}{2\sqrt{4ac - b^2}} \left(-\pi - 2\arctan\left(\frac{b}{\sqrt{4ac - b^2}} \right) \right). \tag{23}$$

Eliminating minus signs and substituting $D = b^2 - 4ac$, we obtain the stated result. ◀

Next, we begin by restating the collision probability between two vectors in terms of half-normal vectors.

▶ **Lemma 11** (Towards three-dimensional large deviations). *Let $\mathcal{H}$ denote the hypercube hash family in d dimensions, and as before, let p be defined as:*

$$p(\theta) = \mathbb{P}_{h \sim \mathcal{H}}(h(\boldsymbol{x}) = h(\boldsymbol{y}) \mid \phi(\boldsymbol{x}, \boldsymbol{y}) = \theta). \tag{24}$$

Let $\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}} \sim \mathcal{H}(0,1)^d$ and let the sequence $\{\boldsymbol{Z}_d\}_{d \in \mathbb{N}} \subset \mathbb{R}^3$ be defined as:

$$\boldsymbol{Z}_d = \frac{1}{d} \left(\sum_{i=1}^d \hat{X}_i \hat{Y}_i,\, \sum_{i=1}^d \hat{X}_i^2,\, \sum_{i=1}^d \hat{Y}_i^2 \right). \tag{25}$$

Then:

$$p(\theta) = \left(\frac{1}{2\sin\theta} \right)^{d + o(d)} \max_{x,y>0} \mathbb{P}(\boldsymbol{Z}_d = (xy\cos\theta,\, x^2,\, y^2)). \tag{26}$$

Proof. First, we write out the definition of the conditional probability in p, and use the fact that each of the 2^d hash regions (orthants) has the same probability mass. Here

$\boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}(0,1)^d$ denote random Gaussian vectors, and subscripts denoting what probabilities are computed over are omitted when implicit.

$$p(\theta) = \mathbb{P}_{h \sim \mathcal{H}}(h(\boldsymbol{x}) = h(\boldsymbol{y}) \mid \phi(\boldsymbol{x},\boldsymbol{y}) = \theta) \tag{27}$$

$$= 2^d \cdot \mathbb{P}_{\boldsymbol{X},\boldsymbol{Y} \sim \mathcal{N}(0,1)^d}(\boldsymbol{X} > 0, \boldsymbol{Y} > 0 \mid \phi(\boldsymbol{X},\boldsymbol{Y}) = \theta) \tag{28}$$

$$= \frac{2^d \cdot \mathbb{P}(\boldsymbol{X} > 0, \boldsymbol{Y} > 0, \phi(\boldsymbol{X},\boldsymbol{Y}) = \theta)}{\mathbb{P}(\phi(\boldsymbol{X},\boldsymbol{Y}) = \theta)} . \tag{29}$$

By Lemma 2, the denominator is equal to $(\sin\theta)^{d+o(d)}$. The numerator of (29) can further be rewritten as a conditional probability on $\{\boldsymbol{X} > 0, \boldsymbol{Y} > 0\}$, multiplied with $\mathbb{P}(\boldsymbol{X} > 0, \boldsymbol{Y} > 0) = 2^{-2d}$. To incorporate the conditionals $\boldsymbol{X}, \boldsymbol{Y} > 0$, we replace $\boldsymbol{X}, \boldsymbol{Y} \sim \mathcal{N}(0,1)^d$ by half-normal vectors $\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}} \sim \mathcal{H}(0,1)^d$, resulting in:

$$p(\theta) = \frac{\mathbb{P}_{\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}} \sim \mathcal{H}(0,1)^d}(\phi(\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}}) = \theta)}{(2\sin\theta)^{d+o(d)}} = \frac{q(\theta)}{(2\sin\theta)^{d+o(d)}} . \tag{30}$$

To incorporate the normalization over the (half-normal) vectors $\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{Y}}$, we introduce dummy variables x, y corresponding to the norms of $\hat{\boldsymbol{X}}/\sqrt{d}$ and $\hat{\boldsymbol{Y}}/\sqrt{d}$, and observe that as the probabilities are exponential in d, the integrals will be dominated by the maximum value of the integrand in the given range:

$$q(\theta) = \int_0^\infty \int_0^\infty \mathbb{P}(\langle \hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}} \rangle = x\, y\, d\, \cos\theta, \|\hat{\boldsymbol{X}}\|^2 = x^2 d, \|\hat{\boldsymbol{Y}}\|^2 = y^2 d)\, dx\, dy \tag{31}$$

$$= 2^{o(d)} \max_{x,y>0} \mathbb{P}\Big(\langle \hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}} \rangle = x\, y\, d\, \cos\theta, \|\hat{\boldsymbol{X}}\|^2 = x^2 d, \|\hat{\boldsymbol{Y}}\|^2 = y^2 d\Big) . \tag{32}$$

Substituting $\boldsymbol{Z}_d = \frac{1}{d}(\langle \hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}} \rangle, \|\hat{\boldsymbol{X}}\|^2, \|\hat{\boldsymbol{Y}}\|^2)$, we obtain the claimed result. ◀

Note that Z_1, Z_2, Z_3 are pairwise but not jointly independent. To compute the density of $\boldsymbol{Z}_d$ at $(xy\cos\theta, x^2, y^2)$ for $d \to \infty$, we use the Gärtner-Ellis theorem stated in Lemma 5.

▶ **Lemma 12** (Applying the Gärtner-Ellis theorem to $\boldsymbol{Z}_d$)**.** *Let $\{\boldsymbol{Z}_d\}_{d \in \mathbb{N}} \subset \mathbb{R}^3$ as in Lemma 11, and let Λ and Λ^* as in Section 2. Then $\mathbf{0}$ lies in the interior of $\mathcal{D}_\Lambda$, and therefore*

$$\mathbb{P}(\boldsymbol{Z}_d = (xy\cos\theta,\, x^2,\, y^2)) = \exp\left(-\Lambda^*(xy\cos\theta, x^2, y^2)d + o(d)\right) . \tag{33}$$

Essentially, all that remains now is computing Λ^* at the appropriate point $\boldsymbol{z}$. To continue, we first compute the logarithmic moment generating function $\Lambda = \Lambda_d$ of $\boldsymbol{Z}_d$:

▶ **Lemma 13** (Computing Λ)**.** *Let $\boldsymbol{Z}_d$ as before, and let $D = D(\lambda_1, \lambda_2, \lambda_3) = \lambda_1^2 - (1 - 2\lambda_2)(1 - 2\lambda_3)$. Then for $\boldsymbol{\lambda} \in \mathcal{D}_\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^3 : \lambda_2, \lambda_3 < \frac{1}{2}, D < 0\}$ we have:*

$$\Lambda(\boldsymbol{\lambda}) = \ln\left(\pi + 2\arctan\left(\frac{\lambda_1}{\sqrt{-D}}\right)\right) - \ln\pi - \tfrac{1}{2}\ln(-D). \tag{34}$$

Proof. By the definition of the LMGF, we have:

$$\Lambda(\boldsymbol{\lambda}) = \ln \mathbb{E}_{\hat{X}_1, \hat{Y}_1 \sim \mathcal{H}(0,1)}\left[\exp\left(\lambda_1 \hat{X}_1 \hat{Y}_1 + \lambda_2 \hat{X}_1^2 + \lambda_3 \hat{Y}_1^2\right)\right] . \tag{35}$$

We next compute the inner expectation over the random variables $\hat{X}_1, \hat{Y}_1$, by writing out the double integral over the product of the argument with the densities of $\hat{X}_1$ and $\hat{Y}_1$.

$$\mathbb{E}_{X_1,Y_1}\left[\exp\left(\lambda_1 X_1 Y_1 + \lambda_2 X_1^2 + \lambda_3 Y_1^2\right)\right] \tag{36}$$

$$= \int_0^\infty \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x^2}{2}\right) dx \int_0^\infty \sqrt{\frac{2}{\pi}} \exp\left(-\frac{y^2}{2}\right) dy \exp\left(\lambda_1 xy + \lambda_2 x^2 + \lambda_3 y^2\right) \tag{37}$$

$$= \frac{2}{\pi} \int_0^\infty \int_0^\infty \exp\left(\lambda_1 xy + \left(\lambda_2 - \tfrac{1}{2}\right) x^2 + \left(\lambda_3 - \tfrac{1}{2}\right) y^2\right)\, dx\, dy\, . \tag{38}$$

Applying Lemma 10 with $(a,b,c) = (\lambda_2 - \frac{1}{2}, \lambda_1, \lambda_3 - \frac{1}{2})$ yields the claimed expression for Λ, as well as the bounds stated in $\mathcal{D}_\Lambda$ which are necessary for the expectation to be finite. ◀

We now continue with computing the Fenchel-Legendre transform of Λ, which involves a rather complicated maximization (supremum) over $\boldsymbol{\lambda} \in \mathbb{R}^3$. The following lemma makes a first step towards computing this supremum.

▶ **Lemma 14** (Computing $\Lambda^*(\boldsymbol{z})$ – General form). *Let $\boldsymbol{z} \in \mathbb{R}^3$ such that $z_2, z_3 > 0$. Then the Fenchel-Legendre transform Λ^* of Λ at $\boldsymbol{z}$ satisfies*

$$\Lambda^*(\boldsymbol{z}) = \ln \pi + \sup_{\substack{\lambda_1, \beta \\ \beta > 1}} \left\{ \frac{z_2}{2} + \frac{z_3}{2} + \lambda_1 z_1 - |\lambda_1| \beta \sqrt{z_2 z_3} + \frac{1}{2} \ln(\beta^2 - 1) + \ln |\lambda_1| \right. \tag{39}$$

$$\left. - \ln \left(\pi + 2 \arctan \left(\frac{\lambda_1}{|\lambda_1| \sqrt{\beta^2 - 1}} \right) \right) \right\}. \tag{40}$$

Proof. First, we recall the definition of Λ^* and substitute the previous expression for Λ:

$$\Lambda^*(\boldsymbol{z}) = \sup_{\boldsymbol{\lambda} \in \mathbb{R}^3} \left\{ \langle \boldsymbol{\lambda}, \boldsymbol{z} \rangle - \Lambda(\boldsymbol{\lambda}) \right\} \tag{41}$$

$$= \ln \pi + \sup_{\boldsymbol{\lambda} \in \mathbb{R}^3} \left\{ \langle \boldsymbol{\lambda}, \boldsymbol{z} \rangle + \ln \sqrt{-D} - \ln \left(\pi + 2 \arctan \left(\frac{\lambda_1}{\sqrt{-D}} \right) \right) \right\}. \tag{42}$$

Here as before $D = \lambda_1^2 - (1 - 2\lambda_2)(1 - 2\lambda_3) < 0$. Let the argument of the supremum above be denoted by $f(\boldsymbol{z}, \boldsymbol{\lambda})$. We make a change of variables by setting $t_2 = 1 - 2\lambda_2 > 0$ and $t_3 = 1 - 2\lambda_3 > 0$, so that D becomes $D = \lambda_1^2 - t_2 t_3 < 0$:

$$f(\boldsymbol{z}, \lambda_1, t_2, t_3) = \frac{z_2}{2} + \frac{z_3}{2} + \lambda_1 z_1 - \frac{t_2 z_2}{2} - \frac{t_3 z_3}{2} \tag{43}$$

$$+ \tfrac{1}{2} \ln(t_2 t_3 - \lambda_1^2) - \ln \left(\pi + 2 \arctan \left(\tfrac{\lambda_1}{\sqrt{t_2 t_3 - \lambda_1^2}} \right) \right). \tag{44}$$

We continue by making a further change of variables $u = t_2 t_3 > \lambda_1^2$ so that $t_2 = u / t_3$. As a result the dependence of f on t_3 is only through the fourth and fifth terms above, from which one can easily deduce that the supremum over t_3 occurs at $t_3 = \sqrt{u z_2 / z_3}$. This also implies that $t_2 = \sqrt{u z_3 / z_2}$. Substituting these values for t_2, t_3, we obtain:

$$f(\boldsymbol{z}, \lambda_1, u) = \tfrac{z_2}{2} + \tfrac{z_3}{2} + \lambda_1 z_1 - \sqrt{u z_2 z_3} + \tfrac{1}{2} \ln(u - \lambda_1^2) - \ln \left(\pi + 2 \arctan \left(\tfrac{\lambda_1}{\sqrt{u - \lambda_1^2}} \right) \right).$$

Finally, we use the substitution $u = \beta^2 \cdot \lambda_1^2$. From $D < 0$ it follows that $u / \lambda_1^2 = \beta > 1$. This substitution and some rewriting of f leads to the claimed result. ◀

The previous simplifications were regardless of z_1, z_2, z_3, where the only assumption that was made during the optimization of t_3 was that $z_2, z_3 > 0$. In our application, we want to compute Λ^* at $\boldsymbol{z} = (xy \cos \theta, x^2, y^2)$ for certain $x, y > 0$ and $\theta \in (0, \frac{\pi}{2})$. Substituting these values for $\boldsymbol{z}$, the expression from Lemma 11 becomes:

$$\Lambda^*(xy \cos \theta, x^2, y^2) = \ln \pi + \frac{x^2}{2} + \frac{y^2}{2} + \sup_{\substack{\lambda_1, \beta \\ \beta > 1}} \left\{ (\lambda_1 \cos \theta - |\lambda_1| \beta) xy + \frac{1}{2} \ln(\beta^2 - 1) \right. \tag{45}$$

$$\left. + \ln |\lambda_1| - \ln \left(\pi + 2 \arctan \left(\frac{\lambda_1}{|\lambda_1| \sqrt{\beta^2 - 1}} \right) \right) \right\}. \tag{46}$$

The remaining optimization over λ_1, β now takes slightly different forms depending on whether $\lambda_1 < 0$ or $\lambda_1 > 0$. We will tackle these two cases separately, based on the identity:

$$\Lambda^*(\boldsymbol{z}) = \max\left\{ \sup_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^3 \\ \lambda_1>0}} \{\langle \boldsymbol{\lambda}, \boldsymbol{z}\rangle - \Lambda(\boldsymbol{\lambda})\}\,,\ \sup_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^3 \\ \lambda_1<0}} \{\langle \boldsymbol{\lambda}, \boldsymbol{z}\rangle - \Lambda(\boldsymbol{\lambda})\} \right\} = \max\{\Lambda^*_+(\boldsymbol{z}), \Lambda^*_-(\boldsymbol{z})\}.$$

▶ **Lemma 15** (Computing $\Lambda^*(\boldsymbol{z})$ for positive λ_1)**.** *Let $\boldsymbol{z} = (xy\cos\theta, x^2, y^2)$ with $x, y > 0$ and $\theta \in (0, \frac{\pi}{2})$. For $\theta \in (0, \arccos\frac{2}{\pi})$, let $\beta_0 = \beta_0(\theta) \in (1, \infty)$ be the unique solution to* (1)*. Then the Fenchel-Legendre transform Λ^* at $\boldsymbol{z}$, restricted to $\lambda_1 > 0$, satisfies*

$$\Lambda^*_+(\boldsymbol{z}) = \frac{x^2}{2} + \frac{y^2}{2} - 1 - \ln(xy) + \begin{cases} \ln\left(\dfrac{\pi\beta_0(\beta_0\cos\theta - 1)}{2(\beta_0 - \cos\theta)^2}\right), & \text{if } \theta \in (0, \arccos\frac{2}{\pi}); \\ 0, & \text{if } \theta \in [\arccos\frac{2}{\pi}, \frac{\pi}{2}). \end{cases} \tag{47}$$

Proof. Substituting $\lambda_1 > 0$ into (46), we obtain:

$$\Lambda^*_+(xy\cos\theta, x^2, y^2) = \ln\pi + \frac{x^2}{2} + \frac{y^2}{2} + \sup_{\substack{\lambda_1>0 \\ \beta>1}} \Big\{ g_+(\lambda_1, \beta) \Big\}, \tag{48}$$

$$g_+(\lambda_1, \beta) = (\cos\theta - \beta)\lambda_1 xy + \frac{\ln(\beta^2-1)}{2} + \ln\lambda_1 - \ln\Big(\pi + 2\arctan\Big(\frac{1}{\sqrt{\beta^2-1}}\Big)\Big). \tag{49}$$

Differentiating w.r.t. λ_1 gives $(\cos\theta - \beta)xy + \frac{1}{\lambda_1}$. Recall that $\beta > 1 > \cos\theta$. For $\lambda_1 \to 0^+$ the derivative is therefore positive, for $\lambda_1 \to \infty$ it is negative, and there is a global maximum at the only root $\lambda_1 = 1/((\beta - \cos\theta)xy)$. In that case, the expression further simplifies and we can pull out more terms that do not depend on β, to obtain:

$$\Lambda^*_+(xy\cos\theta, x^2, y^2) = \ln\pi + \frac{x^2}{2} + \frac{y^2}{2} - 1 - \ln(xy) + \sup_{\beta>1} \Big\{ g_+(\beta) \Big\}, \tag{50}$$

$$g_+(\beta) = \ln\left(\frac{\sqrt{\beta^2-1}}{(\beta - \cos\theta)\left(\pi + 2\arcsin\frac{1}{\beta}\right)} \right) = \ln h_+(\beta). \tag{51}$$

Here we used the identity $\arctan(1/\sqrt{\beta^2-1}) = \arcsin(1/\beta)$. Now, for $\beta \to 1^+$ we have $h_+(\beta) \to 0^+$, while for $\beta \to \infty$, we have

$$h_+(\beta) = \frac{1}{\pi} + \frac{1}{\pi\beta}\left(\cos\theta - \frac{2}{\pi}\right) + O\left(\frac{1}{\beta^2}\right). \tag{52}$$

In other words, if $\cos\theta \le \frac{2}{\pi}$ or $\theta \ge \arccos\frac{2}{\pi}$, we have $h_+(\beta) \to (\frac{1}{\pi})^-$ (the second order term is negative for $\cos\theta = \frac{2}{\pi}$), while for $\theta < \arccos\frac{2}{\pi}$ we approach the same limit from above as $h_+(\beta) \to (\frac{1}{\pi})^+$. For $\theta < \arccos\frac{2}{\pi}$ there is a non-trivial maximum at some value $\beta = \beta_0 \in (1, \infty)$, while for $\theta \ge \arccos\frac{2}{\pi}$, we can see from the derivative $h'_+(\beta)$ that $h_+(\beta)$ is strictly increasing on $(1, \infty)$, and the supremum is attained at $\beta \to \infty$. We therefore obtain two different results, depending on whether $\theta < \arccos\frac{2}{\pi}$ or $\theta \ge \arccos\frac{2}{\pi}$.

Case 1: $\arccos\frac{2}{\pi} \le \theta < \frac{\pi}{2}$. The supremum is attained in the limit of $\beta \to \infty$, which leads to $h_+(\beta) \to \frac{1}{\pi}$ and the stated expression for $\Lambda^*_+(xy\cos\theta, x^2, y^2)$.

Case 2: $0 < \theta < \arccos\frac{2}{\pi}$. In this case there is a non-trivial maximum at some value $\beta = \beta_0$, namely there where the derivative $h'_+(\beta_0) = 0$. After computing the derivative, eliminating the (positive) denominator and rewriting, this condition is equivalent to (1). This allows us to rewrite g and Λ^* in terms of β_0, by substituting the given expression for $\arcsin\left(\frac{1}{\beta_0}\right)$, which ultimately leads to the stated formula for Λ^*_+. ◀

▶ **Lemma 16** (Computing $\Lambda^*(\boldsymbol{z})$ for negative λ_1). *Let $\boldsymbol{z} = (xy\cos\theta, x^2, y^2)$ with $x, y > 0$ and $\theta \in (0, \frac{\pi}{2})$. For $\theta \in (\arccos\frac{2}{\pi}, \frac{\pi}{3})$, let $\beta_1 \in (1, \infty)$ be the unique solution to (1). Then the Fenchel-Legendre transform Λ^* at $\boldsymbol{z}$, restricted to $\lambda_1 < 0$, satisfies*

$$\Lambda^*_-(\boldsymbol{z}) = \frac{x^2}{2} + \frac{y^2}{2} - 1 - \ln(xy) + \begin{cases} 0, & \textit{if } \theta \in (0, \arccos\frac{2}{\pi}]; \\ \ln\left(\dfrac{\pi\beta_1(\beta_1\cos\theta + 1)}{2(\cos\theta + \beta_1)^2}\right), & \textit{if } \theta \in (\arccos\frac{2}{\pi}, \frac{\pi}{3}); \\ \ln\left(\dfrac{\pi}{2(1 + \cos\theta)}\right), & \textit{if } \theta \in [\frac{\pi}{3}, \frac{\pi}{2}). \end{cases} \tag{53}$$

Proof. We again start by substituting $\lambda_1 < 0$ into (46):

$$\Lambda^*_-(xy\cos\theta, x^2, y^2) = \ln\pi + \frac{x^2}{2} + \frac{y^2}{2} + \sup_{\substack{\lambda_1 < 0 \\ \beta > 1}} \Big\{g_-(\lambda_1, \beta)\Big\}, \tag{54}$$

$$g_-(\lambda_1, \beta) = (\cos\theta + \beta)\lambda_1 xy + \frac{\ln(\beta^2 - 1)}{2} + \ln(-\lambda_1) - \ln\left(\pi + 2\arctan\left(\frac{-1}{\sqrt{\beta^2 - 1}}\right)\right).$$

Differentiating w.r.t. λ_1 gives $(\cos\theta + \beta)xy + \frac{1}{\lambda_1}$. For $\lambda_1 \to -\infty$ this is positive, for $\lambda_1 \to 0^-$ this is negative, and so the maximum is at $\lambda_1 = -1/((\cos\theta + \beta)xy)$. Substituting this value for λ_1, and pulling out terms which do not depend on β yields:

$$\Lambda^*_-(xy\cos\theta, x^2, y^2) = \ln\left(\frac{\pi}{2}\right) + \frac{x^2}{2} + \frac{y^2}{2} - 1 - \ln(xy) + \sup_{\beta > 1}\Big\{g_-(\beta)\Big\}, \tag{55}$$

$$g_-(\beta) = \ln\left(\frac{\sqrt{\beta^2 - 1}}{(\cos\theta + \beta)\arccos\frac{1}{\beta}}\right) = \ln h_-(\beta).$$

Above we used the identity $\pi + 2\arctan(-1/\sqrt{\beta^2 - 1}) = 2\arccos\frac{1}{\beta}$, where the factor 2 has been pulled outside the supremum. Now, differentiating h_- w.r.t. β results in:

$$h'_-(\beta) = \frac{\beta\sqrt{\beta^2 - 1}(\beta\cos\theta + 1)\arccos\frac{1}{\beta} - \left(\beta^2 - 1\right)(\cos\theta + \beta)}{\beta\left(\beta^2 - 1\right)(\cos\theta + \beta)^2\arccos\frac{1}{\beta}}. \tag{56}$$

Clearly the denominator is positive, while for $\beta \to 1^+$ the limit is negative iff $\cos\theta < \frac{1}{2}$. For $\beta \to \infty$ we further have $h'_-(\beta) \to 0^-$ for $\cos\theta \leq \frac{2}{\pi}$ and $h'_-(\beta) \to 0^+$ for $\cos\theta > \frac{2}{\pi}$. We therefore analyze three cases separately below.

Case 1: $\frac{\pi}{3} \leq \theta < \frac{\pi}{2}$. In this parameter range, $h'_-(\beta)$ is negative for all $\beta > 1$, and the supremum lies at $\beta \to 1^+$ with limiting value $h_-(\beta) \to \frac{1}{1+\cos\theta}$. This yields the given expression for Λ^*_-.

Case 2: $\arccos\frac{2}{\pi} < \theta < \frac{\pi}{3}$. For θ in this range, $h'_-(\beta)$ is positive for $\beta \to 1^+$ and negative for $\beta \to \infty$, and changes sign exactly once, where it attains its maximum. After some rewriting, we find that this is at the value $\beta = \beta_1(\theta) \in (1, \infty)$ satisfying the relation from (1). Substituting this expression for $\arccos\frac{1}{\beta_1}$ into h_-, we obtain the result for Λ^*_-.

Case 3: $0 < \theta \leq \arccos\frac{2}{\pi}$. In this case h'_- is positive for all $\beta > 1$, and the supremum lies at $\beta \to \infty$. For $\beta \to \infty$ we have $h_-(\beta) \to \frac{2}{\pi}$ (regardless of θ) and we therefore get the final claimed result. ◀

Proof of Theorem 1. Combining the previous two results with Lemma 12 and Equation 47, we obtain explicit asymptotics for $\mathbb{P}(\boldsymbol{Z}_d \approx (xy\cos\theta,\, x^2,\, y^2))$. What remains is a maximization over $x, y > 0$ of p, which translates to a minimization of Λ^*. As $\frac{x^2}{2} + \frac{y^2}{2} - 1 - \ln(xy)$ attains its minimum at $x = y = 1$ with value 0, we obtain Theorem 1. ◀

B Proof of Proposition 9

We will assume the reader is familiar with (the notation from) [23]. Let $t = 2^{c_t d + o(d)}$ denote the number of hash tables, and $n = (4/3)^{d/2+o(d)}$. Going through the proofs of [23, Appendix A] and replacing the explicit instantiation of the collision probabilities $(1 - \theta/\pi)$ by an arbitrary function $p(\theta)$, we get that the optimal number of hash functions concatenated into one function for each hash table, denoted k, satisfies

$$k = \frac{\ln t}{-\ln p(\theta_1)} = \frac{c_t d}{d' \log_2(\pi/\sqrt{3})} \,. \tag{57}$$

The latter equality follows when substituting $\theta_1 = \pi/3$ and substituting the collision probabilities for partial hypercube LSH in some dimension $d' \leq d$. As we need $k \geq 1$, the previous relation translates to a condition on d' as $d' \leq \frac{c_t}{\log_2(\pi/\sqrt{3})} d$. As we expect the collision probabilities to be closer to those of full-dimensional hypercube LSH when d' is closer to d, we replace the above inequality by an equality, and what remains is finding the minimum value c_t satisfying the given constraints.

By carefully checking the proofs of [23, Appendix A.2-A.3], the exact condition on c_t to obtain the minimum asymptotic time complexity is the following:

$$-c_n = \max_{\theta_2 \in (0,\pi)} \left\{ \log_2 \sin \theta_2 + \frac{c_t}{\rho(\frac{\pi}{3}, \theta_2)} \right\} . \tag{58}$$

Here $c_n = \frac{1}{2} \log_2(\frac{4}{3}) \approx 0.20752$, and $\rho(\theta_1, \theta_2) = \ln p(\theta_1) / \ln p(\theta_2)$ corresponds to the exponent ρ for given angles θ_1, θ_2. Note that in the above equation, only c_t is an unknown. Substituting the asymptotic collision probabilities from Theorem 1, we find a solution at $c_t \approx 0.11464$, with maximizing angle $\theta_2 \approx 0.45739\pi$. This corresponds to a time and space complexity of $(n \cdot t)^{1+o(1)} = 2^{(c_n+c_t)d+o(d)} \approx 2^{0.32216d+o(d)}$ as claimed.