Locality Sensitive Hashing for Set-Queries, Motivated by Group Recommendations

Haim Kaplan
School of Computer Science, Tel Aviv University, Israel
haimk@tau.ac.il

Jay Tenenbaum
School of Computer Science, Tel Aviv University, Israel
jaytenenbaum@mail.tau.ac.il

Abstract

Locality Sensitive Hashing (LSH) is an effective method to index a set of points such that we can efficiently find the nearest neighbors of a query point. We extend this method to our novel Set-query LSH (SLSH), such that it can find the nearest neighbors of a set of points, given as a query.

Let $s(x, y)$ be the similarity between two points x and y. We define a similarity between a set Q and a point x by aggregating the similarities $s(p, x)$ for all $p \in Q$. For example, we can take $s(p, x)$ to be the angular similarity between p and x $\left(\text{i.e., } 1 - \frac{\angle(x,p)}{\pi}\right)$, and aggregate by arithmetic or geometric averaging, or taking the lowest similarity.

We develop locality sensitive hash families and data structures for a large set of such arithmetic and geometric averaging similarities, and analyze their collision probabilities. We also establish an analogous framework and hash families for distance functions. Specifically, we give a structure for the euclidean distance aggregated by either averaging or taking the maximum.

We leverage SLSH to solve a geometric extension of the approximate near neighbors problem. In this version, we consider a metric for which the unit ball is an ellipsoid and its orientation is specified with the query.

An important application that motivates our work is group recommendation systems. Such a system embeds movies and users in the same feature space, and the task of recommending a movie for a group to watch together, translates to a set-query Q using an appropriate similarity.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Data structures design and analysis; Information systems → Information retrieval

Keywords and phrases Locality sensitive hashing, nearest neighbors, similarity search, group recommendations, distance functions, similarity functions, ellipsoid

Digital Object Identifier 10.4230/LIPIcs.SWAT.2020.28

Related Version A full version of the paper is available at `https://arxiv.org/abs/2004.07286`.

Funding This work was supported by ISF grant no. 1595/19 and GIF grant no. 1367.

Acknowledgements We want to thank Prof. Micha Sharir and Prof. Edith Cohen for the fruitful discussions.

1 Introduction

The focus of this paper is on similarity search for queries which are sets of points (set-queries), where we aim to efficiently retrieve points with a high aggregated similarity to the points of the set-query.

17th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2020).
Editor: Susanne Albers; Article No. 28; pp. 28:1–28:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Efficient similarity search for massive databases is central in many application areas, such as recommendation systems, content-based image or audio retrieval, machine learning, pattern recognition, and data analysis. The database is often composed of high-dimensional feature vectors of documents, images, etc., and we are interested in finding the near neighbors of a query vector.

Traditional tree-based indexing mechanisms do not scale well to higher dimensions, a phenomenon known as the "curse of dimensionality". To cope with this curse of dimensionality, Indyk and Motwani [11, 10] introduced Locality Sensitive Hashing (LSH), a framework based on hash functions for which the probability of hash collision is higher for similar points than for dissimilar points.

Using such hash functions, one can determine near neighbors by hashing the query point and retrieving the data points stored in its bucket. Typically, multiple LSH functions are concatenated to reduce false positives, and multiple hash tables are needed to reduce false negatives. This gives rise to a data structure which satisfies the following property: for any query point q, if there exists an S-similar data point to q in the database, it retrieves (with constant probability) some cS-similar data point to q for some constant $0 < c < 1$. This data structure is parameterized by a parameter $\rho = \frac{\log(p_1)}{\log(p_2)} < 1$, where p_1 is the minimal collision probability for any two points of similarity at least S, and p_2 is the maximal collision probability for any two points of similarity at most cS. The data structure can be built in time and space $O(1/p_1 \cdot n^{1+\rho})$, and its query time is $O(1/p_1 \cdot n^\rho \log_{1/p_2}(n))$.

Since the seminal paper of Indyk and Motwani [11, 10], many extensions have been considered for the LSH framework [16]. A notable extension is the work of Shrivastava and Li [22], which study the inner product similarity $\textit{ip-sim}(x, y) = x^T y$. They find near neighbors for the inner product similarity by extending the LSH framework to allow asymmetric hashing schemes (ALSH) [20], in which we hash the query and the data points using different hash functions. There is also an analogous LSH framework for distance functions, based on hash functions for which the probability of hash collision is higher for near points than for far points. An important distance function to which the LSH framework has been applied is the ℓ_p distance [19]. Datar et al. [8] study the ℓ_p distance for $p \in (0, 2]$, and present a hash based on p-stable distributions. Andoni and Indyk [2] give a near-optimal (data oblivious) scheme for $p = 2$. Recently, several theoretically superior data dependent schemes have been designed [3, 4].

A noteworthy application of LSH is for *recommendation systems* [15], which are required to recommend points that are similar feature-wise to the user. *Group recommendation systems* [14, 17] are recommendation systems which provide recommendations, not only to an individual, but also to a whole group of people, and are gaining popularity in recent years. The need in such systems arises in many scenarios: when searching for a movie or a TV show for friends to watch together [21, 23], a travel destination for a family to spend a holiday break in [13, 18], or a good restaurant for a group of tourists to have lunch in [5]. In the literature of group recommendation systems, Jameson et al. [14] survey various techniques to aggregate individual user-point similarities s to a group-point similarity s^*. The most famous aggregation techniques are the *average similarity* which defines the aggregated similarity to be $s^*(Q, x) = \frac{1}{|Q|} \sum_{q \in Q} s(q, x)$, and the *center similarity* (sometimes called *Least-Misery*) which defines the aggregated similarity to be $s^*(Q, x) = \min_{q \in Q}(s(q, x))$.

Most of the work to date on group recommendations is experimental on relatively small data sets. In this paper we give (the first to the best of our knowledge) rigorous mathematical treatment of this problem using the LSH framework. LSH-based recommendation schemes are used for individual recommendations but do not naturally support group recommendations. We extend LSH to support set-queries. We formalize this setting by introducing the notions of a set-query-to-point (s2p) similarity function, and of the novel *set-query LSH* (SLSH).

Our novel set-query LSH (SLSH) framework extends the LSH framework to similarities between a set of points and a point (s2p similarities). We define such a similarity between a set-query $Q = \{q_1, \ldots, q_k\} \subset Z$ and a point $x \in Z$ by aggregating (e.g., averaging) point-to-point (p2p) similarities $(s(q_1, x), \ldots, s(q_k, x))$ where $s : Z \times Z \to \mathbb{R}_{\geq 0}$ is a p2p similarity. Specifically, we consider the *ℓ_p similarity* $s_p(Q, x) = \frac{1}{k}\sum_{i=1}^{k}(s(q_i, x))^p$ for a constant $p \in \mathbb{N}$ (of which the *average similarity* $s_{avg}(Q, x) = s_1(Q, x)$ is a special case), the *geometric similarity* $s_{geo}(Q, x) = \prod_{i=1}^{k} s(q_i, x)$, and the *center similarity* $s_{cen}(Q, x) = \min_{q \in Q} s(q, x)$ of s.[1] Analogously, we can define s2p distance functions and SLSH framework for distances. We develop hash families for which the probability of collision between a set-query Q and a point x is higher when Q is similar to x than when Q is dissimilar to x.

Our contribution

We extend the LSH framework to a novel framework for handling set-queries (SLSH) for both distance and similarity functions, and study their set-query extensions. We develop various techniques for designing set-query LSH schemes, either by giving an SLSH family directly for the s2p similarity at hand, or by reducing the problem to a previously solved problem for a different distance or similarity.

Simple SLSH schemes via achievable p2p similarities. We say that a p2p similarity s is *achievable* if there exists a hash family such that the collision probability between x and y is exactly $s(x, y)$. The *angular*, *hamming* and *Jaccard* p2p similarities have this property. We show how to construct SLSH families for the ℓ_p and geometric s2p similarities that are obtained by aggregating a p2p similarity which is achievable.

Many of our SLSH families for s2p similarities can be extended to *weighted* s2p similarity functions, in which the contribution of each individual p2p similarity has a different weight. For example, define the weighted geometric s2p similarity (of a p2p similarity s) of a set-query Q and a data point x to be $s_{wgeo}(Q, x) = \prod_{i=1}^{k}(s(q_i, x))^{w_i}$. These weights are independent of the specific query and are given at preprocessing time. As an example, a solution for the SLSH problem for s_{wgeo} for any *achievable* p2p similarity s appears in Appendix A.2.

Additionally, we present an SLSH scheme for the average euclidean distance which is based upon the shrink-lift transformation (the "lift" refers to the lifting transformation from Bachrach et al. [6]) which approximately reduces euclidean distances to angular distances. We get an average angular distance problem which we then solve using the fact that the angular similarity is achievable and inversely related to the angular distance.[2]

[1] For ease of presenting our ideas, we define the s_p and center similarities to be the p'th and k'th power of their conventional definition in the literature. Note that the results follow for the conventional definitions since maximizing a similarity is equivalent to maximizing a constant power of it.

[2] We note that as the LSH approximation parameter c approaches 1, the required shrink approaches 0. This makes the angles between the lifted points small, which in turn deteriorates the performance of the angular similarity structure (in particular, one can show that the term in $\log_{1/p_2}(n)$ in the query time bound of the LSH structure approaches infinity). Therefore, we conclude that the shrink-lift transformation is useful for values of c which are not too close to 1. However, note that such a property holds for any LSH-based nearest neighbors algorithm, where for approximation ratios $c \to 1$, the performance becomes equivalent or worse than linear scan.

Ellipsoid ALSH. We define the novel *euclidean ellipsoid distance* which naturally extends the regular euclidean distance. We develop an LSH-based near neighbors structure for this distance by a reduction to an SLSH problem with respect to the geometric angular distance. Recall that in the euclidean approximate near neighbor problem, the query specifies the center of two concentric balls such that one is a scaled version of the other. Analogously, in our novel ellipsoid distance, the query specifies the center and orientation of two concentric ellipsoids such that one is a scaled version of the other. If there is a point in the small ellipsoid, we have to return a point in the large one. We reduce this problem to a novel angular ellipsoid distance counterpart via the shrink-lift transformation mentioned before. In this angular distance counterpart, the distance is a weighted sum of squared angles (rather than squared distances in the euclidean ellipsoid distance).

To solve the angular ellipsoid ALSH problem, we make a neat observation that the squared angle that a point creates in the direction of an angular ellipsoid axis, is inversely related to the collision probability of the point with the hyperplane perpendicular to the axis, in the ALSH family of Jain et al. [12]. This observation reduces the problem to a weighted geometric angular similarity SLSH problem, which we finally solve as indicated above using the fact that the angular similarity is achievable.

Center euclidean distance SLSH. The most challenging s2p distance is the center euclidean distance which wants to minimize the maximum distance from the points of the set-query. For this distance function, we obtain an SLSH scheme when the set-query is of size 2, via a reduction to the euclidean ellipsoid ALSH problem. This reduction is based on an observation that the points of center euclidean distance at most r to a set-query of size 2, approximately form an ellipsoid.

We focus on developing techniques to construct SLSH families, but we do not compute closed formulas for ρ as a function of S and c. These expressions can be easily derived for the simpler families but are more challenging to derive for the more complicated ones. We leave the optimization of ρ and testing the method on real recommendation data for future work.

Other related work

Since we study our novel SLSH framework, there is no direct previous work on this. That been said, there is related previous work on LSH, ALSH, and recommendation systems which are as follows. In the literature of recommendation systems, Koren and Volinsky [15] discuss matrix factorization models where user-item interactions are modeled as inner products, and Bachrach et al. [6] propose a transformation that reduces the inner product similarity to euclidean distances. Regarding group recommendation systems, Masthoff and Judith [17] show that humans care about fairness and avoiding individual misery when giving group recommendations, and Yahia et al. [1] formalize semantics that account for item relevance to a group, and disagreements among the group members. Regarding LSH and ALSH, Neyshabur and Srebro [20] study symmetric and asymmetric hashing schemes for the inner product similarity, and show a superior symmetric LSH to that of Shrivastava and Li [22], that uses the transformation of Bachrach et al. [6]. As stated before, we use the ALSH family of Jain et al. [12] to solve the angular ellipsoid ALSH problem. We show that this family can be interpreted as a private case of an SLSH family for an appropriate s2p similarity, however Jain et al. [12] did not need this property, and the connection is coincidental.

2 Preliminaries

We use the following standard definition of a *Locality Sensitive Hash Family (LSH)* with respect to a given point-to-point (p2p) similarity function $s : Z \times Z \to \mathbb{R}_{\geq 0}$.

▶ **Definition 1** (Locality Sensitive Hashing (LSH)). *Let $c < 1$, $S > 0$ and $p_1 > p_2$. A family H of functions $h : Z \to \Gamma$ is an (S, cS, p_1, p_2)-LSH for a p2p similarity function $s : Z \times Z \to \mathbb{R}_{\geq 0}$ if for any $x, y \in Z$,*

1. *If $s(x, y) \geq S$ then $\Pr_{h \in H}[h(x) = h(y)] \geq p_1$, and*
2. *If $s(x, y) \leq cS$ then $\Pr_{h \in H}[h(x) = h(y)] \leq p_2$.*

Note that in the definition above, and in all the following definitions, the hash family H is always sampled uniformly. Following Shrivastava and Li [22] we extend the LSH framework to asymmetric similarities $s : Z_1 \times Z_2 \to \mathbb{R}_{\geq 0}$ (where Z_1 is the domain of the data points and Z_2 is the domain of the queries). Here the (S, cS, p_1, p_2)-ALSH family H consists of pairs of functions $f : Z_1 \to \Gamma$ and $g : Z_2 \to \Gamma$, and the requirement is that $\Pr_{(f,g) \in H}[f(x) = g(y)] \geq p_1$ if $s(x, y) \geq S$, and $\Pr_{(f,g) \in H}[f(x) = g(y)] \leq p_2$ if $s(x, y) \leq cS$.

Set-Query LSH

A special kind of asymmetric similarities are similarities between a set of points and a point (s2p similarities). That is, similarities of the form $s^* : \mathcal{P}(Z, k) \times Z \to \mathbb{R}_{\geq 0}$, where $\mathcal{P}(Z, k)$ is the set of subsets of Z of size k. We focus on s2p similarity functions that are obtained by aggregating the vector of p2p similarities $(s(q_1, x), \ldots, s(q_k, x))$ where $s : Z \times Z \to \mathbb{R}_{\geq 0}$ is a p2p similarity function, as we discussed in the introduction. We call an (S, cS, p_1, p_2)-ALSH for an **s2p** similarity s^*, an (S, cS, p_1, p_2)*-SLSH* for s^*. Our focus is on s2p similarities and SLSH families.

From similarities to distances

For distance functions we wish that close points collide with a higher probability than far points do. Specifically, we require that $\Pr_{h \in H}[h(x) = h(y)] \geq p_1$ if $d(x, y) \leq r$, that $\Pr_{h \in H}[h(x) = h(y)] \leq p_2$ if $d(x, y) \geq cr$, and that $c > 1$. We extend the LSH framework for distances to asymmetric distances and for s2p distances, and define ALSH and SLSH families as we did for similarities. As for similarity functions, we consider s2p distance functions that are defined based on the vector of p2p distances $(d(q_1, x), \ldots, d(q_k, x))$. In particular, we consider the ℓ_p *distance* $d_p(Q, x) = \frac{1}{k} \sum_{q \in Q} (d(q, x))^p$ for a constant $p \in \mathbb{N}$ (of which the *average distance* $d_{avg}(Q, x) = d_1(Q, x)$ is a special case), the *geometric distance* $d_{geo}(Q, x) = \prod_{q \in Q} d(q, x)$, and the *center distance* $d_{cen}(Q, x) = \max_{q \in Q} d(q, x)$ of d, where $d : Z \times Z \to \mathbb{R}_{\geq 0}$ is a p2p distance function.

Additional definitions

We consider the following common p2p similarity functions $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$: 1) The *angular similarity* $\angle sim(x, y) = 1 - \frac{\angle(x,y)}{\pi}$, and 2) The *inner product similarity ip-sim*$(x, y) = x^T y$ [22]. We also consider the following common p2p distance functions $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_{\geq 0}$: 1) The angular distance $\angle(x, y)$, and 2) The euclidean distance $ed(x, y) = \|x - y\|_2$.

We say that a hash family is an (S, cS)*-LSH* for a p2p similarity function s if there exist $p_1 > p_2$ such that it is an (S, cS, p_1, p_2)-LSH. An (S, cS)-LSH family can be used (see [11, 10]) to solve the corresponding (S, cS)-LSH problem of finding an (S, cS)-LSH structure. An (S, cS)*-LSH structure* finds (with constant probability) a neighbor of similarity at least

cS to a query q if there is a neighbor of similarity at least S to q. We define these concepts analogously (and apply analogous versions of [11, 10]) for ALSH and SLSH hash families and for LSH for distances.

We denote the unit ball in $\mathbb{R}^d$ by B_d and the unit sphere in $\mathbb{R}^d$ by S_d. We also denote $[n] := \{1, \ldots, n\}$, and occasionally use the abbreviations $(x_i)_{i=1}^m := (x_1, \ldots, x_m)$ and $\{x_i\}_{i=1}^m := \{x_1, \ldots, x_m\}$. All the missing proofs (from the body of the paper, and from the appendix) appear in the full version of the paper.[3]

3 Similarity schemes

We call a (symmetric or asymmetric) similarity function s *achievable* if there exists a hash family H such that for every query q and point x, $\Pr_{(f,g)\in H}[f(q) = g(x)] = s(q,x)$ (for symmetric p2p similarity functions $f = g$). Clearly, such an H is an (S, cS)-ALSH for s for any S and c. In this section, we show that the ℓ_p and geometric s2p similarity functions of an achievable p2p similarity, is by iteself achievable and therefore has and (S, cS)-SLSH.

Note that many natural p2p similarity functions are achievable. For example, the random hyperplane hash family [2] achieves the angular similarity function $s(x,y) = 1 - \frac{\angle(x,y)}{\pi}$, the random bit hash family [9] achieves the hamming similarity $s\left((x_1, \ldots, x_d), (y_1, \ldots, y_d)\right) = \frac{|\{i | x_i = y_i\}|}{d}$, and MinHash [7] achieves the Jaccard similarity $s(S,T) = \frac{|S\cap T|}{|S\cup T|}$.

In the full version of the paper, we also give a very simple reduction from the average inner product SLSH problem to the regular inner product ALSH problem (which is not achievable).

ℓ_p similarity

In this section, we define *repeat-SLSH*, and prove that it is an SLSH for the ℓ_p s2p similarity s_p of any achievable p2p similarity function s for any constant $p \in \mathbb{N}$. The intuition behind repeat-SLSH is that given an LSH family that achieves a p2p similarity function s, a query point q collides with a data point x on p randomly and independently selected hash functions with probability $(s(Q,x))^p$. Thus, if we uniformly sample a point $q \in Q$ of the set-query,[4] and then compute p consecutive hashes of q, the expected collision probability will be the ℓ_p similarity of Q and x. The formal definition is as follows.

▶ **Definition 2** (Repeat-SLSH). *Let s be an achievable p2p similarity function achieved by a hash family H_s, let k be the size of the set-query, and let $p \in \mathbb{N}$. We define the repeat-SLSH of H_s to be*

$$H = \left\{ \left(Q \to (h_j(q_i))_{j=1}^p, x \to (h_j(x))_{j=1}^p\right) \mid i \in [k],\ (h_1, \ldots, h_p) \in H_s^p \right\},$$

where q_i is the i'th element of the set-query $Q = \{q_1, \ldots, q_k\}$ in some consistent arbitrary order.[5]

▶ **Theorem 3.** *Let s be an achievable p2p similarity function, and let H_s be a family that achieves s. Then for any $S > 0$ and $c < 1$, the repeat-SLSH of H_s is an (S, cS)-SLSH for s_p, the ℓ_p similarity of s.*

Proof. It is clear that $\Pr_{(f,g)\in H}[f(Q) = g(x)] = s_p(Q,x)$ for any set-query $Q = \{q_i\}_{i=1}^k$ and data point x, so it is an (S, cS)-SLSH for any $S > 0$ and $c < 1$. ◀

[3] The link to the full paper appears in the front matter.

[4] Therefore, for repeat-SLSH we do not need to know the set-query size k a-priori.

[5] Let A be a set, and let $p \in \mathbb{N}$. We define $A^p := \{(x_i)_{i=1}^p \mid \forall i, x_i \in A\}$.

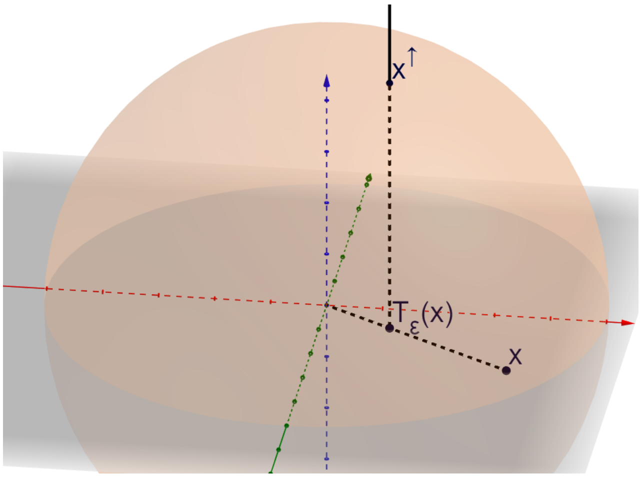

Figure 1 The shrink-lift transformation $x^{\uparrow}$.

Geometric similarity

The geometric similarity is somewhat similar to the center similarity - both similarities are suitable when we want to enforce high similarity to all points of the set-query. Analogously, here a query Q is mapped to $(h_i(q_i))_{i=1}^{k}$ where $h_1, \ldots, h_k$ are random hash functions, each applied to a corresponding item in Q. A data point x is mapped to $(h_i(x))_{i=1}^{k}$. It is not hard to see that the collision probability is $s_{geo}(Q, x)$. In Appendix A, we give a formal theorem analogous to Theorem 3 both for the unweighted and weighted versions of the geometric similarity.

4 Distance schemes

The notion of achievability that allowed us to construct simple SLSH families for s2p similarity functions does not naturally extend to distance functions. Nevertheless, in this section we directly design two important SLSH families for the average angular and the average euclidean distance functions.

We start with the easy observation that repeat-SLSH from Section 3 for $p = 1$ is, as is, an SLSH family for the average angular distance (the easy proof is in Appendix B.1).[6] In the rest of this section we show how to reduce the average euclidean distance SLSH problem to the average angular distance SLSH problem. We assume that all data points x and queries Q are in B_d, and given the parameters $r > 0$ and $c > 1$, we build an (r, cr)-SLSH structure for the average euclidean distance, ed_{avg}, as follows.

We consider the shrink transformation $T_\varepsilon : \mathbb{R}^d \to \mathbb{R}^d$ defined by $T_\varepsilon(x) = \varepsilon x$ for some $\varepsilon < \frac{1}{2}$. Additionally, we use the lifting transformation $L : B_d \to S_{d+1}$ of Bachrach et al. [6], defined by $L(x) = \left(x; \sqrt{1 - \|x\|^2}\right)$. For an ε, which will always be clear from the context, we define the shrink-lift transformation $(\cdot)^{\uparrow} : B_d \to S_{d+1}$, illustrated in Figure 1, by $x^{\uparrow} := L(T_\varepsilon(x))$.

[6] This family hashes a random point from the set-query Q to $\{-1, 1\}$ by a random hyperplane.

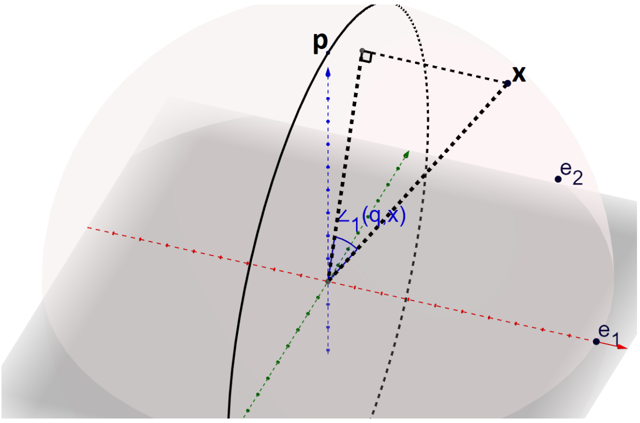

Figure 2 An angular ellipsoid ALSH query $(p, \{e_i\}_{i=1}^d)$ and $\angle_1(q,x)$ for some $x \in S_{d+1}$.

The following lemma specifies the relation between the angle of the lifted points and the euclidean distance between the original points. The exact details of the reduction, including the presentation of an SLSH structure for the average euclidean distance, appear in Appendix B.2.

▶ **Lemma 4.** *Let $x, y \in B_d$ and $\varepsilon \in (0, \frac{1}{2}]$, and define $m(x) = \frac{\sqrt{1+2x^2}}{\sqrt{1-2x^2}}$. Then,*

$$\varepsilon \|x - y\| \leq \angle(x^\uparrow, y^\uparrow) \leq m(\varepsilon) \cdot \varepsilon \|x - y\| .$$

5 Euclidean ellipsoid ALSH

In this section we present our most technically challenging result – an example that leverages SLSH to solve a geometric extension of the approximate near neighbor problem for the euclidean distance. Our structure is built for a specific "shape" of two concentric ellipsoids (specified by the weights of their axis), and their "sizes", r and cr, respectively. Given a query which defines the common center and orientation of these ellipsoids, if there is a data point in the smaller r-ellipsoid, then the structure must return a point in the larger cr-ellipsoid. Specifically, we define the euclidean ellipsoid distance as follows.

Euclidean ellipsoid ALSH

Let $q = (p, \{e_i\}_{i=1}^d)$ be a "query" pair where $p \in B_d$ is a center of an ellipsoid and $\{e_i\}_{i=1}^d$ are orthogonal unit vectors specifying the directions of the ellipsoid axes, let $x \in B_d$ be a data point, and let $\{w_1, \ldots, w_d\}$ be a fixed set of d rational non-negative weights.

We define the *euclidean ellipsoid distance* $d_{\circ}(q, x)$ between q and x with respect to the weights $\{w_1, \ldots, w_d\}$ to be $\sum_{i=1}^d w_i \left(e_i^T(x - p)\right)^2$.

In this section, we describe a structure for the euclidean ellipsoid distance $(r, cr)-$ALSH problem via a sequence of reductions. We reduce this problem to what we call an *angular ellipsoid ALSH* problem, which is then solved via another reduction to the *weighted geometric angular similarity SLSH* problem, which is solved in Appendix A.2.

We give a high level description of these reductions and differ the details to Appendix C. The first reduction is from the euclidean ellipsoid ALSH to what we call the *angular ellipsoid ALSH*. Recall that in Section 4, we have shown that for small values of ε, the shrink-lift transformation approximately reduces euclidean distances in B_d to angular distances on

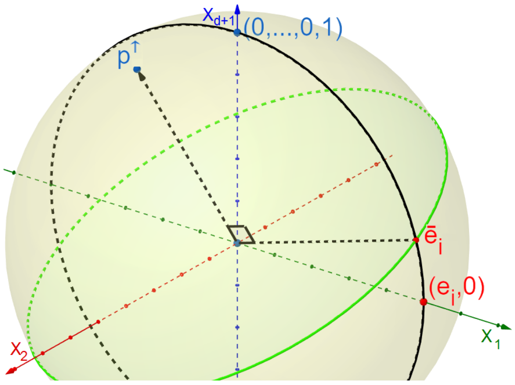

Figure 3 A query $(p, \{e_i\}_{i=1}^d)$ for the euclidean ellipsoid ALSH, and a corresponding angular axis $\overline{e_i}$ of e_i.

S_{d+1}, for which we can use structures for the angular similarity to solve the associated SLSH problems.[7] Here, we apply the same shrink-lift transformation to our data, and transform the ellipsoid queries to an angular counterpart defined as follows. An angular ellipsoid is specified by a center on the unit sphere and axes perpendicular to it. A point is inside it if the weighted sum of the squared **angles** that the point creates with the hyperplanes perpendicular to each axis and passing through the origin is smaller than r. We formalize this as follows.

Angular ellipsoid ALSH

Let $q = (p, \{e_i\}_{i=1}^d)$ be a “query” pair where $p \in S_{d+1}$ is a center of an “*angular ellipsoid*”, and $\{e_i\}_{i=1}^d \subset S_{d+1}$ are unit vectors orthogonal to p (but need not be orthogonal to each other), let $x \in S_{d+1}$ be a data point, and let $\{w_1, \ldots, w_d\}$ be a fixed set of d rational non-negative weights.

Given an index $i \in [d]$, we define $\angle_i(q, x) \in [0, \frac{\pi}{2})$ to be the angle between x and its projection onto the hyperplane through the origin which is orthogonal to e_i. Note that since e_i is orthogonal to p, this hyperplane contains p. This is illustrated in Figure 2, from which we can also observe that $\angle_i(q, x) = \sin^{-1}\left(\left|e_i^T \cdot x\right|\right)$.

We define the *angular ellipsoid distance* $d_{\angle\circ}(q, x)$ between q and x with respect to the weights $\{w_1, \ldots, w_d\}$ to be $\sum_{i=1}^d w_i \cdot \angle_i(q, x)^2$.

We prove that the shrink-lift transformation approximately maps an ellipsoid to an angular ellipsoid with the same weights, and with a center as the shrink-lift of the original ellipsoid's center, and axes which are slight “upwards” (to the direction of the axis x_{d+1}) rotations of the axes of the original ellipsoid, such that they are perpendicular to the angular ellipsoid's center (see Figure 3).

We solve the angular ellipsoid ALSH problem by reducing it to the weighted geometric angular similarity SLSH problem. Our reduction is based on the H-hash of Jain et al. [12], which stores points that reside on S_{d+1} such that for a query hyperplane h through the origin, we can efficiently retrieve the data points that have a small angular distance with their projection on h. H-hash in fact uses an SLSH family for the geometric angular similarity

[7] As stated in the introduction, we do not want to set ε to be too small since this deteriorates the performance of subsequent LSH structures we reduce to.

for sets of size 2, using the following observation which we adapt to our setting. For any direction e and hyperplane h perpendicular to e through the origin, and any $x \in S_d$, it holds that $\angle sim_{geo}(\{e, -e\}, x) = (1 - \angle(x, e)/\pi)(1 - \angle(x, -e)/\pi) = \frac{1}{4} - \frac{\angle(x,h)^2}{\pi^2}$, where $\angle(x, h)$ is the angle between x and its projection on h, and the last step follows by the fact that $\min(\angle(x, e), \angle(x, -e)) = \frac{\pi}{2} - \angle(x, h)$ and $\max(\angle(x, e), \angle(x, -e)) = \frac{\pi}{2} + \angle(x, h)$. Recall that the angular ellipsoid distance between a query $q = (p, \{e_i\}_{i=1}^d)$ and a point x is a weighted sum of $(\angle_i(q, x))^2$. Therefore, if we hash the hyperplane orthogonal to e_i with H-hash, it will collide with higher probability with data points x with a smaller $(\angle_i(q, x))^2$. This suggests that we can answer an angular ellipsoid query $q = (p, \{e_i\}_{i=1}^d)$ by a weighted geometric angular similarity SLSH set-query where the set is the union of the sets $\{e_i, -e_i\}$ for all $i \in [d]$, using the angular ellipsoid weight w_i associated with the axis e_i for each $i \in [d]$. Specifically, the corresponding set-query is $Q = \{e_1, -e_1, e_2, -e_2, \ldots, e_d, -e_d\}$, and the structure is built with the weights $\{w_1, w_1, w_2, w_2, \ldots, w_d, w_d\}$. For the reduction's analysis to hold, we must require that any query $q = (p, \{e_i\}_{i=1}^d)$ and data point x satisfy $\angle(p, x) \leq \sqrt{\frac{c-1}{c}} \cdot \frac{\pi}{4}$. This can be easily guaranteed by taking a sufficiently small value of ε in the previous reduction from euclidean ellipsoids to angular ellipsoids, such that the set of transformed queries and data points has a sufficiently small angular diameter.

Finally, the weighted geometric angular similarity SLSH problem is solved in Appendix A.2.

6 Center euclidean distance for set-queries of size 2

In this section we present a data structure for the center euclidean (r, cr)-SLSH problem. This is among our most technically challenging results. Our data structure receives a set-query $Q = \{q_1, q_2\}$ and returns (with constant probability) a data point v such that
$ed_{cen}(Q, v) = \max(\|v - q_1\|, \|v - q_2\|) \leq cr$, if there is a data point v such that
$ed_{cen}(Q, v) = \max(\|v - q_1\|, \|v - q_2\|) \leq r$.

Our data structure requires that c is larger than c_{min} where $c_{\text{min}} = \frac{3}{2\sqrt{2}} \approx 1.06066$ is a constant slightly larger than 1. We also assume that the possible queries $Q = \{q_1, q_2\}$ are such that $\frac{1}{2}\|q_1 - q_2\| < (1 - \phi)r$, for a parameter $\phi < 1$ that is known to the structure.[8]

We construct our structure via a reduction to the euclidean ellipsoid ALSH from Section 5. Consider the query $Q = \{q_a, q_{-a}\}$ to the center euclidean SLSH structure where $q_a = (a, 0, \ldots, 0)$ and $q_{-a} = (-a, 0, \ldots, 0)$, for some $0 < a < (1 - \phi)r/2$. Let $L^s = \{v \mid \max(\|v - q_a\|, \|v - q_{-a}\|) \leq r\}$ be the set of point of center distance at most r from Q, and let $L^b = \{v \mid \max(\|v - q_a\|, \|v - q_{-a}\|) \leq cr\}$ be the set of point of center distance at most cr from Q. We also define the following two ellipsoids S and B centered at the origin with axes aligned with the standard axes $x_1, \ldots, x_d$:

$$S = \left\{(x_1, \ldots, x_d) \mid \frac{r+a}{r-a}x_1^2 + \sum_{i=2}^{d} x_i^2 \leq r^2 - a^2\right\},$$

$$B = \left\{(x_1, \ldots, x_d) \mid \frac{r+a}{r-a}x_1^2 + \sum_{i=2}^{d} x_i^2 \leq \left(\frac{cr}{c_{\text{min}}}\right)^2 - a^2\right\}.$$

Our reduction depends on the crucial observation stated in the following lemma.

[8] For queries $Q = \{q_1, q_2\}$ such that $\frac{1}{2}\|q_1 - q_2\| > r$, no point v can satisfy $\max(\|v - q_1\|, \|v - q_2\|) \leq r$, and returning no points for such queries satisfies our structure requirements trivially.

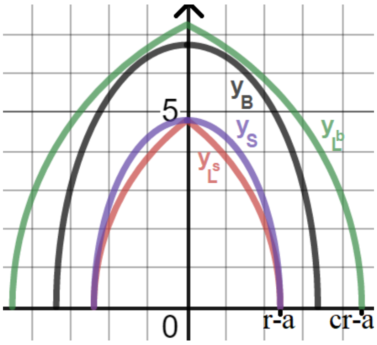

Figure 4 Plots of y_{L^s}, y_S, y_B, and y_{L^b} as functions of x_1. $a = 3.6$, $r = 6$, $c = 1.35$.

▶ **Lemma 5.** *We have that $L^s \subseteq S \subseteq B \subseteq L^b$.*

To illustrate the relation between L^s, S, B, and L^b, we denote the distances of their boundaries from the axis x_1 by $y_{L^s}(x_1)$, $y_S(x_1)$, $y_B(x_1)$ and $y_{L^b}(x_1)$, respectively. These functions are plotted in Figure 4.

Intuitively, our reduction will replace L^s by S and L^b by B: If there is a point x in L_s then x is also in S and the euclidean ellipsoid structure will find a point in B which is in L^b. Specifically, we would like to query with $\{q_a, q_{-a}\}$ a euclidean ellipsoid $(r', c'r')$-ALSH structure where $r' = r^2 - a^2$, c' is set such that $c'r' = \left(\frac{cr}{c_{\min}}\right)^2 - a^2$, and the weights are $\left\{\frac{r+a}{r-a}, 1\ldots, 1\right\}$.

The problem is that a depends on the query (it is half the distance between the query points) and obviously we cannot prepare a different euclidean ellipsoid $(r', c'r')$-ALSH structure for each query. To overcome this we quantize the range of possible values of a and construct a data structure for each quantized value. The range of the possible values for a is $[0, (1-\phi)r]$ and our quantization consists of the values $i \cdot \delta$ for $i = 0, \ldots \lceil \frac{(1-\phi)r}{\delta} \rceil$ where $\delta = \min\left(\frac{1}{2}, 1 - \sqrt{\frac{c_{\min}}{c}}\right)\phi r$.[9,10]

The euclidean ellipsoid $(r', c'r')$-ALSH structure corresponding to the value $i \cdot \delta$ has $r' = \frac{c}{c_{\min}} \cdot \left(r^2 - (i \cdot \delta)^2\right)$, $c' = \frac{c}{c_{\min}}$ and weights $\left\{\frac{r+i\cdot\delta}{r-i\cdot\delta}, 1, \ldots, 1\right\}$. For correctness we will prove that the ellipsoids $S^+ = \left\{(x_1, \ldots, x_d) \mid \frac{r+a'}{r-a'}x_1^2 + \sum_{i=2}^d x_i^2 \le \frac{c}{c_{\min}} \cdot \left(r^2 - (a')^2\right)\right\}$ and $B^- = \left\{(x_1, \ldots, x_d) \mid \frac{r+a'}{r-a'}x_1^2 + \sum_{i=2}^d x_i^2 \le \left(\frac{c}{c_{\min}}\right)^2 \cdot \left(r^2 - (a')^2\right)\right\}$, where $a' = \lceil \frac{a}{\delta} \rceil \cdot \delta$, are such that $S \subseteq S^+ \subset B^- \subseteq B$. One can easily show that $r \ge a' \ge 0$, so the coefficients of x_1^2 and the right hand side of the equations in S^+ and B^- are both non-negative and well-defined.

Query phase

Let $Q = \{q_1, q_2\} \subseteq B_d$ be a set-query where $\|q_1 - q_2\| = 2a$ for $a \in [0, (1-\phi)r)$. Let $a' = \lceil \frac{a}{\delta} \rceil \delta$ as before. To get the answer, we query the euclidean ellipsoid $(r', c'r')$-ALSH structure, where $r' = \frac{c}{c_{\min}} \cdot \left(r^2 - (a')^2\right)$, $c' = \frac{c}{c_{\min}}$ and the weights are $\left\{\frac{r+a'}{r-a'}, 1, \ldots, 1\right\}$ with a query q defined as follows.

[9] To ensure rationality of weights, if δ is irrational, we replace it by $\mathbb{Q}_{>0} \ni \delta' < \delta$.

[10] Intuitively, when c is close to $c_{\min}$, and when ϕ is small, our quantization is finer.

Let R_{q_1,q_2} be a rigid transformation (rotation and translation) such that $R_{q_1,q_2}(q_1) = q_a$ and $R_{q_1,q_2}(q_2) = q_{-a}$ for $q_a = (a, 0 \ldots, 0)$ and $q_{-a} = (-a, 0 \ldots, 0)$. We set $q = (p, \{\overline{e_i}\}_{i=1}^{d})$ where $p = R_{q_1,q_2}^{-1}((0, \ldots, 0)) = \frac{q_1+q_2}{2} \in B_d$ and $\forall i$, $\overline{e_i} = R_{q_1,q_2}^{-1}(e_i)$ where $\{e_i\}_{i=1}^{d}$ is the standard basis of $\mathbb{R}^d$. Our main result is,

▶ **Theorem 6.** *The structure described above is an (r, cr)-SLSH structure for the center euclidean distance and queries of size 2. (For any $c > c_{\min}$, and queries $Q = \{q_1, q_2\}$ such that $\frac{1}{2}\|q_1 - q_2\| < (1 - \phi)r$.)*

7 Conclusions and directions for future work

We present a novel extended LSH framework, motivated by group recommendation systems. We define several set-query extensions for distance and similarity functions, and show how to design SLSH families and data structures for them using different techniques. We use this framework to solve a geometric extension of the euclidean distance approximate near neighbor problem, which we call *euclidean ellipsoid ALSH*, via reduction to an SLSH problem. All the reductions we describe have some performance loss, which (for distance functions) is expressed by a smaller p_1 and p_2, and a worse value of ρ. Estimating the exact performance loss (the value of ρ) and finding more efficient reductions is an interesting line of research. Finding a method for the center euclidean distance for set-queries larger than two is another intriguing open question.

References

1 Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. Group recommendation: Semantics and efficiency. *Proceedings of the VLDB Endowment*, 2(1):754–765, 2009. doi:10.14778/1687627.1687713.

2 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, pages 459–468. IEEE, 2006. doi:10.1109/FOCS.2006.49.

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

4 Alexandr Andoni and Ilya Razenshteyn. Tight lower bounds for data-dependent locality-sensitive hashing. In *SOCG*, pages 1–11. ACM, 2016. doi:10.4230/LIPIcs.SoCG.2016.9.

5 Liliana Ardissono, Anna Goy, Giovanna Petrone, Marino Segnan, and Pietro Torasso. Tailoring the recommendation of tourist information to heterogeneous user groups. In *Workshop on adaptive hypermedia*, pages 280–295. Springer, 2001. doi:10.1007/3-540-45844-1_26.

6 Yoram Bachrach, Yehuda Finkelstein, Ran Gilad-Bachrach, Liran Katzir, Noam Koenigstein, Nir Nice, and Ulrich Paquet. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 257–264. ACM, 2014. doi:10.1145/2645710.2645741.

7 Andrei Z Broder. On the resemblance and containment of documents. In *Compression and complexity of sequences*, pages 21–29. IEEE, 1997. doi:10.1109/SEQUEN.1997.666900.

8 Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SOCG*, pages 253–262. ACM, 2004. doi:10.1145/997817.997857.

9 Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529, 1999. URL: http://www.vldb.org/conf/1999/P49.pdf.

10 Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, 8(1):321–350, 2012. doi:10.4086/toc.2012.v008a014.

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

12 Prateek Jain, Sudheendra Vijayanarasimhan, and Kristen Grauman. Hashing hyperplane queries to near points with applications to large-scale active learning. *Transactions on Pattern Analysis and Machine Intelligence*, pages 276–288, 2014. doi:10.1109/TPAMI.2013.121.

13 Anthony Jameson. More than the sum of its members: challenges for group recommender systems. In *AVI*, pages 48–54. ACM, 2004. doi:10.1145/989863.989869.

14 Anthony Jameson and Barry Smyth. Recommendation to groups. In *The adaptive web*, pages 596–627. Springer, 2007. doi:10.1007/978-3-540-72079-9_20.

15 Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi:10.1109/MC.2009.263.

16 Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *VLDB*, pages 950–961, 2007. URL: http://www.vldb.org/conf/2007/papers/research/p950-lv.pdf.

17 Judith Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. In *Personalized digital television*, pages 93–141. Springer, 2004. doi:10.1023/B:USER.0000010138.79319.fd.

18 Kevin McCarthy, Lorraine McGinty, Barry Smyth, and Maria Salamó. The needs of the many: a case-based group recommender system. In *ECCBR*, pages 196–210. Springer, 2006. doi:10.1007/11805816_16.

19 Rajeev Motwani, Assaf Naor, and Rina Panigrahi. Lower bounds on locality sensitive hashing. In *SOCG*, pages 154–157. ACM, 2006. doi:10.1145/1137856.1137881.

20 Behnam Neyshabur and Nathan Srebro. On symmetric and asymmetric lshs for inner product search. In *ICML*, pages 1926–1934, 2015. URL: http://proceedings.mlr.press/v37/neyshabur15.html.

21 Mark O'connor, Dan Cosley, Joseph A Konstan, and John Riedl. Polylens: a recommender system for groups of users. In *ECSCW*, pages 199–218. Springer, 2001. doi:10.1007/0-306-48019-0_11.

22 Anshumali Shrivastava and Ping Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *NIPS*, pages 2321–2329, 2014. URL: http://arxiv.org/abs/1405.5869.

23 Zhiwen Yu, Xingshe Zhou, Yanbin Hao, and Jianhua Gu. Tv program recommendation for multiple viewers based on user profile merging. *UMUAI*, 16(1):63–82, 2006. doi:10.1007/s11257-006-9005-6.

A Missing parts from Section 3

A.1 Geometric similarity

In this section, we define *exhaustive-SLSH*, and prove that it is an SLSH for the geometric similarity, s_{geo}, of any achievable p2p similarity function s.

Note that the geometric similarity is somewhat similar to the center similarity - both similarities are suitable when we want to enforce high similarity to all points of the set-query. Our scheme for center similarity given in Section 6 is technically challenging. Thus, exhaustive-SLSH could be a simple alternative that somewhat relaxes the requirement to be similar to all points of the query for simplicity.

The intuition behind exhaustive-SLSH is that given an LSH family H that achieves a p2p similarity function s, then for a set-query $Q = \{q_1, \ldots, q_k\}$ and a point x, the expected collision probability of $(h_1(q_1), \ldots, h_k(q_k))$ with $(h_1(x), \ldots, h_k(x))$ when the $\{h_i\}$'s are sampled from H, is $s_{geo}(Q, x)$. The formal definition is as follows.

Exhaustive-SLSH

Let s be an achievable p2p similarity function achieved by a hash family H_s, and let k be the set-query size. We define the exhaustive-SLSH of H_s to be the following family of pairs $H = \left\{ \left(Q \to (h_j(q_j))_{j=1}^{k}, x \to (h_j(x))_{j=1}^{k} \right) \mid (h_1, \ldots, h_k) \in H_s^k \right\}$.

▶ **Theorem 7.** *Let s be an achievable p2p similarity, and H_s be a family that achieves s. Then the exhaustive-SLSH of H_s is an SLSH for the geometric similarity of s.*

Proof. It is clear that $\Pr_{(f,g)\in H}[f(Q) = g(x)] = s_{geo}(Q, x)$ for any set-query $Q = \{q_i\}_{i=1}^{k}$ and data point x, so it is an (S, cS)-SLSH for any $S > 0$ and $c < 1$. ◀

A.2 Weighted geometric similarity

In this section, we define *weighted exhaustive-SLSH*, and prove that it is an SLSH structure for the weighted geometric similarity s_{wgeo} of any achievable p2p similarity function s. So far, we have only considered equal-weighted query points, however, motivated by recommending movies to a set of people, a logical extension would be giving the individuals weights according to their importance, or the strength of their general preferences. To define the *weighted geometric similarity*, we use a sequence of non-negative **rational** weights $W = \{w_1, \ldots, w_k\}$, where each w_i is defined by a pair (a_i, b_i) such that $a_i \in \mathbb{N} \cup \{0\}$, $b_i \in \mathbb{N}$, and $w_i = \frac{a_i}{b_i}$, and k is the set-query size. Given W and a p2p similarity function s, we define the weighted geometric similarity (of s) of a set-query $Q = \{q_1, \ldots, q_k\}$ and a data point x to be $s_{wgeo}(Q, x) = \prod_{i=1}^{k} \left(s\left(q_i, x\right)\right)^{w_i}$.[11] In case the underlying p2p similarity function s is achievable, we reduce the weighted geometric similarity (S, cS)-SLSH problem to the geometric similarity $(S', c'S')$-SLSH problem.

Weighted exhaustive-SLSH

Given $S > 0$, $c < 1$, a p2p similarity function s, the set-query size k, and non-negative rational weights $\{w_i\}_{i=1}^{k}$ as defined above, we define $m = lcm\left(\{b_i\}_{i=1}^{k}\right) \in \mathbb{N}$.[12] The weighted exhaustive-SLSH structure works as follows. In the preprocessing phase, we store all the data points in an $(S^m, c^m S^m)$-SLSH structure for the geometric similarity for a set-query of size $k' = m \cdot \sum_{i=1}^{k} w_i$.[13] Given a set-query $Q = \{q_i\}_{i=1}^{k}$, we query the structure built in the preprocessing phase, with the set-query $T(Q) = \{q_1, \ldots, q_1, \ldots, q_k, \ldots, q_k\}$,[14] where each $q_i \in T(Q)$ is repeated $m \cdot w_i = a_i \cdot \frac{m}{b_i} \in \mathbb{N}$ times.

▶ **Theorem 8.** *Weighted exhaustive-SLSH is an (S, cS)-SLSH structure for the weighted geometric similarity s_{wgeo} of any achievable p2p similarity function s.*

Proof. Observe that for any set-query $Q = \{q_i\}_{i=1}^{k}$ of size k and any data point x, it holds that $s_{geo}(T(Q), x) = \prod_{i=1}^{k} \left(s\left(q_i, x\right)\right)^{m \cdot w_i} = \left(\prod_{i=1}^{k} \left(s\left(q_i, x\right)\right)^{w_i}\right)^{m} = \left(s_{wgeo}(Q, x)\right)^{m}$. Thus, the claim follows since if there is a data point x such that $s_{wgeo}(Q, x) \geq S$, then $s_{geo}(T(Q), x) \geq S^m$, and the $(S^m, c^m S^m)$-SLSH structure finds a data point x such that $s_{geo}(T(Q), x) \geq c^m S^m$, i.e., such that $s_{wgeo}(Q, x) \geq cS$. ◀

[11] For weighted similarities we assume that the set-query is ordered, and this order determines the correspondence between the weights and the points in the set-query.

[12] By lcm we denote the least common multiple.

[13] We can derive such a structure from exhaustive-SLSH (which can be applied since s is achievable).

[14] We allow set-queries that are in fact multi-sets. All our derivations apply to multi set-queries.

B Detailed results from Section 4

B.1 Average angular distance

We warm up with an easy result, and show that repeat-SLSH for the average angular **similarity** (Section 3) is an SLSH family for the average angular **distance** - a fact that follows since the average angular **similarity** is a decreasing function with respect to the average angular **distance**.

▶ **Theorem 9.** *Repeat-SLSH for the average angular* ***similarity*** *is an SLSH for the average angular* ***distance*** $\angle_{avg}$.

Proof. For any set-query Q of size k and data point x, $\angle sim_{avg}(Q,x) = \frac{1}{k}\sum_{q\in Q}\left(1 - \frac{\angle(q,x)}{\pi}\right)$ $= 1 - \frac{\frac{1}{k}\sum_{q\in Q}\angle(q,x)}{\pi} = 1 - \frac{\angle_1(Q,x)}{\pi}$. Thus, the claim follows since for any $r > 0$ and $c > 1$, by Theorem 3, repeat-SLSH for the average angular similarity is an $(1 - \frac{r}{\pi}, 1 - \frac{cr}{\pi}, p_1, p_2)$-SLSH for $\angle sim_1$ for some $p_1 > p_2$, and specifically is an (r, cr, p_1, p_2)-SLSH for $\angle_{avg}$. ◀

B.2 Average euclidean distance

We give a formal definition of Shrink-lift-SLSH, which reduces the average euclidean distance problem to the average angular distance problem. Shrink-lift-SLSH works as follows.

Preprocessing phase. Given the parameters $r > 0,\ c > 1$ and the set-query size k, define $\varepsilon = \frac{1}{2}\sqrt{1 - \frac{2}{1+c^2}} < \frac{1}{2}$. We transform each data point x to $x^{\uparrow}$, and store the transformed data points in an $(r', c'r')$-SLSH structure for the average angular distance, for the parameters $r' = m(\varepsilon)\cdot\varepsilon r,\ c' = \frac{\varepsilon cr}{r'} = \frac{c}{m(\varepsilon)}$ and $k' = k$, where we define $m : \left[0, \frac{1}{2}\right] \to \mathbb{R}$ by $m(x) = \frac{\sqrt{1+2x^2}}{\sqrt{1-2x^2}}$.

Query phase. Let Q be a set-query of size k. We query the average angular distance $(r', c'r')$-SLSH structure constructed in the preprocessing phase with the set-query $Q' = \{q^{\uparrow} \mid q \in Q\}$.

In order to prove that shrink-lift-SLSH is an (r, cr)-SLSH structure for the average euclidean distance, Lemma 10 bounds the angle between the lifted points in terms of their original euclidean distance, using the error function $e(\varepsilon, x, y) := \left(\sqrt{\frac{1}{\varepsilon^2} - \|x\|^2} - \sqrt{\frac{1}{\varepsilon^2} - \|y\|^2}\right)^2$.

▶ **Lemma 10.** *Let $x, y \in B_d$ and $\varepsilon \in (0, 1]$. Then*

$$2\sin^{-1}\left(\frac{\varepsilon}{2}\cdot\|x-y\|\right) \le \angle(x^{\uparrow}, y^{\uparrow}) = 2\sin^{-1}\left(\frac{\varepsilon}{2}\sqrt{\|x-y\|^2 + e(\varepsilon,x,y)}\right).$$

The following lemma bounds the error term.

▶ **Lemma 11.** *For any $x, y \in B_d$ and $\varepsilon \in (0, \frac{1}{2}]$, $0 \le e(\varepsilon, x, y) \le \frac{4}{3}\|x-y\|^2\varepsilon^2$.*

Next, we show the following property of $\sin^{-1}(\cdot)$, which is used in the proof of Lemma 13, and later in the proof of Lemma 15.

▶ **Lemma 12.** $x \le \sin^{-1}(x) \le \frac{x}{\sqrt{1-x^2}}$ *for any $x \in [0, 1)$.*

Then, we use Lemmas 10, 11 and 12 to derive the following important Lemma.

▶ **Lemma 13.** *Let $x, y \in B_d$ and $\varepsilon \in (0, \frac{1}{2}]$. Then, $\varepsilon\|x-y\| \le \angle(x^{\uparrow}, y^{\uparrow}) \le m(\varepsilon)\cdot\varepsilon\|x-y\|$.*

Finally, we use Lemma 13 to prove the following theorem, which is the main result of this section.

▶ **Theorem 14.** *Shrink-lift-SLSH is an (r, cr)-SLSH structure for the average euclidean distance ed_{avg}.*

C Euclidean ellipsoid ALSH detailed presentation

In this section, we give a detailed presentation of the two reductions we use to solve the euclidean ellipsoid ALSH problem from Section 5. Section C.1 gives a reduction from the euclidean ellipsoid ALSH to the angular ellipsoid problem. Section C.2 then reduces this problem to the weighted geometric angular similarity SLSH problem, which is solved in Appendix A.2. We note that this reduction requires that any query $q = (p, \{e_i\}_{i=1}^{d})$ and data point x in the angular ellipsoid structure satisfy $\angle(p, x) \leq \sqrt{\frac{c-1}{c}} \cdot \frac{\pi}{4}$. As we will see, the inputs to the angular ellipsoid structure that we produce by the first reduction (i.e., from the euclidean ellipsoid problem) will satisfy this requirement.

It is worth mentioning that the solution in Appendix A.2 requires that the weights are rational, hence we also require rational weights in both the ellipsoid structures.

C.1 From euclidean ellipsoid ALSH to angular ellipsoid ALSH

In this section, we reduce the euclidean ellipsoid (r, cr)-ALSH problem to an angular ellipsoid $(r', c'r')$-ALSH problem. To do this, we use the shrink-lift transformation $(\cdot)^{\uparrow}$ from Section 4 with an appropriately tuned shrinking parameter ε, to map our data points from B_d to S_{d+1}. For our proofs of Lemma 15 and Theorem 16 to hold, we need that $\varepsilon \leq \frac{1}{8}$. Additionally, to prove that the parameter c' that we use for the angular ellipsoid $(r', c'r')$ structure is larger than 1 (Theorem 16), we need that $\varepsilon \leq \frac{\sqrt[8]{c}-1}{\sqrt[8]{c}+1}$ and $\varepsilon \leq \sqrt{\frac{(c-\sqrt{c})r}{5(\sqrt{c}+1)\cdot\sum_{i=1}^{d} w_i}}$. Finally, to ensure that $\angle(p, x) \leq \sqrt{\frac{c-1}{c}} \cdot \frac{\pi}{4}$ for any query $q = (p, \{e_i\}_{i=1}^{d})$ and data point x in the angular ellipsoid structure (see the proof of Theorem 16 in the full paper), we need that $\varepsilon \leq \sqrt{1 - \frac{1}{\sqrt[4]{c}}} \cdot \frac{\pi}{8\sqrt{2}}$. We therefore set ε to be the minimum of all these upper bounds, that is $\varepsilon = \min\left(\frac{1}{8}, \frac{\sqrt[8]{c}-1}{\sqrt[8]{c}+1}, \sqrt{\frac{(c-\sqrt{c})r}{5(\sqrt{c}+1)\cdot\sum_{i=1}^{d} w_i}}, \sqrt{1 - \frac{1}{\sqrt[4]{c}}} \cdot \frac{\pi}{8\sqrt{2}}\right)$.

We store the images (by the shrink-lift transformation) of our data points in the angular ellipsoid $(r', c'r')$-ALSH structure.[15] We recall (Lemma 13) that for a sufficiently small ε the angular distance between $x^{\uparrow}$ and $y^{\uparrow}$ is approximately equal to ε times the euclidean distance between x and y. We set $r' = \varepsilon^2(1+\varepsilon)^2 \cdot \left(r + 5\beta(\varepsilon) \cdot \sum_{i=1}^{d} w_i\right)$, and $c' = \frac{\varepsilon^2(1-\varepsilon)^2\cdot\left(cr - 5\beta(\varepsilon)\cdot\sum_{i=1}^{d} w_i\right)}{r'}$, where $\beta(\varepsilon) = \frac{1-\sqrt{1-\varepsilon^2}}{\sqrt{1-\varepsilon^2}} \approx \frac{\varepsilon^2}{2} \geq 0$. Our choice of ε guarantees that $\beta(\varepsilon) \cdot \sum_{i=1}^{d} w_i \ll r$ and thereby r' is approximately $\varepsilon^2 \cdot r$, as we expect since the angular ellipsoid distance is a sum of (weighted) squared angular distances each of which is smaller by a factor of ε from its corresponding euclidean distance. Notice also that for our choice of ε, c' is approximately equal to $\sqrt[4]{c}$.[16] The angular ellipsoid structure uses the same weights as of the euclidean ellipsoid structure.

The query

Let $q_0 = (p, \{e_i\}_{i=1}^{d})$ be a euclidean ellipsoid query, where $p \in B_d$ is a center of an ellipsoid and $\{e_i\}_{i=1}^{d}$ are the unit vectors of $\mathbb{R}^d$ in the directions of the ellipsoid axes. We query the angular ellipsoid structure constructed in the preprocessing phase with the angular ellipsoid

[15] We do not want to set ε to be too small since this is likely to deteriorate the performance of the angular ellipsoid structure on these images.

[16] By using a smaller ε we can make c' closer to c.

query $q = (p^{\uparrow}, \{\overline{e_i}\}_{i=1}^{d})$, where each $\overline{e_i}$ is obtained by rotating $(e_i, 0)$ in the direction of $(0, \ldots, 0, 1)$, until its angle with $p^{\uparrow}$ becomes $\frac{\pi}{2}$ (this is illustrated in Figure 3). Formally, we define $\overline{e_i} := (a_i \cdot e_i; \sqrt{1 - a_i^2})$ where $a_i = -sign(p_i) \cdot \sqrt{\frac{1 - \|\varepsilon p\|^2}{\varepsilon^2 p_i^2 + 1 - \|\varepsilon p\|^2}} \in [-1, 1]$, and $sign(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$. To simplify the expression above, we define $z(p, \varepsilon) := \sqrt{\varepsilon^2 p_i^2 + 1 - \|\varepsilon p\|^2}$, so we get that $a_i = -sign(p_i) \cdot \frac{\sqrt{1 - \|\varepsilon p\|^2}}{z(p,\varepsilon)}$, *and* $\sqrt{1 - a_i^2} = \frac{\varepsilon |p_i|}{z(p,\varepsilon)} = \frac{\varepsilon p_i \cdot sign(p_i)}{z(p,\varepsilon)}$.
Note that this definition of $\overline{e_i}$ makes $\overline{e_i}$ orthogonal to $p^{\uparrow}$. Indeed, $\overline{e_i}^T \cdot p^{\uparrow} = a_i \cdot \varepsilon p_i + \sqrt{1 - a_i^2} \cdot \sqrt{1 - \|\varepsilon p\|^2} = -sign(p_i) \cdot \frac{\sqrt{1 - \|\varepsilon p\|^2}}{z(p,\varepsilon)} \cdot \varepsilon p_i + \frac{\varepsilon p_i \cdot sign(p_i)}{z(p,\varepsilon)} \cdot \sqrt{1 - \|\varepsilon p\|^2} = 0$, where the first equality follows from the definition $x^{\uparrow} = (\varepsilon x_1, \ldots, \varepsilon x_d, \sqrt{1 - \|\varepsilon x\|^2})$.

The following Lemma implies the correctness of our structure, stated in Theorem 16.

▶ **Lemma 15.** *Let $\varepsilon \in (0, \frac{1}{2})$, $x, p \in B_d$, and a euclidean ellipsoid query $q_0 = (p, \{e_i\}_{i=1}^{d})$, where $\{e_i\}_{i=1}^{d}$ is the standard basis in $\mathbb{R}^d$. Then taking $q = (p^{\uparrow}, \{\overline{e_i}\}_{i=1}^{d})$ as above, for every $i \in [d]$ we have that* $\max(0, \varepsilon(1 - \varepsilon) \cdot (|x_i - p_i| - \beta(\varepsilon))) \leq \angle_i(q, x^{\uparrow}) \leq \varepsilon(1 + \varepsilon) \cdot (|x_i - p_i| + \beta(\varepsilon))$, *where $\angle_i(q, x)$ is the angular distance between x and its projection on the hyperplane orthogonal to e_i (see Figure 2).*

In Section C.2, we show the existence of an angular ellipsoid $(r', c'r')$-ALSH structure, so we conclude the following theorem.

▶ **Theorem 16.** *The structure above is an (r, cr)-ALSH structure for the euclidean ellipsoid distance $d_{\circ}$.*

Our reduction guarantees that any query $q = (p^{\uparrow}, \{\overline{e_i}\}_{i=1}^{d})$ for the angular ellipsoid structure and any data point $x^{\uparrow}$ stored in it, satisfy $\angle(p, x) \leq \sqrt{\frac{c'-1}{c'}} \cdot \frac{\pi}{4}$ as required.

C.2 From angular ellipsoid ALSH to weighted geometric angular similarity SLSH

In this section, we reduce the angular ellipsoid (r, cr)-ALSH problem that we have studied in Section C.1, to a weighted geometric angular similarity $(r', c'r')$-SLSH problem.

C.2.1 H-hash - the LSH scheme of Jain et al.

Our data structure is based on the H-hash of Jain et al. [12]. The H-hash stores points which reside on S_{d+1} such that for a query hyperplane h through the origin, we can efficiently retrieve the data points that have a small angular distance with their projection on h.

H-hash in fact uses an SLSH family for the s2p geometric angular similarity for sets of size 2. That is, a hash function is defined by two random directions u and v. We hash a point x to the concatenation of $sign(x^T u)$ and $sign(x^T v)$ and we represent a query hyperplane h, perpendicular to e, by the set $\{e, -e\}$, which is hashed to the concatenation of $sign(e^T u)$ and $sign((-e)^T v)$.

The probability that a data point x collides with the hyperplane h perpendicular to e is equal to $\angle sim(x, e) \cdot \angle sim(x, -e) = (1 - \angle(x, e)/\pi)(1 - \angle(x, -e)/\pi)$. This collision probability increases with the angle between x and its projection on h, and attains its maximum when x is on h.

Recall that the angular ellipsoid distance between a query $q = (p, \{e_i\}_{i=1}^{d})$ and a point x is a weighted sum of the terms $(\angle_i(q, x))^2$. Therefore, if we hash the hyperplane orthogonal to e_i with H-hash, it will collide with higher probability with data points x with a smaller

$\angle_i(q,x)$. This suggests that we can answer an angular ellipsoid query $q = (p, \{e_i\}_{i=1}^d)$ by a weighted geometric angular similarity SLSH set-query where the set is the union of the sets $\{e_i, -e_i\}$ for all $i \in [d]$, using an appropriate weight w_i for each $i \in [d]$. Specifically, given the parameters $r > 0$ and $c > 1$, we store the data points in an $(S', c'S')$-SLSH structure for the weighted geometric angular s2p similarity for queries of size $k' = 2d$ and with the weights $\{w_1, w_1, w_2, w_2, \ldots, w_d, w_d\}$.[17] We define c' and S' as follows

$$S' = e^{\sum_{i=1}^d w_i \cdot \ln\left(\frac{1}{4}\right) - \frac{4r}{\pi^2 - 4\psi_c^2}}, \text{ and } c' = \frac{e^{\sum_{i=1}^d w_i \ln\left(\frac{1}{4}\right) - \frac{4cr}{\pi^2}}}{S'} = e^{-4r\left(\frac{c}{\pi^2} - \frac{1}{\pi^2 - 4\psi_c^2}\right)},$$

where we define $\psi_c = \sqrt{\frac{c-1}{c}} \cdot \frac{\pi}{4}$.[18] To answer an angular ellipsoid query $q = (p, \{e_i\}_{i=1}^d)$, we query our structure with the set-query $Q = \{e_1, -e_1, e_2, -e_2, \ldots, e_d, -e_d\}$. For the reduction to succeed, we require that any query $q = (p, \{e_i\}_{i=1}^d)$ and data point x satisfy $\angle(p,x) \leq \sqrt{\frac{c-1}{c}} \cdot \frac{\pi}{4}$.

Correctness of our structure follows from the following two theorems.

▶ **Theorem 17.** *Let $x \in S_{d+1}$ and $q = (p, \{e_i\}_{i=1}^d)$ be an angular ellipsoid query. Then, $\angle sim_{geo}(\{e_i, -e_i\}, x) = \frac{1}{4} - \frac{\angle_i(q,x)^2}{\pi^2}$ for all $i \in [d]$.*

▶ **Theorem 18.** *The structure above is an (r, cr)-ALSH structure for the angular ellipsoid distance $d_{\angle\circ}$.*

[17] Such a structure is given in Appendix A.2.

[18] In the full paper we prove that $c' < 1$.