Accurate and Fast Asymmetric Locality-Sensitive Hashing Scheme for Maximum Inner Product Search

Qiang Huang
School of Data and Computer Science, Sun Yat-Sen University
Guangzhou, China
huangq25@mail2.sysu.edu.cn

Guihong Ma
School of Data and Computer Science, Sun Yat-Sen University
Guangzhou, China
maguihong@vip.qq.com

Jianlin Feng
School of Data and Computer Science, Sun Yat-Sen University
Guangzhou, China
fengjlin@mail.sysu.edu.cn

Qiong Fang
School of Software Engineering, South China University of Technology
Guangzhou, China
sefangq@scut.edu.cn

Anthony K. H. Tung
School of Computing, National University of Singapore
Singapore
atung@comp.nus.edu.sg

ABSTRACT

The problem of Approximate Maximum Inner Product (AMIP) search has received increasing attention due to its wide applications. Interestingly, based on asymmetric transformation, the problem can be reduced to the Approximate Nearest Neighbor (ANN) search, and hence leverage Locality-Sensitive Hashing (LSH) to find solution. However, existing asymmetric transformations such as L2-ALSH and XBOX, suffer from large distortion error in reducing AMIP search to ANN search, such that the results of AMIP search can be arbitrarily bad.

In this paper, we propose a novel Asymmetric LSH scheme based on Homocentric Hypersphere partition (H2-ALSH) for high-dimensional AMIP search. On the one hand, we propose a novel Query Normalized First (QNF) transformation to significantly reduce the distortion error. On the other hand, by adopting the homocentric hypersphere partition strategy, we can not only improve the search efficiency with early stop pruning, but also get higher search accuracy by further reducing the distortion error with limited data range. Our theoretical studies show that H2-ALSH enjoys a guarantee on search accuracy. Experimental results over four real datasets demonstrate that H2-ALSH significantly outperforms the state-of-the-art schemes.

CCS CONCEPTS

• **Information systems** → *Information retrieval*; Retrieval tasks and goals;

KEYWORDS

Maximum Inner Product Search, Locality-Sensitive Hashing, Nearest Neighbor Search

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219971

ACM Reference Format:
Qiang Huang, Guihong Ma, Jianlin Feng, Qiong Fang, and Anthony K. H. Tung. 2018. Accurate and Fast Asymmetric Locality-Sensitive Hashing Scheme for Maximum Inner Product Search. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3219819.3219971

1 INTRODUCTION

The Maximum Inner Product (MIP) search in high dimensional space is a fundamental problem which has wide applications in various fields, such as matrix factorization based Recommendation systems [18–20, 26], multi-class label prediction [10, 16], structural SVM [17], and Deep Learning [25]. In most of applications, data objects are typically represented as vectors (or points). Given a database D of n data objects and a query object q in Euclidean space $\mathcal{R}^d$, the problem of MIP search is to find the object $o^* \in D$ maximizing the inner product with q:

$$o^* = \arg\max_{o \in D} \langle o, q \rangle .$$

Due to the difficulty of finding exact query answers in high-dimensional space, the approximate version of the problem, named c-Approximate MIP (c-AMIP) search, has attracted extensive studies [2, 3, 13, 21, 23, 24, 31]. Given an approximation ratio c ($0 < c < 1$) and a query q, the problem of c-AMIP search aims to find an object $o \in D$ such that $\langle o, q \rangle \geq c \langle o^*, q \rangle$, where o^* is the MIP object of q.

The most popular solutions for c-AMIP search are to leverage the power of Locality-Sensitive Hashing (LSH) for solving the problem of Nearest Neighbor (NN) search or Maximum Cosine Similarity (MCS) search. The NN search is to find $o^* \in D$ such that:

$$o^* = \arg\min_{o \in D} \|o - q\|^2 = \arg\max_{o \in D} (\langle o, q \rangle - \frac{\|o\|^2}{2}),$$

and the MCS search is to find $o^* \in D$ such that:

$$o^* = \arg\max_{o \in D} \frac{\langle o, q \rangle}{\|o\|\|q\|} = \arg\max_{o \in D} \frac{\langle o, q \rangle}{\|o\|}.$$

Notice that the results of these three search problems are independent of the norm $\|q\|$.

However, since inner product $\langle \cdot, \cdot \rangle$ is not a metric, LSH schemes [1, 5, 9, 12, 14, 15, 27, 28, 33] cannot be directly adapted to c-AMIP

search. As pointed out by [21], "over the entire space $\mathcal{R}^d$, not only is there no symmetric LSH, but there is also no asymmetric LSH either." Nevertheless, these three search problems are equivalent if all objects $o \in D$ have the same Euclidean norm $\|o\|$. In fact, the state-of-the-art schemes [3, 21, 23, 24] make use of this observation and convert MIP search into NN search or MCS search. Specifically, they apply two vector transformations $P : \mathcal{R}^d \rightarrow \mathcal{R}^{d'}$ and $Q : \mathcal{R}^d \rightarrow \mathcal{R}^{d'}$ on data objects and queries, respectively, to convert MIP search in $\mathcal{R}^d$ into NN search or MCS search in $\mathcal{R}^{d'}$, where $d' > d$. If $P \neq Q$, the transformation is called an asymmetric transformation. For ease of reference, from now on we call a hashing scheme and its corresponding transformation interchangeably.

The state-of-the-art schemes, such as L2-ALSH [23], Sign-ALSH [24], and Simple-LSH [21], solve the c-AMIP search with sub-linear query time. However, there are two limitations of these hashing schemes. We first consider the existing asymmetric transformations which convert MIP search into NN search, i.e., L2-ALSH and XBOX [3]. No matter L2-ALSH or XBOX, a large constant will be added to the Euclidean distance $\|P(o) - Q(q)\|$ between any $P(o)$ and $Q(q)$ after their transformations, which will introduce a distortion error for NN search, in the sense that the Euclidean distance between any $P(o)$ and $Q(q)$ will be close to each other. Any object $P(o)$ can be an approximate NN search result of $Q(q)$, even though the inner product $\langle o, q\rangle$ between o and q is small. Thus, the results of MIP search can be arbitrarily bad.

In addition, L2-ALSH, Sign-ALSH, and Simple-LSH only enjoy a quality guarantee based on two assumptions: (1) data objects are bounded inside the unit sphere and (2) queries are normalized. For the general datasets and queries, we should rescale the data objects and queries to satisfy these two assumptions. Even though this operation will not change the order of MIP results, it will change the distance between data objects and queries after converting MIP search in $\mathcal{R}^d$ into NN/MCS search in $\mathcal{R}^{d'}$, which leads to a loss of accuracy for the approximate NN/MCS search. XBOX does not rely on any assumptions on datasets and queries, but it is a heuristic method without any guarantee.

Motivated by the above limitations, we introduce a novel Asymmetric LSH scheme based on Homocentric Hypersphere partition (H2-ALSH) for high-dimensional c-AMIP search. On the one hand, we propose a novel Query Normalized First (QNF) transformation to significantly reduce the distortion error. On the other hand, to answer c-AMIP queries quickly, we propose a homocentric hypersphere partition strategy to divide the data objects into several disjoint sets to bound the range of norms $\|o\|$. According to Equation 1 as will be introduced in Section 2, a large norm $\|o\|$ leads to a large $\langle o, q\rangle$ value with high probability. Thus, we prefer to first search the set with the largest norm and apply early stop pruning to avoid searching all disjoint sets. In addition, due to this partition strategy, we get higher accuracy by further reducing the distortion error with limited data range.

In summary, we introduce a novel hashing scheme H2-ALSH for high-dimensional c-AMIP search. H2-ALSH not only largely reduces the distortion error, but also accelerates the c-AMIP search with early stop pruning. We demonstrate that H2-ALSH enjoys a guarantee on search accuracy. H2-ALSH also solves the problem of c-approximate k-MIP (c-k-AMIP) search. Extensive experiments show that H2-ALSH significantly outperforms the state-of-the-art schemes, such as L2-ALSH, XBOX, Sign-ALSH, and Simple-LSH.

The rest of the paper is organized as follows. We first review the preliminaries in Section 2. The H2-ALSH scheme is presented in Section 3 and its theoretical analysis is given in Section 4. Experimental studies are depicted in Section 5. Related work is discussed in Section 6. Finally, we conclude our work in Section 7.

2 PRELIMINARIES

2.1 Problem Definition

Let D be a database of n data objects in d-dimensional Euclidean space $\mathcal{R}^d$. For two objects $o = (o_1, o_2, \ldots, o_d)$ and $q = (q_1, q_2, \ldots, q_d)$, the inner product $\langle o, q\rangle$ can be computed as follows:

$$\langle o, q\rangle = \|o\|\|q\| \cos\beta, \tag{1}$$

where $\|o\| = \sqrt{\sum_{i=1}^{d} o_i^2}$ is the Euclidean norm of o; β is the angle between o and q. Formally, the c-AMIP search is defined as follows:

Definition 2.1 (c-AMIP search problem). Given an approximation ratio c ($0 < c < 1$), the problem of c-AMIP search is to construct a data structure which, for any $q \in \mathcal{R}^d$, finds an object $o \in D$ such that $\langle o, q\rangle \geq c\langle o^*, q\rangle$, where o^* is the MIP object of q in D.

Similarly, the problem of c-k-AMIP search is to construct a data structure which, for any query $q \in \mathcal{R}^d$, finds k objects $o_i \in D$ ($1 \leq i \leq k$) such that $\langle o_i, q\rangle \geq c\left\langle o_i^*, q\right\rangle$, where o_i^* is the i^{th} MIP object of q in D.

In this paper, we focus on the transformation which converts c-AMIP search into c_0-Approximate Nearest Neighbor (c_0-ANN) search with Euclidean distance. Formally, the c_0-ANN search is defined as follows:

Definition 2.2 (c_0-ANN search problem). Given an approximation ratio c_0 ($c_0 > 1$), the problem of c_0-ANN search is to construct a data structure which, for any query $q \in \mathcal{R}^d$, finds an object $o \in D$ such that $\|o - q\| \leq c_0\|o^* - q\|$, where o^* is the NN of q in D.

2.2 From MIP Search to NN Search

In this section, we first define the ALSH family. Then, we review two common transformations which reduce MIP search to NN search, i.e., L2-ALSH [23] and XBOX [3].

2.2.1 ALSH family. An ALSH function family (or simply ALSH family) is expected to have the following property: the probability with which any two objects o and q are partitioned into the same bucket increases monotonically as their inner product $\langle o, q\rangle$ increases. Formally, the ALSH family is defined as follows:

Definition 2.3 (ALSH family [23]). Given an inner product threshold S_0 and an approximation ratio c, a hash family $\mathcal{H}$, along with two vector transformations $P : \mathcal{R}^d \rightarrow \mathcal{R}^{d'}$ (Pre-processing transformation) and $Q : \mathcal{R}^d \rightarrow \mathcal{R}^{d'}$ (Query transformation), is said to be (S_0, cS_0, p_1, p_2)-sensitive, if for any $o, q \in \mathcal{R}^d$, $\mathcal{H}$ satisfies the following conditions:

- If $\langle o, q\rangle \geq S_0$, then $\Pr_{h\in\mathcal{H}}[h(P(o)) = h(Q(q))] \geq p_1$.
- If $\langle o, q\rangle \leq cS_0$, then $\Pr_{h\in\mathcal{H}}[h(P(o)) = h(Q(q))] \leq p_2$.
- $0 < c < 1$ and $p_1 > p_2$.

Notice that P is only applied to $o \in D$ in the pre-processing phase and Q is only applied to q in the query phase. For any object o, if $P(o) = Q(o) = o$, the ALSH function is simply an LSH function; if $P(o) = Q(o) \neq o$, the transformation is symmetric. In general, the transformation is asymmetric, i.e., $P(o) \neq Q(o) \neq o$.

2.2.2 L2-ALSH transformation. The L2-ALSH transformation [23] is based on two assumptions on data objects and queries:

- Data objects are bounded inside the unit sphere, i.e., $\|o\| \leq U < 1$ for all $o \in D$, where U is a fixed constant;
- Queries are normalized, i.e., $\|q\| = 1$ for all queries q.

If the 1^{st} or 2^{nd} assumption is not satisfied, we rescale all data objects or normalize q first without changing the order of MIP results. Suppose $[;]$ is the concatenation. The vector transformations $P : \mathcal{R}^d \to \mathcal{R}^{d+m}$ and $Q : \mathcal{R}^d \to \mathcal{R}^{d+m}$ are defined as follows:

$$P(o) = [o; \|o\|^2; \|o\|^4; \ldots; \|o\|^{2^m}], \tag{2}$$

$$Q(q) = [q; 1/2; 1/2; \ldots; 1/2]. \tag{3}$$

Using Equations 2 and 3, we have:

$$\|Q(q) - P(o)\|^2 = 1 + \tfrac{m}{4} - 2\langle o, q\rangle + \|o\|^{2^{m+1}}. \tag{4}$$

Since $\|o\| \leq U < 1$, $\|o\|^{2^{m+1}} \mapsto 0$ as long as m is not too small, i.e., $m \geq 3$. The term $(1 + \frac{m}{4})$ is a fixed constant. Thus, using Equations 2 and 3, the c-MIP search in $\mathcal{R}^d$ can be reduced to the c_0-ANN search in $\mathcal{R}^{d+m}$. However, since there exists a variable term $\|o\|^{2^{m+1}}$ in Equation 4, this transformation is not strictly lossless. L2-ALSH will introduce a *transformation error*, in the sense that the order of c-AMIP results in $\mathcal{R}^d$ cannot be preserved by the order of c_0-ANN results in $\mathcal{R}^{d+m}$.

2.2.3 XBOX transformation. In order to avoid the transformation error, Bachrach et al. propose an asymmetric transformation named XBOX [3]. Suppose M is the maximum norm of all data objects, i.e., $M = \max_{o \in D} \|o\|$. The vector transformations $P : \mathcal{R}^d \mapsto \mathcal{R}^{d+1}$ and $Q : \mathcal{R}^d \mapsto \mathcal{R}^{d+1}$ are defined as follows:

$$P(o) = [o; \sqrt{M^2 - \|o\|^2}], \tag{5}$$

$$Q(q) = [q; 0]. \tag{6}$$

Using Equations 5 and 6, we have:

$$\|Q(q) - P(o)\|^2 = \|q\|^2 + M^2 - 2\langle o, q\rangle. \tag{7}$$

For a specific query q, the term $(\|q\|^2 + M^2)$ is a fixed constant. Thus, compared with L2-ALSH, XBOX is an exact transformation without any loss. Furthermore, since XBOX only introduces one more dimension and it does not require any assumptions on dataset and queries, XBOX is simpler and more accurate than L2-ALSH.

3 THE H2-ALSH SCHEME

In this section, we introduce a novel Asymmetric LSH scheme based on Homocentric Hypersphere partition (H2-ALSH) for the problem of c-AMIP search. We first give an overview of H2-ALSH. Then, we describe the Query Normalized First (QNF) transformation which reduces c-AMIP search to c_0-ANN search. Finally, we introduce the details of the pre-processing phase and query phase of H2-ALSH.

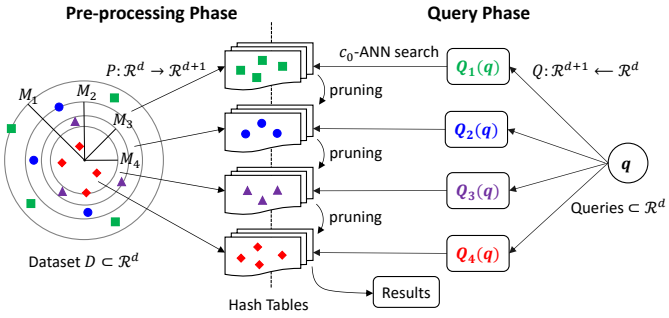


Figure 1: An overview of the H2-ALSH scheme

3.1 Overview

Given a data object o and a query q, based on Equation 1, the inner product $\langle o, q\rangle$ is related to the norm $\|o\|$ and the angle β between o and q. In general, we assume that q is not known beforehand. Thus, it is hard to design a reasonable way to bound the range of β to accelerate the c-AMIP search. However, the norm $\|o\|$ of all $o \in D$ is known in advance. Intuitively, a large norm $\|o\|$ leads to a large value of $\langle o, q\rangle$ with high probability.

Based on the above observation, before applying the vector transformations (i.e., P and Q) on datasets and queries, we propose a homocentric hypersphere partition strategy to bound the range of norm $\|o\|$ for all $o \in D$. Specifically, we partition the data objects into several disjoint sets according to their norms in the pre-processing phase. When the query arrives, we search the disjoint sets from the one with the largest norm to the one with the smallest norm, and apply the early stop pruning strategy for acceleration. An overview of the H2-ALSH scheme is depicted in Figure 1.

3.2 QNF Transformation

3.2.1 Drawbacks of existing transformations. Before we present the QNF transformation, we first revisit the existing transformations L2-ALSH and XBOX of the same kind, which convert c-AMIP search into c_0-ANN search. According to Equations 4 and 7, no matter L2-ALSH or XBOX, a large constant (i.e., $(1 + \frac{m}{4})$ or $(\|q\|^2 + M^2)$) will be added to the Euclidean distance $\|Q(q) - P(o)\|$ between any $P(o)$ and $Q(q)$, which will introduce the *distortion error*, in the sense that the Euclidean distance $\|Q(q) - P(o)\|$ between different $P(o)$ and $Q(q)$ will be close to each other and any objects $P(o)$ can be a c_0-ANN search result of $Q(q)$.

Example 3.1. We now use XBOX as an example to explain the distortion error. Referring to Figure 2, by applying P transformation (i.e., Equation 5), data objects o_1 and o_2 in $\mathcal{R}$ are mapped to $P(o_1)$ and $P(o_2)$ on the circle in $\mathcal{R}^2$ respectively, where the circle is centered at the origin and with a radius M. However, for the queries q_1 and q_2, after Q transformation (i.e., Equation 6), $Q(q_1)$ and $Q(q_2)$ stay in the same position. Consider the query q_1 such that $\|q_1\| \ll M$, the Euclidean distance $\|Q(q_1) - P(o_1)\|$ and $\|Q(q_1) - P(o_2)\|$ are close to each other. Under this situation, XBOX will introduce a large distortion error, because any object of $P(o_1)$ and $P(o_2)$ can be a c_0-ANN of $Q(q_1)$. Similarly, for the query q_2 such that $\|q_2\| \gg M$, the c-AMIP search result returned by XBOX can be arbitrarily bad either.

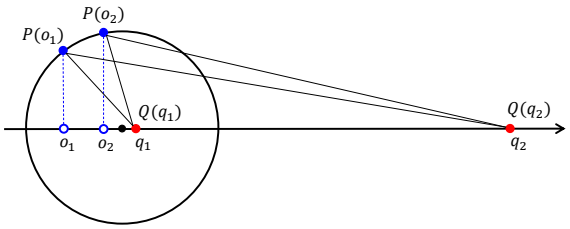


Figure 2: An example of XBOX transformation

3.2.2 QNF transformation. We now introduce the QNF transformation. Suppose M is the maximum norm of all data objects, i.e., $M = \max_{o \in D} \|o\|$. Given a data object $o = (o_1, o_2, \ldots, o_d)$ and a query $q = (q_1, q_2, \ldots, q_d)$, the vector transformations $P : \mathcal{R}^d \mapsto \mathcal{R}^{d+1}$ and $Q : \mathcal{R}^d \mapsto \mathcal{R}^{d+1}$ are defined as follows:

$$P(o) = [o_1, o_2, \ldots, o_d; \sqrt{M^2 - \|o\|^2}], \tag{8}$$

$$Q(q) = [\lambda q_1, \lambda q_2, \ldots, \lambda q_d; 0], \text{where } \lambda = \frac{M}{\|q\|}. \tag{9}$$

Compared with XBOX, we add a parameter λ for the vector transformation Q to tune the norm of $Q(q)$, i.e., $\|Q(q)\| = \lambda \|q\|$. According to Equations 8 and 9, we have:

$$\|Q(q) - P(o)\|^2 = M^2 + \lambda^2 \|q\|^2 - 2\lambda \langle o, q \rangle. \tag{10}$$

For a specific query q, $\lambda = \frac{M}{\|q\|}$ is a fixed constant. Thus, our QNF transformation is exact without transformation error. In Section 4.1, we will demonstrate that by setting $\lambda = \frac{M}{\|q\|}$, the distortion error introduced by our QNF transformation is minimized.

Notice that under the setting of $\lambda = \frac{M}{\|q\|}$, our QNF transformation is logically equivalent to Simple-LSH [21] in the sense of sharing the form of transformation formula. While Simple-LSH is originally designed to reduce MIP search to MCS search, our QNF transformation is an asymmetric transformation which converts MIP search into NN search and it does not require any assumptions on data objects and queries. Thus, it is easier to use. From the perspective of space analytic geometry, since $\|P(o)\| = \|Q(q)\| = M$, our QNF transformation maps all data objects and queries on the $(d+1)$-dimensional hypersphere which is centered at the origin and with a radius M. Example 3.2 shows that, compared with L2-ALSH and XBOX, the distortion error can be significantly reduced by our QNF transformation due to our normalizing query first by setting $\lambda = \frac{M}{\|q\|}$.

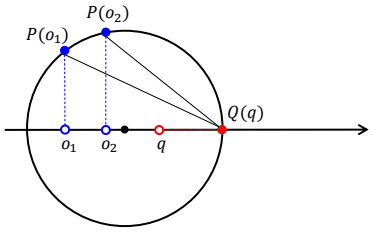


Figure 3: An example of QNF transformation

Example 3.2. Referring to Figure 3, after the QNF transformation, all data objects $P(o_1)$, $P(o_2)$, and the query $Q(q)$ are on the circle in $\mathcal{R}^2$. Under this situation, even though $\|q\| \ll M$ or $\|q\| \gg M$, since q will be mapped to $Q(q)$ which is always on the circle in $\mathcal{R}^2$, the Euclidean distance $\|Q(q) - P(o_1)\|$ and $\|Q(q) - P(o_2)\|$ are no longer close to each other. Thus, the distortion error can be greatly reduced.

3.3 Pre-processing Phase

Given a database D of n data objects, we first compute the Euclidean norm $\|o\|$ for all $o \in D$ and sort the data objects in ascending order according to their norms $\|o\|$. In order to effectively partition the data objects into different disjoint sets $\{S_1, S_2, \ldots, S_K\}$, we first introduce a parameter of interval ratio b, which is used to bound the range of the Euclidean norm $\|o\|$ for each S_i. Given an approximation ratio c ($0 < c < 1$) for c-AMIP search and an approximation ratio c_0 ($c_0 > 1$) for c_0-ANN search, b is computed as follows:

$$b = \sqrt{1 - \frac{1 - c}{c_0^4 - c}}, \tag{11}$$

where $0 < b < 1$. We will explain the setting of b in Theorem 4.2.

Now, we introduce how to partition data objects into disjoint sets. Notice that each S_i corresponds to a maximum norm M_i. At the beginning, since all data objects are sorted in ascending order, M_1 is determined by the norm of the last object in D. We partition data objects $o \in D$ into S_1 such that $bM_1 < \|o\| \leq M_1$ and remove them from D. Then, based on Equation 8, we apply $P : \mathcal{R}^d \mapsto \mathcal{R}^{d+1}$ transformation with M_1, to map all $o \in S_1$ on the $(d+1)$-dimensional hypersphere which is centered at the origin and with a raidus M_1. If the number of objects in S_1 is larger than a prespecified threshold N_0, i.e., $|S_1| \geq N_0$, we apply QALSH [15] to build hash tables for S_1; Otherwise, since $|S_1|$ is small enough, we store all $o \in S_1$ directly without any index. The partitioning process

Algorithm 1: Indexing of H2-ALSH

Input: a database D of n data objects $\{o_1, o_2, \ldots, o_n\}$, an interval ratio b, and a threshold N_0.
Output: the number of disjoint sets K, K disjoint sets $\{S_1, S_2, \ldots, S_K\}$, and their maximum norms $\{M_1, M_2, \ldots, M_K\}$.

```
Compute ||o|| for all o ∈ D;
Sort ||o|| in ascending order;
i = 1; j = n;
while j ≥ 1 do
    M_i = ||o_j||;
    S_i = ∅;
    while j ≥ 1 ∧ ||o_j|| > bM_i do
        P(o_j) = (o_j; sqrt(M_i^2 − ||o_j||^2));
        S_i = S_i ∪ {P(o_j)}; j = j − 1;
    if |S_i| > N_0 then
        Build hash tables for S_i using QALSH;
    i = i + 1;
K = i − 1;
return K, {S_1, S_2, ..., S_K}, and {M_1, M_2, ..., M_K};
```

continues for the remaining objects until all $o \in D$ have already been partitioned into a set S_i. Finally, we return the number of disjoint sets K, the disjoint sets $\{S_1, S_2, \ldots, S_K\}$, and their maximum norms $\{M_1, M_2, \ldots, M_K\}$. Since we partition data objects into different S_i and map them on the $(d+1)$-dimensional hyperspheres which are centered at the same origin but with different radius M_i, we call this way of partition as homocentric hypersphere partition.

The pseudo-code of indexing of H2-ALSH is depicted in Algorithm 1. Notice that $\{M_1, M_2, \ldots, M_K\}$ are sorted in descending order. The K value is automatically determined by the interval ratio b and the distribution of the Euclidean norm of data objects. Instead of mapping all data objects on the $(d+1)$-dimensional hypersphere with the maximum norm M_1, by adopting the homocentric hypersphere partition strategy, the distortion error for the data objects with small Euclidean norm can be further reduced.

3.4 Query Phase

3.4.1 c-AMIP search. To answer a c-AMIP query q, we first compute $\|q\|$ and set up the MIP value φ as $\varphi = -\infty$. Since the set S_i with a larger M_i is more likely to contain the MIP object, H2-ALSH searches the sets from S_1 to S_K according to the descending order of $\{M_1, M_2, \ldots, M_K\}$. For each S_i, it consists of three steps. Firstly, we estimate the upper bound ub of S_i and q. Since all $o \in S_i$ satisfy $bM_i < \|o\| \leq M_i$, based on the Cauchy-Schwarz inequality, we have $\langle o, q \rangle = \|o\|\|q\| \cos\beta \leq \|o\|\|q\| \leq M_i \cdot \|q\|$. Thus, ub can be computed as follows:

$$ub = M_i \cdot \|q\|. \tag{12}$$

Secondly, we use ub for pruning: (1) If $ub \leq \varphi$, we stop searching on S_i and return the MIP object o_{mip} we found so far. Because $\{M_{i+1}, \ldots, M_K\}$ for the remaining sets $\{S_{i+1}, \ldots, S_K\}$ are smaller than M_i and their upper bounds are smaller than ub either. (2) If $ub > \varphi$, we search on S_i. If the number of objects in S_i is small enough, i.e., $|S_i| \leq N_0$, we simply apply the linear scan method on S_i to find the candidate o; Otherwise, we compute $Q(q)$ according to Equation 9 and use QALSH [15] to find the c_0-ANN of $Q(q)$. Thirdly, we add o into the candidate set C, and update φ and o_{mip} accordingly. We return o_{mip} as the final answer. The pseudo-code of H2-ALSH for c-AMIP search is shown in Algorithm 2.

3.4.2 c-k-AMIP search. In order to answer a c-k-AMIP query, we only need to make the following modifications in Algorithm 2:

- We use the k^{th} MIP value φ_k instead of φ for pruning (referring to Line 5);
- For each S_i, we add k objects $\{o_1, o_2, ..., o_k\}$ into C (referring to Lines 8, 11, and 12), and update φ_k and top-k MIP objects $\{o^1_{mip}, o^2_{mip}, \ldots, o^k_{mip}\}$ accordingly (referring to Line 13);
- We return $\{o^1_{mip}, o^2_{mip}, \ldots, o^k_{mip}\}$ (referring to Line 14).

3.4.3 Discussions. The homocentric hypersphere partition strategy is sensitive to the distribution of the Euclidean norm of data objects. If it is a uniform distribution or a skew distribution where most of Euclidean norms of data objects are much smaller than M, our strategy is able to improve the search efficiency significantly with early stop pruning. However, if most of Euclidean norms of data objects are close to M, our strategy may be not effective since most of data objects will fall in S_1 and the early stop pruning will be not effective. We can limit the maximum number of objects in each S_i and apply the homocentric hypersphere partition in a fine granularity (i.e., increase the number of disjoint sets) to alleviate this concentration problem.

Algorithm 2: c-AMIP search of H2-ALSH

Input: a query q, a threshold N_0, the number of disjoint sets K, K disjoint sets $\{S_1, S_2, \ldots, S_K\}$, and their maximum norms $\{M_1, M_2, \ldots, M_K\}$ (descending order).
Output: the MIP object $o_{mip} \subset C$.

1 Compute $\|q\|$;
2 $C = \emptyset$; $\varphi = -\infty$;
3 **for** $i = 1$ *to* K **do**
4 $ub = M_i \cdot \|q\|$;
5 **if** $ub \leq \varphi$ **then**
6 **break**;
7 **if** $|S_i| \leq N_0$ **then**
8 $\{o\}$ = linear_scan(S_i, q);
9 **else**
10 $\lambda = \frac{M_i}{\|q\|}$; $Q(q) = (\lambda q_1, \lambda q_2, \ldots, \lambda q_d; 0)$;
11 $\{o\}$ = QALSH($Q(q)$);
12 $C = C \cup \{o\}$;
13 (φ, o_{mip}) = update(C);
14 **return** o_{mip};

For the c_0-ANN search, we choose QALSH based on two considerations: (1) QALSH is a state-of-the-art scheme which is independent of d; (2) QALSH enjoys a quality guarantee and works with any $c_0 > 1$, which helps us to establish a quality guarantee of H2-ALSH.

4 THEORETICAL ANALYSIS

4.1 Quality Guarantee

We first show that under the setting of $\lambda = \frac{M}{\|q\|}$, the distortion error introduced by our QNF transformation is minimized.

Lemma 4.1. *Given an approximation ratio c ($0 < c < 1$) for c-AMIP search and an approximation ratio c_0 ($c_0 > 1$) for c_0-ANN search, under the setting of $\lambda = \frac{M}{\|q\|}$, the distortion error introduced by the QNF transformation defined by Equations 8 and 9 is minimized.*

Proof. Suppose o^* is the MIP object of q. According to Equation 10, we have:

$$\|Q(q) - P(o^*)\| = \sqrt{M^2 + \lambda^2\|q\|^2 - 2\lambda\langle o^*, q\rangle}.$$

After converting c-AMIP search into c_0-ANN search, we can apply state-of-the-art LSH scheme (i.e., QALSH [15]) and return a c_0^2-ANN of $Q(q)$. Thus, we have:

$$\frac{\|Q(q) - P(o)\|}{\|Q(q) - P(o^*)\|} = \sqrt{\frac{M^2 + \lambda^2\|q\|^2 - 2\lambda\langle o, q\rangle}{M^2 + \lambda^2\|q\|^2 - 2\lambda\langle o^*, q\rangle}} \leq c_0^2.$$

Then, we obtain:

$$\frac{\langle o, q\rangle}{\langle o^*, q\rangle} \geq c_0^4 - \frac{(c_0^4 - 1)(M^2 + \lambda^2\|q\|^2)}{2\lambda\langle o^*, q\rangle}. \tag{13}$$

For c-AMIP search, we aim to return an object o such that $\langle o, q\rangle$ is as large as possible, i.e., $\max \langle o, q\rangle$. Intuitively, the larger $\langle o, q\rangle$ for the MIP object o we return, the smaller the distortion error our QNF transformation introduces. For a specific query q, since $\langle o^*, q\rangle$ is constant, $\max \langle o, q\rangle \Leftrightarrow \max \frac{\langle o,q\rangle}{\langle o^*,q\rangle}$. In other words, if the ratio $\frac{\langle o,q\rangle}{\langle o^*,q\rangle}$ is maximized, the distortion error introduced by our QNF transformation is minimized.

According to Equation 13, we infer that:

$$\begin{aligned} \max \frac{\langle o,q\rangle}{\langle o^*,q\rangle} &\Leftrightarrow \min \frac{(c^4-1)(M^2+\lambda^2\|q\|^2)}{2\lambda\langle o^*,q\rangle} \\ &\Leftrightarrow \min \frac{M^2+\lambda^2\|q\|^2}{\lambda} \end{aligned}$$

Let $f(\lambda) = \frac{M^2+\lambda^2\|q\|^2}{\lambda}$. We take the derivative on $f(\lambda)$ and obtain $f'(\lambda) = -\frac{M^2}{\lambda^2} + \|q\|^2$. Let $f'(\lambda) = 0$. Since $\lambda > 0, \lambda = \frac{M}{\|q\|}$. We derive that when $\lambda = \frac{M}{\|q\|}$, $f(\lambda)$ is minimized and the corresponding ratio $\frac{\langle o,q\rangle}{\langle o^*,q\rangle}$ is maximized. Therefore, the distortion error introduced by our QNF transformation is minimized when $\lambda = \frac{M}{\|q\|}$. □

We now establish a quality guarantee of Algorithm 2.

Theorem 4.2. *Given an approximation ratio c ($0 < c < 1$) for c-AMIP search and an approximation ratio c_0 ($c_0 > 1$) for c_0-ANN search, by setting b according to Equation 11, Algorithm 2 returns a c-AMIP object with probability at least $(\frac{1}{2} - \frac{1}{e})$.*

Proof. We first derive an expression for $\frac{\langle o,q\rangle}{\langle o^*,q\rangle}$. Suppose o^* is the MIP object of q and o^* falls in a set S_i, i.e., $bM_i < \|o^*\| \le M_i$ and $\lambda = \frac{M_i}{\|q\|}$. According to Equation 10, we have

$$\|Q(q) - P(o^*)\|^2 = 2{M_i}^2 - \frac{2M_i}{\|q\|}\langle o^*, q\rangle .$$

After reduced to c_0-ANN search, according to Theorem 1 in [15], QALSH returns a c_0^2-ANN object $P(o)$ of $Q(q)$. Thus, $\frac{\|Q(q)-P(o)\|}{\|Q(q)-P(o^*)\|} \le c_0^2$. Let β^* be the angle between o^* and q. According to Equation 1, we have:

$$\frac{\langle o,q\rangle}{\langle o^*,q\rangle} \ge c_0^4 - \frac{(c_0^4-1)\cdot M_i\cdot\|q\|}{\langle o^*,q\rangle} \ge c_0^4 - \frac{c_0^4-1}{b\cos\beta^*}.$$

$\frac{\langle o,q\rangle}{\langle o^*,q\rangle}$ is the approximation ratio c. According to Definition 2.1, we need $c_0^4 - \frac{c_0^4-1}{b\cos\beta^*} \ge 0$, i.e., $\cos\beta^* \in [\frac{c_0^4-1}{bc_0^4}, 1]$. However, since β^* is related to q, the value of $\cos\beta^*$ always varies by q. Thus, we estimate c in the average case instead, i.e., $c = E\left[\frac{\langle o,q\rangle}{\langle o^*,q\rangle}\right]$.

Secondly, we derive an expression for $E\left[\frac{\langle o,q\rangle}{\langle o^*,q\rangle}\right]$. Suppose that o_0 is the object with minimum angle of q and β_0 is the angle between o_0 and q. Since o^* is the MIP object of q, $\langle o^*, q\rangle \ge \langle o_0, q\rangle$. According to Equation 1, $\|o^*\|\|q\|\cos\beta^* \ge \|o_0\|\|q\|\cos\beta_0$. Thus, we have:

$$\cos\beta^* \ge \frac{\|o_0\|}{\|o^*\|}\cos\beta_0 \ge \frac{bM_i}{M_i}\cos\beta_0 = b\cos\beta_0.$$

Now, we obtain a lower bound of c by computing $E[\frac{1}{\cos\beta^*}]$:

$$\begin{aligned} E\left[\frac{\langle o,q\rangle}{\langle o^*,q\rangle}\right] &= c_0^4 - \frac{c_0^4-1}{b}E[\tfrac{1}{\cos\beta^*}], \cos\beta^* \in [\tfrac{c_0^4-1}{bc_0^4}, 1], \\ &\ge c_0^4 - \frac{c_0^4-1}{b^2}E[\tfrac{1}{\cos\beta_0}], \cos\beta_0 \in [\tfrac{c_0^4-1}{c_0^4}, 1]. \end{aligned}$$

Since $0 < b < 1$, $\frac{c_0^4-1}{bc_0^4} \ge \frac{c_0^4-1}{c_0^4}$ and the interval $[\frac{c_0^4-1}{c_0^4}, 1]$ is larger than $[\frac{c_0^4-1}{bc_0^4}, 1]$. Thus, the integral of $E[\frac{1}{\cos\beta_0}]$ under $[\frac{c_0^4-1}{c_0^4}, 1]$ is larger than that of $E[\frac{1}{\cos\beta^*}]$ under $[\frac{c_0^4-1}{bc_0^4}, 1]$. Thus, the range of β_0 is determined, i.e., $\beta_0 \in [0, \arccos((c_0^4-1)/c_0^4)]$.

Finally, we derive a lower bound for $E\left[\frac{\langle o,q\rangle}{\langle o^*,q\rangle}\right]$. Let β be the angle between any object $o \in S_i$ and q. According to Definition 2.1, since $c > 0$, we only consider the objects whose angle β with q are less than $\frac{\pi}{2}$. Suppose β is distributed uniformly in $[0, \frac{\pi}{2}]$, i.e., $\Pr[\beta \le \alpha] = \frac{\alpha}{\pi/2}$. Thus, $\Pr[\beta > \alpha] = 1 - \Pr[\beta \le \alpha] = 1 - \frac{\alpha}{\pi/2}$.

Since β_0 is the minimum angle between all $o \in S_i$ and q and $|S_i| \le n$, we have $\Pr[\beta_0 > \alpha] = \Pr[\beta_1 > \alpha, \ldots, \beta_n > \alpha] = (1 - \frac{\alpha}{\pi/2})^n$. Then, we obtain the cumulative density function $F_{\beta_0}(\alpha)$ of β_0:

$$F_{\beta_0}(\alpha) = \Pr[0 \le \beta_0 \le \alpha] = 1 - (1 - \tfrac{\alpha}{\pi/2})^n.$$

We take the derivative on $F_{\beta_0}(\alpha)$ and obtain the probability distribution function $f_{\beta_0}(\alpha)$ of β_0:

$$f_{\beta_0}(\alpha) = F'_{\beta_0}(\alpha) = \tfrac{n}{\pi/2}(1 - \tfrac{\alpha}{\pi/2})^{n-1}.$$

Since $\beta_0 \in [0, \arccos((c_0^4-1)/c_0^4)]$, we have

$$\begin{aligned} E[\tfrac{1}{\cos\beta_0}] &= \int_0^{\arccos((c_0^4-1)/c_0^4)} \tfrac{1}{\cos\alpha} f_{\beta_0}(\alpha)\, d\alpha \\ &= \int_0^{\arccos((c_0^4-1)/c_0^4)} \tfrac{1}{\cos\alpha}\tfrac{n}{\pi/2}(1 - \tfrac{\alpha}{\pi/2})^{n-1}\, d\alpha. \end{aligned}$$

Notice that the exact value of $E[\frac{1}{\cos\beta_0}]$ is hard to compute. Thus, we compute its upper bound instead. Since $0 \le \alpha \le \arccos((c_0^4 - 1)/c_0^4) < \frac{\pi}{2}$, $\cos\alpha \ge 1 - \frac{\alpha}{\pi/2}$. Thus, we have:

$$\begin{aligned} E[\tfrac{1}{\cos\beta_0}] &= \int_0^{\arccos((c_0^4-1)/c_0^4)} \tfrac{1}{\cos\alpha}\tfrac{n}{\pi/2}(1 - \tfrac{\alpha}{\pi/2})^{n-1}\, d\alpha \\ &\le \int_0^{\arccos((c_0^4-1)/c_0^4)} \tfrac{1}{1-\frac{\alpha}{\pi/2}}\tfrac{n}{\pi/2}(1 - \tfrac{\alpha}{\pi/2})^{n-1}\, d\alpha \\ &= \tfrac{n}{n-1}\left[1 - \left(1 - \tfrac{\arccos((c_0^4-1)/c_0^4)}{\pi/2}\right)^{n-1}\right] \le \tfrac{n}{n-1}. \end{aligned}$$

Then, we have:

$$E\left[\frac{\langle o,q\rangle}{\langle o^*,q\rangle}\right] \ge c_0^4 - \frac{c_0^4-1}{b^2}E\left[\frac{1}{\cos\beta_0}\right] \ge c_0^4 - \frac{c_0^4-1}{b^2}\frac{n}{n-1}.$$

When n is large, $\frac{n}{n-1} \mapsto 1$. Thus, $c = c_0^4 - \frac{c_0^4-1}{b^2}$. The interval ratio b can be computed as follows:

$$b = \sqrt{1 - \frac{1-c}{c_0^4 - c}}.$$

According to Theorem 1 in [15], QALSH returns a c_0^2-ANN object o with probability at least $(\frac{1}{2} - \frac{1}{e})$ if we fix the error probability of QALSH to be $\frac{1}{e}$. Then, Algorithm 2 could return a c-MIP object with the same probability. Therefore, Theorem 4.2 is proved. □

Corollary 4.3. *Algorithm 2 works with any approximation ratio $0 < c < 1$.*

Proof. Referring to Theorem 4.2. During the proof of Theorem 4.2, there is no limitation on the approximation ratio c. Thus, Algorithm 2 works with any $0 < c < 1$. □

Notice that H2-ALSH correctly answers c-AMIP queries with probability at least $(\frac{1}{2} - \frac{1}{e})$, which is not high enough. Using the standard boosting strategy, we can increase the success probability of Algorithm 2 to $(1 - \tau)$ by repeating it $O(\log \tau)$ times.

The two values c and c_0 are set independently. However, the search performance of H2-ALSH is still influenced by c_0. According to Equation 11, the interval ratio b increases monotonically as c_0 increases. If c_0 is close to 1, b will be close to 0. Then, most of data objects will fall in S_1 and all of them can be the c-AMIP candidate of q. Thus, in order to keep the search accuracy for c, we require a low value of c_0 to return a high accurate c_0-ANN result. However, a low c_0 value will increase the query time of QALSH [15]. Thus, we recommend to set up a moderate value of c_0 (i.e., $c_0 = 2.0$) to get a large b value, so that we can reduce the query time of QALSH and utilize the early stop pruning for further acceleration.

4.2 Space and Query Time Complexities

Theorem 4.4. *Given an approximation ratio c ($0 < c < 1$) for c-AMIP search and an approximation ratio c_0 ($c_0 > 1$) for c_0-ANN search, Algorithm 2 returns a c-AMIP object which uses $O(nd+n\log n)$ space and $O(n\log n)$ query time.*

Proof. The space overhead of Algorithm 2 consists of two parts: the space of dataset $O(nd)$ and the space to store hash tables for K disjoint sets $\{S_1, S_2, \ldots, S_K\}$, where $\sum_{i=1}^{K} |S_i| = n$. According to [15], QALSH builds hash tables for each S_i with $O(|S_i| \log |S_i|)$. Since $|S_i| \leq n$, we have $\sum_{i=1}^{K} |S_i| \log |S_i| \leq \sum_{i=1}^{K} |S_i| \log n \leq n \log n$. Thus, the space overhead of Algorithm 2 is $O(nd + n \log n)$.

To answer a c-AMIP query, in the worst case, Algorithm 2 needs to check objects in all K disjoint sets, or all data objects fall in S_1. According to [15], QALSH uses $O(|S_i| \log |S_i|)$ to answer a c_0-ANN query for each S_i. Since $|S_i| \leq n$, we have $\sum_{i=1}^{K} |S_i| \log |S_i| \leq \sum_{i=1}^{K} |S_i| \log n \leq n \log n$. Therefore, the query time complexity of Algorithm 2 is $O(n \log n)$. □

Notably, the query time complexity of H2-ALSH is analysed under the worst case. However, for the real datasets in high-dimensional space, data objects are sparse and often lie on a manifold. Thus, in most cases, H2-ALSH stops early in the first few disjoint sets. Even for the concentration problem where all data objects fall in S_1, i.e., the dataset Sift[1] used in Section 5, our results show that H2-ALSH is more efficient than the competitors. Thus, the actual query time of H2-ALSH will be much better than the theoretical bound.

5 EXPERIMENTS

In this section, we study the performance of H2-ALSH using four real datasets. All methods are implemented in C++ using -O3 optimization. We conduct all experiments on a machine with Intel(R) Xeon(R) CPU E5-2603 1.70GHz, 16 GB main memory, and 1 TB hard disk, running under Ubuntu 16.04.

5.1 Experiment Setup

5.1.1 Benchmark methods. We compare **H2-ALSH** with four state-of-the-art schemes: **L2-ALSH** [23], **XBOX** [3], **Sign-ALSH** [24], and **Simple-LSH** [21]. Since their implementations are not available, we implement all of them according to [3, 21, 23, 24]. To make a fair comparison with H2-ALSH, we adapt L2-ALSH and XBOX to use QALSH for c_0-ANN search after their asymmetric transformations. We also run another version of H2-ALSH named **H2-ALSH$^-$** without homocentric hypersphere partition to evaluate the effectiveness of this strategy.

Table 1: Statistics of datasets

Datasets	#Objects	#Queries	d	Data Size
Sift	1,000,000	1,000	128	337.8 MB
Netflix	17,770	1,000	300	50.8 MB
Yahoo	624,961	1,000	300	2.3 GB
Gist	1,000,000	1,000	960	4.0 GB

5.1.2 Datasets and queries. We use four real datasets **Sift**, **Netflix** [4], **Yahoo** [11], and **Gist**[2] in our experiments. For the collaborative filtering datasets Netflix and Yahoo, each one is a sparse user-item rating matrix R, where $R(i, j)$ is the rating of user i for item j. We apply PureSVD [6] on R to generate user vectors and latent item vectors, and follow [21, 23, 24] to set up the latent dimension d, i.e., $d = 300$. Then, we randomly select 1,000 user vectors from the user matrix and use them as queries. For Sift and Gist, we randomly select 1,000 objects from their test sets and use them as queries. The statistics of the datasets are summarized in Table 1.

5.1.3 Evaluation metrics. We use the following metrics for performance evaluation. They are averaged over all queries.

- **Recall.** We follow L2-ALSH [23], Sign-ALSH [24], and Simple-LSH [21] to use recall to measure the accuracy of a method.
- **Overall Ratio.** Since H2-ALSH enjoys a quality guarantee on approximation ratio c, we use overall ratio to access the accuracy of a method either. For the c-k-AMIP search, it is defined as: $\frac{1}{k}\sum_{i=1}^{k} \frac{\langle o_i, q\rangle}{\langle o_i^*, q\rangle}$, where o_i is i^{th} MIP object returned by a method and o_i^* is the exact i^{th} MIP object, $i \in \{1, 2, \ldots, k\}$. Intuitively, a larger overall ratio means a higher accuracy.
- **Running Time.** We use the running time to evaluate the efficiency of a method. It is defined as the wall clock time for a method to solve the c-k-AMIP problem.

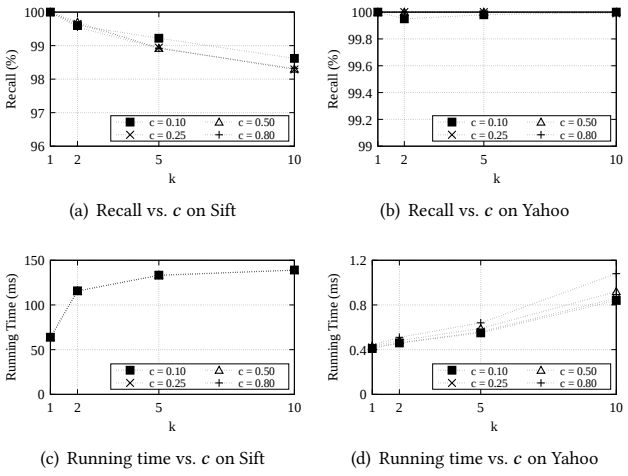
5.2 Parameter Settings

We study the performance of H2-ALSH under different parameter settings, i.e., approximation ratio c for c-AMIP search and approximation ratio c_0 for c_0-ANN search. For QALSH, we use its default settings [15]. We set the threshold $N_0 = 100$ so that both QALSH and linear scan check the same number of candidates for c_0-ANN search for each S_i.

5.2.1 Impact of approximation ratio c. We study how the performance of H2-ALSH varies with c. We fix $c_0 = 2.0$ and consider $c = \{0.10, 0.25, 0.50, 0.80\}$. To be concise, we only show results on the datasets Sift and Yahoo. We observe similar trends from the results on Netflix and Gist. From Figure 4, the recalls under the

[1] http://corpus-texmex.irisa.fr/

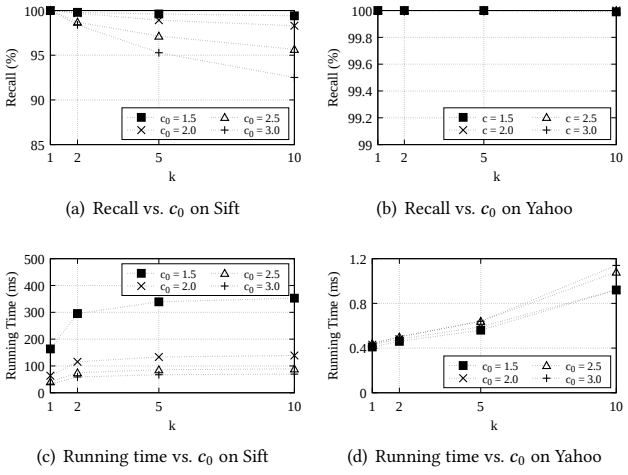
[2] http://corpus-texmex.irisa.fr/



(a) Recall vs. c on Sift (b) Recall vs. c on Yahoo

(c) Running time vs. c on Sift (d) Running time vs. c on Yahoo

Figure 4: H2-ALSH vs. approximation ratio c



(a) Recall vs. c_0 on Sift (b) Recall vs. c_0 on Yahoo

(c) Running time vs. c_0 on Sift (d) Running time vs. c_0 on Yahoo

Figure 5: H2-ALSH vs. approximation ratio c_0

four settings of c are close to each other. Meanwhile, the running time under $c = 0.5$ on Yahoo is smaller than that under $c = 0.8$ and is close to that under $c = 0.1$. Thus, under the setting $c = 0.5$, H2-ALSH enjoys a better trade-off between search accuracy and search efficiency.

5.2.2 Impact of approximation ratio c_0. We study how the performance of H2-ALSH varies with c_0. We fix $c = 0.5$ and consider $c_0 = \{1.5, 2.0, 2.5, 3.0\}$. Due to space limitations, we only show results on the datasets Sift and Yahoo. From Figure 5, the results on Sift are more sensitive to c_0 than those on Yahoo. From the results on Sift, we observe that H2-ALSH has a better trade-off between search accuracy and search efficiency under the setting $c_0 = 2.0$.

5.2.3 Summary. Based on the above results, we use the settings $c = 0.5$ and $c_0 = 2$ for H2-ALSH and apply the same parameter settings for H2-ALSH$^-$ in the subsequent experiments. For the benchmark methods, we use the settings suggested by the authors [21, 23, 24] to achieve the best performance, i.e., $m = 3$ and $U = 0.83$ for L2-ALSH, $m = 3, U = 0.85, K = 512$ for Sign-ALSH, and $K = 512$ for Simple-LSH. XBOX does not require any parameters.

5.3 Results and Analysis

We study the performance of all methods for 0.5-k-AMIP search by varying k from 1 to 10. The results are presented in Figure 6.

5.3.1 Recall and overall ratio. From Figures 6(a)-6(d), H2-ALSH enjoys highest recalls, which are close to 100%. The recalls of H2-ALSH$^-$ are higher than those of L2-ALSH and XBOX either. These results demonstrate that, compared with L2-ALSH and XBOX, the distortion error can be reduced by our QNF transformation. In addition, the homocentric hypersphere partition strategy we design for H2-ALSH can further reduce the distortion error. This also explains why the recalls of H2-ALSH are higher than those of Simple-LSH, even though our QNF transformation is logically equivalent to Simple-LSH.

From Figures 6(e)-6(h), the results on overall ratio show the same trends as those on recall. The overall ratios of H2-ALSH are close to 1, which are much larger than the theoretical bound $c = 0.5$.

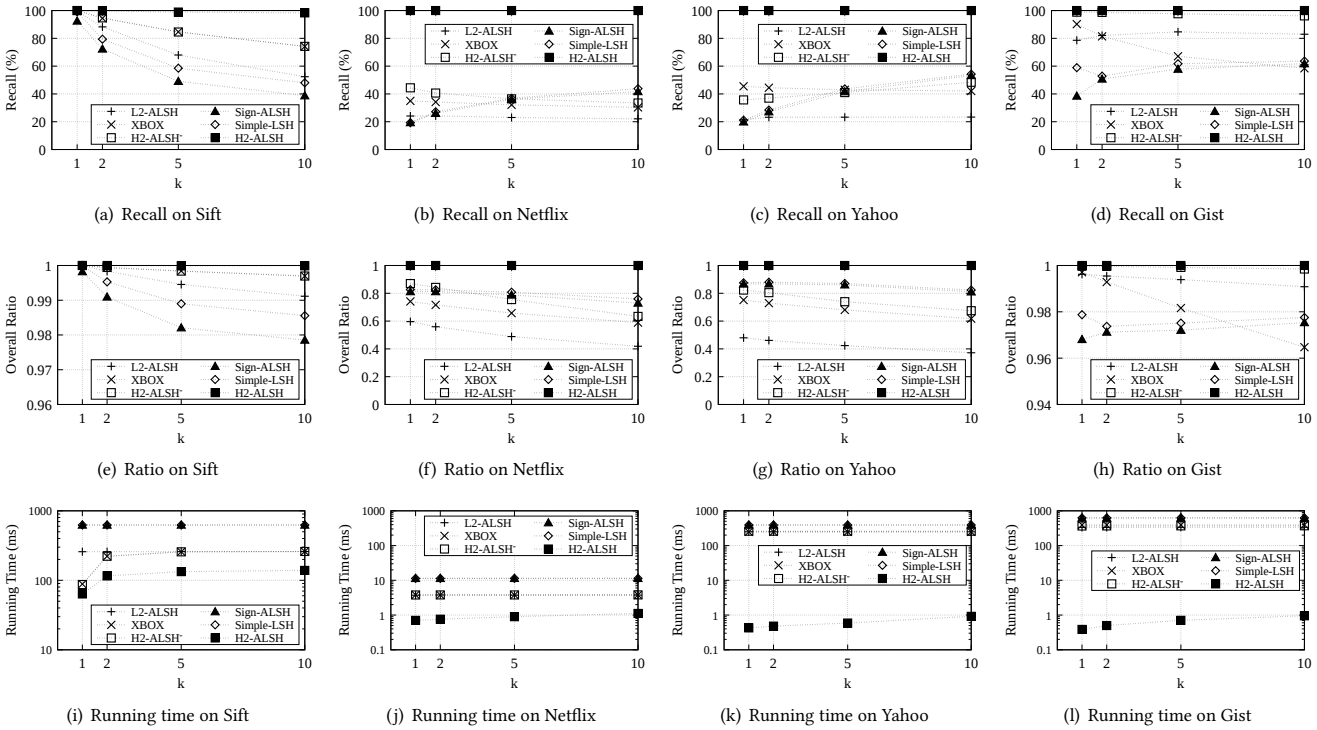
5.3.2 Running time. From Figures 6(i)-6(l), the running time of H2-ALSH is the smallest among all methods. Specifically, for Yahoo and Gist, the running time of H2-ALSH is much smaller than that of other schemes by about three orders of magnitude. However, the advantage of H2-ALSH over Sift is less apparent. This can be explained by the distribution of Euclidean norm of data objects for the datasets. From Figure 7, for Yahoo and Gist, only a little percentage of data objects fall in the first few largest norm. Thus, we can apply the homocentric hypersphere partition strategy with effectively early stop pruning for acceleration. However, since all data objects in Sift fall in S_1, the early stop pruning is not effective. Thus, the advantage of H2-ALSH over Sift is less apparent.

The running time of H2-ALSH$^-$ is close to that of L2-ALSH and XBOX. Because we did not implement the homocentric hypersphere partition on H2-ALSH$^-$ and all of them use QALSH for c_0-ANN search. For Sign-ALSH and Simple-LSH, according to [21, 24], they require to compute Hamming distance between all data objects and q and sort all of them. Therefore, the running time of Sign-ALSH and Simple-LSH is larger than that of other schemes.

5.3.3 Precision-recall curve. In order to verify the effectiveness of our implementations of Sign-ALSH and Simple-LSH, we follow [21, 24] and add an experiment to plot the precision-recall curves under two collaborative filtering datasets Netflix and Yahoo.

From Figure 8, the precision-recall curves of Sign-ALSH and Simple-LSH under Netflix are consistent with the results presented in [21, 24]. Furthermore, we observe that the precision-recall curves of H2-ALSH are much higher than those of Sign-ALSH and Simple-LSH. The results further demonstrate that the search accuracy of H2-ALSH is much higher than that of Sign-ALSH and Simple-LSH.

5.3.4 Summary. Based on the experimental results, we have two important observations. Firstly, H2-ALSH Significantly outperforms the state-of-the-art methods, such as L2-ALSH, XBOX, Sign-ALSH, and Simple-LSH. The recalls of H2-ALSH are close to 100%, which are much higher than those of the competitors. Due to the use of

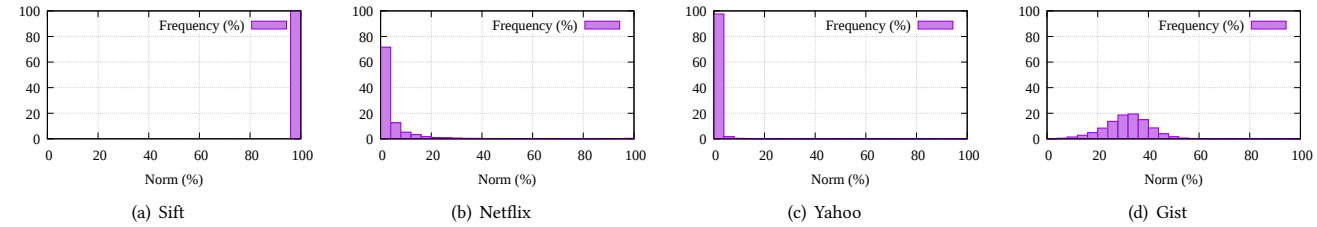


(a) Recall on Sift (b) Recall on Netflix (c) Recall on Yahoo (d) Recall on Gist

(e) Ratio on Sift (f) Ratio on Netflix (g) Ratio on Yahoo (h) Ratio on Gist

(i) Running time on Sift (j) Running time on Netflix (k) Running time on Yahoo (l) Running time on Gist

Figure 6: Experimental results



(a) Sift (b) Netflix (c) Yahoo (d) Gist

Figure 7: Distribution of the Euclidean norm of the data objects

homocentric hypersphere partition strategy, the running time of H2-ALSH is much smaller than that of other schemes by up to three orders of magnitude. Secondly, H2-ALSH works with any approximation ratio $0 < c < 1$.

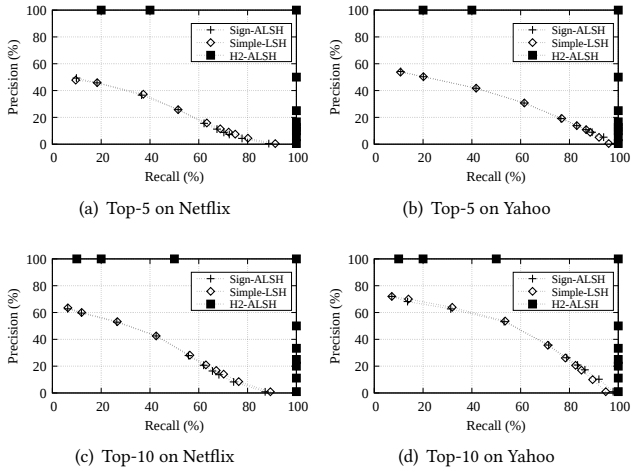
6 RELATED WORKS

The Maximum Inner Product (MIP) search is a fundamental problem. The most popular type of methods for exact MIP search is the tree-based methods, such as ball tree [22], metric-tree [18], and cover-tree [7, 8]. Another popular solutions is the linear scan based methods, i.e., LEMP [29, 30] and FEXIPRO [20]. These exact methods are suitable for the low or moderate data dimension. However, their performance will degrade rapidly as the dimensionality of dataset increases [20, 23, 32].

Due to the hardness of exact MIP search in high-dimensional space, c-AMIP search has attracted increased studies [2, 3, 13, 21, 23, 24, 31] recently. Since inner product is not a metric, the approximate methods usually apply a transformation to convert MIP search into NN search or MCS search.

Shrivastava and Li introduce the concept of Asymmetric LSH (ALSH) and propose the first ALSH scheme named L2-ALSH [23] with provable sub-linear query time. L2-ALSH converts MIP search into NN search and then solves this problem by E2LSH [9]. Later, they develop another asymmetric transformation named Sign-ALSH [24] which reduces MIP search to MCS search, and then solves the MCS search problem by SimHash [5]. However, both L2-ALSH and Sign-ALSH are not the exact transformations. They will introduce the transformation error such that the order of MIP search results cannot be preserved by the order of NN/MCS search results.

To avoid the transformation error, Bachrach et al. propose an exact asymmetric transformation named XBOX [3]. XBOX converts MIP search into NN search and solves this problem by PCA-Tree [3]. However, XBOX and L2-ALSH will introduce distortion error such



(a) Top-5 on Netflix (b) Top-5 on Yahoo

(c) Top-10 on Netflix (d) Top-10 on Yahoo

Figure 8: Precision-recall curves of Top-k MIP search results (higher is better)

that the Euclidean distance between data objects and the query after the transformations will be close to each other.

The above transformations are asymmetric. Recently, Neyshabur et al. propose a symmetric transformation named Simple-LSH [21]. Simple-LSH reduces MIP search to MCS search and uses SimHash [5] to solve this problem. Notice that our QNF transformation is logically equivalent to Simple-LSH in the sense of sharing the form of transformation formula. However, our QNF transformation is an asymmetric transformation which converts MIP search into NN search. Furthermore, we design a homocentric hypersphere partition strategy to partition the data objects into disjoint sets to limit the range of data, so that we can apply the QNF tranformation to further reduce the distortion error. Thus, H2-ALSH enjoys much higher accuracy than Simple-LSH and others schemes. In fact, since our homocentric hypersphere partition strategy is an isolated step before the transformation we apply, the transformations of the same kind, i.e., L2-ALSH and XBOX, can combine with this strategy to increase the search accuracy.

7 CONCLUSIONS

In this paper, we introduce a novel hashing scheme H2-ALSH for high-dimensional c-AMIP search. Compared with L2-ALSH and XBOX, H2-ALSH reduces the distortion error significantly by using the QNF transformation and the homocentric hypersphere partition strategy we design. In addition, by using the homocentric hypersphere partition strategy, H2-ALSH can accelerate the c-AMIP search with early stop pruning. H2-ALSH enjoys a guarantee on search accuracy and works with any approximation ratio $0 < c < 1$. Extensive experiments over four real datasets demonstrate the superior performance of H2-ALSH.

ACKNOWLEDGMENTS

This work was supported in part by NSFC under Grant No: 61772563 and 61602186, and the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO, at the SeSaMe Centre.

REFERENCES

[1] Alexandr Andoni and Piotr Indyk. 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*. 459–468.
[2] Alex Auvolat, Sarath Chandar, Pascal Vincent, Hugo Larochelle, and Yoshua Bengio. 2015. Clustering is efficient for approximate maximum inner product search. *arXiv preprint arXiv:1507.05910* (2015).
[3] Yoram Bachrach, Yehuda Finkelstein, Ran Gilad-Bachrach, Liran Katzir, Noam Koenigstein, Nir Nice, and Ulrich Paquet. 2014. Speeding up the xbox recommender system using a euclidean transformation for inner-product spaces. In *RecSys*. 257–264.
[4] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*. 35.
[5] Moses S Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *STOC*. 380–388.
[6] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*. 39–46.
[7] Ryan R Curtin and Parikshit Ram. 2014. Dual-tree fast exact max-kernel search. *Statistical Analysis and Data Mining* 7, 4 (2014), 229–253.
[8] Ryan R Curtin, Parikshit Ram, and Alexander G Gray. 2013. Fast exact max-kernel search. In *ICDM*. 1–9.
[9] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni. 2004. Locality-sensitive hashing scheme based on p-stable distributions. In *SoCG*. 253–262.
[10] Thomas Dean, Mark A Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. 2013. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*. 1814–1821.
[11] Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. 2012. The Yahoo! Music Dataset and KDD-Cup'11.. In *KDD Cup*. 8–18.
[12] Junhao Gan, Jianlin Feng, Qiong Fang, and Wilfred Ng. 2012. Locality-sensitive hashing scheme based on dynamic collision counting. In *SIGMOD*. 541–552.
[13] Ruiqi Guo, Sanjiv Kumar, Krzysztof Choromanski, and David Simcha. 2016. Quantization based fast inner product search. In *Artificial Intelligence and Statistics*. 482–490.
[14] Qiang Huang, Jianlin Feng, Qiong Fang, et al. 2017. Query-aware locality-sensitive hashing scheme for l_p norm. *The VLDB Journal* 26, 5 (2017), 683–708.
[15] Qiang Huang, Jianlin Feng, Yikai Zhang, Qiong Fang, and Wilfred Ng. 2015. Query-aware locality-sensitive hashing for approximate nearest neighbor search. *Proceedings of the VLDB Endowment* 9, 1 (2015), 1–12.
[16] Prateek Jain and Ashish Kapoor. 2009. Active learning for large multi-class problems. In *CVPR*. 762–769.
[17] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. 2009. Cutting-plane training of structural SVMs. *Machine Learning* 77, 1 (2009), 27–59.
[18] Noam Koenigstein, Parikshit Ram, and Yuval Shavitt. 2012. Efficient retrieval of recommendations in a matrix factorization framework. In *CIKM*. 535–544.
[19] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).
[20] Hui Li, Tsz Nam Chan, Man Lung Yiu, and Nikos Mamoulis. 2017. FEXIPRO: Fast and Exact Inner Product Retrieval in Recommender Systems. In *SIGMOD*. 835–850.
[21] Behnam Neyshabur and Nathan Srebro. 2015. On Symmetric and Asymmetric LSHs for Inner Product Search. In *ICML*. 1926–1934.
[22] Parikshit Ram and Alexander G Gray. 2012. Maximum inner-product search using cone trees. In *SIGKDD*. 931–939.
[23] Anshumali Shrivastava and Ping Li. 2014. Asymmetric LSH (ALSH) for sublinear time Maximum Inner Product Search (MIPS). In *NIPS*. 2321–2329.
[24] Anshumali Shrivastava and Ping Li. 2015. Improved asymmetric locality sensitive hashing (ALSH) for Maximum Inner Product Search (MIPS). In *UAI*. 812–821.
[25] Ryan Spring and Anshumali Shrivastava. 2017. Scalable and sustainable deep learning via randomized hashing. In *SIGKDD*. 445–454.
[26] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. 2005. Maximum-margin matrix factorization. In *NIPS*. 1329–1336.
[27] Yifang Sun, Wei Wang, Jianbin Qin, Ying Zhang, and Xuemin Lin. 2014. SRS: solving c-approximate nearest neighbor queries in high dimensional euclidean space with a tiny index. *Proceedings of the VLDB Endowment* 8, 1 (2014), 1–12.
[28] Yufei Tao, Ke Yi, Cheng Sheng, and Panos Kalnis. 2009. Quality and efficiency in high dimensional nearest neighbor search. In *SIGMOD*. 563–576.
[29] Christina Teflioudi and Rainer Gemulla. 2016. Exact and approximate maximum inner product search with lemp. *ACM TODS* 42, 1 (2016), 5.
[30] Christina Teflioudi, Rainer Gemulla, and Olga Mykytiuk. 2015. LEMP: Fast retrieval of large entries in a matrix product. In *SIGMOD*. 107–122.
[31] Sudheendra Vijayanarasimhan et al. 2014. Deep networks with large output spaces. *arXiv preprint arXiv:1412.7479* (2014).
[32] Roger Weber, Hans-Jörg Schek, and Stephen Blott. 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, Vol. 98. 194–205.
[33] Yuxin Zheng, Qi Guo, Anthony KH Tung, and Sai Wu. 2016. Lazylsh: Approximate nearest neighbor search for multiple distance functions with a single index. In *SIGMOD*. 2023–2037.