Sublinear Time Nearest Neighbor Search over Generalized Weighted Manhattan Distance

Huan Hu  
Harbin Institute of Technology, China  
hit_huhuan@foxmail.com

Jianzhong Li  
Harbin Institute of Technology, China  
lijzh@hit.edu.cn

Abstract

Nearest Neighbor Search (NNS) over generalized weighted distances is fundamental to a wide range of applications. The problem of NNS over the generalized weighted square Euclidean distance has been studied in previous work. However, numerous studies have shown that the Manhattan distance could be more effective than the Euclidean distance for high-dimensional NNS, which indicates that the generalized weighted Manhattan distance is possibly more practical than the generalized weighted square Euclidean distance in high dimensions. To the best of our knowledge, no prior work solves the problem of NNS over the generalized weighted Manhattan distance in sublinear time. This paper achieves the goal by proposing two novel hashing schemes ($d_{l_1} w$, $l_2$)-ALSH and ($d_{l_1} w$, $\theta$)-ALSH.

1 Introduction

Nearest Neighbor Search (NNS) over generalized weighted distances is fundamental to a wide variety of applications, such as personalized recommendation [11][14][18] and kNN classification [3][19]. Given a set of $n$ data points $D \subset \mathbb{R}^d$ and a query point $q \in \mathbb{R}^d$ with a weight vector $w \in \mathbb{R}^d$, NNS over a generalized weighted distance, denoted by $d_w$, is to find a point $o^* \in D$ such that $o^*$ is the closest data point to $q$ for $d_w$. Formally, the goal of NNS over $d_w$ is to return

$$o^* = \arg \min_{o \in D} d_w(o, q).$$

(1)

Note that the weight vector $w$ is specified along with $q$ rather than pre-specified. Moreover, each element of $w$ can be either positive or non-positive.

The generalized weighted Manhattan distance, denoted by $d_{l_1} w$, and the generalized weighted square Euclidean distance, denoted by $d_{l_2} w$, are two typical generalized weighted distances which are derived from the Manhattan distance and the Euclidean distance, respectively. For any two points $o = (o_1, o_2, \ldots, o_d) \in \mathbb{R}^d$ and $q = (q_1, q_2, \ldots, q_d) \in \mathbb{R}^d$, the distances $d_{l_1} w(o, q)$ and $d_{l_2} w(o, q)$ are respectively computed as follows:

$$d_{l_1} w(o, q) = \sum_{i=1}^{d} w_i |o_i - q_i|$$

$$d_{l_2} w(o, q) = \sum_{i=1}^{d} w_i (o_i - q_i)^2,$$

(2)

where $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$. A recent paper [16] studied the problem of NNS over $d_{l_2} w$ and provided two sublinear time solutions for it. However, to the best of our knowledge, there is no prior work that solves the problem of NNS over $d_{l_1} w$ in sublinear time. Actually, plenty of studies [1][12] have shown that the Manhattan distance could be more effective than the Euclidean distance.
for producing meaningful NNS results in high-dimensional spaces. It indicates that NNS over $d^{l_1}_w$ is possibly more practical than NNS over $d^{l_2}_w$ in many real scenarios. In this paper, we target to propose sublinear time methods for efficiently solving the problem of NNS over $d^{l_1}_w$.

As a matter of fact, existing methods can not handle NNS over $d^{l_1}_w$ well. Specifically, the brute-force linear scan scales linearly with data size and thus may yield unsatisfactory performance. The conventional spatial index-based methods \cite{2,5,22} can only perform well for NNS in low dimensions due to the “curve of dimensionality” \cite{5}. Locality-Sensitive Hashing (LSH) \cite{27} is a popular approach for approximate NNS and exhibits good performance for high-dimensional cases. In the literature, a number of efficient LSH schemes \cite{7,9,10,13,15,17,26,28} have been proposed based on LSH for approximate NNS and exhibits good performance for high-dimensional cases. In the literature, there is no ALSH family proposed for NNS over $d^{l_1}_w$, and some of them can answer NNS queries even in sublinear time. Unfortunately, they can not be applied to answer the NNS queries over $d^{l_1}_w$ unless $w$ is fixed to an all-1 vector.

Recently, Asymmetric Locality-Sensitive Hashing (ALSH) was extended from LSH so that the problems for NNS over $d^{l_1}_w$ is no ALSH family proposed for NNS over $d^{l_1}_w$ in previous work. To provide sublinear time solutions for NNS over $d^{l_1}_w$ in this paper, we follow the ALSH approach to propose ALSH schemes by introducing ALSH families that are suitable for NNS over $d^{l_1}_w$.

Outline. In Section \ref{sec:2} we review the approaches of LSH and ALSH. In Section \ref{sec:3}, we show that there is no LSH or ALSH family for NNS over $d^{l_1}_w$ over the entire space $\mathbb{R}^d$ and there is no LSH family for NNS over $d^{l_1}_w$ over the bounded spaces in $\mathbb{R}^d$. Then we seek to find ALSH families for NNS over $d^{l_1}_w$ over the bounded spaces in $\mathbb{R}^d$. As a result, we propose two suitable ALSH families and further obtain two sublinear time ALSH schemes $(d^{l_1}_w, l_2)$-ALSH and $(d^{l_1}_w, \theta)$-ALSH in Section \ref{sec:4}.

2 Preliminaries

Before introducing our proposed solutions to the problem of NNS over $d^{l_1}_w$, we first present the preliminaries on LSH and ALSH.

2.1 Locality-Sensitive Hashing

Let $d(\cdot, \cdot)$ be a distance function and $Z$ be the space where $d(\cdot, \cdot)$ is defined. Assume that data points and query points are located in $\mathcal{X} \subseteq Z$ and $\mathcal{Y} \subseteq Z$, respectively. Then, an LSH family is formally defined as follows.

Definition 1 (LSH Family) An LSH family $\mathcal{H}(h) = \{h : Z \rightarrow \text{BucketIDs}\}$ is called $(R_1, R_2, P_1, P_2)$-sensitive if for any $o \in \mathcal{X}$ and $q \in \mathcal{Y}$, the following conditions are satisfied:

- If $d(o, q) \leq R_1$, then $Pr[h(o) = h(q)] \geq P_1$;
- If $d(o, q) \geq R_2$, then $Pr[h(o) = h(q)] \leq P_2$;
- $R_1 < R_2$ and $P_1 > P_2$.

As we can see from Definition\ref{def:LSH}, an LSH family is essentially a set of hash functions that can hash closer points into the same bucket with higher probability. Thus, the basic idea of an LSH scheme is to use an LSH family to hash points such that only the data points that have the same hash code as the query point are likely to be retrieved to find approximate nearest neighbors. In the following, we review two popular LSH families that were proposed for the $l_2$ distance (a.k.a. the Euclidean distance) and the Angular distance, respectively.

The $l_2$ distance between any two points $o, q \in \mathbb{R}^d$ is computed as $d^{l_2}(o, q) = \|o - q\|_2$, where $\|\cdot\|_2$ is the $l_2$-norm of a vector. The LSH family proposed for the $l_2$ distance in \cite{7} is $\mathcal{H}(h^{l_2}) = \{h^{l_2} : \mathbb{R}^d \rightarrow \mathbb{Z}\}$, where

$$h^{l_2}(x) = \left\lfloor \frac{a^Tx + b}{w} \right\rfloor,$$

(3)

$a$ is a $d$-dimensional vector where each element is chosen independently from the standard normal distribution, $b$ is a real number chosen uniformly at random from $[0, w]$, and $w$ is a user-specified
positive constant. Let \( r = d^{l_2}(o, q) \). The collision probability function is

\[
P^{l_2}(r) = Pr[h^{l_2}(o) = h^{l_2}(q)] = 1 - 2\Phi(-w/r) - \frac{2}{\sqrt{2\pi}(w/r)}(1 - e^{-(w^2/2r^2)}),
\]

where \( \Phi(\cdot) \) is the cumulative distribution function of the standard normal distribution [7].

The Angular distance between any two points \( o, q \in \mathbb{R}^d \) is computed as \( d^\theta(o, q) = \arccos(\frac{a^T o}{\|a\|_2 \|o\|_2}) \).

The LSH family proposed for the Angular distance in [4] is \( H^{o, q} \), solving the \( d^\theta \) problem.

In this section, we present some negative results on the existence of LSH and ALSH families for \( d^\theta \). Theorem 1 states that for any \( (R_1, R_2) \)-NNS problem, one can construct a data structure for solving the problem of \( (R_1, R_2) \)-NNS with \( O(n^{1+\rho}d\log n) \) query time and \( O(n^{1+\rho}) \) space, where \( \rho = \frac{\log R_1}{\log R_2} < 1 \).

### Negative Results

In this section, we present some negative results on the existence of LSH and ALSH families for NNS over \( d^{l_1} \). The following theorem indicates that it is impossible to find an LSH or ALSH family for NNS over \( d^{l_1} \) over \( X = \mathcal{Y} = \mathbb{R}^d \) (\( d \geq 3 \)).
Theorem 2  For any $d \geq 3$, $R_1 < R_2$ and $P_1 > P_2$, there is no $(R_1, R_2, P_1, P_2)$-sensitive LSH or ALSH family for NNS over $d_w^1$ over $X = Y = \mathbb{R}^d$.

Proof.  An LSH (or ALSH) family for NNS over $d_w^1$ over $\mathbb{R}^d$ ($d > 3$) is also an LSH (or ALSH) family for NNS over $d_w^1$ over a three-dimensional subspace, i.e., over $\mathbb{R}^3$. Hence, we only need to prove that there is no LSH or ALSH family for NNS over $d_w^1$ over $\mathbb{R}^3$. Assume by contradiction that for some $R_1 < R_2$ and $P_1 > P_2$ there exists an $(R_1, R_2, P_1, P_2)$-sensitive LSH family $\mathcal{H}(h)$ or ALSH family $\mathcal{H}(f,g)$ for NNS over $d_w^1$ over $\mathbb{R}^3$. Consider a set of $N$ data points $\{o^1, o^2, \ldots, o^N\} \subset \mathbb{R}^3$ and a set of $N$ query points $\{q^1, q^2, \ldots, q^N\} \subset \mathbb{R}^3$, where for $1 \leq i, j \leq N$,

\[
o^i = (iR_1 - iR_2, 0, 0) \quad q^j = (0, jR_1 - jR_2, R_1).
\]

The weight vector specified along with each query point is set as follows:

\[
w = \begin{cases} 
(1, -1, -1) & \text{if } R_1 < 0 \\
(1, -1, 1) & \text{if } R_1 \geq 0.
\end{cases}
\]

Thus, $d_w^1(o^i, q^j) = (i - j)(R_2 - R_1) + R_1$ for $1 \leq i, j \leq N$. As can be seen, $d_w^1(o^i, q^j) \leq R_1$ if $1 \leq i \leq j \leq N$ and $d_w^1(o^i, q^j) \geq R_2$ if $1 \leq j < i \leq N$. Let $A \in \mathbb{R}^{N \times N}$ be a sign matrix where each element is

\[
A(i, j) = \begin{cases} 
1 & \text{if } d_w^1(o^i, q^j) \leq R_1 \\
-1 & \text{if } d_w^1(o^i, q^j) \geq R_2 \\
0 & \text{otherwise.}
\end{cases}
\]

Obviously, $A$ is triangular with $+1$ on and above the diagonal and $-1$ below it. Consider also the matrix $B \in \mathbb{R}^{N \times N}$ of collision probabilities $B(i, j) = \Pr[h(o^i) = h(q^j)]$ (for $\mathcal{H}(h)$) or $B(i, j) = \Pr[f(o^i) = g(q^j)]$ (for $\mathcal{H}(f,g)$). Let $\theta = \frac{P_1 + P_2}{2} < 1$ and $\epsilon = \frac{P_1 - P_2}{2} > 0$. It is easy to know that $A(i, j)(B(i, j) - \theta) \geq \epsilon$ for $1 \leq i, j \leq N$. That is,

\[
A \odot (B - \theta) \geq \epsilon,
\]

where $\odot$ denotes the Hadamard (element-wise) product. From [25], the margin complexity of the sign matrix $A$ is $mc(A) = \inf_{A \odot C \geq \epsilon} \|C\|_{\max}$, where $\| \cdot \|_{\max}$ is the max-norm of a matrix. Since $A$ is also an $N \times N$ triangular matrix, the margin complexity of $A$ is bounded by $mc(A) = \Omega(\log N)$ according to [8]. Therefore, from Equation (10), we can obtain

\[
\|B - \theta\|_{\max} = \Omega(\log N).
\]

Since $B$ is a collision probability matrix, the max-norm of $B$ satisfies $\|B\|_{\max} \leq 1$ (20). Shifting $B$ by $0 < \theta < 1$ changes $\|B\|_{\max}$ by at most $\theta$. Thus, we have

\[
\|B - \theta\|_{\max} < 2.
\]

Combining Equations (11) and (12), we can easily derive that $\epsilon = O(\frac{1}{\log N})$. For any $\epsilon = \frac{P_1 - P_2}{2} > 0$, we get a contradiction by selecting a large enough $N$. The proof of Theorem 2 is similar to that of Theorem 3 in [21]. Due to space limitations, for the details of the max-norm and margin complexity involved in the proof of Theorem 2, please refer to [http://proceedings.mlr.press/v37/neyshabur15-supp.pdf](http://proceedings.mlr.press/v37/neyshabur15-supp.pdf).

Actually, in real scenarios data points and query points are usually located in bounded spaces. Consider the typical case of $X = Y = S$, where $S \subset \mathbb{R}^d$ is a bounded space. The following theorem shows nonexistence of an LSH family for NNS over $d_w^1$ over $X = Y = S$.

Theorem 3  For any $d > 0$, $R_1 < R_2$ and $P_1 > P_2$, there is no $(R_1, R_2, P_1, P_2)$-sensitive LSH family for NNS over $d_w^1$ over $X = Y = S$.

Proof. Assume by contradiction that for some $d > 0$, $R_1 < R_2$ and $P_1 > P_2$ there exists an $(R_1, R_2, P_1, P_2)$-sensitive LSH family $\mathcal{H}(h)$ for NNS over $d_w^1$ over $S$. Let $o, q \in S$ where $o \neq q$.

\[\text{[Ignore the trivial case that } S \text{ contains only a single point.]}\]
As $w \in \mathbb{R}^d$, we can always set $w$ to a value such that $d_w^H(o, q) = R_1$ and thus $Pr[h(o) = h(q)] \geq P_1$. Moreover, we can always set $w$ to another value such that $d_w^H(o, q) = R_2$ and thus $Pr[h(o) = h(q)] \leq P_2$. However, since data points should be hashed before queries arrive, $H(h)$ can not involve $w$. So $Pr[h(o) = h(q)]$ is not affected by $w$, which leads to a contradiction. 

Due to the negative results in Theorems 2 and 3 we seek to propose ALSH families for NNS over $d_w^H$ over bounded spaces in Section 4. Notice that if an ALSH family is suitable for NNS over $d_w^H$ over $\mathcal{X} = \mathcal{Y} = [M_t, M_u]^d$ ($M_t < M_u$), it must also be suitable for NNS over $d_w^H$ over $\mathcal{X} = \mathcal{Y} = \mathcal{S}$ for any $\mathcal{S} \subseteq [M_t, M_u]^d$. Thus, it is sufficient to deal with the case of $\mathcal{X} = \mathcal{Y} = [M_t, M_u]^d$. Further, suppose $\mathcal{X} = \mathcal{Y} = [0, M_u - M_t]^d$. Otherwise, it can be satisfied by shifting $o, q \in [M_t, M_u]^d$ without changing the results of NNS over $d_w^H$.

4 Our Solutions

Let $M = \{(M_u - M_t)t \mid t > 0\}$. The following Observation 1 indicates that if we find an ALSH family for NNS over $d_w^H$ over $\mathcal{X} = \mathcal{Y} = \{0, 1, \ldots, M\}^d$, a similar ALSH family can be immediately obtained for NNS over $d_w^H$ over $\mathcal{X} = \mathcal{Y} = [0, M_u - M_t]^d$. Thus, we only need to consider NNS over $d_w^H$ over $\mathcal{X} = \mathcal{Y} = \{0, 1, \ldots, M\}^d$ in the rest of the paper. Note that in our solutions $M$ can be an arbitrary positive integer.

**Observation 1** Define a vector function $u_k(x) = [xt] = [x_1t, x_2t, \ldots, x_dt]$, where $x = (x_1, x_2, \ldots, x_d) \in [0, M_u - M_t]^d$ and $t > 0$. For any $d > 0$, $R_1 < R_2$ and $P_1 > P_2$, if $H(f, g)$ is an ALSH family for NNS over $d_w^H$ over $\mathcal{X} = \mathcal{Y} = \{0, 1, \ldots, (M_u - M_t)t\}^d$, then $H(f_{out}, g_{out})$ must be an ALSH family for NNS over $d_w^H$ over $\mathcal{X} = \mathcal{Y} = [0, M_u - M_t]^d$, where $R_1' = (R_1 - \sum_{i=1}^d |w_i|) / t$ and $R_2' = (R_2 + \sum_{i=1}^d |w_i|) / t$.

**Proof.** Let $o, q \in [0, M_u - M_t]^d$. Then $[ot], [qt] \in \{0, 1, \ldots, (M_u - M_t)t\}^d$. A simple calculation shows that $d_w^H([ot], [qt]) \leq \sum_{i=1}^d |w_i| \leq d_w^H((ot, qt)) \leq d_w^H([ot], [qt]) + \sum_{i=1}^d |w_i|$ holds. Thus, we have that $d_w^H([ot], [qt]) \leq R_1'$ if $d_w^H((ot, qt)) \leq R_1'$ and $d_w^H([ot], [qt]) \geq R_2'$ if $d_w^H((ot, qt)) \geq R_2'$. Further, since $f([ot]) = f(u_k(o)) = (f \circ u_k)(o) = g([qt]) = g(u_k(q)) = (g \circ u_k)(q)$, we have that $Pr[(f \circ u_k)(o) = (g \circ u_k)(q)] \geq P_1$ if $d_w^H((ot, qt)) \leq R_1'$ and $Pr[(f \circ u_k)(o) = (g \circ u_k)(q)] \leq P_2$ if $d_w^H((ot, qt)) \geq R_2'$. As a result, $H(f_{out}, g_{out})$ is an ALSH family for NNS over $d_w^H$ over $\mathcal{X} = \mathcal{Y} = [0, M_u - M_t]^d$ (note: $R_1' < R_2'$ always holds since $R_1 < R_2$). 

4.1 From NNS over $d_w^H$ to MIPS

In the following, we take two steps to convert the problem of NNS over $d_w^H$ into the problem of MIPS. As a result of these steps, a novel preprocessing transformation and query transformation are introduced for data points and query points, respectively. The two transformations are essential to our solutions.

**Step 1: Convert NNS over $d_w^H$ into NNS over $d_w^H$**

The generalized weighted Hamming distance $d_w^H$ is defined on the Hamming space and computed in the same way as the generalized weighted Manhattan distance $d_w^H$. That is, $d_w^H(o, q) = \sum_{i=1}^d |o_i - q_i|$ for any $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$, $o = (o_1, o_2, \ldots, o_d) \in \{0, 1\}^d$ and $q = (q_1, q_2, \ldots, q_d) \in \{0, 1\}^d$.

Inspired by [10], we complete this step by applying unary coding. Specifically, each point $x = (x_1, x_2, \ldots, x_d) \in \{0, 1, \ldots, M\}^d$ is mapped into a binary vector $v(x) = \text{Unary}(x_1); \text{Unary}(x_2); \ldots; \text{Unary}(x_d) \in \{0, 1\}^{Md}$, where $()$ is the concatenation and each $\text{Unary}(x_i) = (x_{i1}, x_{i2}, \ldots, x_{iM})$ is the unary representation of $x_i$, i.e., a sequence of $x_i$’s followed by $(M - x_i)$’s. Then $|o_i - q_i| = \sum_{j=1}^M |o_j - q_j|$ for $o = (o_1, o_2, \ldots, o_d) \in \{0, 1, \ldots, M\}^d$, $q = (q_1, q_2, \ldots, q_d) \in \{0, 1, \ldots, M\}^d$ and $1 \leq i \leq d$. Moreover, the weight vector $w = (w_1, w_2, \ldots, w_d)$ is mapped into $I(w) = (I(w_1); I(w_2); \ldots; I(w_d))$, where each $I(w_i)$ is a se-
we only need to check two cases. Case 1: If $w_i |o_i - q_i| = w_i - (\cos(\frac{\pi}{2}o_{ij}), \sin(\frac{\pi}{2}o_{ij}))^T (w_i \cos(\frac{\pi}{2}q_{ij}), w_i \sin(\frac{\pi}{2}q_{ij}))$ always holds.

**Proof.** We only need to check two cases. Case 1: If $o_{ij} = q_{ij}$, then $w_i |o_i - q_i| = 0 = w_i - (\cos(\frac{\pi}{2}o_{ij}), \sin(\frac{\pi}{2}o_{ij}))^T (w_i \cos(\frac{\pi}{2}q_{ij}), w_i \sin(\frac{\pi}{2}q_{ij}))$. Case 2: If $o_{ij} \neq q_{ij}$, then $w_i |o_i - q_i| = w_i = w_i - (\cos(\frac{\pi}{2}o_{ij}), \sin(\frac{\pi}{2}o_{ij}))^T (w_i \cos(\frac{\pi}{2}q_{ij}), w_i \sin(\frac{\pi}{2}q_{ij})).$

For any $x = (x_1, x_2, \ldots, x_d) \in \{0, 1, \ldots, M\}^d$, we define $\cos(\frac{\pi}{2}v(x))$ and $\sin(\frac{\pi}{2}v(x))$ as follows:

$$\cos(\frac{\pi}{2}v(x)) = \left(\cos\left(\frac{\pi}{2}v_1\right), \cos\left(\frac{\pi}{2}v_2\right), \ldots, \cos\left(\frac{\pi}{2}v_d\right)\right)$$

$$\sin(\frac{\pi}{2}v(x)) = \left(\sin\left(\frac{\pi}{2}v_1\right), \sin\left(\frac{\pi}{2}v_2\right), \ldots, \sin\left(\frac{\pi}{2}v_d\right)\right),$$

where

$$\cos\left(\frac{\pi}{2}v_1\right) = \cos\left(\frac{\pi}{2}x_1\right), \cos\left(\frac{\pi}{2}x_2\right), \ldots, \cos\left(\frac{\pi}{2}x_d\right)$$

$$\sin\left(\frac{\pi}{2}v_1\right) = \sin\left(\frac{\pi}{2}x_1\right), \sin\left(\frac{\pi}{2}x_2\right), \ldots, \sin\left(\frac{\pi}{2}x_d\right).$$

According to Observation 2, we have

$$d_w^H(v(o), v(q)) \leq \sum_{i=1}^{d} \sum_{j=1}^{M} w_i |o_{ij} - q_{ij}|

= \sum_{i=1}^{d} \sum_{j=1}^{M} \left( w_i - \left(\cos(\frac{\pi}{2}o_{ij}), \sin(\frac{\pi}{2}o_{ij})\right)^T \left( w_i \cos(\frac{\pi}{2}q_{ij}), w_i \sin(\frac{\pi}{2}q_{ij})\right) \right)

= M \sum_{i=1}^{d} w_i - d^{IP}(T^1(v(o)), T^2(v(q))),$$

where $w = (w_1, w_2, \ldots, w_d) \in \mathbb{R}^d$, $o = (o_1, o_2, \ldots, o_d) \in \{0, 1, \ldots, M\}^d$, $q = (q_1, q_2, \ldots, q_d) \in \{0, 1, \ldots, M\}^d$, and $d^{IP}(\cdot, \cdot)$ is the inner product of two vectors, and $T^1(v(o))$ and $T^2(v(q))$ are respectively as follows:

$$T^1(v(o)) = (\cos(\frac{\pi}{2}v(o)); \sin(\frac{\pi}{2}v(o)))$$

$$T^2(v(q)) = (I(w) \circ \cos(\frac{\pi}{2}v(q)); I(w) \circ \sin(\frac{\pi}{2}v(q))).$$
To sum up, after Steps 1 and 2, we convert NNS over \( d_w^H \) over \( X = \{ v(x) \mid x \in \{0, 1, \ldots, M\}^d \} \subset \{0, 1\}^{Md} \) can be converted into MIPS over \( X = \{ T^1(v(x)) \mid x \in \{0, 1, \ldots, M\}^d \} \subset \{0, 1\}^{2Md} \) and \( Y = \{ T^2(v(x)) \mid x \in \{0, 1, \ldots, M\}^d \} \subset \mathbb{R}^{2Md} \).

Next, we formally present two ALSH schemes for NNS over \( d_w^1 \): the first one is called \((d_w^1, l_2)\)-ALSH and the second one is called \((d_w^1, \theta)\)-ALSH. \((d_w^1, l_2)\)-ALSH solves the problem of NNS over \( d_w^1 \) by reducing it to the problem of NNS over the \( l_2 \) distance, while \((d_w^1, \theta)\)-ALSH solves the problem of NNS over \( d_w^1 \) by reducing it to the problem of NNS over the Angular distance.

### 4.2 ALSH Schemes for NNS over \( d_w^1 \)

Based on the transformations \( P(\cdot) \) and \( Q_w(\cdot) \) and the LSH family \( \mathcal{H}(h, l_2) \) introduced in Section 2.1, \((d_w^1, l_2)\)-ALSH uses the ALSH family \( \mathcal{H}_{f(d_w^1, l_2), g(d_w^1, l_2)} = \{ f(d_w^1, l_2) : \{0, 1, \ldots, M\}^d \rightarrow \mathbb{Z} \} \cup \{ g(d_w^1, l_2) : \{0, 1, \ldots, M\}^d \rightarrow \mathbb{Z} \}, \) where \( f(d_w^1, l_2)(x) = h^2(P(x)) \) and \( g(d_w^3, l_2)(x) = h^2(Q_w(x)) \) for \( x \in \{0, 1, \ldots, M\}^d \). Combining Equations (13) and (18) we obtain

\[
d_w^1(o, q) = M \sum_{i=1}^{d} w_i - d_H(P(o), Q_w(q)). \tag{21}
\]

It is easy to know

\[
\|P(o)\|_2^2 = M d \tag{22}
\]

\[
\|Q_w(q)\|_2^2 = M \sum_{i=1}^{d} w_i^2. \tag{23}
\]

Thus, we have

\[
d^l_2(P(o), Q_w(q)) = \|P(o) - Q_w(q)\|_2
\]

\[
= \sqrt{\|P(o)\|_2^2 + \|Q_w(q)\|_2^2 - 2 d_H(P(o), Q_w(q))}
\]

\[
= \sqrt{M \left( d + \sum_{i=1}^{d} w_i^2 \right) - 2 \left( M \sum_{i=1}^{d} w_i - d_w^1(o, q) \right)}. \tag{24}
\]

Let \( r = d_w^1(o, q) \). According to Equations (4) and (24), the collision probability function with respect to \( \mathcal{H}_{f(d_w^1, l_2), g(d_w^1, l_2)} \) is

\[
P(d_w^1, l_2)(r) = P_r[f(d_w^1, l_2)(o) = g(d_w^1, l_2)(q)]
\]

\[
= P_r[h^2(P(o)) = h^2(Q_w(q))]
\]

\[
= P^l_2(\|P(o) - Q_w(q)\|_2)
\]

\[
= P^l_2 \left( \sqrt{M \left( d + \sum_{i=1}^{d} w_i^2 \right) - 2 \left( M \sum_{i=1}^{d} w_i - r \right)} \right). \tag{25}
\]

Since \( P^l_2(\cdot) \) is a decreasing function, \( P(d_w^1, l_2)(R_1) > P(d_w^1, l_2)(R_2) \) holds for any \( R_1 < R_2 \). Therefore, we obtain the following Lemma [1]

**Lemma 1** \( \mathcal{H}_{f(d_w^1, l_2), g(d_w^1, l_2)} \) is \((R_1, R_2, P(d_w^1, l_2)(R_1), P(d_w^1, l_2)(R_2))\)-sensitive for any \( R_1 < R_2 \).
According to Theorem 1 and Lemma 1, we have the following Theorem 4.

**Theorem 4** \((d_{w^1}^l, l_2)-ALSH\) can solve the problem of \((R_1, R_2)\)-NNS over \(d_{w^1}^l\) with \(O(n^{d_{w^1}^l \cdot l_2})\) \(d\log n\) query time and \(O(n^{1 + \rho(d_{w^1}^l \cdot l_2)})\) space, where \(\rho(d_{w^1}^l, l_2) = \frac{\log \left( \frac{p(d_{w^1}^l, l_2)}{p(d_{w^1}^l, l_2)(R_1)} \right)}{\log \left( \frac{p(d_{w^1}^l, l_2)}{p(d_{w^1}^l, l_2)(R_2)} \right)} < 1.\)

### 4.2.2 \((d_{w^1}^l, \theta)-ALSH\)

Now we introduce the scheme of \((d_{w^1}^l, \theta)-ALSH\). Based on the transformations \(P(\cdot)\) and \(Q_w(\cdot)\) and the LSH family \(H_{(h, \theta)}\) introduced in Section 2.1, \((d_{w^1}^l, \theta)-ALSH\) uses the ALSH family \(H_{(f(d_{w^1}^l, \theta), g(d_{w^1}^l, \theta))} = \{f(d_{w^1}^l, \theta) : \{0, 1, \ldots, M\}^d \to \{0, 1\}\} \cup \{g(d_{w^1}^l, \theta) : \{0, 1, \ldots, M\}^d \to \{0, 1\}\},\)

where \(f(d_{w^1}^l, \theta)(x) = h^\theta(P(x))\) and \(g(d_{w^1}^l, \theta)(x) = h^\theta(Q_w(x))\) for \(x \in \{0, 1, \ldots, M\}^d\). According to Equations 21, 22 and 23, the relationship between \(d_{w^1}^l(o, q)\) and \(d^\theta(P(o), Q_w(q))\) is as follows:

\[
d^\theta(P(o), Q_w(q)) = \arccos \left( \frac{P(o)^T Q_w(q)}{\|P(o)\|_2 \|Q_w(q)\|_2} \right) = \arccos \left( \frac{d_1^P(P(o), Q_w(q))}{\|P(o)\|_2 \|Q_w(q)\|_2} \right) = \arccos \left( \frac{M \sum_{i=1}^{d} w_i^1 - d_{w^1}^l(o, q)}{M \sqrt{d \sum_{i=1}^{d} w_i^2}} \right). \tag{26}\]

Let \(r = d_{w^1}^l(o, q)\). From Equations 6 and 26, it can be seen that the collision probability function with respect to \(H_{(f(d_{w^1}^l, \theta), g(d_{w^1}^l, \theta))}\) is

\[
P(d_{w^1}^l, \theta)(r) = Pr[f(d_{w^1}^l, \theta)(o) = g(d_{w^1}^l, \theta)(q)] = Pr[h^\theta(P(o)) = h^\theta(Q_w(q))] = 1 - \frac{1}{\pi} \arccos \left( \frac{P(o)^T Q_w(q)}{\|P(o)\|_2 \|Q_w(q)\|_2} \right) = 1 - \frac{1}{\pi} \arccos \left( \frac{M \sum_{i=1}^{d} w_i^1 - r}{M \sqrt{d \sum_{i=1}^{d} w_i^2}} \right). \tag{27}\]

It is easy to know that \(P(d_{w^1}^l, \theta)(\cdot)\) is a decreasing function. Thus, \(P(d_{w^1}^l, \theta)(R_1) > P(d_{w^1}^l, \theta)(R_2)\) holds for any \(R_1 < R_2\). Then we obtain the following Lemma 2.

**Lemma 2** \(H_{(f(d_{w^1}^l, \theta), g(d_{w^1}^l, \theta))}\) is \((R_1, R_2, P(d_{w^1}^l, \theta)(R_1), P(d_{w^1}^l, \theta)(R_2))\)-sensitive for any \(R_1 < R_2\).

Combining Theorem 1 and Lemma 2, we have the following Theorem 5.

**Theorem 5** \((d_{w^1}^l, \theta)-ALSH\) can solve the problem of \((R_1, R_2)\)-NNS over \(d_{w^1}^l\) with \(O(n^{d_{w^1}^l \cdot l_2})\) \(d\log n\) query time and \(O(n^{1 + \rho(d_{w^1}^l \cdot \theta)})\) space, where \(\rho(d_{w^1}^l, \theta) = \frac{\log \left( \frac{p(d_{w^1}^l, \theta)}{p(d_{w^1}^l, \theta)(R_1)} \right)}{\log \left( \frac{p(d_{w^1}^l, \theta)}{p(d_{w^1}^l, \theta)(R_2)} \right)} < 1.\)

### 4.2.3 Implementation Skills of \((d_{w^1}^l, l_2)-ALSH\) and \((d_{w^1}^l, \theta)-ALSH\)

The scheme of \((d_{w^1}^l, l_2)-ALSH\) (or \((d_{w^1}^l, \theta)-ALSH\)) needs to compute the hash values \(h^{l_2}(P(o))\) and \(h^{l_2}(Q_w(q))\) (or \(h^\theta(P(o))\) and \(h^\theta(Q_w(q))\)). We can easily know that the running time of computing \(h^{l_2}(P(o))\) (or \(h^\theta(P(o))\)) is dominated by the time cost of obtaining \(a^T P(o)\) and the running time of computing \(h^{l_2}(Q_w(q))\) (or \(h^\theta(Q_w(q))\)) is dominated by the time cost of obtaining \(a^T Q_w(q)\), where \(a\) is a \(2Md\)-dimensional vector where each entry is chosen independently from the standard normal distribution.
distribution. The naive approach to obtain $a^T P(o)$ or $a^T Q_w(q)$ is to compute the inner product of the two corresponding vectors. However, it will require $2Md$ multiplications and $2Md - 1$ additions, which is expensive when $M$ is large.

Next, we show how to obtain $a^T P(o)$ with only $2d - 1$ additions and obtain $a^T Q_w(q)$ with only $2d - 1$ additions and $d$ multiplications. Suppose $a = (a_1; a_2; \ldots; a_d; a_{d+1}; a_{d+2}; \ldots; a_{2d})$, where $a_i = (a_{1i}; a_{2i}; \ldots; a_{Mi}) \in \mathbb{R}^M$. According to Equations 14-17, 19 and 20, we have $a^T P(o) = \sum_{i=1}^{d} a_i^T \cos(\frac{\pi}{2} \text{Unary}(o_i)) + \sum_{i=1}^{d} a_i^T \sin(\frac{\pi}{2} \text{Unary}(o_i))$ and $a^T Q_w(q) = \sum_{i=1}^{d} a_i^T \cos(\frac{\pi}{2} \text{Unary}(q_i)) w_i + \sum_{i=1}^{d} a_i^T \sin(\frac{\pi}{2} \text{Unary}(q_i)) w_i$. Since $\cos(\frac{\pi}{2} \text{Unary}(o_i))$ is a sequence of $a_i$'s followed by $(M - a_i)$ 1's and $\sin(\frac{\pi}{2} \text{Unary}(o_i))$ is a sequence of $a_i$'s followed by $(M - a_i)$ 0's, it is easy to know that $a_i^T \cos(\frac{\pi}{2} \text{Unary}(o_i))$ is the sum of the last $M - a_i$ elements of $a_i$ and $a_i^T \sin(\frac{\pi}{2} \text{Unary}(o_i))$ is the sum of the first $a_i$ elements of $a_{d+i}$. Thus, we preprocess the vector $a$ to obtain $a' = (a_1'; a_2'; \ldots; a_d'; a_{d+1}'; a_{d+2}'; \ldots; a_{2d}')$, where $a_i' = (a_{i1}'; a_{i2}'; \ldots; a_{iM'; a_{i(M+1)}}$) and

$$a_{i,j} = \begin{cases} \sum_{k=j}^{M} a_{ik} & \text{if } 1 \leq i \leq d \text{ and } 1 \leq j \leq M \\ 0 & \text{if } 1 \leq i \leq d \text{ and } j = M + 1 \\ 0 & \text{if } d + 1 \leq i \leq 2d \text{ and } j = 1 \\ \sum_{k=1}^{j-1} a_{ik} & \text{if } d + 1 \leq i \leq 2d \text{ and } 2 \leq j \leq M + 1. \end{cases} (28)$$

Then we have $a^T P(o) = \sum_{i=1}^{d} a_{i'(o_i+1)} + \sum_{i=1}^{d} a_{i'(d+i)(o_i+1)}$. It can be seen that $a^T P(o)$ can be obtained with $2d - 1$ additions by using $a'$. Similarly, we have $a^T Q_w(q) = \sum_{i=1}^{d} w_i a_{i'(q_i+1)} + \sum_{i=1}^{d} w_i a_{i'(q_i+1)} + a_{i'(d+i)(q_i+1)}$. Therefore, $a^T Q_w(q)$ can be obtained with $2d - 1$ additions and $d$ multiplications by using $a'$.

5 Conclusion

This paper studies the fundamental problem of Nearest Neighbor Search (NNS) over the generalized weighted Manhattan distance ($d^1_w$). As far as we know, there is no prior work that solves the problem in sublinear time. In this paper, we first prove that there is no LSH or ALSH family for $d^1_w$ over the entire space $\mathbb{R}^d$. Then, we prove that there is still no LSH family suitable for $d^1_w$ over a bounded space. After that, we propose two ALSH families for $d^1_w$ over a bounded space. Based on these ALSH families, two ALSH schemes ($d^1_w; l_2$)-ALSH and ($d^1_w; \theta$)-ALSH are proposed for solving NNS over $d^1_w$ in sublinear time.

References

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In ICDT, 2001.
[2] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. Commun. ACM, 1975.
[3] Gautam Bhattacharya, Koushik Ghosh, and Ananda S. Chowdhury. Granger causality driven AHP for feature weighted knn. Pattern Recognit., 2017.
[4] Moses Charikar. Similarity estimation techniques from rounding algorithms. In STOC, 2002.
[5] Lei Chen. Curse of dimensionality. In Encyclopedia of Database Systems. 2009.
[6] King Lum Cheung and Ada Wai-Chee Fu. Enhanced nearest neighbour search on the r-tree. SIGMOD Rec., 1998.
[7] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In SCG, 2004.
[8] Jürgen Forster, Niels Schmitt, Hans Ulrich Simon, and Thorsten Suttrop. Estimating the optimal margins of embeddings in euclidean half spaces. Mach. Learn., 2003.
[9] Junhao Gan, Jianlin Feng, Qiong Fang, and Wilfred Ng. Locality-sensitive hashing scheme based on dynamic collision counting. In SIGMOD, 2012.
[10] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *VLDB*, 1999.

[11] Yupeng Gu, Bo Zhao, David Hardtke, and Yizhou Sun. Learning global term weights for content-based recommender systems. In *WWW*, 2016.

[12] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *VLDB*, 2000.

[13] Qiang Huang, Jianlin Feng, Qiong Fang, Wilfred Ng, and Wei Wang. Query-aware locality-sensitive hashing scheme for lp norm. *VLDB J.*, 2017.

[14] Chein-Shung Hwang, Yi-Ching Su, and Kuo-Cheng Tseng. Using genetic algorithms for personalized recommendation. In *ICCCI*, Lecture Notes in Computer Science. Springer, 2010.

[15] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, 1998.

[16] Yifan Lei, Qiang Huang, Mohan S. Kankanhalli, and Anthony K. H. Tung. Sublinear time nearest neighbor search over generalized weighted space. In *ICML*, 2019.

[17] Kejing Lu, Hongya Wang, Wei Wang, and Mineichi Kudo. VHP: approximate nearest neighbor search via virtual hypersphere partitioning. *Proc. VLDB Endow.*, 2020.

[18] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, 2015.

[19] Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. Learning to weight for text classification. *IEEE Trans. Knowl. Data Eng.*, 2020.

[20] Behnam Neyshabur, Yury Makarychev, and Nathan Srebro. Clustering, hamming embedding, generalized LSH and the max norm. In *ALT*, 2014.

[21] Behnam Neyshabur and Nathan Srebro. On symmetric and asymmetric lshs for inner product search. In *ICML*, 2015.

[22] Hanan Samet. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, 2006.

[23] Anshumali Shrivastava and Ping Li. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *NeurIPS*, 2014.

[24] Anshumali Shrivastava and Ping Li. Improved asymmetric locality sensitive hashing (ALSH) for maximum inner product search (MIPS). In *UAI*, 2015.

[25] Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *COLT*, Lecture Notes in Computer Science. Springer, 2005.

[26] Yufei Tao, Ke Yi, Cheng Sheng, and Panos Kalnis. Efficient and accurate nearest neighbor and closest pair search in high-dimensional space. *TODS*, 2010.

[27] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *CoRR*, 2014.

[28] Bolong Zheng, Xi Zhao, Lianggui Weng, Nguyen Quoc Viet Hung, Hang Liu, and Christian S. Jensen. PM-LSH: A fast and accurate LSH framework for high-dimensional approximate NN search. *Proc. VLDB Endow.*, 2020.