Sublinear Time Nearest Neighbor Search over Generalized Weighted Space

Yifan Lei [1] **Qiang Huang** [1] **Mohan Kankanhalli** [1] **Anthony K. H. Tung** [1]

Abstract

Nearest Neighbor Search (NNS) over generalized weighted space is a fundamental problem which has many applications in various fields. However, to the best of our knowledge, there is no sublinear time solution to this problem. Based on the idea of Asymmetric Locality-Sensitive Hashing (ALSH), we introduce a novel spherical asymmetric transformation and propose the first two novel weight-oblivious hashing schemes SL-ALSH and S2-ALSH accordingly. We further show that both schemes enjoy a quality guarantee and can answer the NNS queries in sublinear time. Evaluations over three real datasets demonstrate the superior performance of the two proposed schemes.

1. Introduction

In this paper, we study a fundamental problem of Nearest Neighbor Search (NNS) over the Generalized Weighted Square Euclidean Distance (GWSED) d_w. Given a dataset D of n data objects and a query object q with a weight vector w in the Euclidean space $\mathbb{R}^d$, the problem of NNS over d_w is to find the nearest object $o^* \in D$ of q such that

$$o^* = \arg\min_{o \in D} d_w(o, q). \tag{1}$$

For any two objects $o = (o_1, o_2, \ldots, o_d)$ and $q = (q_1, q_2, \ldots, q_d) \in \mathbb{R}^d$, the GWSED $d_w(o, q)$ between o and q is computed as follows:

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

where $w = (w_1, w_2, \ldots, w_d)$ is a weight vector. Here w is specified only when q arrives and we consider the generalized setting of $w \in \mathbb{R}^d$ without any constraints such as $w_i \geq 0, \forall i$. Although this generalization makes the distance d_w no longer a metric if $w_i < 0$, it is fundamental and has a variety of diverse applications.

[1]School of Computing, National University of Singapore, Singapore. Correspondence to: Qiang Huang <huangq@comp.nus.edu.sg>, Anthony K. H. Tung <atung@comp.nus.edu.sg>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

For example, under the specific setting that $w_i = 1, \forall i$, the problem of NNS over L_2 distance (or Euclidean distance) can be reduced to NNS over d_w. Similarly, by setting $w_i = -1, \forall i$, the Furthest Neighbor Search (FNS) over L_2 distance can be converted into NNS over d_w. In addition, the problem of Maximum Inner Product Search (MIPS) which has attracted increasing interests in recent times, as we will discuss in Sect. 3, can also be reduced to NNS over d_w. Thus, NNS over d_w is a very fundamental problem.

There are many scenarios in which the problem of NNS over d_w arises naturally, especially when the negative weight $w_i < 0$ is allowed. For example, for the Recommendation systems (Wang et al., 2015; Gu et al., 2016), the weight vector w makes it possible for users to adjust the importance of different dimensions of retrieved items, and the negative weights indicate that users do not expect the search results which are similar to query item in those particular dimensions. Another popular scenario is the kNN classifier (Fernandez et al., 2018; Bhattacharya et al., 2017; Liu et al., 2015). It is common to set different weights for different dimensions. An efficient method for NNS over d_w can largely reduce the tuning time of kNN classifier for the weights of different dimensions.

The naive solution for NNS over d_w is linear scan, which sequentially compares data objects $o \in D$ to the query q with its corresponding weight vector w. However, the query time complexity is $O(nd)$, which is not efficient if n or d is large. Due to the "curse of dimensionality," the space and/or query time complexities of traditional space partition based methods (Guttman, 1984; Bentley, 1990; Katayama & Satoh, 1997) for exact NNS are exponential in d (Weber et al., 1998; Beyer et al., 1999). Locality-Sensitive Hashing (LSH) and its variants (Indyk & Motwani, 1998; Datar et al., 2004; Andoni & Indyk, 2006; Gan et al., 2012; Sun et al., 2014; Huang et al., 2015; Zheng et al., 2016; Huang et al., 2017; Andoni et al., 2018) are the sublinear time methods for approximate NNS in high-dimensional space, but they are only suitable for NNS over d_w for the specific type of w with $w_i = 1, \forall i$. For the generalized setting of w, it is not sufficient for NNS over d_w. We are interested in the methods with a *weight-oblivious* data structure that knows no information of w during the preprocessing phase but can answer the NNS queries over d_w for arbitrary w of various types, which can be seen as "one index for all generalized

weighted space." However, to the best of our knowledge, there is no sublinear time method with weight-oblivious data structure for NNS over d_w.

On the other hand, even though inner product is not a metric which is similar to d_w, starting from the pioneering work of Shrivastava & Li (2014), many sublinear time methods based on Asymmetric LSH (ALSH) have been proposed for MIPS (Shrivastava & Li, 2015b; Neyshabur & Srebro, 2015; Shrivastava & Li, 2015a; Huang et al., 2018; Yan et al., 2018). Compared with LSH-based schemes, ALSH allows asymmetric transformations for data objects and queries so that they can compute the collision probability of inner product similar to LSH, which motivates us to consider the ALSH-based methods for NNS over d_w.

Based on the idea of ALSH, we propose the first two sublinear time ALSH schemes SL-ALSH and S2-ALSH for NNS over d_w. Firstly, we show that there is no ALSH for NNS over d_w in $\mathbb{R}^d$ (Sect. 3). Secondly, we propose a novel spherical asymmetric transformation which converts the data objects and queries from $\mathbb{R}^d$ to $\mathbb{R}^{2d}$. Then, based on this transformation, we introduce two novel weight-oblivious ALSH schemes SL-ALSH and S2-ALSH which convert the problem of NNS over d_w into the problem of NNS over L_2 distance and NNS over Angular distance, respectively. Furthermore, we show that the two proposed schemes enjoy sublinear query time for NNS over d_w (Sect. 4). Experimental evaluations over three real-life datasets show that SL-ALSH and S2-ALSH lead to significant computational saving over linear scan and support various types of weight vectors w (Sect. 5).

2. Preliminaries

Before we introduce the schemes for NNS over d_w, we first review some preliminary knowledge about LSH and ALSH.

2.1. Locality-Sensitive Hashing

LSH schemes are the most popular methods for the problem of Approximate NNS (ANNS). Formally, an LSH function family (or simply LSH family) is defined as follows:

Definition 1 (LSH family)**.** *A family of hash functions $\mathcal{H}$ is said to be (R_1, R_2, p_1, p_2)-sensitive for a distance function $Dist(\cdot,\cdot)$ if, for any $o, q \in \mathbb{R}^d$ and $h \in \mathcal{H}$, $\mathcal{H}$ satisfies the following conditions:*

- *If $Dist(o,q) \leq R_1$, then $\Pr_{\mathcal{H}}[h(o) = h(q)] \geq p_1$;*
- *If $Dist(o,q) \geq R_2$, then $\Pr_{\mathcal{H}}[h(o) = h(q)] \leq p_2$;*
- *$R_1 < R_2$ and $p_1 > p_2$.*

Under the generalized setting of w, the distance d_w can be negative if $w_i < 0$. Thus, we analyse the theoretical guarantee using the (R_1, R_2, p_1, p_2)-sensitive setting and consider the problem of (R_1, R_2)-Near Neighbor Search ((R_1, R_2)-NNS), which is defined as follows:

Definition 2 ((R_1, R_2)-NNS)**.** *Given a distance function $Dist(\cdot,\cdot)$ and two distance thresholds R_1 and R_2 ($R_1 < R_2$), the problem of (R_1, R_2)-NNS is to construct a data structure which, for any query $q \in \mathbb{R}^d$, returns an object $o \in D$ such that $Dist(o,q) \leq R_2$ if there exists any $o' \in D$ such that $Dist(o',q) \leq R_1$.*

With an (R_1, R_2, p_1, p_2)-sensitive hash family, we have Theorem 1 (Datar et al., 2004) as follows:

Theorem 1. *Given an (R_1, R_2, p_1, p_2)-sensitive family $\mathcal{H}$, one can build a data structure for the problem of (R_1, R_2)-NNS which uses $O(n^{1+\rho})$ space and $O(dn^{\rho}\log_{1/p_2}(n))$ query time, where $\rho = \ln p_1 / \ln p_2$.*

Notice that (R_1, R_2)-NNS is simply a decision version of the ANNS problem. One can reduce the ANNS problem to (R_1, R_2)-NNS via a binary-search-like method. Then, the query time complexity of ANNS is the same (within a log factor for binary search) as that of the (R_1, R_2)-NNS problem. Next, we review two popular LSH schemes E2LSH and SimHash for NNS over L_2 distance and Angular distance, respectively.

E2LSH was proposed by Datar et al. (2004). Let $\|o\|_2$ be the L_2 norm of an object o. Given any two objects $o, q \in \mathbb{R}^d$, the L_2 distance can be computed as $\|o-q\|_2 = \sqrt{\sum_{i=1}^{d}(o_i - q_i)^2}$. The LSH function is defined as follows:

$$h_{l_2}(o) = \left\lfloor \frac{a^T o + b}{r} \right\rfloor, \tag{3}$$

where a is a d-dimensional vector with each entry chosen independently and uniformly at random from standard Gaussian distribution $\mathcal{N}(0,1)$; r is a pre-specified bucket width; b is a random offset chosen uniformly at random from $[0, r)$.

Given any two objects $o, q \in \mathbb{R}^d$, let $\delta = \|o-q\|_2$. The collision probability $p^{(l_2)}(\delta)$ is computed as follows:

$$\begin{aligned} p^{(l_2)}(\delta) &= Pr[h_{l_2}(o) = h_{l_2}(q)] \\ &= 1 - 2\Phi(-r/\delta) - \frac{2}{\sqrt{2\pi}(r/\delta)}(1 - e^{-(r/\delta)^2/2}), \end{aligned} \tag{4}$$

where $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx$ (Datar et al., 2004).

SimHash was proposed by Charikar (2002). Given two objects $o, q \in \mathbb{R}^d$, the Angular distance is computed as $\theta(o,q) = cos^{-1}(\frac{o^T q}{\|o\|_2 \|q\|_2})$. Its LSH function is called Sign Random Projection (SRP), which is defined as follows:

$$h_{srp}(o) = sign(a^T o), \tag{5}$$

where a is a d-dimensional vector with each entry generated independently and uniformly at random from $\mathcal{N}(0,1)$.

Given any two objects $o, q \in \mathbb{R}^d$, let $\delta = \theta(o,q)$. The collision probability $p^{(srp)}(\delta)$ is computed as follows:

$$p^{(srp)}(\delta) = Pr[h_{srp}(o) = h_{srp}(q)] = 1 - \frac{\delta}{\pi}. \tag{6}$$

2.2. Asymmetric LSH

ALSH schemes can solve the MIPS with sublinear query time. Formally, the ALSH family is defined as follows:

Definition 3 (ALSH family)**.** *A family of hash functions $\mathcal{H}$, along with two vector transformations $P : \mathbb{R}^d \to \mathbb{R}^{d'}$ (Preprocessing transformation) and $Q : \mathbb{R}^d \to \mathbb{R}^{d'}$ (Query transformation), is said to be (R_1, R_2, p_1, p_2)-sensitive for a distance function $Dist(\cdot, \cdot)$ if, for any $o, q \in \mathbb{R}^d$ and $h \in \mathcal{H}$, $\mathcal{H}$ satisfies the following conditions:*

- *If $Dist(o, q) \leq R_1$, then $\Pr_{\mathcal{H}}[h(P(o)) = h(Q(q))] \geq p_1$;*
- *If $Dist(o, q) \geq R_2$, then $\Pr_{\mathcal{H}}[h(P(o)) = h(Q(q))] \leq p_2$;*
- *$R_1 < R_2$ and $p_1 > p_2$.*

Simple-LSH (Neyshabur & Srebro, 2015) is one of the state-of-the-art ALSH schemes. It has the same preprocessing transformation P and query transformation Q as follows:

$$P(x) = Q(x) = [x; \sqrt{1 - \|x\|_2^2}].$$

Notice that they assume $\|o\|_2 \leq 1$ for all $o \in D$ and limit $\|q\|_2 = 1$ for the query q. Thus, different normalizations are used for data objects and queries. From this perspective, it also shares the same nature of ALSH schemes.

3. No ALSH over $\mathbb{R}^d$

Neyshabur & Srebro (2015) demonstrated that there is no ALSH for MIPS over $\mathbb{R}^d$. By leveraging their results, we show that there is no ALSH for NNS over d_w over $\mathbb{R}^d$.

At first, we show that the problem of MIPS can be reduced to the problem of NNS over d_w.

Lemma 1. *If we have an (R_1, R_2, p_1, p_2)-sensitive ALSH family for NNS over d_w for any $R_1 < R_2$ and $p_1 > p_2$ over $\mathbb{R}^d, d \geq 5$, we can then construct an (S, cS, p_1, p_2)-sensitive ALSH family for MIPS over $\mathbb{R}^{\lfloor (d-1)/2 \rfloor}$ for any $S > 0$ and $0 < c < 1$.*

Proof. Suppose $\mathcal{H}$ along with two vector transformations P and Q is an (R_1, R_2, p_1, p_2)-sensitive ALSH family for NNS over d_w over $\mathbb{R}^d$.

We first consider the case of odd $d \geq 5$. Consider the problem of MIPS. For any $o, q \in \mathbb{R}^{\lfloor (d-1)/2 \rfloor}$, we construct two vector transformations $f : \mathbb{R}^{\lfloor (d-1)/2 \rfloor} \to \mathbb{R}^d$ and $g : \mathbb{R}^{\lfloor (d-1)/2 \rfloor} \to \mathbb{R}^d$:

$$f(o) = [o; o; 0],$$
$$g(q) = [q; -q; 1],$$

and the weight vector $w = [\frac{R_2 - R_1}{4(1-c)S}; -\frac{R_2 - R_1}{4(1-c)S}; \frac{R_2 - cR_1}{1-c}]$. Then, we have

$$d_w(f(o), g(q)) = \frac{(R_1 - R_2)}{(1-c)S} o^T q + \frac{R_2 - cR_1}{1-c},$$

and

$$\begin{aligned} d_w(f(o), g(q)) \leq R_1 &\iff o^T q \geq S, \\ d_w(f(o), g(q)) \geq R_2 &\iff o^T q \leq cS. \end{aligned} \tag{7}$$

Combining Eq. 7 with Definition 3, we infer that

$$o^T q \geq S \implies \Pr_{\mathcal{H}}[h(P \circ f(o)) = h(Q \circ g(q))] \geq p_1,$$
$$o^T q \leq cS \implies \Pr_{\mathcal{H}}[h(P \circ f(o)) = h(Q \circ g(q))] \leq p_2.$$

Thus, $\mathcal{H}$ along with $P \circ f$ and $Q \circ g$ is an (S, cS, p_1, p_2)-sensitive ALSH family for MIPS over $\mathbb{R}^{\lfloor (d-1)/2 \rfloor}$.

For the case of even $d \geq 6$, we could add one more dimension to w, f, g with 0, and construct an (S, cS, p_1, p_2)-sensitive ALSH family $\mathcal{H}$ for MIPS over $\mathbb{R}^{\lfloor (d-1)/2 \rfloor}$ as well.

Therefore, Lemma 1 is proved. □

Based on Lemma 1, we have Theorem 2 as follows.

Theorem 2. *For any $d \geq 5$, $R_1 < R_2$, and $p_1 > p_2$, there is no (R_1, R_2, p_1, p_2)-sensitive ALSH family for NNS over d_w over $\mathbb{R}^d$.*

Proof. Suppose there exists an (R_1, R_2, p_1, p_2)-sensitive ALSH family for NNS over d_w over $\mathbb{R}^d$ for any $d \geq 5$. According to Lemma 1, we are able to construct an (S, cS, p_1, p_2)-sensitive ALSH family for MIPS over $\mathbb{R}^{\lfloor (d-1)/2 \rfloor}$ for any $S > 0$ and $0 < c < 1$, where $\lfloor (d-1)/2 \rfloor \geq 2$. This contradicts Theorem 3.1 in (Neyshabur & Srebro, 2015) which states that there is no (S, cS, p_1, p_2)-ALSH family for MIPS over $\mathbb{R}^d$ for any $d \geq 2$. Thus, Theorem 2 is proved. □

Theorem 2 formally demonstrates that there is no ALSH family for NNS over d_w over $\mathbb{R}^d$. Fortunately, this is not required for the problem of NNS over d_w. Next, we will introduce a spherical asymmetric transformation which converts the data objects and queries from $\mathbb{R}^d$ to $\mathbb{R}^{2d}$ and propose two ALSH schemes for NNS over d_w accordingly.

4. Our Proposed Methods

4.1. Spherical Asymmetric Transformation

We now introduce the spherical asymmetric transformation. Suppose $o, q \in [0, U]^d \subset \mathbb{R}^d$ for all $o \in D$ and q for a fixed $0 < U \leq \pi$. Otherwise, we can shift and/or rescale $o \in D$ and q without changing the order of the NNS results. In addition, unless $w = [0, 0, \ldots, 0]$ where the problem of NNS over d_w is trivial, we assume $\|w\|_1 = 1$, where $\|w\|_1 = \sum_i |w_i|$ is the L_1 norm of w. Otherwise, we can also rescale w to get the same order of the NNS results.

For any object $o = (o_1, o_2, \ldots, o_d)$, let COS and SIN be the element-wise cos and sin, respectively, i.e.,

$$COS(o) = [\cos(o_1), \cos(o_2), \ldots, \cos(o_d)],$$
$$SIN(o) = [\sin(o_1), \sin(o_2), \ldots, \sin(o_d)].$$

Given a data object $o = (o_1, o_2, \ldots, o_d)$ and a query $q = (q_1, q_2, \ldots, q_d)$ with a weight vector $w = (w_1, w_2, \ldots, w_d)$, the vector transformations $P : \mathbb{R}^d \to \mathbb{R}^{2d}$ and $Q : \mathbb{R}^d \to \mathbb{R}^{2d}$ are defined as follows:

$$P(o) = [COS(o); SIN(o)], \tag{8}$$

$$Q(q, w) = [w \otimes COS(q); w \otimes SIN(q)], \tag{9}$$

where $\otimes$ denotes element-wise product, i.e., $w \otimes COS(q) = (w_1 \cos(q_1), w_2 \cos(q_2), \ldots, w_d \cos(q_d))$.

Notice that $\|P(o)\|_2 = \sqrt{d}$ for all $o \in D$ and $\|Q(q, w)\|_2 = \|w\|_2$ for each q. This asymmetric transformation maps o and q over $\mathbb{R}^d$ to the hypersphere over $\mathbb{R}^{2d}$ with radii $\sqrt{d}$ and $\|w\|_2$, respectively. Thus, we call it *spherical asymmetric transformation*. In addition, since the preprocessing transformation P is independent of w, this asymmetric transformation is *weight-oblivious*.

The intuition of the spherical asymmetric transformation comes from the fact that $1 - \cos(\delta) \approx \frac{\delta^2}{2}$ when $\delta \mapsto 0$. Based on the Taylor expansion, we have Fact 1 as follows:

Fact 1. *For any $\delta \in \mathbb{R}$, $\frac{\delta^2}{2} - \frac{\delta^4}{24} \leq 1 - cos(\delta) \leq \frac{\delta^2}{2}$.*

Let w^- be

$$w^- = \sum_{i:w_i \leq 0} w_i. \tag{10}$$

Then, we have

$$\sum_{i:w_i \geq 0} w_i = \sum_i |w_i| + w^- = 1 + w^-, \tag{11}$$

$$\sum_i w_i = \sum_i |w_i| + 2w^- = 1 + 2w^-. \tag{12}$$

According to Fact 1, based on Eqs. 10 and 11, we have:

Lemma 2. *Given a fixed $U > 0$, for any $x \in [-U, U]^d$ and $w \in \mathbb{R}^d$, we have*

$$\sum_i w_i(1-cos(x_i)) \geq \frac{1}{2}\sum_i w_i x_i^2 - \frac{1}{24}(1+w^-)U^4, \tag{13}$$

$$\sum_i w_i(1 - cos(x_i)) \leq \frac{1}{2}\sum_i w_i x_i^2 - \frac{1}{24} w^- U^4. \tag{14}$$

Let $x = o - q$. Lemma 2 shows that $\sum_i w_i(1-cos(o_i - q_i))$ is a good approximation to $d_w(o, q)$.

Until now, we have presented the spherical asymmetric transformation and discussed some theoretical properties behind its intuition. Next, we will discuss the choice of LSH functions after the spherical asymmetric transformation, and present two ALSH schemes Spherical L2-ALSH (or simply SL-ALSH) and Spherical SRP-ALSH (or simply S2-ALSH) which convert NNS over d_w into NNS over L_2 distance and NNS over Angular distance, respectively.

4.2. SL-ALSH

We first introduce SL-ALSH which converts NNS over d_w into NNS over L_2 distance. We apply the LSH function $h_{l_2}(\cdot)$ after the spherical asymmetric transformation.

According to Eqs. 8 and 9, we have:

$$\begin{aligned} &\|P(o) - Q(q, w)\|_2^2 \\ &= \sum_i (1 - w_i)^2 + 2\sum_i w_i(1 - cos(o_i - q_i)). \end{aligned} \tag{15}$$

Thus, based on Eqs. 4 and 15, the collision probability for certain o, q, w is computed as follows:

$$\begin{aligned} &Pr[h_{l_2}(P(o)) = h_{l_2}(Q(q, w))] \\ &= p^{(l_2)}\left(\sqrt{\sum_i (1 - w_i)^2 + 2\sum_i w_i(1 - cos(o_i - q_i))}\right). \end{aligned} \tag{16}$$

Let $R_1 < R_2$. Based on Lemma 2, if $d_w(o, q) \leq R_1$, then

$$\begin{aligned} &Pr[h_{l_2}(P(o)) = h_{l_2}(Q(q, w))] \\ &\geq p^{(l_2)}\left(\sqrt{\sum_i (1 - w_i)^2 + R_1 - \frac{1}{12} w^- U^4}\right); \end{aligned} \tag{17}$$

If $d_w(o, q) \geq R_2$, then

$$\begin{aligned} &Pr[h_{l_2}(P(o)) = h_{l_2}(Q(q, w))] \\ &\leq p^{(l_2)}\left(\sqrt{\sum_i (1 - w_i)^2 + R_2 - \frac{1}{12}(1 + w^-) U^4}\right). \end{aligned} \tag{18}$$

Let $p_1^{(l_2)}$ and $p_2^{(l_2)}$ be the right-hand side of Inequalities 17 and 18, respectively. In order to satisfy $p_1^{(l_2)} > p_2^{(l_2)}$, we require

$$U < \sqrt[4]{12(R_2 - R_1)}, \tag{19}$$

which can be satisfied by selecting a small U.

Thus, we have Lemma 3 as follows:

Lemma 3. *A family of hash functions $h_{l_2}(\cdot)$, along with $P : \mathbb{R}^d \to \mathbb{R}^{2d}$ and $Q : \mathbb{R}^d \to \mathbb{R}^{2d}$ as defined by Eqs. 8 and 9, is $(R_1, R_2, p_1^{(l_2)}, p_2^{(l_2)})$-sensitive for the distance d_w, where $R_1 < R_2$ and $p_1^{(l_2)} > p_2^{(l_2)}$.*

According to Theorem 1, the hashing quality of SL-ALSH can be computed as $\rho^{(l_2)} = \frac{\ln(p_1^{(l_2)})}{\ln(p_2^{(l_2)})}$. As we discussed, rescaling w does not change the order of the NNS results, but it will change the $\rho^{(l_2)}$ value. Since we assume $\|w\|_1 = 1$, we introduce a scale factor λ ($\lambda > 0$) to rescale w and consider the weight vector $w' = \lambda w$. We discover that $\rho^{(l_2)}$ is minimized when $\lambda = \sqrt{d}/\|w\|_2$. The detailed proof can be found in the supplementary file. Let $\eta = \sqrt{d}\|w\|_2$.

Based on Eqs. 4 and 16 and Inequalities 17 and 18, the minimum value of $\rho^{(l_2)}$ is computed as follows:

$$\rho_{min}^{(l_2)} = \min_{r,U} \frac{\ln(p^{(l_2)}(\sqrt{2\eta - 2(1+2w^-) + R_1 - \frac{1}{12}w^- U^4}))}{\ln(p^{(l_2)}(\sqrt{2\eta - 2(1+2w^-) + R_2 - \frac{1}{12}(1+w^-)U^4}))}. \tag{20}$$

Thus, based on Lemma 3, we have Theorem 3 as follows:

Theorem 3. *Given a family of hash functions $h_{l_2}(\cdot)$, along with P and Q as defined by Eqs. 8 and 9, which is $(R_1, R_2, p_1^{(l_2)}, p_2^{(l_2)})$-sensitive, SL-ALSH is a weight-oblivious data structure for (R_1, R_2)-NNS over d_w with $O(n^{1+\rho_{min}^{(l_2)}})$ space and $O(dn^{\rho_{min}^{(l_2)}} \log_{1/p_2}(n))$ query time, where $\rho_{min}^{(l_2)}$ is defined by Eq. 20.*

Even though $\rho_{min}^{(l_2)}$ is achieved when $\lambda = \sqrt{d}/\|w\|_2$, we still need to check a large number of combinations of r and U to find $\rho_{min}^{(l_2)}$. Next, we will introduce a simple ALSH scheme S2-ALSH which is independent of r.

4.3. S2-ALSH

We now introduce S2-ALSH which reduces NNS over d_w to NNS over Angular distance. We apply the LSH function $h_{srp}(\cdot)$ after the spherical asymmetric transformation.

According to Eqs. 8 and 9, we have

$$\frac{P(o)^T Q(q,w)}{\|P(o)\|_2 \|Q(q,w)\|_2} = \frac{\sum_i w_i - \sum_i w_i(1 - cos(o_i - q_i))}{\sqrt{d}\|w\|_2}. \tag{21}$$

Based on Eqs. 6, 12, and 21, the collision probability for certain o, q, w is computed as follows:

$$Pr[h_{srp}(P(o)) = h_{srp}(Q(q,w))] = 1 - \frac{1}{\pi} cos^{-1}\left(\frac{1 + 2w^- - \sum_i w_i(1 - cos(o_i - q_i))}{\eta}\right). \tag{22}$$

Let $R_1 < R_2$. Based on Lemma 2, if $d_w(o,q) \le R_1$, then

$$Pr[h_{srp}(P(o)) = h_{srp}(Q(q,w))] \ge 1 - \frac{1}{\pi} cos^{-1}\left(\frac{1 + 2w^- - \frac{1}{2}R_1 + \frac{1}{24}w^- U^4}{\eta}\right); \tag{23}$$

If $d_w(o,q) \ge R_2$, then

$$Pr[h_{srp}(P(o)) = h_{srp}(Q(q,w))] \le 1 - \frac{1}{\pi} cos^{-1}\left(\frac{1 + 2w^- - \frac{1}{2}R_2 + \frac{1}{24}(1+w^-)U^4}{\eta}\right). \tag{24}$$

Let $p_1^{(srp)}$ and $p_2^{(srp)}$ be the right-hand side of Inequalities 23 and 24, respectively. In order to satisfy $p_1^{(srp)} > p_2^{(srp)}$, we have the same condition as defined in Inequality 19.

Thus, we have Lemma 4 as follows:

Lemma 4. *A family of hash functions $h_{srp}(\cdot)$, along with $P : \mathbb{R}^d \to \mathbb{R}^{2d}$ and $Q : \mathbb{R}^d \to \mathbb{R}^{2d}$ as defined by Eqs. 8 and 9, is $(R_1, R_2, p_1^{(srp)}, p_2^{(srp)})$-sensitive for the distance d_w, where $R_1 < R_2$ and $p_1^{(srp)} > p_2^{(srp)}$.*

According to Theorem 1, the hashing quality of S2-ALSH is computed as $\rho^{(srp)} = \frac{\ln(p_1^{(srp)})}{\ln(p_2^{(srp)})}$. Based on Eq. 22 and Inequalities 23 and 24, we compute the minimum value of $\rho^{(srp)}$ as follows:

$$\rho_{min}^{(srp)} = \min_U \frac{\ln\left(1 - \frac{1}{\pi} cos^{-1}\left(\frac{1+2w^- - \frac{1}{2}R_1 + \frac{1}{24}w^- U^4}{\eta}\right)\right)}{\ln\left(1 - \frac{1}{\pi} cos^{-1}\left(\frac{1+2w^- - \frac{1}{2}R_2 + \frac{1}{24}(1+w^-)U^4}{\eta}\right)\right)}. \tag{25}$$

Thus, based on Lemma 4, we have Theorem 4 as follows:

Theorem 4. *Given a family of hash functions $h_{srp}(\cdot)$, along with P and Q as defined by Eqs. 8 and 9, which is $(R_1, R_2, p_1^{(srp)}, p_2^{(srp)})$-sensitive, S2-ALSH is a weight-oblivious data structure for (R_1, R_2)-NNS over d_w with $O(n^{1+\rho_{min}^{(srp)}})$ space and $O(dn^{\rho_{min}^{(srp)}} \log_{1/p_2}(n))$ query time, where $\rho_{min}^{(srp)}$ is defined by Eq. 25.*

4.4. Computational Analysis of Hashing Quality

According to Theorems 3 and 4, the query time complexities of SL-ALSH and S2-ALSH depend on the hashing quality ρ. A smaller value of ρ leads to less query time of a method.

We now present a computational analysis of $\rho_{min}^{(l_2)}$ and $\rho_{min}^{(srp)}$ between SL-ALSH and S2-ALSH. According to Eqs. 20 and 25, the values of $\rho_{min}^{(l_2)}$ and $\rho_{min}^{(srp)}$ depend on R_1, R_2, w^-, and η only. R_1 and R_2 are the pre-specified distance threshold. Since $o, q \in [0, U]^d$, we derive that $w^- U^2 \le R_1 < R_2 \le (1 + w^-)U^2$. w^- represents the negativity of w, where $w^- \in [-1, 0]$. Intuitively, η represents the sparsity of w, where $\eta \in [1, \sqrt{d}]$. For example, η gets the minimum value 1 when w is a dense vector with $w_i = 1/d, \forall i$, while for a sparse case where 80% of w_i are 0 and 20% of w_i are $5/d$ (we assume $\|w\|_1 = 1$), $\eta = \sqrt{5}$ which is independent of d.

To show the relationship between $\rho_{min}^{(l_2)}$ and $\rho_{min}^{(srp)}$, we plot the contour diagram of $\rho_{min}^{(l_2)}$ and $\rho_{min}^{(srp)}$ w.r.t R_1 and R_2 for SL-ALSH and S2-ALSH under different settings of w^- and η. We consider three special cases $\rho_{min}^{(l_2)}, \rho_{min}^{(srp)} \in \{0.3, 0.6, 0.9\}$ for all combinations of $w^- \in \{0.0, -0.25, -0.5, -0.75, -1.0\}$ and $\eta \in \{1.0, 1.5, 2.0\}$, where $w^- U^2 \le R_1 < R_2 \le (1 + w^-)U^2$. From Fig. 1, for the same R_1 and R_2, $\rho_{min}^{(l_2)}$ and $\rho_{min}^{(srp)}$ have the maximum value when $w^- = 0.5$. Moreover, $\rho_{min}^{(l_2)}$ and $\rho_{min}^{(srp)}$ increase

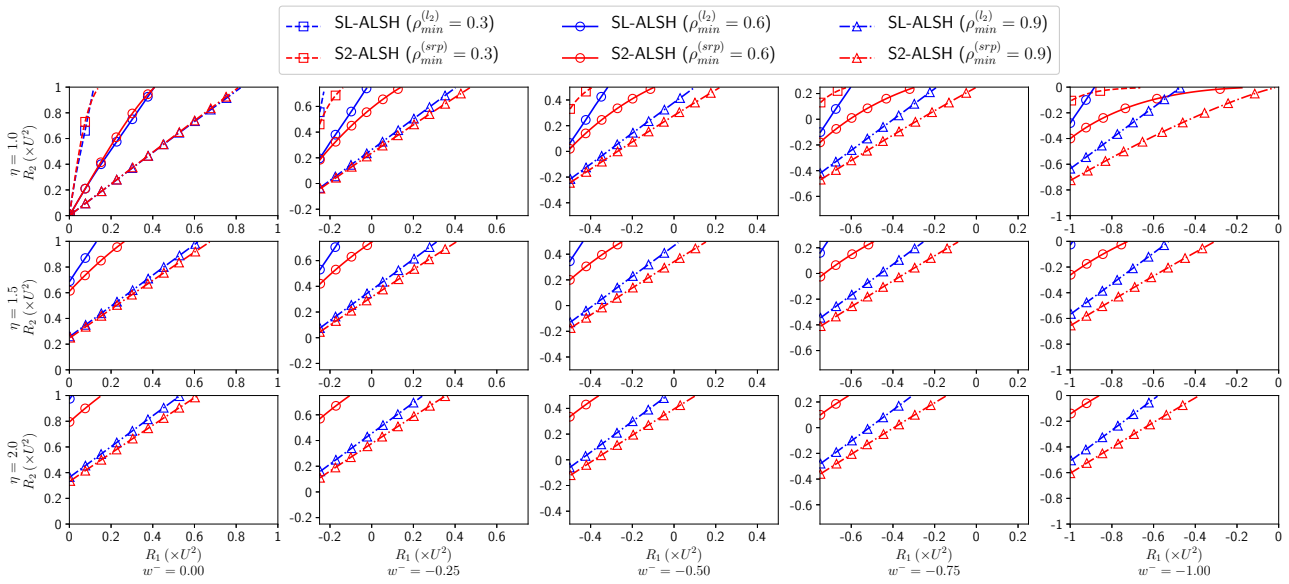

Figure 1. Contour diagram of $\rho_{min}^{(l_2)}$ and $\rho_{min}^{(srp)}$ of SL-ALSH and S2-ALSH w.r.t R_1 and R_2 under different settings of w^- and η.

as η increases. Except for the case $\eta = 1$ and $w^- = 0$, for certain R_1 and R_2, $\rho_{min}^{(l_2)} > \rho_{min}^{(srp)}$. Thus, we have two important findings: Firstly, the closer w^- is to -0.5 and the larger η is, the worse performance SL-ALSH and S2-ALSH have. Secondly, for certain R_1 and R_2, S2-ALSH outperforms SL-ALSH in most cases of w^- and η.

5. Evaluation

In this section, we study the performance of SL-ALSH and S2-ALSH for NNS over d_w on three real-life datasets, i.e., Mnist,[1] Sift,[2] and MovieLens Full[3] (or simply MovieLens). For Mnist and Sift, we randomly sample 1,000 objects from their test sets as queries. For the collaborative filtering dataset MovieLens, we follow the standard pureSVD procedure (Cremonesi et al., 2010) to generate user and item latent vectors and set the latent dimension $d = 150$ (Cremonesi et al., 2010; Shrivastava & Li, 2014; Neyshabur & Srebro, 2015). We randomly sample 1,000 vectors from item vectors as queries and use the rest item vectors as dataset. The statistics of datasets are displayed in Table 1.

In order to demonstrate that both SL-ALSH and S2-ALSH are weight-oblivious, we generate w from five typical distributions, which are illustrated in Table 2.

We follow Shrivastava & Li (2014); Neyshabur & Srebro (2015) and evaluate the performance of SL-ALSH and S2-ALSH with a precision-recall curve. For this task, given a query q with its weight vector w, we first compute the top-10 exact NNs based on $d_w(o, q)$. Then, we compute K different hash values for each q. Finally, we sort and scan all data objects according to the common hash values they match with q. The results are averaged over all queries.

Table 1. Statistics of datasets and queries

Datasets	#Objects	#Queries	d
Mnist	60,000	1,000	784
Sift	1,000,000	1,000	128
MovieLens	52,889	1,000	150

Table 2. Illustrations of five types of weight vectors w

Types	Illustrations
identical	all "1"s
binary	uniformly distributed in $\{0, 1\}^d$
normal	d-dimensional normal distribution $\mathcal{N}(0, I)$
uniform	uniformly distributed in $[0, 1]^d$
negative	all "-1"s

5.1. Impact of Parameters

We study the impact of parameters of SL-ALSH and S2-ALSH. We mainly focus on U and the number of hash functions K.[4] To be concise, we show results on Sift for the binary and normal types of w only. Similar trends can be observed from other datasets and other types of w.

We first study the impact of U. We fix $K = 256$ and plot

[1] http://yann.lecun.com/exdb/mnist/
[2] http://corpus-texmex.irisa.fr/
[3] https://grouplens.org/datasets/movielens/

[4] Compared with S2-ALSH, there is an extra parameter r for SL-ALSH. We set the bucket width r to be 56, 23, and 3 for Mnist, Sift, and MovieLens, respectively, to achieve their best results.

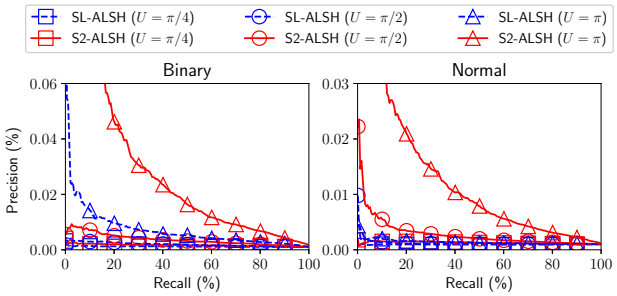

Figure 2. Impact of U

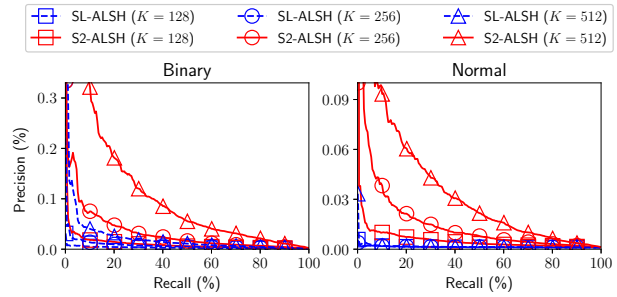

Figure 3. Impact of K

the precision-recall curves of SL-ALSH and S2-ALSH with $U \in \{\pi/4, \pi/2, \pi\}$. From Fig. 2, both schemes achieve the (nearly) best accuracy when $U = \pi$. It seems counter-intuitive because we need to select a small U to satisfy Inequality 19 as we discussed in Sects. 4.2 and 4.3. However, for most real-life datasets, Inequality 19 is too strict, in this case, a large U which violates Inequality 19 is still sufficient to preserve the LSH property. Thus, the (nearly) best precision-recall curves are achieved when $U = \pi$ over the three real-life datasets.

We then study the impact of K. We fix $U = \pi$ and consider $K \in \{128, 256, 512\}$. From Fig. 3, the accuracy increases as K increases. However, the number of hash tables and the corresponding running time increase as K increases. Thus, both schemes have a better trade-off between accuracy and efficiency when $K = 256$.

Based on the above results, we use the settings of $U = \pi$ and $K = 256$ for both schemes in the subsequent experiments.

5.2. Ranking Experiments

In this section, we study the hash code quality of SL-ALSH and S2-ALSH over five types of w. For the identical type of w, we plot the precision-recall curves of E2LSH as the baseline in comparison with SL-ALSH and S2-ALSH.[5]

From Fig. 4, S2-ALSH outperforms SL-ALSH by a large margin in most types of w. These results are consistent with the computational analysis on $\rho_{min}^{(l_2)}$ and $\rho_{min}^{(srp)}$ in Sect. 4.4. Moreover, both schemes work better on Mnist and Sift than on MovieLens, because there exist a tiny fraction of data objects in MovieLens whose L_2 norms are much larger than others. Thus, most of data objects locate inside the center of bounded space after data normalization, and hence the ALSH functions cannot distinguish them. For the negative type of w, we observe an interesting step-like pattern on MovieLens. The reason is that the data objects with much large L_2 norms can be distinguished easily.

In addition, E2LSH outperforms SL-ALSH and S2-ALSH, because SL-ALSH and S2-ALSH which use P and Q for asymmetric transformation will introduce a bit distortion for L_2 distance compared to using E2LSH directly. However, the gap is small on Mnist and Sift, which verifies our explanation that $U = \pi$ satisfies the LSH property in Sect. 5.1. Furthermore, E2LSH only works with the identical type of w, but SL-ALSH and S2-ALSH are weight-oblivious.

5.3. Bucketing Experiments

Although the precision-recall curve is a good indicator of the accuracy of SL-ALSH and S2-ALSH, it, however, does not always reflect their efficiency. Thus, we further study the performance of SL-ALSH and S2-ALSH with the implementation of the vanilla (m, L)-bucketing algorithm (Indyk & Motwani, 1998; Datar et al., 2004). For the identical type of w, we also implement E2LSH as a benchmark method.

The performance of an (m, L)-bucketing algorithm is highly sensitive to the settings of m and L for different datasets. To reduce the impacts of m and L, we follow Shrivastava & Li (2014) and construct 300 hash tables for different $m \in \{1, 2, \ldots, 30\}$. We then check the candidate set formed by the first L out of 300 hash tables for each query, where $L \in \{1, 2, \ldots, 300\}$. We plot the smallest fraction of data objects to scan, i.e., the ratio between the size of candidate set and the cardinality of dataset, to achieve certain level of recall for top-10 NNs among all combinations of m and L.

From Fig. 5, clear sublinear time curves of SL-ALSH and S2-ALSH are observed from most types of w. Specifically, for Mnist and Sift with identical, binary, and uniform types of w, both schemes outperform linear scan by one to two orders of magnitudes. Moreover, S2-ALSH outperforms SL-ALSH in most types of w, which is consistent with the results from Sect. 5.2. For MovieLens with the normal and negative types of w, there is an interesting pattern that a part of NNs can be found easily. These results are consistent with those from Sect. 5.2 and can also be explained by the long-tailed L_2 norm property of MovieLens.

Furthermore, S2-ALSH is comparable to E2LSH on Mnist and Sift, because despite accuracy loss introduced by P and Q, the binary signatures of S2-ALSH have better hashing quality than the integer ones of E2LSH. For MovieLens,

[5] For E2LSH, we shift and rescale the datasets into $[0, 1]^d$, and set the bucket width r to 8, 1, and 0.002 for Mnist, Sift, and MovieLens, respectively, to achieve their best results.

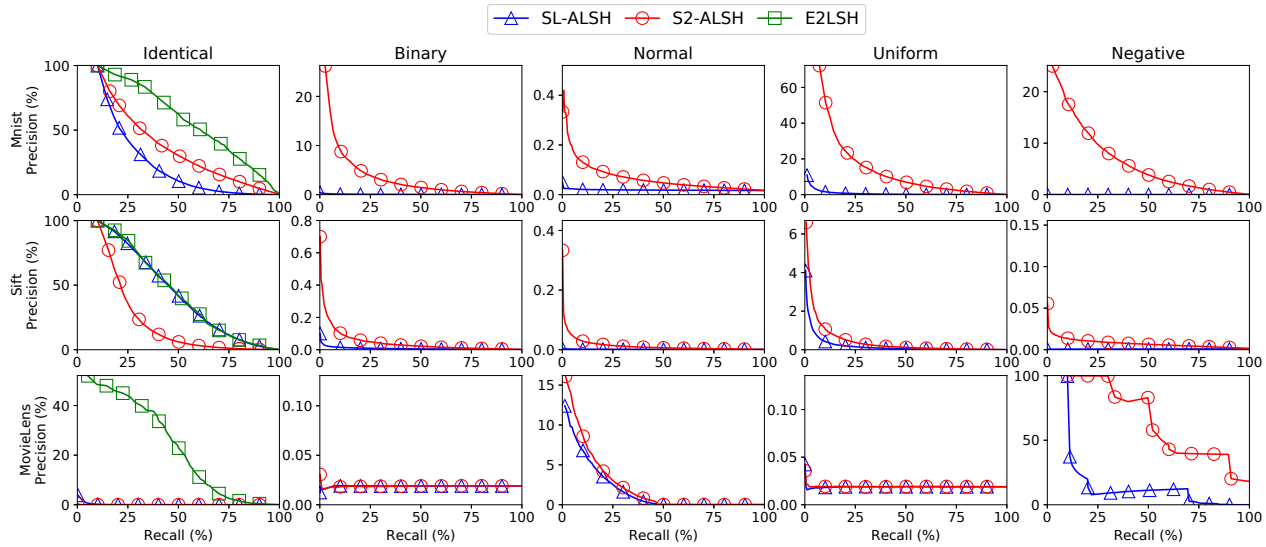

Figure 4. Ranking Experiments. Precision-recall curves of retrieving top-10 NNs (higher is better).

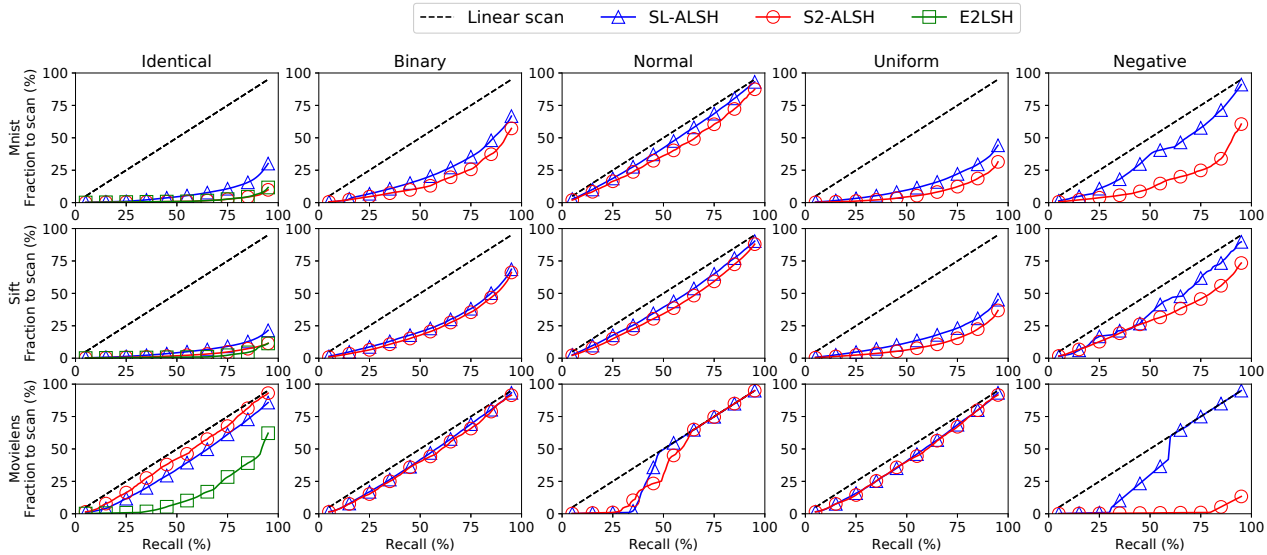

Figure 5. Bucketing Experiments. The best fraction of dataset to scan to achieve certain level of recalls among all combinations of $m \in \{1, 2, \ldots, 30\}$ and $L \in \{1, 2, \ldots, 300\}$ (Lower is better).

E2LSH outperforms both schemes by a large margin due to the long-tailed L_2 norm property of MovieLens.

5.4. Summary

Based on the above results, we have three important observations: Firstly, SL-ALSH and S2-ALSH answer the NNS queries in sublinear time. Specifically, their running time is much less than that of linear scan by up to two orders of magnitudes. Secondly, both schemes are weight-oblivious, which support five different types of w, while E2LSH only works with the identical type of w. Finally, for most types of w, S2-ALSH outperforms SL-ALSH in terms of accuracy.

6. Conclusion

In this paper, we study a fundamental problem of NNS over generalized weighted space. We first demonstrate that there is no ALSH scheme for NNS over d_w in $\mathbb{R}^d$. To address this challenging problem, we introduce a novel spherical asymmetric transformation and propose the first two sublinear time ALSH schemes SL-ALSH and S2-ALSH which are weight-oblivious data structures. Both SL-ALSH and S2-ALSH enjoy a quality guarantee. Extensive experiments over three real-life datasets verify that SL-ALSH and S2-ALSH answer the NNS queries in sublinear time and support various types of weight vectors.

Acknowledgements

This research is supported by the National Research Foundation Singapore under its AI Singapore Programme and the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative.

References

Andoni, A. and Indyk, P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, pp. 459–468, 2006.

Andoni, A., Indyk, P., and Razenshteyn, I. Approximate nearest neighbor search in high dimensions. *arXiv preprint arXiv:1806.09823*, 2018.

Bentley, J. L. K-d trees for semidynamic point sets. In *SoCG*, pp. 187–197, 1990.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. When is "nearest neighbor" meaningful? In *ICDT*, pp. 217–235, 1999.

Bhattacharya, G., Ghosh, K., and Chowdhury, A. S. Granger causality driven AHP for feature weighted knn. *Pattern Recognition*, 66:425–436, 2017.

Charikar, M. S. Similarity estimation techniques from rounding algorithms. In *STOC*, pp. 380–388, 2002.

Cremonesi, P., Koren, Y., and Turrin, R. Performance of recommender algorithms on top-n recommendation tasks. In *RecSys*, pp. 39–46, 2010.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. Locality-sensitive hashing scheme based on p-stable distributions. In *SoCG*, pp. 253–262, 2004.

Fernandez, A. M., Esuli, A., and Sebastiani, F. Learning to weight for text classification. *TKDE*, 2018.

Gan, J., Feng, J., Fang, Q., and Ng, W. Locality-sensitive hashing scheme based on dynamic collision counting. In *SIGMOD*, pp. 541–552, 2012.

Gu, Y., Zhao, B., Hardtke, D., and Sun, Y. Learning global term weights for content-based recommender systems. In *WWW*, pp. 391–400, 2016.

Guttman, A. R-trees: A dynamic index structure for spatial searching. In *SIGMOD*, pp. 47–57, 1984.

Huang, Q., Feng, J., Zhang, Y., Fang, Q., and Ng, W. Query-aware locality-sensitive hashing for approximate nearest neighbor search. *Proc. VLDB*, 9(1):1–12, 2015.

Huang, Q., Feng, J., Fang, Q., Ng, W., and Wang, W. Query-aware locality-sensitive hashing scheme for l_p norm. *The VLDB Journal*, 26(5):683–708, 2017.

Huang, Q., Ma, G., Feng, J., Fang, Q., and Tung, A. K. Accurate and fast asymmetric locality-sensitive hashing scheme for maximum inner product search. In *SIGKDD*, pp. 1561–1570, 2018.

Indyk, P. and Motwani, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC*, pp. 604–613, 1998.

Katayama, N. and Satoh, S. The sr-tree: An index structure for high-dimensional nearest neighbor queries. *ACM SIGMOD Record*, 26(2):369–380, 1997.

Liu, Z., Huang, H., He, Q., Chiew, K., and Gao, Y. Rare category exploration on linear time complexity. In *DASFAA*, pp. 37–54, 2015.

Neyshabur, B. and Srebro, N. On symmetric and asymmetric lshs for inner product search. In *ICML*, pp. 1926–1934, 2015.

Shrivastava, A. and Li, P. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *NeurIPS*, pp. 2321–2329, 2014.

Shrivastava, A. and Li, P. Asymmetric minwise hashing for indexing binary inner products and set containment. In *WWW*, pp. 981–991, 2015a.

Shrivastava, A. and Li, P. Improved asymmetric locality sensitive hashing (alsh) for maximum inner product search (mips). In *UAI*, pp. 812–821, 2015b.

Sun, Y., Wang, W., Qin, J., Zhang, Y., and Lin, X. Srs: solving c-approximate nearest neighbor queries in high dimensional euclidean space with a tiny index. *Proc. VLDB*, 8(1):1–12, 2014.

Wang, H., Wang, N., and Yeung, D.-Y. Collaborative deep learning for recommender systems. In *SIGKDD*, pp. 1235–1244, 2015.

Weber, R., Schek, H.-J., and Blott, S. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, volume 98, pp. 194–205, 1998.

Yan, X., Li, J., Dai, X., Chen, H., and Cheng, J. Norm-ranging lsh for maximum inner product search. In *NeurIPS*, pp. 2956–2965, 2018.

Zheng, Y., Guo, Q., Tung, A. K., and Wu, S. Lazylsh: Approximate nearest neighbor search for multiple distance functions with a single index. In *SIGMOD*, pp. 2023–2037, 2016.