On the Complexity of Maximum Inner Product Search *

Thomas D. Ahle*, Rasmus Pagh*, Ilya Razenshteyn**, and Francesco Silvestri*

*IT University of Copenhagen, {thdy,pagh,fras}@itu.dk
**MIT CSAIL, ilyaraz@mit.edu

Abstract

This paper investigates the complexity of Maximum Inner Product Search (MIPS) and considers both the signed and unsigned versions of the problem. Given a collection $\mathcal{P} \subset \mathbb{R}^d$ of n vectors and a query vector $q \in \mathbb{R}^d$, the signed MIPS returns the vector $p \in \mathcal{P}$ that maximizes the inner product $q^T p$, that is $\arg\max_{p\in\mathcal{P}}\{q^T p\}$. On the other hand, the unsigned MIPS returns the vector $p \in \mathcal{P}$ that maximizes the absolute value of the inner product, that is $\arg\max_{p\in\mathcal{P}}\{|q^T p|\}$.

We first show that the approximate versions of the signed and unsigned MIPS problems are SETH hard by providing a reduction from the Orthogonal Vectors Problem (OVP). Suppose that OVP cannot be solved in $O\left(n^{2-\epsilon}d^{O(1)}\right)$ for any constant $0 < \epsilon < 1$. Then, if the approximation factor is $1/2^{\log^{1/2-\gamma} n} < c \leq 1$ for some constant $0 < \gamma \leq 1/2$, it is not possible to obtain a data structure with $O\left(n^{1+\alpha}d^{O(1)}\right)$ construction time and $O\left(n^{1-\epsilon}d^{O(1)}\right)$ query time for c-approximate signed and unsigned MIPS, for any constants $0 < \epsilon \leq 1$ and $\alpha \geq 0$.

Then, we focus on a particular approach for solving MIPS, namely Locality Sensitive Hashing. Consider any asymmetric LSH where vectors with inner product larger than S collide with probability at least P_1, and vectors with inner product smaller than cS collide with probability at most P_2. Suppose that data and query domains are balls of radius one and $U > 0$ respectively, then we have $P_1 - P_2 = O\left(1/\log(U/(S(1-c)))\right)$ for $d \geq 2$ dimensions.

Finally, we propose efficient data structures for signed and unsigned MIPS. In particular, we propose a data structure based on linear sketches for unsigned MIPS that yields a $c = 1/n^\delta$ approximation with $\tilde{O}\left(dn^{2-2/\delta}\right)$ construction time and $\tilde{O}\left(dn^{1-2\delta}\right)$ query time, for every $0 < \delta \leq 1/2$. The previous conditional lower bound gives evidence that we cannot significantly improve the approximation factor with similar query and construction bounds.

*The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no. 614331.

1 Introduction

The *Maximum Inner Product Search* (MIPS) is defined as follows. Given a set $\mathcal{P} \subset \mathbb{R}^d$ of n vectors, construct a data structure that efficiently returns the vector in $\mathcal{P}$ that maximizes a suitable function of the inner product[1] with a query vector $q \in \mathbb{R}^d$. MIPS arises in different data-mining and machine-learning applications. A well-know example is provided by recommendation systems based on matrix-factorization [14, 23]. In these settings, an user and the available items are represented as vectors and the preference of an user for an item is given by the inner product of the two associated vectors; the retrieval of a recommendation for an user is equivalent to a MIPS where $\mathcal{P}$ is the set of items' vectors and the query is the user's vector. Other examples of applications for MIPS are object detection [10] and multi-class prediction [9, 12].

This paper investigates the complexity of MIPS. We are interested in two variants of the problem that slightly differ on the formulation of the objective function. The first variant is the *Signed Maximum Inner Product Search* (signed MIPS), which returns the vector $\hat{p}$ in $\mathcal{P}$ that maximizes the inner product with any given query q: $\hat{p} = \arg\max_{p \in \mathcal{P}} q^T p$. The second variant is the *Unsigned Maximum Inner Product Search* (unsigned MIPS), which returns the vector $\hat{p}$ in $\mathcal{P}$ that maximizes the *absolute value* of the inner product with any given query q: $\hat{p} = \arg\max_{p \in \mathcal{P}} |q^T p|$. We observe that unsigned MIPS for a query q can be solved with signed MIPS by performing queries q and $-q$ and then returning the output vector with the largest absolute scalar product. The signed MIPS variant is of interest when searching for similar or preferred items with a positive correlation, like in recommendation systems. On the other hand, the unsigned MIPS can be used when studying relations among some phenomena, where even a large negative correlations is of interest.

Our focus is on approximate algorithms for signed and unsigned MIPS. That is, given a value $0 < c < 1$, the *c-approximate signed MIPS* (resp., *c-approximate unsigned MIPS*) returns a vector p' which is at most a factor c from the value of the optimal solution $\hat{p}$, that is $q^T p' \geq cSq^T \hat{p}$ (resp., $|q^T p'| \geq |cSq^T \hat{p}|$). Approximate algorithms allow to overcome the curse of dimensionality without significantly affecting the final results.

1.1 Previous results

An efficient solution for signed MIPS is proposed in [20, 13] which is based on tree data structures combined with a branch-and-bound space partitioning technique similar to k-d trees. However, as many similarity search problems, the exact version suffers from the curse of dimensionality [26]. The efficiency of approximate approaches for MIPS based on Locality Sensitive Hashing (LSH) is studied in [21, 18]. These papers show that a traditional LSH does exist when the data domain is the unit ball and the query domain is the unit sphere, while it does not exist when the two domains coincide. On the other hand an asymmetric LSH exists in this case, but it cannot be extended to the unbounded domain $\mathbb{R}^d$. An asymmetric LSH for binary inner product is proposed in [22]. Unsigned MIPS is clearly equivalent to signed MIPS when the vectors are non-negative and the most related problem is the estimation of the ℓ_∞ norm (see e.g., [11, 2]) of Aq, where A represents the $n \times d$ matrix of the input set $\mathcal{P}$. However, the goal of unsigned MIPS is to return the vector that maximizes the absolute value of the inner product, and not the value itself as for ℓ_∞ norm. Finally we remark that, when data and query vectors have the same norm, signed/unsigned MIPS can be solved with techniques for ℓ_2 norm and for cosine similarity [20].

[1] MIPS is usually used in literature for denoting the problem that we call here Signed MIPS. However, in this paper we prefer to use MIPS for denoting a generic search problem based on a suitable function of the inner product.

1.2 Our results

Similarly to the literature on nearest neighbor search, the results in the paper are provided for "threshold versions" of the approximate problems. For $S > 0$ and $0 < c < 1$, the (c, S)-*signed MIPS* (resp., (c, S)-*unsigned MIPS*) problem requires to construct a data structure that reports, for any query q, a vector $p \in \mathcal{P}$ with $q^T p \geq cS$ (resp., $|q^T p| \geq cS$) if there exists a vector $\hat{p} \in \mathcal{P}$ with $q^T \hat{p} \geq S$ (resp., $|q^T \hat{p}| \geq S$).

The first results of the paper are conditional lower bounds for approximate signed and unsigned MIPSs that rely on the *Orthogonal Vectors Problem* (OVP). This problem consists in determining if two sets A and B, each one with n d-dimensional vectors, contain two vectors $x \in A$ and $y \in B$ such that $x^T y = 0$. It is known [27] that OVP cannot be solved in $O\left(n^{2-\epsilon} d^{O(1)}\right)$ time, for any constant $0 < \epsilon \leq 1$, if the Strong Exponential Time Hypothesis (SETH) is true. Our results give evidence that getting $O\left(n^{1-\epsilon} d^{O(1)}\right)$ query time and $O\left(n^{1+\alpha} d^{O(1)}\right)$ construction time, for any constants $0 < \epsilon \leq 1$, $\alpha \geq 0$, is hard for (c, S)-signed and unsigned MIPS if the approximation constant is $1/2^{\log^{1/2-\gamma} n} \leq c \leq 1$ for an arbitrary $0 < \gamma < 1/2$ and a suitable small threshold $0 < S \leq 1$. From a technical point of view, the most interesting part is in the reduction from OVP to unsigned MIPS: a Chebyshev embedding [25] is used for mapping two vectors x and y of the OVP input sets to vectors x' and y' such that $|x'^T y'| \geq S$ if $x^T y = 0$ and $|x'^T y'| \leq cS$ otherwise, for a given threshold S and approximation c.

Then, we investigate the limitations of asymmetric LSH for signed and unsigned MIPS by extending the aforementioned impossibility results in [18, 21]. We provide a lower bound on the gap between the collision probability P_1 of vectors with inner product larger than S and the collision probability P_2 of vectors with inner product smaller than cS. Specifically we show, that for any asymmetric LSH for data vectors in the unit ball and query vectors in a ball of radius U, we have $P_1 - P_2 = O\left(1/\log(U/(S(1-c)))\right)$ when $d \geq 2$ and $P_1 - P_2 = O\left(1/\log\log_{1/c}(U/S)\right)$ when $d = 1$. This result shows that there cannot exist an asymmetric LSH when query vectors are unbounded, then including the aforementioned result in [18].

Finally we provide some insights on the upper bounds. We first show that it is possible to improve the asymmetric LSH in [18, 22] by just plugging the best known data structure for Approximate Near Neighbor for ℓ_2 on a sphere [7] into the reduction in [18]. Then we show how to circumvent the impossibility results in [18, 21] by showing that there exists a symmetric LSH when the data and query space coincide by allowing the bounds on collision probability to not hold for a few pairs of vectors. We conclude by describing a data structure based on the linear sketches for ℓ_p in [3] for c-approximate unsigned MIPS, which yields a $c = 1/n^{\kappa}$ approximation with $\tilde{O}\left(dn^{2-2/\kappa}\right)$ construction time and $\tilde{O}\left(dn^{1-2/\kappa}\right)$ query time, for every $0 < \kappa \leq 1/2$. Although this result is not that strong, the conditional lower bound shows that we cannot improve the approximation with similar construction and query performance.

2 Preliminaries

2.1 Approximate MIPS

We let $||p||$ and $||x||_\infty$ denote the ℓ_2 and ℓ_∞ norms of a vector $p \in \mathbb{R}^d$, and with $|x|$ the absolute value of the scalar $x \in R$.

As already mentioned, many technical results in the paper are for (c, S)-signed/unsigned MIPS, since the c-approximate problem can be reduced to the (c, S)-one with a logarithmic slowdown. Indeed, c-signed MIPS (similarly for unsigned MIPS) can be solved with $k = \log_{1/c} \Delta$ data structures for (c, c^i)-signed MIPS, for any $0 \leq i < k$ and where $1/\Delta$ is the smallest distance that can be

stored in an unit word. Then for any query q, the k data structures are iteratively (starting from $i = 0$) queried with $q/||q||$ until a vector with inner product larger than c^{i+1} is returned.

For notational simplicity, we assume the data domain to be a ball with radius one. Since the data vectors are given as input of the construction process and do not change dynamically, it is always possible to scale down the vectors and to adjust accordingly the query vector. On the other hand, we assume the query domain to be a ball of radius $U > 0$. Stated differently U can be seen as the ratio between the radii of the smallest balls enclosing the query and data domains. Although the value of U is unimportant for the c-approximation signed/unsigned MIPS (the norm of the query vector does not affect the final solution), we will see that the value U is crucial in LSH approaches and determines the quality of collision probabilities.

We use the following definition of asymmetric LSH based on the definition in [21]. Let $\mathcal{U}_p$ denote the data domain and $\mathcal{U}_q$ the query domain. Consider a family $\mathcal{H}$ of pairs of hash functions $h = (h_p(\cdot), h_q(\cdot))$. Then $\mathcal{H}$ is said (S, cS, P_1, P_2)-*asymmetric LSH* for a similarity function *sim* if for any $p \in \mathcal{U}_p$ and $q \in \mathcal{U}_q$ we have: 1) if $sim(p, q) \geq S$ then $\Pr_{\mathcal{H}}[h_p(p) = h_q(q)] \geq P_1$; 2) if $sim(p, q) < cS$ then $\Pr_{\mathcal{H}}[h_p(p) = h_q(q)] \leq P_2$. When $h_p(\cdot) = h_q(\cdot)$, we get the traditional LSH definition, which we denote with *symmetric LSH*. The ρ value of an (asymmetric) LSH is defined as usual with $\rho = \log(1/P_1)/\log(1/P_2)$ [4]. Finally, two vectors $p \in \mathcal{U}_p$ and $q \in \mathcal{U}_q$ are said to collide under an hash function of $\mathcal{H}$ if $h_p(p) = h_q(q)$.

2.2 Orthogonal Vectors Problem (OVP)

The Orthogonal Vectors Problem (OVP) is defined as follows. Given two sets A and B, each one containing n vectors in $\{0, 1\}^d$, detect if there exist vectors $a \in A$ and $b \in B$ such that $a^T b = 0$. OVP derives its hardness from the Strong Exponential Time Hypothesis, and the connection was proved by Williams [27]. We remark that the conjectures are assumed to hold even against randomized algorithms [1]. We will therefore assume the following plausible conjecture:

Conjecture 1 ([27])**.** *The Orthogonal Vectors Problem on n vectors with $d = O(\log n)$ cannot be solved in time $O\left(n^{2-\varepsilon}\right)$, for all $0 < \varepsilon \leq 1$.*

2.3 Chebyshev embedding

In the reduction from OVP to unsigned MIPS we will exploit the Chebyshev embedding introducted by G. Valiant [25]. We recall here the results that will be used in the reduction. Let denote with $\Lambda_q(x)$ the first-kind Chebyshev polynomial of degree q. Consider two sets A and B, each one containing n vectors in $\{-1, 1\}^d$. Valiant showed that there exists an asymmetric embedding $(f^c_{A,B}(\cdot), f^c_{B,A}(\cdot))$, that maps each vector $x \in A$ and $y \in B$ to vectors $f^c_{A,B}(x)$ and $f^c_{A,B}(y)$ in $\{-1, 1\}^{d'}$, for any $d' > 0$, with the following guarantees:

Lemma 1 (Proposition 6 in [25])**.** *Consider the embedding $(f^c_{A,B}(\cdot), f^c_{B,A}(\cdot))$ in [25] with input sets $A = \{a_0, \ldots, a_{n-1}\}$ and $B = \{b_0, \ldots, b_{n-1}\}$ and input parameters d', $\tau_-, \tau_+ \in [-1, 1]$ for $\tau_- < \tau_+$. The embedding requires time $O(nqd')$ and, with probability $1 - o(1)$ over the randomness in the construction, we have for every $i, j \in [0, n]$*

$$\left| f^c_{A,B}(a_i)^T f^c_{A,B}(b_j) - \Lambda_q\left(\frac{(a_i^T b_j)/d - \tau_-}{\tau_+ - \tau_-}\right) \cdot d' \cdot (\tau_+ - \tau_-)^q \cdot \frac{1}{2^{3q-1}} \right| \leq \sqrt{d'} \log n.$$

Since $|\Lambda_q(x)| \leq 1$ for any $x \in [-1, 1]$ and that $\Lambda_q(1 + \delta) \geq e^{q\sqrt{\delta}}/2$ [25], the above embedding allows to increase by a factor $\Lambda_q(\epsilon) \geq e^{q\sqrt{\epsilon}}/2$ the gap between inner products of value w and $(1 + \epsilon)w$, for any $w > 0$ and $0 < \epsilon < 1/2$, by suitably tuning the input parameters.

3 Reductions from OVP

In this section we propose conditional lower bounds for signed and unsigned MIPS problems that rely on the OVP conjecture, which is known to be SETH hard [27]. We show that, if the OVP conjecture is true and the approximation factor is $c \geq 1/2^{\log^{1/2-\gamma} n}$ for an arbitrary $0 < \gamma \leq 1/2$, then it is not possible to construct a data structure for (c, S)-signed and unsigned MIPS that require $n^{1+\alpha}d^{O(1)}$ construction time and $n^{1-\epsilon}d^{O(1)}$ query time for any constant $\alpha > 0$ and $0 < \epsilon \leq 1$.

Although the conditional lower bound on unsigned MIPS implies the one for the signed version, we start as a warm-up with a simple reduction from OVP to signed MIPS in Theorem 1. The reduction maps the d-dimensional vectors of an OVP instance to $(d+1)$-dimensional vectors so that orthogonal vectors are mapped to vectors with inner product larger than a value S, while all the remaining vectors are mapped to vectors with inner product smaller than cS. Then a (c, S)-signed MIPS data structure is used for finding a pair with a inner product larger than S.

We provide the reduction to unsigned MIPS in Theorem 2. The structure of the reduction is the same of the previous one, however a different mapping is used for dealing with the absolute value in the objective function of unsigned MIPS. Vectors are mapped to $2^{\Theta\left(\sqrt{d}\log n\right)}$-dimension vectors with a Chebyshev embedding [25], which guarantees that orthogonal vectors are mapped to vectors with inner product larger than a suitable value S, while all the remaining vectors are mapped to vectors with the absolute value of the inner product smaller than cS. In addition, the Chebyshev embedding allows to derive a conditional lower bound that applies for a wider range of the approximation factor c than the mapping used for signed MIPS.

Theorem 1. *Consider a data structure $\mathcal{D}$ for the (c, S)-signed MIPS in the unit ball, with $0 < S \leq 1$ and $1 - (1-S)/(Sd) \leq c \leq 1$. Then, the OVP conjecture implies that $\mathcal{D}$ cannot have construction time $n^{1+\alpha}d^{O(1)}$ and query time $n^{1-\varepsilon}d^{O(1)}$ for any constant value $\alpha \geq 0$ and $0 < \epsilon \leq 1$.*

Proof. Suppose that there exists a data structure $\mathcal{D}$ with construction time $n^{1+\alpha}d^{O(1)}$ and query time $n^{1-\varepsilon}d^{O(1)}$ for some constant values $\alpha \geq 0$ and $0 < \epsilon \leq 1$. Consider an instance of the OVP problem with input sets A, B with size n and dimension $d = O(\log n)$.

Let denote with A' (resp., B') the set containing the mapping of each vector $x = (x_0, \ldots, x_{d-1})$ in A (resp., in B) to the $(d+1)$-dimension vector $f_A(x) = (f(x_0), \ldots, f(x_{d-1}), \sqrt{S})$ (resp., $f_B(x) = (-f(x_0), \ldots, -f(x_{d-1}), \sqrt{S})$), where $f(x_i) = x_i\sqrt{S(1-c)}$. For any $x \in A$ and $y \in B$, we have that $f_A(x)^T f_B(y) = S\left(1 - x^T y(1-c)\right)$.

Partition the set B' into subsets $B'_0, \ldots, B'_{n^{1-\delta}-1}$ of size n^δ for a value $0 < \delta < 1$ that will be set later. For each subset B'_i we construct a data structure D_i for (c, s)-signed MIPS. Then, for each vector $x' \in A'$, we perform a query x' in D_i for each $0 \leq i < n^{1-\delta}$. If and only if there exists a pair of vectors $x \in A$ and $y \in B$ with $x^T y = 0$, then at least one query must return a point with inner product larger than S. Indeed, we have $f_A(x)^T f_B(y) = S$ if $x^T y = 0$ and $f_A(x)^T f_B(y) \leq cS$ otherwise (note that the inner product can be even smaller than $-S$ in this case, which prevents us from using a data structure for (c, S)-unsigned MIPS). We observe that it is possible to use the data structure $\mathcal{D}$ with vectors in A' and B' since all vectors in A' and B' are contained in the unit ball, being $||f_A(x)||^2 = S(1-c)||x|| + S \leq 1$ when $c \geq 1 - (1-S)/(Sd)$ (similarly for $f_B(\cdot)$).

Since there are $n^{1-\delta}$ data structures with n^δ vectors and each one is queried n times, the reduction requires total time:

$$T(n,d) = n^{1-\delta}\left(\left(n^\delta d^{O(1)}\right)^{1+\alpha} + n^{1+\delta(1-\epsilon)}d^{O(1)}\right) = \left(n^{1+\delta\alpha} + n^{2-\epsilon\delta}\right)d^{O(1)}.$$

By setting $\delta = 1/(\alpha+\epsilon)$, we get that $T(n,d) = O\left(n^{1+\alpha/(\alpha+\epsilon)}d^{O(1)}\right)$, which contradicts the OVP conjecture. □

We observe that the mapping used in the previous proof guarantees that orthogonal vectors are mapped to vectors with inner product larger than a value S. However, all the remaining vectors are mapped to vectors with inner product smaller than cS, which can be even smaller than $-S$. Therefore a (c, S)-unsigned MIPS cannot be used, unless a different mapping is used, as the next theorem shows.

Theorem 2. *Consider a data structure $\mathcal{D}$ for the (c, S)-unsigned MIPS in the unit ball, for a sufficiently small S and $1/2^{\log^{1/2-\gamma} n} \leq c \leq 1$ for an arbitrary $0 < \gamma < 1/2$. Then, the OVP conjecture implies that $\mathcal{D}$ cannot have construction time $n^{1+\alpha} d^{O(1)}$ and query time $n^{1-\varepsilon} d^{O(1)}$ for any constant value $\alpha \geq 0$ and $0 < \epsilon \leq 1$.*

Proof. Let A and B denote the input sets of OVP, each one containing n vectors with $d = O(\log n)$ dimensions. We suppose that all vectors within a set have the same weight a and b. If this is not the case, the problem can be decomposed into d^2 smaller OVPs where vectors in A with weight a are compared with vectors in B with weight b for each $0 < a, b \leq d$. This increases the running time by a multiplicative term d^2 which is negligible, given the aimed bounds.

We map A and B using the following three-step mapping:

1. Each vector $x \in A$ (resp., $y \in B$) is mapped to a $(6d - 2a - 2b)$-dimension vector as follows: the first d positions contain $2x - 1$ (resp., $1 - 2x$); the remaining $5d - 2a - 2b$ positions are set to 1. Note that the domain of the output vectors is $\{-1, 1\}^{6d-2a-2b}$, and that a pair $x \in A$ and $y \in B$ is mapped to a pair with inner product $4(d - x^T y)$. (We note that the $\{-1, 1\}$ domain is required by the Chebyshev embedding used in the next step.)

2. A Chebyshev embedding is then applied to the output vectors of the previous step. We apply Lemma 1 with $t_- = 0$, $t_+ = \Theta((d-1)/d)$, $q = \Theta\left(\sqrt{d}\ln(1/c)\right)$, and $d' = \Theta\left((\Psi \log n)^2\right)$ with $\Psi = e^{\Theta\left(\sqrt{d}\log(1/c)\right)}$. We let $f_A(x)$ and $f_B(y)$ denote the vectors associated with $x \in A$ and $y \in B$ at the end of this step. For each $x \in A$ and $y \in B$ with $x^T y = 0$, we have with high probability that

$$f_A(x)^T f_B(y) \geq \Lambda_q (1 + 1/d) \frac{d'}{\Psi} - \sqrt{d'} \log n \sim \Lambda_q (1 + 1/d) \frac{d'}{\Psi} = \frac{d'}{c\Psi}$$

   On the other hand, for each $x \in A$ and $y \in B$ with $x^T y \geq 1$, we get with high probability that

$$0 \leq \frac{d'}{\Psi} - \sqrt{d'} \log n \leq f_A(x)^T f_B(y) \leq \frac{d'}{\Psi} + \sqrt{d'} \log n \sim \frac{d'}{\Psi}.$$

   We observe that the first value is a factor $1/c$ larger than the second.

3. Finally, vectors returned in the last step are normalized to be in the unit sphere. This last step guarantees that $(f_A(x)^T f_B(y))/\Delta^2 \geq 1/(c\Psi)$ if $x^T y = 0$ and $0 \leq f_A(x)^T f_B(y)/\Delta^2 \leq d'/\Psi$ otherwise.

Suppose now that $S = 1/\Psi$. We let A' and B' denote the sets with the d'-dimensional vectors returned by the above mapping, for $d' = \left(e^{\Theta\left(\sqrt{d}\log(1/c)\right)} \log n\right)^2$. If there exists a pair of vectors $x \in A$ and $y \in B$ with $x^T y = 0$, then there exists a pair $x' \in A'$ and $y' \in B'$ with $x'^T y' \geq S$. On the other hand, each pair $x \in A$ and $y \in B$ with $x^T y > 0$ is associated with a pair $x' \in A'$ and $y' \in B'$ with $0 < x'^T y' \leq cS$. Therefore a data structure for (c, S)-unsigned MIPS suffices for finding an orthogonal input pair in A and B. As in the previous reduction, we split B' into subsets

$B'_0, \dots B'_{n/\delta-1}$ of size n^δ and then insert each B'_i in a (c,S)-MIPS data structure D_i. Then, for each $x' \in A'$, we perform a query x' in each D_i. If a pair with inner product larger than S is returned by at least a query, then the initial OVP instance contains an orthogonal pair. The total time of the reduction is

$$T(n,d') = O\left(nqd'\right) + \left(n^{1+\delta\alpha} + n^{2-\epsilon\delta}\right) d^{O(1)}.$$

The first term is the Chebyshev embedding construction time, while the second term is the cost of the second part of the reduction (see the proof of Theorem 1). Being $q = \sqrt{d}\ln(2/c)$ and $d' = \left(e^{\Theta\left(\sqrt{d}\log(1/c)\right)}\log n\right)^2$, OVP can be solved with high probability in $O\left(n^{2-\epsilon'}d^{O(1)}\right)$ for some $0 \leq \epsilon' < 1$ if $c \geq 1/2^{\log^{1/2-\gamma} n}$ for an arbitrary $0 < \gamma \leq 1/2$. □

4 Limitations of LSH for MIPS

We now focus our attention on approaches for signed/unsigned MIPS based on LSH. We provide a lower bound on the gap between P_1 and P_2 for an (S, cS, P_1, P_2)-asymmetric LSH when data and query domains are respectively the unit ball and a ball of radius U and dimension $d \geq 1$. Specifically, we show that $P_1 - P_2 = O\left(1/\log\log_{1/c}(U/S)\right)$ if $d = 1$ and $P_1 - P_2 = O\left(1/\log(U/(S(1-c)))\right)$ otherwise. The lower bound applies also to data dependent LSH [7]. A consequence of this bound is that there cannot exist an asymmetric LSH for any dimension $d \geq 1$ when the set of query points are unbounded, getting a similar result of [18], which however requires even the data space to be unbounded and $d \geq 2$.

Our proof and the one in [18] are in some sense related since they both rely on a collision matrix (here, we use the term grid) given by two sequences of data and query vectors that force the gap to be small. The proof in [18] then applies an asymptotic analysis of the margin complexity of this matrix [24], and it shows that for any given value of $P_1 - P_2$ there are sufficiently large data and query domains for which the gap must be smaller. Unfortunately, due to their asymptotic analysis, it is not possible to provide an upper bound on the gap for any radius U of the query domain, and it then does not rule out very large gaps for small domains. In addition, our proof holds for $d = 1$ and only uses simple combinatorial arguments.

The following Theorem 3 shows that $P_1 - P_2 = O\left(1/\log h\right)$, where h is the length of two suitable sequences of query and data points with some collision properties. Then, within Corollary 1 we show that the length is $\Theta\left(\log_{1/c}(U/S)\right)$ if $d = 1$ and $\Theta\left(\sqrt{U/(S(1-c))}\right)$ otherwise.

Theorem 3. *Suppose that there exists a sequence of data vectors $\mathcal{P} = \{p_0, \dots, p_{h-1}\}$ and a sequence of query vectors $\mathcal{Q} = \{q_0, \dots, q_{h-1}\}$ such that $q_i^T p_j \geq S$ if $j \geq i$ and $q_i^T p_j \leq cS$ otherwise. Then any (S, cS, P_1, P_2)-asymmetric LSH for these points must satisfy $P_1 - P_2 \leq 1/(8\log h)$.*

Proof. For the sake of simplicity we assume that $h = 2^\ell - 1$ for some $\ell \geq 1$; the assumption can be removed by introducing floor and ceiling operations in the proof. Let $\mathcal{H}$ denote an (S, cS, P_1, P_2)-asymmetric LSH family of hash functions, and let h be a function in $\mathcal{H}$.

Consider the $h \times h$ grid representing the collisions between $\mathcal{Q} \times \mathcal{P}$, that is, a node (i,j) denotes the query-data vectors q_i and p_j. We say that a node (i,j), with $0 \leq i, j < h$, collides under h if vectors q_i and p_j collide under h. By definition of asymmetric LSH, all nodes with $j \geq i$ must collide with probability at least P_1, while the remaining nodes collide with probability at most P_2. We denote with *lower triangle* the part of the grid with $j \geq i$ and with *P_1-nodes* the nodes within it; we refer to the remaining nodes with *P_2-nodes*.

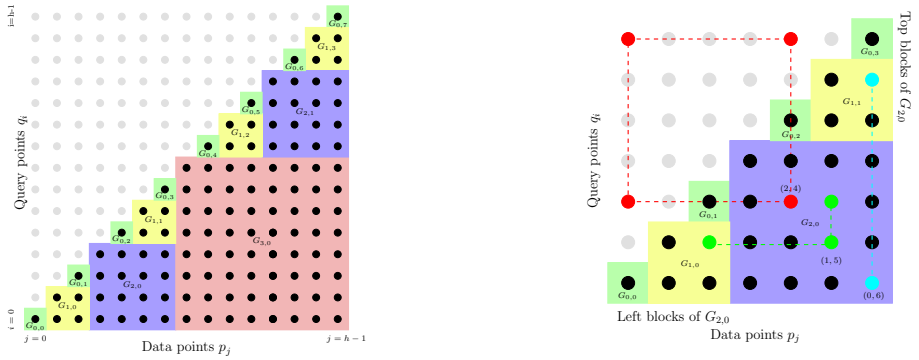

Figure 1: On the left, a 15×15 grid: black nodes are P_1-nodes, gray nodes are P_2-nodes; the colored blocks denote the partitioning of the lower triangle into squares. On the right, a zoom of the $G_{2,0}$ square and of its left and top squares: the red nodes collide under a $(2,4)$-shared function; the green nodes collide under a $(1,5)$-partially shared function; the cyan node collide under a $(0,6)$-proper function (specifically, row proper).

We partition the lower triangle into squares of exponentially increasing side as shown in Figure 1. Specifically, we split the lower triangle into *squares* $G_{r,s}$ for every $0 \leq r < \log(h+1) = \ell$ and $0 \leq s < (h+1)/2^{r+1} = 2^{\ell-r-1}$, where $G_{r,s}$ includes all nodes in the square of side 2^r and top-left node $((2s+1)2^r - 1, (2s+1)2^r - 1)$. For a given square $G_{r,s}$, we define the *left squares* (resp., *top squares*) to be the set of squares that are on the left (resp., top) of $G_{r,s}$. We note that the left squares (resp., top squares) contain 2^{r-i-1} squares of side 2^i for any $0 \leq i < r$ and all P_1-nodes with $s2^{r+1} \leq i, j < (2s+1)2^r - 1$ (resp., $(2s+1)2^r - 1 < i, j \leq (s+1)2^{r+1} - 2$) .

We define the *mass* $m_{i,j}$ of a node (i,j) to be the collision probability, under $\mathcal{H}$, of q_i and p_j. We split the mass of a P_1-node into three contributions called shared mass, partially shared mass and proper mass, which we define below. Consider each P_1-node (i,j) and each function $h \in \mathcal{H}$ where (i,j) collides. Let $G_{r,s}$ be the square containing (i,j) and let $K_{h,i,j}$ denote the set of P_1-nodes (i',j') on the same row or column of (i,j) (i.e., $i' = i$ and $j < j' \leq i$, or $j' = j$ and $i \leq i' < j$) that have the same hash value of (i,j) under h (i.e., $h(i) = h(j) = h(i') = h(j')$). Clearly all nodes in $K_{h,i,j}$ collide under h. For the given node (i,j), we classify h as follows (see Figure 1 for an example):

- (i,j)-*shared function.* $K_{h,i,j}$ contains at least a node (i,j') in a left square, and at least a node (i',j) in a top square.
- (i,j)-*partially shared function.* function h is not in case 1 and $K_{h,i,j}$ contains at least a node node (i,j') with $j' < j$, and at least a node (i',j) with $i' > i$. That is, $K_{h,i,j}$ contains only nodes in $G_{r,s}$ and in the left blocks, or only nodes in $G_{r,s}$ and in the top blocks.
- (i,j)-*proper function.* $K_{h,i,j}$ contains no points (i,j') for any $i \leq j' < j$ or contains no points (i',j) for any $i < i' \leq j$. That is, $K_{h,i,j}$ contains only nodes in the same row or only nodes in the same column of (i,j). Function h is said row (resp., column) proper if there are no nodes in the same row (resp., column). We break ties arbitrary but consistently if $K_{h,i,j}$ is empty

The *shared mass* $m^s_{i,j}$ is the sum of probabilities of all (i,j)-shared functions. The *partially shared mass* $m^{ps}_{i,j}$ is the sum of probabilities of all (i,j)-partially shared functions. The *proper mass* $m^p_{i,j}$ is the sum of probabilities of all functions of (i,j)-proper functions (the row/column proper mass includes only row/column proper functions). We have $m_{i,j} = m^p_{i,j} + m^{ps}_{i,j} + m^s_{i,j}$. The *mass* $M_{r,s}$ of a square $G_{r,s}$ is the sum of the masses of all its nodes, while the *proper mass* $M^p_{r,s}$ is the sum of proper masses of all its nodes. The sum of row proper masses of all nodes in a row is at most

one since a function h is row proper for at most one node in a row. Similarly, the sum of column proper masses of all nodes in a column is at most one. Therefore, we have that $\sum_{r,s} M^p_{r,s} \leq 2h$.

We now show that $\sum_{(i,j)\in G_{r,s}} m^s_{i,j} \leq 2^{2r} P_2$ for every $G_{r,s}$. Consider a node (i,j) in a given $G_{r,s}$. For each (i,j)-shared function h there is a P_2-node colliding under f: indeed, $K_{h,i,j}$ contains nodes (i,j') in the left blocks and (i',j) in the top blocks with $h(i) = h(j) = h(i') = h(j')$ (i.e., $s2^{r+1} \leq j' < (2s+1)2^r - 1$ and $(2s+1)2^r - 1 < i' \leq (s+1)2^{r+1} - 2$); then node (i',j') is a P_2-node since $i' > j'$ and collides under h. By considering all nodes in $G_{r,s}$, we get that all the P_2-nodes that collide in a shared function are in the square of side 2^{r-1} and bottom-right node in $((2s+1)2^r, (2s+1)2^r - 2)$. Since these P_2-nodes have total mass at most $2^{2r}P_2$, the claim follows.

We now prove that $\sum_{(i,j)\in G_{r,s}} m^{ps}_{i,j} \leq 2^{r+1} M^p_{r,s}$. A (i,j)-partially shared function is (i',j) or (i,j')-proper shared h for some $i' < i$ and $j' > j$, otherwise there would be a node in left blocks and a node in top blocks that collide with (i,j) under h, implying that h cannot be partially shared. Since an (i,j)-proper function is partially shared for at most 2^{r+1} nodes in $G_{r,s}$, we get

$$\sum_{(i,j)\in G_{r,s}} m^{ps}_{i,j} \leq 2^{r+1} \sum_{(i,j)\in G_{r,s}} m^p_{i,j} = 2^{r+1} M^p_{r,s}.$$

By the above two bounds, we get

$$M_{r,s} \leq \sum_{(i,j)\in G_{r,s}} m^p_{i,j} + m^{ps}_{i,j} + m^s_{i,j} \leq (2^{r+1}+1) M^p_{r,s} + 2^{2r} P_2.$$

Being $M_{r,s} \geq 2^{2r} P_1$, we get $M^p_{r,s} \geq (2^{r-1} - 1)(P_1 - P_2)$. By summing among all squares, we get

$$2h \geq \sum_{r=0}^{\ell-1} \sum_{s=0}^{2^{\ell-r-1}-1} M^p_{r,s} > (P_1 - P_2) \frac{h \log h}{4}$$

from which follows the claim. □

Corollary 1. *Consider an (S, cS, P_1, P_2)-asymmetric LSH when data and query domains are d-dimensional balls with unit radius and radius U respectively. Then, we must have $P_1 - P_2 = O\left(\frac{1}{\log \log_{1/c}(U/S)}\right)$ if $d = 1$ and $P_1 - P_2 = O\left(\frac{1}{\log(U/S(1-c))}\right)$ otherwise. This result implies that there cannot exist an asymmetric LSH when the query domain is unbounded.*

Proof. When $d = 1$, we use Theorem 3 with the following sequences of length $h = 1 + \lfloor \log_{1/c}(U/S) \rfloor$ of query and data points: $\mathcal{Q} = \{Uc^i, 0 \leq i \leq h-1\}$ and $\mathcal{P} = \{S/(Uc^j), 0 \leq j \leq h-1\}$.

For $d \geq 2$ we use Theorem 3 with the sets $\mathcal{P} = \{p_i, 0 \leq i < h\}$ and $\mathcal{Q} = \{q_j, 0 \leq j < h\}$ where $h = \Theta\left(U/(S(1-c))\right)$ and p_i and q_j are d-dimensional points, each one consisting of two values repeated $d/2$ times. That is:

$$q_i = \left(\frac{2\sqrt{U}}{d\sqrt{2}}(1 - \beta i), \ldots, \frac{2\sqrt{U}}{d\sqrt{2}}, \ldots \right), \tag{1}$$

$$p_j = \left(\frac{2\sqrt{2}}{d\sqrt{U}} \frac{S(1-c)}{\beta}, \ldots, \frac{2\sqrt{2}}{d\sqrt{U}} \left(S(1-c)j + S(\beta + c - 1)/\beta \right), \ldots \right) \tag{2}$$

where $\beta = S(1-c)/U$. We have that $q_i^T p_j \geq S$ if $j \geq i$, and $q_i^T p_j \leq cS$ otherwise. If d is odd, we add a dimension with value 0 to the points in Equations 1 and 2, and then repeat each value $(d-1)/2$ times. We observe that the above sequences may generate large negative scalar product. However, by applying a suitable rotation and compression □

The above upper bounds for $P_1 - P_2$ translate into lower bounds for the ρ factor, as soon as P_2 is fixed. To the best of our knowledge, this is the first lower bound on ρ that holds for asymmetric LSH. Indeed, previous results [16, 19] have investigated lower bounds for symmetric LSH and it is not clear if they can be extended to the asymmetric case. We believe that our bound on the gap can be significantly reduced: indeed, we conjecture that longer sequences can be obtained by considering many rotations of the sequences in Equations 1 and 2.

5 Upper bounds

This section is essentially a sequence of three observations. We first observe that by plugging the best known data structure for Approximate Near Neighbor (ANN) for ℓ_2 on a sphere in [7] into the reduction in [18], we get a data structure based on LSH for signed MIPS with $\rho = (1-S)/(1+(1-2c)S)$. This result is:

- stronger then the one from [18] in all regimes;
- stronger then the version of the one in [22] tailored to binary data in most of the parameter settings.

The latter conclusion is somewhat surprising, since the data structure we obtain works for non-binary vectors as well. We point out that in practice one may want to use a recent LSH family from [5] that—both in theory and in practice—is superior to the hyperplane LSH from [8] used in [18]. More details are provided in Appendix A.

Then, in Section 5.1, we circumvent the impossibility results in [18, 21] that show that symmetric LSH is not possible when the data and query domain coincide. Indeed, we show that such an LSH exists if we allow the bounds on collision probability to not hold for a few pairs of vectors. We accomplish this by using explicit incoherent matrices that are built using Reed-Solomon codes [17].

Finally, in Section 5.2, we show a data structure for c-approximate unsigned MIPS that uses linear sketches for ℓ_p norms from [3]. We obtain a $c \geq 1/n^{1/\kappa}$ approximation with $\tilde{O}\left(dn^{2-2/\kappa}\right)$ construction time and $\tilde{O}\left(dn^{1-2/\kappa}\right)$ query time, for any $\kappa \geq 2$. Although the resulting guarantees are not that strong, they are not far from the conditional lower bound in Theorem 2.

5.1 Symmetric LSH for Signed MIPS for almost all vectors

Neyshabur and Srebro [18] shows that an asymmetric view on LSH for signed MIPS is required. They indeed shows that a symmetric LSH does not exist when data and query domains coincide with a ball of same radius, while an asymmetric LSH does exist. On the other hand, when the data domain is a ball of given radius U and the query domain is the sphere of same radius, a symmetric LSH does exist. In this section, we complete this scenario by showing that a symmetric LSH does exist even when data and query space coincides if we allow LSH to fail on some pairs of vectors. That is, we use an asymmetric definition of LSH that does not provide any collision bound for two overlapping vectors (i.e., $\Pr[h(p) = h(q)]$ is unknown when $p = q$).

We first show how to reduce (c, S)-MIPS to the case where data and query vectors lie on a unit sphere. The reduction is deterministic and maintains inner products up to an additive error ε for all vectors x, y with $x \neq y$. We then plug in any data structure for ANN on the sphere, like [7]. This reduction treats data and query vectors equally, unlike the one from [18], and thus we are able to obtain a symmetric LSH.

Assume that all the coordinates of all the data and queries are encoded as s-bit numbers and that the data and query vectors are in the unit ball. The idea is the following. There are at

most $N = 2^{O(ds)}$ possible data vectors and queries. Let us imagine a collection of N unit vectors $v_1, \ldots, v_N$ such that for every $i \neq j$ one has $|v_i^T v_j| \leq \varepsilon$. Then, it is easy to check that a map of a vector p to $f(p) = [p, \sqrt{1 - \|p\|^2} \cdot v_p]$ maps a vector from a unit ball to a unit sphere and, moreover, for $p \neq q$ one has $|f(p)^T f(q) - p^T q| \leq \varepsilon$.

It is left to construct such a collection of vectors v_i. Moreover, our collection of vectors must be explicit in a strong sense: we should be able to compute v_u given a vector u (after interpreting is as a sd-bit string). It turns out that there is such a construction: in [17] it is shown how to build such vectors using Reed-Solomon codes. The resulting dimension is $O\left(\log N/\varepsilon^2\right) = O\left(sd/\varepsilon^2\right)$ [15, 17].

After performing such a reduction, we can apply any state-of-the-art data structure for ANN, like [7, 5], for the ℓ_2 norm on a sphere, with distance threshold $r^2 = 2(1 - S + \epsilon)$, approximation factor $c'^2 = (1 - cS/ - \epsilon)/r^2$. If ε is sufficiently small we get a ρ value close to the one in Equation 3. The final result is therefore a symmetric LSH for symmetric domains that does not provide any collision bound for all pairs (q, p) with $q = p$ since the guarantees on the inner product fail for these pairs. This LSH can used for solving (c, S)-signed MIPS as a traditional LSH [4], although it is required an initial step that verifies whether a query vector is in the input set and, if this is the case, returns the vector q itself if $q^T q \geq S$.

5.2 Unsigned MIPS via linear sketches

In this section we propose a data structure for c-undisgned MIPS, and hence for (c, S)-unsigned MIPS. Our data structure requires $\tilde{O}\left(dn^{2-2/\kappa}\right)$ construction time and $\tilde{O}\left(dn^{1-2/\kappa}\right)$ query time and provide a $c \geq 1/n^{1/\kappa}$ approximation with high probability, for any $\kappa \geq 2$. As shown in Theorem 2, we cannot hope to improve further the approximation factor since we cannot have polynomial construction time and subpolynomial query time when $1/2^{\sqrt{\log n}} \leq c \leq 1$, if the OVP conjecture is true.

First, suppose we are only interested in approximating the value of $\max_p |q^t p|$ and not to find the corresponding vector. Then, the problem is equivalent to estimating $\|Aq\|_\infty$, where A is an $n \times d$ matrix, whose rows are data vectors. This problem can be tackled using linear sketches (for an overview see [28, 6]). More specifically, we use the following result from [3]: for every $2 \leq \kappa \leq \infty$ there exists a distribution over $\widetilde{O}(n^{1-2/\kappa}) \times n$ matrices Π such that for every $x \in \mathbb{R}^n$ one has:

$$\Pr_{\Pi}\left[(1-c)\|x\|_\kappa \leq \|\Pi x\|_\infty \leq (1+c)\|x\|_\kappa\right] \geq 0.99$$

for a suitable constant $0 < c < 1$. Thus, to build a data structure for computing $\|Aq\|_\infty$, we sample a matrix Π according to the the aforementioned result in [3] and compute the $\tilde{O}\left(n^{1-2/\kappa}\right) \times d$ matrix $A_s = \Pi A$. Then, for every query q, we compute $\|A_s q\|_\infty$ in time $\tilde{O}\left(d \cdot n^{1-2/\kappa}\right)$, which is a $O\left(n^{1/\kappa}\right)$-approximation to $\|Aq\|_\infty$ with probability at least 0.99. Note that we can reduce the probability of error from 0.01 to $\delta > 0$ as usual, by building $O(\log(1/\delta))$ independent copies of the above data structure.

We now consider the recovery of the vector that almost maximizes $|p^t q|$. We recover the index of the desired vector bit by bit. That is, for every bit index $0 \leq i < \log n$, we consider every binary sequence b of length i and build a data structure for the dataset containing only the vectors in $\mathcal{P}$ for which the binary representations of their indexes have prefix b. Although the number of data structures is n, the total required space is still $\tilde{O}\left(dn^{1-2/\kappa}\right)$ since each vector appears in only $\log n$ data structures. The claim stated at the beginning follows.

References

[1] Amir Abboud, Virginia Vassilevska Williams, and Huacheng Yu. Matching triangles and basing hardness on an extremely popular conjecture. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 41–50, 2015.

[2] A. Andoni, D. Croitoru, and M. Patrascu. Hardness of nearest neighbor under l-infinity. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on*, pages 424–433, Oct 2008.

[3] Alexandr Andoni. High frequency moments via max-stability. Unpublished manuscript, 2012.

[4] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

[5] Alexandr Andoni, Piotr Indyk, Michael Kapralov, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal LSH for angular distance. Unpublished manuscript, 2015.

[6] Alexandr Andoni, Robert Krauthgamer, and Ilya P. Razenshteyn. Sketching and embedding are equivalent for norms. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 479–488, 2015.

[7] Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 793–801, 2015.

[8] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, 2002.

[9] Thomas Dean, Mark Ruzon, Mark Segal, Jonathon Shlens, Sudheendra Vijayanarasimhan, and Jay Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2013.

[10] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[11] Piotr Indyk. On approximate nearest neighbors under l norm. *Journal of Computer and System Sciences*, 63(4):627 – 638, 2001.

[12] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural svms. *Mach. Learn.*, 77(1):27–59, 2009.

[13] Noam Koenigstein, Parikshit Ram, and Yuval Shavitt. Efficient retrieval of recommendations in a matrix factorization framework. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 535–544, 2012.

[14] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[15] Kasper Green Larsen and Jelani Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. *CoRR*, abs/1411.2404, 2014.

[16] Rajeev Motwani, Assaf Naor, and Rina Panigrahi. Lower bounds on locality sensitive hashing. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, pages 154–157, 2006.

[17] Jelani Nelson, Huy L. Nguyen, and David P. Woodruff. On deterministic sketching and streaming for sparse recovery and norm estimation. *Linear Algebra and its Applications*, 441(0):152 – 167, 2014. Special Issue on Sparse Approximate Solution of Linear Systems.

[18] Behnam Neyshabur and Nathan Srebro. On symmetric and asymmetric lshs for inner product search. In *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, 2015.

[19] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Trans. Comput. Theory*, 6(1):5:1–5:13, 2014.

[20] Parikshit Ram and Alexander G. Gray. Maximum inner-product search using cone trees. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 931–939, 2012.

[21] Anshumali Shrivastava and Ping Li. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Procs. of 27th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2321–2329, 2014.

[22] Anshumali Shrivastava and Ping Li. Asymmetric minwise hashing for indexing binary inner products and set containment. In *Proceedings of the 24th International Conference on World Wide Web*, pages 981–991. International World Wide Web Conferences Steering Committee, 2015.

[23] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336. MIT Press, 2005.

[24] Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *Proceedings 18th Annual Conference on Learning Theory, COLT*, volume 3559 of *LNCS*, pages 545–560, 2005.

[25] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *J. ACM*, 62(2):13:1–13:45, 2015.

[26] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, pages 194–205, 1998.

[27] Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2):357–365, 2005.

[28] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10:1–157, 2014.

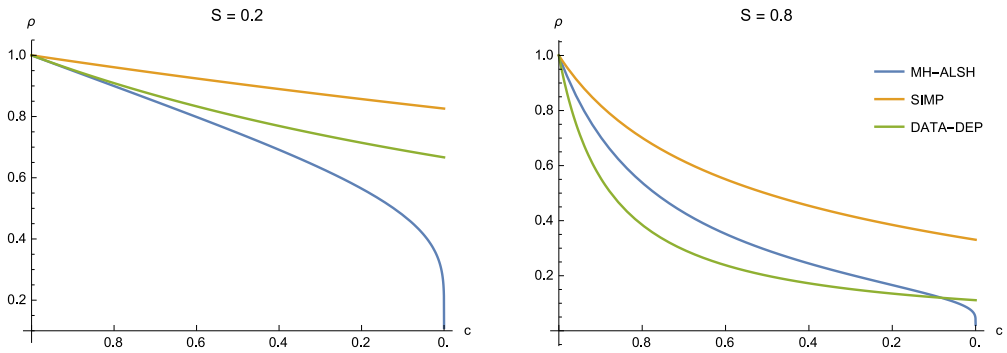

Figure 2: Our ρ value (DATA-DEP) compared to that of [18] (SIMP) and the binary data only of [22] (MH-ALSH).

A Appendix: Asymmetric LSH for Signed MIPS

In this section we plug the best known data structure for ANN for the Euclidean distance from [7] into the reduction from [18] to get a data structure for (c, S)-signed MIPS. As usual we assume the data and query domains to be d-dimensional balls with respective radius 1 and U. Vectors are embedded to a $(d+2)$-dimension unit sphere using the asymmetric map in [18]: a data vector p is mapped in $[p, \sqrt{||p||^2}, 0]$, while a query q in $[q/U, 0, \sqrt{||q||^2/U}]$. This transformation does not change inner products and then (c, S)-signed MIPS can be seen as an instance of ANN in ℓ_2 with distance thresholds $r = \sqrt{2(1-S)}$ and approximation $c' = \sqrt{(1-cS)/(1-S)}$. The latter can be solved in space $O(n^{1+\rho} + dn)$ and query time $O(n^{\rho})$ using a data structure from [7], where

$$\rho = \frac{1}{2c'^2 - 1} = \frac{1-S}{1+(1-2c)S}. \tag{3}$$

The data structure in [7] is a data-dependent LSH, that is an LSH where the hash family depends on the input set. Therefore, the final data structure for (c, S)-signed MIPS is an asymmetric data-dependent LSH.

In Figure 2, we plot the ρ value of three asymmetric LSHs: the one proposed here, the one from [18], and the one from [22]. The latter works only for binary vectors. We point out that our bound is always stronger than the one from [18] and sometimes stronger than the one from [22], despite the latter is tailored for binary vectors.