Subsets and Supermajorities: Unifying Hashing-based Set Similarity Search

Thomas Dybdahl Ahle
IT University of Copenhagen, BARC

April 8, 2019

Abstract

We consider the problem of designing Locality Sensitive Filters (LSF) for set overlaps, also known as maximum inner product search on binary data. We give a simple data structure that generalizes and outperforms previous algorithms such as MinHash [J. Discrete Algorithms 1998], SimHash [STOC 2002], Spherical LSF [SODA 2017] and Chosen Path [STOC 2017]; and we show matching lower bounds using hypercontractive inequalities for a wide range of parameters. This answers the main open question in Christiani and Pagh [STOC 2017] on unifying the landscape of Locality Sensitive (non-data-dependent) set similarity search.

While the previous algorithms consider different similarity measures, they are comparable when the weights of the sets are known. [1] We thus define the generalized problem of (w_q, w_u, w_1, w_2)-Gap Set Similarity Search, GapSS, on a universe U, where $1 \geq w_q, w_u \geq w_1 > w_2 \geq 0$; to be constructing a data structure with n sets of weight $w_u|U|$ such that given a query, $q \subseteq U$, with weight $|q| = w_q|U|$ we can efficiently find y in the dataset such that $|q \cap y| \geq w_1|U|$, given the intersection with all other sets is smaller than $w_2|U|$.

The specific case $w_1 = w_q$ corresponds to a "subset query" (also known as partial match) studied at least since Ronald L. Rivest's PhD thesis. Our data structure gives a complete space/time trade-off, as an example for $w_q = 0.02, w_u = 0.1, w_1 = 0.02, w_2 = 0.002$ we give the first data structures for each of the following:

- Space $n^{1.283}$ and query time $n^{0.283}$, (MinHash has space $n^{1.394}$ and query time $n^{0.394}$)
- Space $n^{1.599}$ and query time $n^{o(1)}$, (Spherical LSF has space $n^{4.44}$ at this query time)
- Space $n^{1+o(1)}$ and query time $n^{0.808}$ (Spherical has query time $n^{0.816}$ at this space usage).

We give a number of lower bounds, showing that our ρ_u, ρ_q trade-off is optimal for data structures in the Locality Sensitive framework when $w_q = w_u$ and either $w_2 = w_q^2$ or w_q is small. The bounds are based on p-biased boolean analysis, generalizing previous methods by O'Donnell et al. [TOCT 2014], Motwani et al. [JDM 2007] and others. We finally conjecture a new hypercontractive inequality, which if true, would show optimality also in the case $w_q \neq w_u$, and imply many new inequalities for boolean functions.

Contents

[1] Building separate data structures for sets with different norms, "norm ranging", was shown in [Yan et. al, NeurIPS 2018] to also be the fastest in practice when solving the maximum inner product search problem.

1 Introduction

Sparse boolean vectors arises from the classical representation of documents as "bags of words", where non-zero vector entries correspond to occurrences of words (or shingles). Another example is one-hot encoding of categorical data, such as which movies a user has watched. See e.g. [30] for a recent survey of applications.

Data structures for such data has been constructed for queries such as Superset / Subset / Containment, Partial Match, Jaccard similarity and maximum inner product search (MIPS). Early work goes back to Ronald Rivest's thesis [43] and many later papers have tackled the issue [15, 22]. Unfortunately these problems are equivalent to the Orthogonal Vectors problem [17], which means that we can't do much better than a brute force search through the database.

Hence recent research has focused on approximate versions of the problem, with MinHash (a.k.a. min-wise hashing) by Broder et al.[13, 12] being a landmark result. These problems are usually defined over some "simmilarity measure" like Jaccard similarity or Inner Product, with Chosen Path for Braun Blanquet similarity [21] being a recent break through. It has been observed however, that knowing the size of the sets in the database and queries makes all of these equivalent [21], including more than 76 binary similarity (and distance) measures defined in the survey [19]. This method, sometimes known as "norm ranging" is also practical, giving state of the art results at NeurIPS 2018 [48].

We thus define the Gap Similarity Search problem, as the approximate set similarity search problem that is aware of the set weights. Recall the definition from the abstract:

Definition 1. *The (w_q, w_u, w_1, w_2)-GapSS problem is to, pre-process n sets $Y \subseteq \binom{U}{w_u|U|}$ such that given a query $q \in \binom{U}{w_q|U|}$ we can efficiently return $y' \in Y$ with $|y' \cap q| > w_2|U|$ or determine that there is no $y \in Y$ with $|y \cap q| \geq w_1|U|$.*

Here U is some universe set, which we can assume to be larger than $\omega(\log n)$ by duplication if necessary. Note that GapSS includes approximate subset/superset queries by setting $w_1 = w_u$ or $w_1 = w_q$. The classical setting of a planted similar point on a background of random data [43], is included with $w_2 = w_q w_u$.

MinHash, Chosen Path and the extremely studied Spherical LSH [7] all solve the GapSS problem, with different algorithms being more efficient for different ranges of parameters. While the sketching problem for sets have been studied extensively, with faster MinHash algorithms such as [23], the search problem is less well understood. In [21] it was left as an the main open problem to unify the above methods, ideally finding the optimal LSH algorithm for set data. That is the problem we tackle in this paper.

Approach The proposed data-structure is a simple "list-of-points" data structure: For some constants $t_q, t_u \in [0, 1]$, we sample m sets $S_i \subseteq U$ independently and with replacement.

We make m lists and each set y from the database is stored in list $i \in [m]$ if the t_u-majority of S_i is in y. In other words, if $F_u^{(i)}(y) = 1$ where

$$F_u^{(i)}(y) = \begin{cases} [|y \cap S_i| \geq t_u |S_i|] & \text{if } t_u \geq w_u, \\ [|y \cap S_i| \leq t_u |S_i|] & \text{if } t_u < w_u, \end{cases} \quad \text{and} \quad F_q^{(i)}(q) = \begin{cases} [|q \cap S_i| \geq t_q |S_i|] & \text{if } t_q \geq w_q, \\ [|q \cap S_i| \leq t_q |S_i|] & \text{if } t_q < w_q, \end{cases} \tag{1}$$

When performing a query, q, we search each list i such that a t_q-majority of S_i is in q ($F_q^{(i)}(q) = 1$) and compare q to each of the ys in those lists, returning the first with $\langle q, y \rangle \geq w_2$. Since t_u (resp. t_q) is usually greater than the expectation of $|y \cap S_i|/|S_i|$ (w_u, resp. w_q) we call these boolean functions supermajorities, taken from social choice theory — "a qualified majority must vote in favour". [2]

While the above algorithm is a simple enough to be described in (roughly) a paragraph, the resulting bounds are complicated, and it is not obvious at first that they would be optimal. Perhaps this is why the scheme hasn't (to our knowledge) been described earlier in the literature. We do however show a number of lower bounds proving that given the right choices of t_u and t_q the scheme is indeed optimal over all choices of functions F_u and F_q for a large range of parameters w_q, w_u, w_1 and w_2. We conjecture that it is optimal over the entire space. For this reason the relative complication is inherent, and researchers as well as practitioners should not shy away from using supermajorities more widely.

A limitation of our result is the assumption that w_q and w_u be constants $\in [0, 1]$. It is common in the literature [23, 21] to assume that the similarities (s_1, s_2) (e.g. Jaccard) are constants, but usually arbitrarily smalls sets are allowed. We believe this is mainly an artefact of the analysis, and something it would be interesting future work to get rid of. In the meantime it also means that we can always hash down to a universe size of $\approx w_2^{-1} \log n$, which removes the need for a factor $|U|$ in the query time.

Intuitively our approach is similar to the Chosen Path algorithm, which fits in the above framework by taking $F_q^{(i)}(q) = [S_i \subseteq q]$ and $F_u^{(i)}(y) = [S_i \subseteq y]$. If $|q|$ is not equal to $|u|$ however (lets say $w_q > w_u$), the queries will have to look in many more lists than each data point is stored in, which gives a non-balanced space/time trade-off. Chosen Path handles this by a certain symmetrization technique (which we will study later), but perhaps the most natural approach is simply to slack the requirement of F_u, including also some lists where S_i is not completely contained in y.

[2] The Chosen Path filters of [21] are similar, but use the 'Consensus' function or 'ALL-function' for both F_u and F_q. The spherical LSF in [7] uses linear threshold functions, but for boolean data it can also use simple majorities (or $1 + o(1)$ fraction majorities.)

In our results we include a comprehensive comparison to a number of other LSH based approaches, which in addition to unifying the space of algorithms also gives a lot more intuition for why supermajorities are the right space partition for sparse boolean data.

1.1 Related Work

Work on Set Similarity Search has focused on a number of seemingly disparate problems: (1) Super-/Subset queries (2) Partial Match, (3) Jaccard/Braun Blanquet/Cosine similarity queries, and (4) maximum inner product search (MIPS).

The problems all have all traditionally been studied in their exact form:

Super-/Subset queries Pre-process a database D of n points in $\{0,1\}^d$ such that, for all query of the form $q \in \{0,1\}^d$, either report a point $x \in D$ such that $x \subseteq q$ (resp. $q \subseteq x$) or report that no such x exists.

Partial Match Pre-process a database D of n points in $\{0,1\}^d$ such that, for all query of the form $q \in \{0,1,*\}^d$, either report a point $x \in D$ matching all non-$*$ characters in q or report that no such x exists.

Similarity Search Given a similarity measure $S : \{0,1\}^d \times \{0,1\}^d \to [0,1]$, pre-process a database D of n points in $\{0,1\}^d$ such that, for all query of the form $q \in \{0,1\}^d$ return the point $x \in D$ maximizing $S(q,x)$.

Maximum Inner Product Search Same as Similarity Search, but $S(x,y) = \langle x,y \rangle$.

These problems are all part of an equivalence class of hard problems, known as Orthogonal Vectors [17]. This means that we don't expect the existence of polynomial space data structures that can solve either of these problems faster than a linear scan through the entire database. See also [3, 1, 44].

For this reason people have studied approximate versions of each problem. While the exact definition of the approximation differs in the literature, once we fix the weight of the input vectors, they all become essentially equal to GapSS as defined in this paper. This allows us to compare the best algorithms from each category against each other, as well as against our suggested Supermajorities algorithm. It should be noted that the hardness results mentioned above also holds for approximate variations, so the gap will have to be sufficiently large for any approach to work.

Partial Match The problem is equivalent to the subset query problem by the following well known reductions: (PM $\to$ SQ) Replace each $x \in D$ by the set $\{(i, p_i) : i \in [d]\}$. Then replace each query q by $\{(i, q_i) : q_i = *\}$. (SQ $\to$ PM) Keep the sets in the database as vectors and replace in each query each 0 by an $*$.

The classic approach, studied by Rivest [43], is to split up database strings like *supermajority* and file them under s, u, p etc. Then when given query like *set* we take the intersection of the lists s, e and t. Sometimes this can be done faster than brute force searching each list. He also considered the space heavy solution of storing all subsets, and showed that that when $d \leq 2 \log n$, the trivial space bound of 2^d can be somewhat improved. Rivest finally studied approaches based on tries and in particular the case where most of the database was random strings. The later case is in some ways similar to the LSH based methods we will describe below.

Indyk, Charikar and Panigrahi [15] also studied the exact version of the problem, and gave algorithms with

1. $n2^{(O(d \log^2 d \sqrt{c/\log n})}$ space and $O(n/2^c)$ time.

2. nd^c space and $O(dn/c)$ query time.

for any $c \in [n]$. Their approach was a mix between the shingling method of Rivest, building a look-up table of size $\approx 2^{\Omega(d)}$, and a brute force search. These bounds manage to be non-trivial for $d = \omega(\log n)$, however only slightly, since otherwise they would break the mentioned OVC lower bounds.

There has also been a large number of practical papers written about Partial Match / Subset queries or the equivalent batch problem of subset joins [42, 34, 27, 2, 25]. Most of these use similar methods to the above, but save time and space in various places by using bloom filters and sketches such as MinHash [13] and HyperLogLog [26].

Maximum Inner Product For exact algorithms, most work has been done in the batch version (n data points, n queries). Here Alman et al. [4] gave an $n^{2-1/\tilde{O}(\sqrt{k})}$ algorithm, when $d = k \log n$.

An approximative version can be defined as: Given $c > 1$, pre-process a database D of n points in $\{0,1\}^d$ such that, for all query of the form $q \in \{0,1\}^d$ return a point $x \in D$ such that $\langle q, x\rangle \geq \frac{1}{c} \max_{x' \in D} \langle q, x'\rangle$. Here [3] gives a data-structure with query time $\approx \tilde{O}(n/c^2)$, and [17] solves the batch problem in time $n^{2-1/O(\log c)}$ (both when d is $n^{o(1)}$.)

There are a large number of practical papers on this problem as well. Many are based on the Locality Sensitive Hashing framework (discussed below) and have names such as SIMPLE-LSH [36] and L2-ALSH [45]. The main problem for these algorithms is usually that no hash family of functions $h : \{0,1\}^d \times \{0,1\}^d \to [m]$ such that $\Pr[h(q) = h(x)] = \langle q, x\rangle/d$ [3] and various embeddings and asymmetries are suggested as solutions.

The state of the art is a paper from NeurIPS 2018 [48] which suggests partitioning data by the vector norm, such that the inner product can be more easily estimated by LSH-able similarities such as Jaccard. This is curiously very similar to what we suggest in this paper.

We will not discuss these approaches further since, for GapSS, they are all dominated by the three LSH approaches we study next.

Similarity Search The problem is usually studied as an approximate problem: Given a similarity measure $S : \{0,1\}^d \times \{0,1\}^d \to [0,1]$ and $s_1 > s_2 \in [0,1]$, pre-process a database D of n points in $\{0,1\}^d$ such that for queries $q \in \{0,1\}^d$ we return a point $x \in D$ with $S(q,x) \geq s_2$ given there is $x' \in D$ with $S(q,x') \geq s_1$.

This naturally generalizes MIPS as defined above. The formulation allows the use of Indyk and Motwani's LSH framework [29]. Here we define a family, $\mathcal{H}$, of functions $h : \{0,1\}^d \times \{0,1\}^d \to [m]$ such that

1. $\Pr_{h \sim \mathcal{H}}[h(q) = h(x)] \geq p_1$ when $S(q,x) \geq s_1$, and

2. $\Pr_{h \sim \mathcal{H}}[h(q) = h(x)] < p_2$ when $S(q,x) < s_2$.

The constructions in [29, 28] then give an algorithm for the approximate similarity search problem with space $n^{1+\rho} + dn$ and query time dominated by n^ρ evaluations of h, where $\rho = \log p_1 / \log p_2$.

If $\mathcal{H}$ exists such that $\Pr_{h \sim \mathcal{H}}[h(q) = h(x)] = S(q,x)$ is achievable (see [18] for a study of when this is the case) then such a family is an obvious choice. An example of this is Broder's MinHash algorithm, which has $\Pr_{h \sim \mathcal{H}}[h(q) = h(x)] = |q \cap x|/|q \cup x|$ where $S(q,x) = |q \cap x|/|q \cup x|$ is the Jaccard similarity.

Choosing $\mathcal{H}$ like this is however not always optimal, as Christiani and Pagh [21] shows by constructing a data structure with $\rho = \frac{\log 2s_1/(1+s_1)}{\log 2s_2/(1+s_2)} < \frac{\log s_1}{\log s_2}$ when the size of sets is equal, $|q| = |x|$. In general they get $\rho = \frac{\log b_1}{\log b_2}$ where $b_1 > b_2 \in [0, 1]$ are Blanquet Similarities $B(q, x) = |q \cap x| / \max\{|q|, |x|\}$.

The most studied variant is LSH on the sphere. Here, given $\alpha > \beta \in [-1, 1]$, we pre-process a database D of n points in S^{d-1} and for a query $q \in S^{d-1}$ return $x' \in D$ with $\langle q, x' \rangle \geq \beta$ given the promise that there is $x \in D$ with $\langle q, x \rangle \geq \alpha$. In [5] they show how to get $\rho_{\text{sp}} = \frac{1-\alpha}{1+\alpha}\frac{1+\beta}{1-\beta}$.[3]

While it is clear that both MinHash and Chosen Path can solve GapSS when w_q and w_u is known in advance, using spherical LSH requires that we embed the binary vectors onto the sphere. Multiple ways come to mind, such as mapping $0 \mapsto -1/\sqrt{d}, 1 \mapsto 1/\sqrt{d}$ or $0 \mapsto 0$, $1 \mapsto 1/\sqrt{w_q d}$ (for queries, resp. $1/\sqrt{w_u d}$ for data points). Depending on how we do it, the algorithm of [5] will naturally return different results, however given knowledge of w_q and w_u there is an optimal embedding[4], as we will show in this paper. This gives $\alpha = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$ and $\beta = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$ which is better than the two previous methods when w_q and w_u are not too small.

Two other classic methods are Bit Sampling [29] and SimHash (Hyperplane rounding) [16], which give $\rho_{\text{bs}} = \frac{\log(1-w_q-w_u+2w_1)}{\log(1-w_q-w_u+2w_2)}$ and $\rho_{\text{hp}} = \frac{\log(1-\arccos(\alpha)/\pi)}{\log(1-\arccos(\beta)/\pi)}$ respectively. (SimHash also works on the sphere, but has the same optimal embedding as spherical LSH.) These ρ-values however turn out to always be larger than ρ_{sp}, so we won't study them as much.

While Chosen Path and Spherical LSH both have proofs of optimality [21, 8, 39, 35] in the LSH model, these optimality proofs consider specific ranges, like when w_q, w_u or w_1 goes to zero. Hence they are not necessarily optimal when used in all the ranges of parameters in which GapSS is interesting. In fact they each have regions of optimality, as was observed in [21] who proposed as an open problem to find an LSF scheme that unified all of the above. This is what we do in this paper, as well as showing matching lower bounds in a wider range of parameters.

Trade-offs and Data Dependency The above algorithms, based on the LSH framework, all had space usage roughly $n^{1+\rho}$ and query time n^{ρ} for the same constant ρ. This is known as the "balanced regime" or the "LSH regime". Time/space trade-offs are important, since $n^{1+\rho}$ can sometimes be too much space, even for relatively small ρ. Early work on this was done by Panigrahy [40] and Kapralov [31] who gave smooth trade-offs ranging from space $n^{1+o(1)}$ to query time $n^{o(1)}$. A breakthrough was the use of LSF, which allowed time/space trade-offs with sublinear query time even for near linear space and small approximation [33, 20, 8].

Prior to this article, the only way to achieve trade-offs for set data was to embed it into the above spherical algorithms. In this paper we show that it is often possible to do much better, and in some cases get query time less than that of balanced spherical LSH, even with near-linear space.

Arguably the largest break-through in LSH based data-structures was the introduction of data-dependent LSH[6, 9, 10]. It was shown how to reduce the general case of α, β similarity search as described above, to the case $\beta = 0$ (and $\alpha \mapsto \frac{\alpha-\beta}{1-\beta}$), in which many LSH schemes work better. Using those data structures on GapSS with $w_2 > w_q w_u$ will often yield better performance than the algorithms described in this paper. However, since the data-dependent methods are equivalent to Spherical LSH for $w_2 = w_u w_q$, we always dominate this case, and it is an exciting open problem to create similar reductions directly for set data, possibly using the space partitioning proposed in this algorithm as a building block.

[3] For $\beta \to 1$ this approaches $\log \alpha / \log \beta$, which would be like an LSH-able family for inner product on the sphere, but unfortunately this is not achievable with LSH. For the batch problem it was shown possible in [32].

[4] optimal for embeddings on the form $0 \mapsto a, 1 \mapsto b$.

1.2 Results

We split our results in upper bounds, lower bounds and comparisons with other approaches. We provide fig. 1 to guide the intuition, as well as some corollaries with results for specific parameters.

1.2.1 Upper bounds

We show the existence of a data-structure for (w_q, w_u, w_1, w_2)-GapSS with space usage $n^{1+\rho_u+o(1)} + O(n\, w_u|U|)$ and query time $n^{\rho_q+o(1)}$ for some ρ_q and ρ_u which are functions of w_q, w_u, w_1, w_2 as well as $t_u, t_q \in [0,1]$ as described in the introduction.

Our main upper bound, theorem 4, is a bit to intricate to be stated yet, but we note the following corollaries. Each of them follows easily by inserting the giving values into theorem 4.

Corollary 1 (Near balanced)**.** *For any choice of constants $w_q, w_u \geq w_1 \geq w_2 \geq 0$ we can solve the (w_q, w_u, w_1, w_2)-GapSS problem over universe U with query time $n^{\rho_q+o(1)}$ and space usage $n^{\rho_u+o(1)} + O(n\, w_u|U|)$,*

$$\rho_q = \frac{H(w_1) - D(w_u, 1-w_q)}{H(w_2) - D(w_u, 1-w_q)}, \quad \rho_u = \frac{H(w_1) - D(w_q, 1-w_u)}{H(w_2) - D(w_u, 1-w_q)}. \tag{2}$$

where $H(w_i) = (1 - w_q - w_u)\log(\frac{1-w_q-w_u+w_i}{w_i})$, and $D(a,b) = a\log\frac{a}{b} + (1-a)\log\frac{1-a}{1-b}$ is the Kullback–Leibler divergence. (We define $D(0,b) = \log 1/(1-b)$, $D(1,b) = \log 1/b$ as is standard.)

Proof. This follows by setting $t_q = 1 - w_u, t_u = 1 - w_q$ in theorem 4. □

If $w_q = w_u = w$ this simplifies to:

$$\rho_q = \rho_u = \log\left(\frac{w}{w_1}\frac{1-2w+w_1}{1-w}\right) \Big/ \log\left(\frac{w}{w_2}\frac{1-2w+w_2}{1-w}\right) = \frac{\log S(q,x)}{\log S(q,x')}, \tag{3}$$

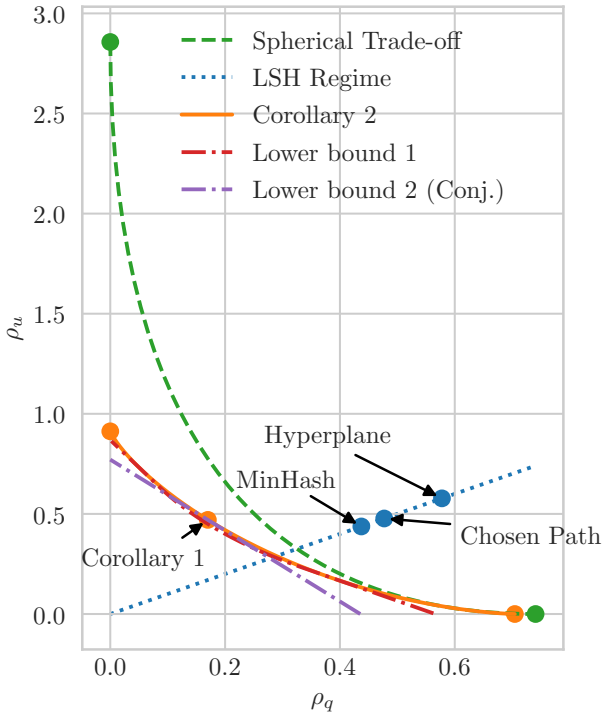
where $S(q,x) = \frac{\langle q,x\rangle}{\|q\|_2\|x\|_2} \Big/ \frac{\langle \bar{q},\bar{x}\rangle}{\|\bar{q}\|_2\|\bar{x}\|_2}$ is the cosine similarity divided by the cosine similarity on the complement sets.

In the case of small sets, $w, w_1, w_2 \to 0$, equation (3) reduces to $\log(\frac{w}{w_1}) \big/ \log(\frac{w}{w_2})$, which is the ρ-value of Chosen Path on balanced sets, and which was shown in [21] to be optimal in this range. In the next section we generalize their lower bound to hold generally for all values w, w_1, w_2 though still only asymptotically sharp for small sets.

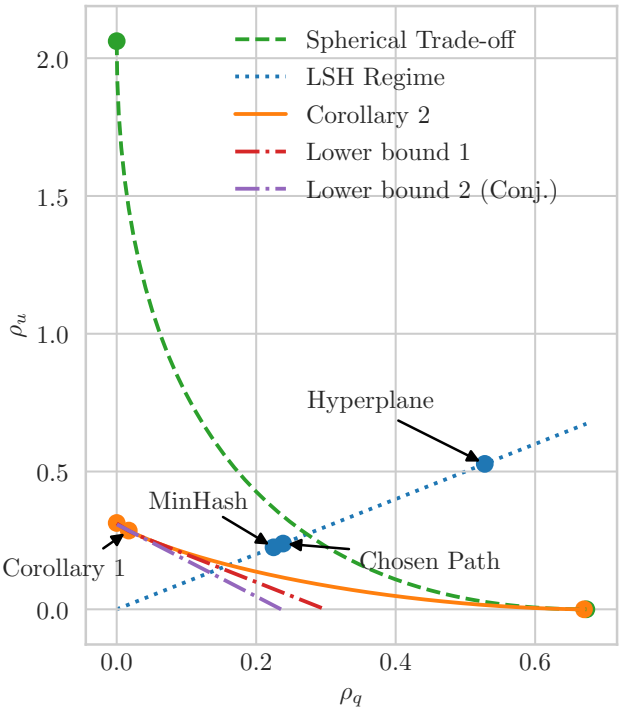
Corollary 1 is special, because the optimal values of t_u and t_q depend only on w_u and w_q, while in general they will also depend on w_1 and w_2. We will show (3) is optimal for the case $w_2 = w_q^2$ for all choices of w_q and w_1. Conditioned on a conjectured hypercontractive inequality we will even show eq. (2) is optimal for all choices of w_q, w_u and w_1 at the particular trade-off.

Corollary 2 (Subset/superset queries)**.** *If $w_1 = \min\{w_u, w_q\}$, $w_2 = w_u w_q$ we can take*

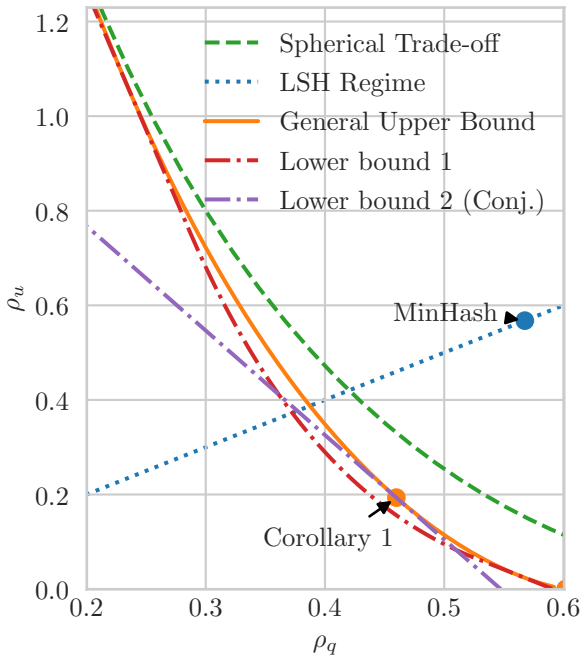
$$t_q = -\frac{w_u(1-w_u)w_q(1-w_q)}{w_q - w_u}\alpha + \frac{w_q(1-w_u)}{w_q - w_u}$$
$$\text{and } t_u = \frac{1}{w_q - w_u}\alpha^{-1} - \frac{w_u(1-w_q)}{w_q - w_u}$$
$$\text{for any } \alpha \in [\min\{w_u, w_q\} - w_q w_u, \max\{w_u, w_q\} - w_q w_u]$$



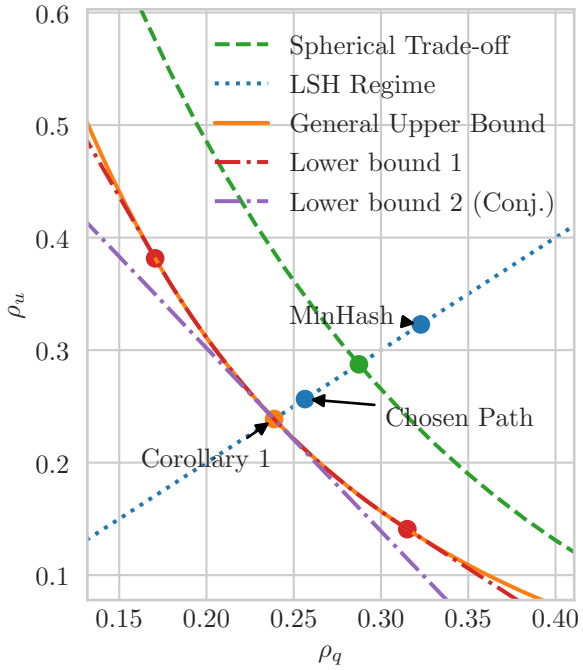
(a) Superset queries with $w_q = 0.1$, $w_u = 0.3$, $w_1 = 0.1$ and $w_2 = w_q w_u$. As the sets are relatively large, Spherical LSH beats MinHash and Chosen Path.



(b) Superset queries at smaller scale with $w_q = 0.01$, $w_u = 0.03$, $w_1 = 0.01$ and $w_2 = w_q w_u$.



(c) Zoom in on $w_q = 0.4$, $w_u = 0.1$ $w_1 = 0.1$ $w_2 = w_q w_u$. We see that theorem 1 is not tight when $w_q \neq w_u$. However the conjectured Lower bound 2 matches the upper bound exactly.



(d) Zoom in on a "regular" example $w_q = w_u = 0.16$ with $w_1 = 0.1$, $w_2 = w_q w_u$. The red dots show the segment within where r, s can be chosen optimally without exceeding 1/2.

Figure 1: Examples of ρ-values obtained from theorem 4 for various parameter settings, compared to that of other algorithms, and to the lower bounds theorem 1 and conjecture 1. See the appendix fig. 3 for more comparisons.

to get data structures with

$$\rho_q = \frac{t_q \log \frac{1-t_u}{1-w_u} - t_u \log \frac{1-t_q}{1-w_q}}{D(t_u, w_u)} \quad \rho_u = \frac{(1-t_u)\log \frac{t_q}{w_q} - (1-t_q)\log \frac{t_u}{w_u}}{D(t_u, w_u)} \quad \text{if } w_1 = w_u,$$

$$\rho_q = \frac{-(1-t_u)\log \frac{t_q}{w_q} + (1-t_q)\log \frac{t_u}{w_u}}{D(t_u, w_u)} \quad \rho_u = \frac{-t_q \log \frac{1-t_u}{1-w_u} + t_u \log \frac{1-t_q}{1-w_q}}{D(t_u, w_u)} \quad \text{if } w_1 = w_q.$$

As it turns out, the optimal values for t_u and t_q, when doing superset/subset queries lie on the hyperbola $t_u w_q(1-w_u) - t_q(1-w_q)w_u = t_u t_q(w_q - w_u)$ with $t_u, t_q = 0$ in one end and $t_q, t_u = 1$ in the other. This means that Chosen Path (without symmetrization) is indeed equivalent to our approach for these problems, when we are interested in near linear space or near constant query time.

1.2.2 Lower bounds

For the lower bounds we will assume $|U| = \omega(\log n)$ (like we reduce to in the upper bounds). This follows all previous LSH-lower bounds, and it is indeed known from [11] that this is necessary since it is possible to do better in the "medium dimension regime" when $|U| = O(\log n)$. In that regime classical data structures such as KD-trees are also competitive, see e.g. [14].

The lower bounds are all in the restricted model of Locality Sensitive Filters (definition 2) or the "list of points" model (definition 4). In other words, the data structure is presumed to be as described in the introduction, and the only part we are allowed to change is how the F functions from equation (1) are defined. There is a stronger kind of LSH, in which the filter distribution is allowed to depend on the dataset [6, 9, 7] which does better than data-independent LSF in general. However most of our lower bounds are for the "random case" $w_2 = w_q w_u$ in which no difference is known between the two approaches.

The first two lower bounds follow from hypercontractive inequalities, similar to the original LSH lower bound by Motwani et al. [35], though we need to use the less well studied p-biased variation.[5] The idea to upper bound the expected inner product of two boolean functions when given only slightly correlated inputs. To handle the inner product between functions on "far points" we need that they are completely uncorrelated, $w_2 = w_q w_u$, so we may act as if the boolean functions were independent. Another approach is that of O'Donnell et al. [39] which gives an inequality connecting the probability of colliding with a close point to that of colliding with a far point. We show how this can be generalized to biased spaces as well.

Theorem 1 (Lower bound 1)**.** *Given $\alpha \geq 0$ and $0 \leq r, s \leq 1/2$, let $u_q = \log \frac{1-w_q}{w_q}$, $u_u = \log \frac{1-w_u}{w_u}$, $\sigma = \frac{\sinh(u_q(1-1/r))}{\sinh(u_q/r)}$, and $\upsilon = \frac{\sinh(u_u(1-1/s))}{\sinh(u_u/s)}$. If r and s are such that $\sqrt{\sigma\upsilon} = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$, then any list-of-points data structure must use space $n^{1+\rho_u}$ and have query time n^{ρ_q} where*

$$\rho_q \geq (1/r - 1)\alpha + 1/s \quad \text{or} \quad \rho_u \geq \alpha/r + 1/s - 1,$$

To get the most out of the lower bound, we will also want r and s such that

$$\frac{\sigma u_q \sinh(u_q)\sinh(u_u(1-1/s))\sinh(u_u/s)}{\upsilon u_u \sinh(u_u)\sinh(u_q(1-1/r))\sinh(u_q/r)} = \alpha,$$

[5] Plugging into the framework of Panigrahy et al. [41] this also gives a cell probe lower bound for 1 probe data structures.

however due to the limitation $r, s \geq 2$, this is not always possible.[6]

For $w_u = w_q$ we can take $\alpha = 1$ and $r = s$ to get $\rho_q, \rho_u \geq \log \frac{w_q}{w_1} \frac{1-2w_q+w_1}{1-w_q} \Big/ \log \frac{1-w_q}{w_q}$ which exactly matches corollary 1 and shows it is optimal for all w_1 and w_q when $w_u = w_q$, $w_2 = w_q^2$. When $w_q \neq w_u$ the bound is unfortunately not sharp, as can be seen in fig. 1.

Theorem 1 is based on the p-biased hypercontractive inequalities of Oleszkiewicz and Krzysztof [38]. Their inequality, while sharp, only handles the case of a single boolean function, and we have expanded it using Cauchy Schwartz to get our more general theorem. This approach, while good enough for sharp space/time trade-offs on the sphere, turns out to be insufficient for sets when $w_q \neq w_u$.

To overcome this, we conjecture a new two-function p-biased hypercontractive inequality for which we have much evidence (see the lower bounds section). This inequality implies the following lower bound:

Conjecture 1 (Lower bound 2)**.** *Let $r = \log \frac{(1-w_q)(1-w_u)}{w_q w_u} \Big/ \log \frac{1-w_q-w_u+w}{w}$ and $\alpha \geq 0$ then any list-of-points data structure must use space $n^{1+\rho_u}$ and have query time n^{ρ_q} where*

$$\rho_q \geq (\alpha+1)/r - \alpha \quad \text{or} \quad \rho_u \geq (\alpha+1)/r - 1.$$

Setting $\alpha = 1$ this immediately shows corollary 1 is tight for all w_q, w_u, w_1 and $w_2 = w_q w_u$. We believe it should be possible to extend this further to separate r and s values, as in theorem 1, which would show theorem 4 to be tight for all w_q, w_u, w_1 and the entire time/space trade-offs, but this is work for the future.

Our previous lower bounds have assumed $w_2 = w_q w_u$. To extend to general values of w_2 we show the following bound:

Theorem 2 (Lower bound 3)**.** *If $w_q = w_u$, any LSF data structure that uses the same functions for updates and queries ($F_u^{(i)} = F_q^{(i)}$) must have $\rho_u = \rho_q$ and use space $n^{1+\rho_u}$ and have query time n^{ρ_q} where*

$$\rho_q \geq \log\left(\frac{w_1 - w_q^2}{w_q(1-w_q)}\right) \Big/ \log\left(\frac{w_2 - w_q^2}{w_q(1-w_q)}\right).$$

Taking $w, w_1, w_2 \to 0$ recovers Pagh and Christiani's $\rho \geq \log(w_1/w) \big/ \log(w_2/w)$ bound for Braun Blanquet similarity [21].[7] For larger values of w_q, w_1, w_2 the bound is however not tight. Showing any lower bound that holds for $w_2 \neq w_u w_q$ and for large distances is an open problem in the LSH world.

1.2.3 Comparison to previous approaches

Since our lower bounds don't cover the entire range of parameters w_q, w_u, w_1, w_2 (no LSH lower bounds do), we need to compare our ρ values with those achieved by previous methods and show that we get lower values on the entire range.

We show two results towards this: (1) For Spherical LSH we show how to most optimally embed GapSS onto the sphere, and that our ρ values are at least as small as with Spherical LSH in this

[6] This is not just an artefact of the proof, since computations of theorem 1 with r, s outside the given range shows that the theorem as stated is indeed false in that case, as it goes above our upper bound. The limitation might be removed by using the more general p-biased inequalities by Wolff [47], but unfortunately those are for asymptotically small sets.

[7] Their bound also implicitly had the same function for queries and updates.

setting. (2) For MinHash we show a dominating family of Chosen Path like algorithms, which it is natural to conjecture is again dominated by supermajorities. The first result is quite interesting on its own right, since many previous algorithms for Maximum Inner Product Search consisted of various embeddings onto the sphere. The second result is also interesting in that it sheds more light on why MinHash is sometimes faster than Chosen Path, which is a question raised in [21]. The result shows that in fact on can change Chosen Path only slightly for it to consistently beat MinHash.

Lemma 1.1 (Best Binary Embedding)**.** *Let $g, h : \{0,1\}^d \to \mathbb{R}$ be functions on the form $g(x) = a_1 x + b_1$ and $h(y) = a_2 y + b_2$. Let $\rho(x, y, y') = f(\alpha(x, y))/f(\alpha(x, y'))$ where $\alpha(x, y) = \langle x, y\rangle / \|x\|\|y\|$ be such that*

$$f(z) \geq 0, \quad \tfrac{d}{dz}\left((\pm 1 - z)\tfrac{d}{dz}\log f(z)\right) \geq 0 \quad \text{and} \quad \tfrac{d^3}{dz^3}\log f(z) \leq 0$$

for all $z \in [-1, 1]$. Assume we know that $\|x\|_2^2 = w_q d$, $\|y\|_2^2 = w_u d$, $\langle x, y'\rangle = w_1 d$ and $\langle x, y\rangle = w_2 d$, then

$$\arg\min_{a_1,b_1,a_2,b_2} \rho(g(x), h(y), h(y')) = (1, -w_q, 1, -w_u).$$

See proof in section 3.2. Since α, as defined above, scales the vectors down by their norm to make sure they are on the sphere, the lemma indeed says that we should subtract the mean and divide by the standard deviation of our vectors before we use LSH. We show that Spherical LSH and Hyperplane LSH [16] (a.k.a. SimHash) satisfy this lemma, given their ρ values for distinguishing between inner products $\alpha > \beta$:

$$\rho_{\text{hp}} = \frac{\log(1 - \arccos(\alpha)/\pi)}{\log(1 - \arccos(\beta)/\pi)}, \quad \rho_{\text{sp}} = \frac{1-\alpha}{1+\alpha}\frac{1+\beta}{1-\beta}.$$

This implies we should take $\alpha = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$ and $\beta = \frac{w_2 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$.

See also fig. 1 where we have plotted theorem 4 against Chosen Path, MinHash, Spherical LSF and Hyperplane LSH.

Comparison to MinHash Consider the LSF family, $\mathcal{F}$, formed by one of the functions

$$F_0(x) = [s_0 \in x], F_1(x) \quad = [s_1 \in x \wedge s_0 \notin x], \ldots, F_i(x) = [s_i \in x \wedge s_0 \notin x \wedge \cdots \wedge s_{i-1} \notin x], \ldots$$

where $(s_i \in U)_{i\in\mathbb{N}}$ is a random sequence by sampling elements of U with replacement. Note that while the sequence is infinite, it the functions eventually all become 0 as we get a prefix including all of U, hence we can sample from $\mathcal{F}$ efficiently. Also note that then $h(x) = \min\{i \mid F_i(x) = 1\}$ is the usual MinHash function.

While MinHash is balanced, $\rho_u = \rho_q$, most of the F_i's are on their own not balanced if $w_q \neq w_u$. We can fix this by applying a symmetrization technique implicit in [21]. Using that we get

$$\rho_i = \log \frac{(1 - w_q - w_u + w_1)^i w_1}{\max\{(1-w_q)^i w_q,\, (1-w_u)^i w_u\}} \Big/ \log \frac{(1 - w_q - w_u + w_2)^i w_2}{\max\{(1-w_q)^i w_q,\, (1-w_u)^i w_u\}}$$

for the LSF data structure using only F_i. Note that $\rho_0 = \log \frac{w_1}{\max\{w_q, w_u\}} / \log \frac{w_2}{\max\{w_q, w_u\}}$ is exactly the same as ρ_{cp} achieved by Chosen Path. This makes sense, since it is exactly the Chosen Path function with the Chosen Path symmetrization technique.

We show that in section 3.3 that $\rho_{\text{mh}} = \log \frac{w_1}{w_q + w_u - w_1} / \log \frac{w_2}{w_q + w_u - w_2} \geq \min_{i\geq 0} \rho_i$. In fact we can restrict this to $i \in \{0, \infty, \log(w_q/w_u)/\log((1-w_q)/(1-w_u))\}$, where the first gives Chosen Path, the second gives Chosen Path on the complemented sets, and the last gives two concatenated Chosen Path's in a balanced trade-off where $(1-w_q)^i w_q = (1-w_u)^i w_u$.

2 Preliminaries

The Locality Sensitive Filter approach to similarity search is an extension by Becker et al. [11] to the Locality Sensitive Hashing framework by Indyk and Motwani [29]. We will use the following definition by Christiani [20], which we have slightly extended to support separate universes for query and data points:

Definition 2 (LSF)**.** *Let X and Y be some universes, let $S : X \times Y \to \mathbb{R}$ be a similarity function, and let $\mathcal{F}$ be a probability distribution over $\{(Q, U) \mid Q \subseteq X, U \subseteq Y\}$. We say that F is $(s_1, s_2, p_1, p_2, p_q, p_u)$-sensitive if for all points $x \in X, y \in Y$ and (Q, U) sampled randomly from $\mathcal{F}$ the following holds:*

1. *If $S(x, y) \geq s_1$ then $\Pr[x \in Q, y \in U] \geq p_1$.*
2. *If $S(x, y) \leq s_2$ then $\Pr[x \in Q, y \in U] \leq p_2$.*
3. *$\Pr[x \in Q] \leq p_q$ and $\Pr[x \in U] \leq p_u$.*

We refer to (Q, U) as a filter and to Q as the query filter and U as the update filter.

The main theorem from [20] which we will use for our upper bounds is (paraphrasing):

Theorem 3 (LSF theorem)**.** *Suppose we have access to a family of filters that is $(s_1, s_2, p_1, p_2, p_q, p_u)$-sensitive. Then we can construct a fully dynamic data structure that solves the (s_1, s_2)-similarity search problem with query time $dn^{\rho_q + o(1)}$, update time $dn^{\rho_u + o(1)}$, and space usage $dn + n^{1+\rho_u+o(1)}$ where $\rho_q = \log(p_q/p_1)/\log(p_q/p_2)$ and $\rho_u = \log(p_u/p_1)/\log(p_q/p_2)$.*

We must be able to sample, store, and evaluate filters from $\mathcal{F}$ in time $dn^{o(1)}$.

We will use definition 2 with $S(x, y) = |x \cap y|$. For given values of w_q and w_u, the (s_1, s_2)-similarity search problem then corresponds to the (w_u, w_q, w_1, w_2)-gap similarity search problem.

3 Upper bounds

To state our results we first need to define the following functions:

Definition 3 (Entropy Functions)**.** *The relative entropy function (or Kullback–Leibler divergence) is defined for $a, b \in [0, 1]$ by: $D(a, b) = a \log \frac{a}{b} + (1 - a) \log \frac{1-a}{1-b}$ and $D(0, b) = \log 1/(1 - b)$, $D(1, b) = \log 1/b$.*

For $x, y \in (0, 1)$, $t_1, t_2 \in (0, 1)$ and $b \in (0, \min(x, y))$ we define the following pair-relative entropy function:

$$\Lambda(x, y, b, t_1, t_2) = t_1\lambda_1 + t_2\lambda_2 - \log v \quad \textit{where}$$

$$\lambda_1 = \log\left(\frac{v(1-t_2) - (1-x-y+b)}{x-b}\right),\ \lambda_2 = \log\left(\frac{v(1-t_1) - (1-x-y+b)}{y-b}\right),\ v = \frac{v_1 + \sqrt{v_1^2 - v_2}}{2b(1-t_1)(1-t_2)},$$

and $v_1 = (1 - t_1 - t_2)(b - xy) + b(1 - x - y + b)$, $v_2 = 4(1 - t_1)(1 - t_2)(b - xy)b(1 - x - y + b)$.

At $t_1 = 1$, $t_2 = 1$ or $b = \min(x, y)$, Λ is defined as follows:

$$\Lambda(x, y, xy, t_1, t_2) = D(t_1, x) + D(t_2, y)$$

$$\Lambda(x, y, b, 1, t_2) = \log\frac{1}{x} + D(t_2, y) \tag{4}$$

$$\Lambda(x, y, y, t_1, t_2) = t_2 \log\frac{t_2}{y} + (1 - t_1)\log\frac{1 - t_1}{1 - x} + (t_1 - t_2)\log\frac{t_1 - t_2}{x - y}. \tag{5}$$

The cases $t_2 = 1$ and $b = x$ are symmetric to eq. (4) and eq. (5).

The goal of this section is to prove the following general upper bound:

Theorem 4 (General Upper Bound)**.** *For any choice of constants $w_q, w_u \geq w_1 \geq w_2 \geq 0$ and $1 \geq t_q, t_u \geq 0$ we can solve the (w_q, w_u, w_1, w_2)-GapSS problem over universe U with query time $n^{\rho_q + o(1)}$ and space usage $n^{\rho_u + o(1)} + O(n\, w_u |U|)$, where*

$$\rho_q = \frac{\Lambda(w_q, w_u, w_1, t_q, t_u) - D(t_q, w_q)}{\Lambda(w_q, w_u, w_2, t_q, t_u) - D(t_q, w_q)}, \quad \rho_u = \frac{\Lambda(w_q, w_u, w_1, t_q, t_u) - D(t_u, w_u)}{\Lambda(w_q, w_u, w_2, t_q, t_u) - D(t_q, w_q)}.$$

The theorem defines the entire space/time trade-off of fig. 1 by choices of t_q and t_u. By Lagrangian multipliers we can compute the optimal t_u for any t_q. An easy corollary is

Corollary 3 (Linear space / constant time)**.**

$$\begin{aligned} &\textit{If} \quad t_q\, w_u(1-w_u) + w_1 w_u = t_u\,(w_1 - w_q w_u) + w_q w_u && \textit{then} \quad \rho_u = 0. \\ &\textit{If} \quad t_u w_q(1-w_q) + w_1 w_q = t_q(w_1 - w_q w_u) + w_u w_q && \textit{then} \quad \rho_q = 0. \end{aligned}$$

Proof. Insert said values into theorem 4 to make the numerators 0. □

We will use the LSF theorem 3 as the basis for our upper bound with the filter family described in the introduction. Note that we can reduce the universe to $O(\epsilon^{-2} w_2^{-1} \log n)$ by sampling. By a union bound this preserves w_1, w_2, w_q and w_u within a factor $1 \pm \epsilon$. Taking $\epsilon = 1/\log n$ this is absorbed into the $n^{o(1)}$ factor in our bounds. If $|U|$ is too small, we can simply replicate the elements to ensure $|U| = \omega(\log n)$.

We restate the filter family: The functions are constructed by sampling a random subset $S \subseteq U, |S| = \omega(\log n)$ with replacement, and picking two thresholds, $1 \geq t_u \geq 0, 1 \geq t_q \geq 0$. Then

$$F_u(y) = \begin{cases} [|y \cap S| \geq t_u |S|] & \text{if } t_u \geq w_u \\ [|y \cap S| \leq t_u |S|] & \text{if } t_u < w_u \end{cases} \quad \text{and} \quad F_q(x) = \begin{cases} [|x \cap S| \geq t_q |S|] & \text{if } t_q \geq w_q \\ [|x \cap S| \leq t_q |S|] & \text{if } t_q < w_q \end{cases}.$$

To use theorem 3 we then need to compute $p_u = \Pr[F_u(y) = 1]$, $p_q = \Pr[F_q(x) = 1]$ and $p_1 = \Pr[F_u(y) = 1 \wedge F_q(x) = 1]$. The two first follow from the standard Entropy Chernoff bound: $\log p_u = -D(t_u, w_u)|S|(1+o(1))$ and $\log p_q = -D(t_q, w_q)|S|(1+o(1))$. where D is from definition 3. Note that this form of the Chernoff bound holds in the $t_u \geq w_u$ case as well as the $t_u < w_u$ case.

The joined probability p_1 (and p_2) is more tricky. Note that we need a bound which is tight up to a factor $1 + o(1)$ in the exponent. We will do this using the Large Deviations theorem by Gartner Ellis (see below) on the sequence $\{X_i\}_{i \in [|S|]} \subseteq \{0,1\}^2$ of outcomes when sampling S, where $X_{i,1} = F_q^{(i)}(x)$ and $X_{i,2} = F_u^{(i)}(x)$. This has joint Bernoulli distribution $\sim \left[\begin{smallmatrix} w_1 & w_u - w_1 \\ w_q - w_1 & 1 - w_q - w_u + w_1 \end{smallmatrix}\right]$ or concretely: $\Pr[X_i = (1,1)] = w_1$, $\Pr[X_i = (1,0)] = w_q - w_1$, $\Pr[X_i = (0,1)] = w_u - w_1$ and $\Pr[X_i = (0,0)] = 1 - w_q - w_u + w_1$.

Using Gartner Ellis we will show the following lemma, from which theorem 4 follows:

Lemma 3.1. *there is a $(w_q, w_u, w_1,\ w_2,\ p_1,\ p_2,\ p_q,\ p_u)$-sensitive filter, where*

$$\begin{aligned} |S|^{-1} \log 1/p_1 &= \Lambda(w_q, w_u, w_1, t_q, t_u) + o(1), \\ |S|^{-1} \log 1/p_2 &= \Lambda(w_q, w_u, w_2, t_q, t_u) + o(1), \\ |S|^{-1} \log 1/p_q &= D(t_q, w_q) + o(1), \\ |S|^{-1} \log 1/p_u &= D(t_u, w_u) + o(1). \end{aligned}$$

3.1 Large Deviations

Theorem 5 (Gartner-Ellis theorem [DZ10, Theorem 2.3.6 and Corollary 6.1.6])**.** *Let $\{X_i\}_{i\in\mathbb{N}} \subseteq \mathbb{R}^k$ be a sequence of iid. random vectors. Let $S_n = \frac{1}{n}\sum_{i=0}^{n} X_i$ be the empirical means. Define the logarithmic generating function $\Lambda(\lambda) = \log E \exp\langle\lambda, X_1\rangle$, and the rate function $\Lambda^*(z) = \sup_{\lambda\in\mathbb{R}^k}\{\langle\lambda, z\rangle - \Lambda(\lambda)\}$. If $\Lambda(\varepsilon) < \infty$ for all $\varepsilon \in \mathbb{R}^k$ with $\|\epsilon\|_2 < \delta$ for some $\delta > 0$ small enough, then for any set $F \subseteq \mathbb{R}^k$:*

$$\lim_{n\to\infty} \log \Pr\left[S_n \in F\right] = -\inf_{z\in F} \Lambda^*(z).$$

From this we can derive the more simple:

Lemma 3.2 (Multi Dimensional Cramer)**.** *Let $X_i \in \mathbb{R}^k$ be a sequence of iid. random variables, and let $t \in \mathbb{R}^k$ be a list of values such that $E[X_1] \le t \le \max X_1$. Let $\Lambda(\lambda) = \log E[\exp\langle X_1, \lambda\rangle]$ be finite for all $\lambda \in \mathbb{R}^k$ then*

$$\frac{1}{n}\log \Pr\left[\frac{1}{n}\sum_{i=1}^{n} X_i \ge t\right] = -\Lambda^*(t) + o(1)$$

where $\Lambda^*(t) = \langle t, \lambda\rangle - \Lambda(\lambda)$ and $\nabla\Lambda(\lambda) = t$.

Proof. We use the Gartner-Ellis theorem. Since we assume $\Lambda(z)$ is finite everywhere, it is also so an epsilon ball around 0. Next note that $\Lambda(\lambda)$ is convex so $\langle\lambda, z\rangle - \Lambda(\lambda)$ is maximized at $\nabla\Lambda(\lambda) = z$.

We need to show $\inf_{z\ge t}\Lambda^*(z) = \Lambda^*(t)$. Let $\mu = E[X_1]$.

Note $\frac{d\Lambda}{d\lambda_i}(0) = \mu_i$ (since Λ is a mgf.), thus if $z_i = \mu_i$ then $\lambda_i = 0$ and so $\frac{d\Lambda^*}{dz_i}(\mu_i) = 0$. From this, and the convexity of Λ^* we get $\langle\nabla\Lambda^*(z), z - \mu\rangle \ge 0$, and so for any point in $\{z \ge t\}$ we can always decrease $\Lambda^*(z)$ be moving towards μ, showing that the minimum is achieved at $z = t$. □

See also the details in the appendix and in [24]. Finally we can get the specific version we need:

Lemma 3.3. *Let $X_i \in \{0,1\}^2$, $i \in [m]$, have joint probability distribution $P \in [0,1]^{2\times 2}$ such that $\Pr[X_i = (a,b)] = P_{a,b}$. Let $w = P_{1,1}$, $\mu_1 = P_{1,1} + P_{1,2}$ and $\mu_2 = P_{1,1} + P_{2,1}$, if $\mu_1 < t_1 < 1, \mu_2 < t_2 < 1$, then*

$$\frac{1}{m}\log \Pr\left[\tfrac{1}{m}\sum_{i=1}^{n} X_{i,1} \ge t_1 \wedge \tfrac{1}{m}\sum_{i=1}^{n} X_{i,2} \ge t_2\right] = -\Lambda(\mu_1, \mu_2, w, t_1, t_2) + o(1),$$

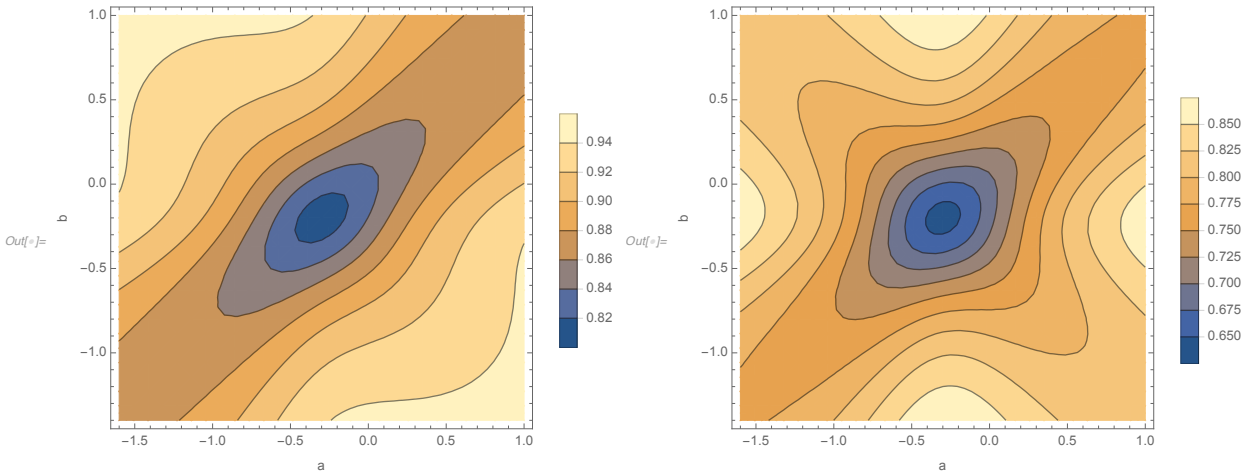
(where Λ is define in definition 3.) If $t_1 < \mu_1$ then ($\ge$) above is replaced by $\le$ and similarly for $t_2 < \mu_2$.

Proof. This follows directly by lemma 3.2, when we plug-in λ_1, λ_2 to check that indeed

$$t = \nabla\Lambda(\lambda) = \frac{1}{be^{\lambda_1+\lambda_2} + (x-b)e^{\lambda_1} + (y-b)e^{\lambda_2} + (1-x-y+b)} \begin{bmatrix} be^{\lambda_1+\lambda_2} + (x-b)e^{\lambda_1} \\ be^{\lambda_1+\lambda_2} + (y-b)e^{\lambda_2} \end{bmatrix}.$$

□

This proves theorem 4.



(a) ρ-hyperplane (b) ρ-spherical

Figure 2: ρ-values for hyperplane and spherical LSH under different shifts.

3.2 Embedding onto the Sphere

Recall lemma 1.1: Let $g, h : \{0,1\}^d \to \mathbb{R}$ be function on the form $g(1) = a_1 x + b_1$ and $h(y) = a_2 y + b_2$. Let $\rho(x, y, y') = f(\alpha(x, y))/f(\alpha(x, y'))$ where $\alpha(x, y) = \langle x, y\rangle / \|x\|\|y\|$ be such that

$$f(z) \geq 0, \quad \tfrac{d}{dz}\left((\pm 1 - z)\tfrac{d}{dz}\log f(z)\right) \geq 0 \quad \text{and} \quad \tfrac{d^3}{dz^3}\log f(z) \leq 0$$

for all $z \in [-1, 1]$. Assume we know that $\|x\|_2^2 = w_q d$, $\|y\|_2^2 = w_u d$, $\langle x, y'\rangle = w_1 d$ and $\langle x, y\rangle = w_2 d$, then $\arg\min_{a_1,b_1,a_2,b_2} \rho(g(x), h(y), h(y')) = (1, -w_q, 1, -w_u)$.

In this section we will show that Hyperplane [16] and Spherical [8] LSH both satisfy the requirements of the lemma. Hence we get two algorithms with ρ-values:

$$\rho_{\text{hp}} = \frac{\log(1 - \arccos(\alpha)/\pi)}{\log(1 - \arccos(\beta)/\pi)}, \quad \rho_{\text{sp}} = \frac{1-\alpha}{1+\alpha}\frac{1+\beta}{1-\beta}.$$

where $\alpha = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$ and $\beta = \frac{w_2 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$, and space/time trade-offs using the ρ_q, ρ_u values in [20]. [8] Figure 2 shows how ρ varies with different translations a, b.

Taking $t_q = w_q(1+o(1))$ and $t_u = w_u(1+o(1))$ in theorem 4 recovers ρ_{sp} by standard arguments. This implies that theorem 4 dominates Spherical LSH (for binary data).

Lemma 3.4. *The functions $f(z) = (1-z)/(1+z)$ for Spherical LSH and $f(z) = -\log(1 - \arccos(z)/\pi)$ for Hyperplane LSH satisfy lemma 1.1.*

Proof. For Spherical LSH we have $f(z) = (1-z)/(1+z)$ and get

$$\tfrac{d}{dz}\left((\pm 1 - z)\tfrac{d}{dz}\log f(z)\right) = 2(1 \mp 2z + z^2)/(1-z^2)^2 \geq 0,$$
$$\tfrac{d^3}{dz^3}\log f(z) = -4(1+3z^2)/(1-z^2)^3 \leq 0.$$

[8] Unfortunately the space/time aren't on a form applicable to lemma 1.1. From numerical experiments we however still conjecture that the embedding is optimal for those as well.

For Hyperplane LSH we have $f(z) = -\log(1 - \arccos(z)/\pi)$ and get

$$\frac{d}{dz}\left((1-z)\frac{d}{dz}\log f(z)\right) = \frac{(\arccos(z) - \sqrt{1-z^2} - \pi)\log(1-\arccos(z)/\pi) - \sqrt{1-z^2}}{(1+z)\sqrt{1-z^2}(\pi - \arccos(z))^2\log(1-\arccos(z)/\pi)^2},$$
$$\frac{d}{dz}\left((-1-z)\frac{d}{dz}\log f(z)\right) = \frac{(\arccos(z) + \sqrt{1-z^2} - \pi)\log(1-\arccos(z)/\pi) + \sqrt{1-z^2}}{(1-z)\sqrt{1-z^2}(\pi - \arccos(z))^2\log(1-\arccos(z)/\pi)^2}.$$

In both cases the denominator is positive, and the numerator can be shown to be likewise by applying the inequalities $\sqrt{1-z^2} \le \arccos(z)$, $\sqrt{1-z^2} + \arccos(z) \le \pi$ and $x \le \log(1+x)$.

The $\frac{d^3}{dz^3}\log f(z) \le 0$ requirement is a bit trickier, but a numerical optimization shows that it's in fact less than -1.53. □

Finally we prove the embedding lemma:

Proof of lemma 1.1. We have

$$\alpha = \frac{\langle x+a, y+b\rangle}{\|x+a\|\|y+b\|} = \frac{w_1 + w_q b + w_u a + ab}{\sqrt{(w_q(1+a)^2 + (1-w_q)a^2)(w_u(1+b)^2 + (1-w_u)b^2)}}$$

and equivalent with w_2 for β. We'd like to show that $a = -w_q$, $b = -w_u$ is a minimum for $\rho = \frac{1-\alpha}{1+\alpha}\frac{1-\beta}{1+\beta}$.

Unfortunately ρ is not convex, so it is not even clear that there is just one minimum. To proceed, we make the following substitution $a \to (c+d)\sqrt{w_q(1-w_q)} - w_q$, $b \to (c-d)\sqrt{w_u(1-w_u)} - w_u$ to get

$$\alpha(c,d) = \frac{cd + \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}}{\sqrt{(1+c^2)(1+d^2)}}.$$

We can further substitute $cd \mapsto rs$ and $\sqrt{(1+c^2)(1+d^2)} \mapsto r+1$ or $r \ge 0$, $-1 \le s \le 1$, since $1 + cd \le \sqrt{(1+c^2)(1+d^2)}$ by Cauchy Schwartz, and $(cd, \sqrt{(1+c^2)(1+d^2)})$ can take all values in this region.

The goal is now to show that $h = f\left(\frac{rs+x}{r+1}\right)/f\left(\frac{rs+y}{r+1}\right)$, where $1 \ge x \ge y \ge -1$, is increasing in r. This will imply that the optimal value for c and d is 0, which further implies that $a = -w_q$, $b = -w_u$ for the lemma.

We first show that h is quasi-concave in s, so we may limit ourselves to $s = \pm 1$. Note that $\log h = \log f\left(\frac{rs+x}{r+1}\right) - \log f\left(\frac{rs+y}{r+1}\right)$, and that $\frac{d^2}{ds^2}\log f\left(\frac{rs+x}{r+1}\right) = \left(\frac{r}{1+r}\right)^2 \frac{d^2}{dz^2}\log f(z)$ by the chain rule. Hence it follows from the assumptions that h is log-concave, which implies quasi-concavity as needed.

We now consider $s = \pm 1$ to be a constant. We need to show that $\frac{d}{dr}h \ge 0$. Calculating,

$$\frac{d}{dr}f\left(\frac{rs+x}{r+1}\right)/f\left(\frac{rs+y}{r+1}\right) = \frac{(s-x)f\left(\frac{rs+y}{r+1}\right)f'\left(\frac{rs+x}{r+1}\right) - (s-y)f\left(\frac{rs+x}{r+1}\right)f'\left(\frac{rs+y}{r+1}\right)}{(1+r)^2 f\left(\frac{rs+y}{r+1}\right)^2}.$$

Since $f \ge 0$ it suffices to show $\frac{d}{dx}(s-x)f'\left(\frac{rs+x}{r+1}\right)/f\left(\frac{rs+x}{r+1}\right) \ge 0$. If we substitute $z = \frac{rs+x}{r+1}$, $z \in [-1,1]$, we can write the requirement as $\frac{d}{dz}(s-z)f'(z)/f(z) \ge 0$ or $\frac{d}{dz}\left((\pm 1 - z)\frac{d}{dz}\log f(z)\right) \ge 0$. □

3.3 A MinHash dominating family

We complete the arguments from section 1.2.3.

We first state the LSF-Symmetrization lemma implicit in [21]:

Lemma 3.5 (LSF-Symmetrization)**.** *Given a (p_1, p_2, p_q, p_u)-sensitive LSF-family, we can create a new family that is $(p_1 q/p, p_2 q/p, q, q)$-sensitive, where $p = \max\{p_q, p_u\}$ and $q = \min\{p_q, p_u\}$.*

For some values of p_1, p_2, p_q, p_u this will be better than simply taking $\max(\rho_u, \rho_q)$. In particular when symmetrization may reduce ρ_u by a lot by reducing its denominator.

Proof. W.l.o.g. assume $p_q \geq p_u$. When sampling a query filter, $Q \subseteq U$, pick a random number $\varrho \in [0, 1]$. If $\varrho > p_u/p_q$ use $\emptyset$ instead of Q. The new family then has $p'_q = p_q \cdot p_u/p_q$ and so on giving the lemma. □

Using this lemma it is easy to make a version of supermajority LSF that always beats Chosen Path: Simply take $t_q = t_u = 1$ and apply lemma 3.5. Then we have exactly the same ρ value as Chosen Path. We do however conjecture that symmetrization is not necessary for supermajorities, since we have another (presumably more efficient) form of symmetrization via asymmetric $t_u \neq t_q$.

Now recall the filter family from the introduction:

$$F_0(x) = [s_0 \in x], F_1(x) \quad = [s_1 \in x \wedge s_0 \notin x], \ldots, F_i(x) = [s_i \in x \wedge s_0 \notin x \wedge \cdots \wedge s_{i-1} \notin x], \ldots$$

where $(s_i \in U)_{i \in \mathbb{N}}$ is a random sequence by sampling elements of U with replacement.

Using just one of these functions, combined with symmetrization, gives the ρ value:

$$\rho_i = \log \frac{(1 - w_q - w_u + w_1)^i w_1}{\max\{(1 - w_q)^i w_q,\, (1 - w_u)^i w_u\}} \Big/ \log \frac{(1 - w_q - w_u + w_2)^i w_2}{\max\{(1 - w_q)^i w_q,\, (1 - w_u)^i w_u\}}.$$

We want to show $\rho_{\text{mh}} = \log \frac{w_1}{w_q + w_u - w_1} / \log \frac{w_2}{w_q + w_u - w_2} \geq \min_{i \geq 0} \rho_i$. For this we show the following lemma, which intuitively says that it is never advantageous to combine multiple filter families:

Lemma 3.6. *The function $f(x, y, z, t) = \log(\max\{x, y\}/z)/\log(\max\{x, y\}/t)$, defined for $\min\{x, y\} \geq z \geq t > 0$, is quasi-concave.*

This means in particular that

$$\frac{\log(\max\{x + x', y + y'\}/(z + z'))}{\log(\max\{x + x', y + y'\}/(t + t'))} \geq \min\left\{ \frac{\log(\max\{x, y\}/z)}{\log(\max\{x, y\}/t)}, \frac{\log(\max\{x', y'\}/z')}{\log(\max\{x', y'\}/t')} \right\},$$

when the variables are in the range of the lemma.

Proof. We need to show that the set

$$\{(x, y, z, t) : \log(\max\{x, y\}/z)/\log(\max\{x, y\}/t) \geq \alpha\} = \{(x, y, z, t) : \max\{x, y\}^{1-\alpha} t^\alpha \geq z\}$$

is convex for all $\alpha \in [0, 1]$ (since $z \geq t$ so $f(x, y, z, t) \in [0, 1]$). This would follow if $g(x, y, t) = \max\{x, y\}^{1-\alpha} t^\alpha$ would be quasi-concave itself, and the eigenvalues of the Hessian of g are exactly 0, 0 and $-(1 - \alpha)\alpha t^{\alpha-2} \max\{x, y\}^{-\alpha-1} \left(\max\{x, y\}^2 + t^2\right)$ so g is even concave! □

We can then show that MinHash is always dominated by one of the filters described, as

$$\rho_{\text{mh}} = \frac{\log \frac{w_1}{w_q+w_u-w_1}}{\log \frac{w_2}{w_q+w_u-w_2}} = \frac{\log \frac{\sum_{i\geq 0}(1-w_q-w_u+w_1)^i w_1}{\max\{\sum_{i\geq 0}(1-w_q)^i w_q, \sum_{i\geq 0}(1-w_u)^i w_u\}}}{\log \frac{\sum_{i\geq 0}(1-w_q-w_u+w_2)^i w_2}{\max\{\sum_{i\geq 0}(1-w_q)^i w_q, \sum_{i\geq 0}(1-w_u)^i w_u\}}} \geq \min_{i\geq 0} \frac{\log \frac{(1-w_q-w_u+w_1)^i w_1}{\max\{(1-w_q)^i w_q, (1-w_u)^i w_u\}}}{\log \frac{(1-w_q-w_u+w_2)^i w_2}{\max\{(1-w_q)^i w_q, (1-w_u)^i w_u\}}},$$

where the right hand side is exactly the symmetrization of the filters $F^{(i)}$. By monotonicity of $(1-w_q)^i w_q$ and $(1-w_u)^i w_u$ we can further argue that it is even possible to limit ourselves to one of $i \in \{0, \infty, \log(w_q/w_u)/\log((1-w_q)/(1-w_u))\}$, where the first gives Chosen Path, the second gives Chosen Path on the complemented sets, and the last gives a balanced trade-off where $(1-w_q)^i w_q = (1-w_u)^i w_u$.

4 Lower bounds

As we discussed in the introduction, it is necessary for our lower bounds to assume $d = \omega(\log n)$. We will also assume w_q, w_u, w_1, w_2 are constants, like we do four our upper bounds, though we don't believe this to be necessary.

We proceed to define the hard distributions for all further lower bounds.

1. A query $x \in \{0,1\}^d$ is created by sampling d random independent bits with Bernoulli(w_q) distribution.

2. A dataset $P \subseteq \{0,1\}^d$ is constructed by sampling $n-1$ vectors with random independent bits from such that $y_i \sim$ Bernoulli(w_2/w_q) if $x_i = 1$ and $y_i \sim$ Bernoulli($(w_u - w_2)/(1-w_q)$) otherwise, for all $y \in P$.

3. A 'close point', y', is created by $y'_i \sim$ Bernoulli(w_1/w_q) if $x_i = 1$ and $y'_i \sim$ Bernoulli($(w_u - w_1)/(1-w_q)$) otherwise. This point is also added to P.

The values are chosen such that $E|x \cap y|/d = w_1$, $E|y|/d = w_u$ for all $y \in P$, and $E|x \cap y'|/d = w_1$ and $E|x \cap y|/d = w_u$ for all $y \in P \setminus \{y'\}$. By a union bound over P, the actual values are within factors $1 + o(1)$ of their expectations with high probability. Changing at most $o(\log n)$ coordinates we ensure the weights of queries/database points is exactly their expected value, while only changing the inner products by factors $1+o(1)$. Since the changes don't contain any new information, we can assume for lower bounds that entries are independent. Thus any $(w_q, w_u, w_1(1-o(1)), w_2(1+o(1)))$-GapSS data structure on P must thus be able to return y' with at least constant probability when given the query x.

Model For the first bound follow O'Donnel et al. [39] and Christiani [20] and directly lower bound the quantity $\frac{\log(p_1/\min\{p_u,p_q\})}{\log(p_2/\min\{p_u,p_q\})}$ which lower bounds ρ_u and ρ_q in definition 2.

For the second and third lower bound we follow Andoni et al. [8] and lower bound a general so called "list-of-points" data structures (see definition 4). This is a slightly more general model, though no it is believed that all bounds for the first model can be shown in the list-of-points model as well.

The second and third bound are shown using so called Hypercontractive inequalities and can be extended to show cell probe lower bounds by the arguments in [41].

4.1 p-biased analysis

In the analysis of boolean functions is is common to use $\{-1,1\}^d$ as the function domain. We'll map 1 to -1 (true) and 0 to 1 (false).

Given functions $f, g : \{-1,1\}^n \to \{0,1\}$, we write

$$f(x) = \sum_{S\subseteq[n]} \hat{f}(S)\phi_S(x), \qquad g(y) = \sum_{S\subseteq[n]} \hat{g}(S)\gamma_S(y) \tag{6}$$

where $\hat{f}, \hat{g} : \mathcal{P}([n]) \to \mathbb{R}$ and $\phi(x_i) = \frac{x_i-\mu_q}{\sigma_q}$, $\gamma(y_i) = \frac{y_i-\mu_u}{\sigma_u}$ for $\mu_q = 1-2w_q, \sigma_q = 2\sqrt{w_q(1-w_q)}$ and $\mu_u = 1-2w_u, \sigma_u = 2\sqrt{w_u(1-w_u)}$. Finally $\phi_S, \gamma_S : \{-1,1\}^n \to \mathbb{R}$ are defined $\phi_S(x) = \prod_{i\in S}\phi(x_i)$ and respectively for y.

Any boolean function can be expanded as (6), but it is particularly useful in our case. To see why, let π be the probability distribution, with the following probability mass function: $\pi(-1) = w_q, \pi(1) = 1-w_q$, and let $\pi^n : \{-1,1\}^n \to [0,1]$ be the product distribution on $\{-1,1\}^n$. We then have the useful properties:

$$\begin{aligned} p_q &= \Pr_{x\sim\pi^n}[f(x)=1] = E_{x\sim\pi^n}[f(x)] = E_{x\sim\pi^n}[\sum_{S\subseteq[n]} \hat{f}(S)\phi_S(x)] = \hat{f}(\emptyset) \\ &= E_{x\sim\pi^n_{w_q}}[f(x)^2] = E_{x\sim\pi^n_{w_q}}[\sum_{S,T\subseteq[n]} \hat{f}(S)\hat{f}(T)\phi_S(x)\phi_T(x)] = \sum_{S\subseteq[n]} \hat{f}(S)^2. \end{aligned}$$

If we think of f as an LSF-filter, $\Pr_{x\sim\pi^n}[f(x)=1]$ is the probability that the filter accepts a random point with expected weight w_q (nw_q of coordinates being -1).

Next, we let ψ be the probability distribution, with the following probability mass function:

$$\begin{aligned} \psi(-1,-1) &= w & \psi(-1,1) &= w_q - w \\ \psi(1,-1) &= w_u - w & \psi(1,1) &= 1 - w_q - w_u + w, \end{aligned}$$

then $p = \Pr_{x,y\sim\psi^n}[f(x)=1, g(y)=1]$ is the probability that a random query point, x, and a random data point, y, with $E\langle x,y\rangle/d = w$ both get caught by their respective filter. ($\Pr[x\in Q, y\in U]$ in the language of definition 2.) This has the following nice form:

$$\begin{aligned} p &= E_{x,y\sim\psi^n}[f(x)g(y)] \\ &= E_{x,y\sim\psi^n}[\sum_{S,T\subseteq[n]} \hat{f}(S)\hat{g}(T)\phi_S(x)\gamma_T(x)] \\ &= \sum_{S\subseteq[n]} \hat{f}(S)\hat{g}(S)E_{x,y\sim\psi^n}[\phi_S(x)\gamma_T(x)] \\ &= \sum_{S\subseteq[n]} \hat{f}(S)\hat{g}(S)E_{x,y\sim\psi^n}\left[\prod_{i\in S}\frac{x_i-\mu_q}{\sigma_q}\frac{y_i-\mu_u}{\sigma_u}\right] \\ &= \sum_{S\subseteq[n]} \hat{f}(S)\hat{g}(S)\left(\frac{E_{x,y\sim\psi^n}[x_iy_i]-\mu_q\mu_u}{\sigma_q\sigma_u}\right)^{|S|} \\ &= \sum_{S\subseteq[n]} \hat{f}(S)\hat{g}(S)\left(\frac{w-w_qw_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}\right)^{|S|}. \end{aligned}$$

Note that in the case $w = w_q w_u$ all terms except $(S = \emptyset)$ are 0, so we have $\Pr_{x,y\sim\psi^n}[f(x) = 1, g(y) = 1] = \Pr[f(x) = 1]\Pr[g(y) = 1]$ as we would expect for x and y independent.

We will define the norm $\|f\|_q = (E_{x\sim\pi^n} f(x)^q)^{1/q}$ and equivalently for g with its respective distribution. Note that since f and g are boolean, we have $\|f\|_q = (E_{x\sim\pi^n} f(x))^{1/q} = \Pr_q[f(x) = 1]^{1/q}$. This will turn out to be very useful.

4.2 Symmetric Lower bound

The simplest approach is to use the expansion directly. This is what O'Donnel used [39] to prove the first optimal LSH lower bounds of $\rho \geq 1/c$ for data-independent hashing. Besides handling the case of set similarity with filters rather than hash functions, we slightly generalize the approach a big by using the power-means inequality rather than log-concavity. [9]

We will show an inequality on the form

$$\left(\frac{\Pr_{x,y',f,g}[f(x) = 1, g(y') = 1]}{\min\{\Pr_{x,f}[f(x) = 1], \Pr_{y,g}[g(y) = 1]\}}\right)^{1/\log\alpha} \leq \left(\log\frac{\Pr_{x,y,f,g}[f(x) = 1, g(y) = 1]}{\min\{\Pr_{x,f}[f(x) = 1], \Pr_{y,g}[g(y) = 1]\}}\right)^{1/\log\beta}$$

where $\alpha = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$ and $\beta = \frac{w_2 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$, and y' and y are sampled as respectively a close and a far point (see the top of the section).

If we knew the filter family $\mathcal{F}$ was regular, that is $(Q, U) \sim \mathcal{F}$ have fixed $|Q|$ and $|U|$, we wouldn't have to take the expectation over f and g. However doing so is only a minor syntactic annoyance in the proof below, and ensures that one cannot circumvent the proof by letting $|Q|$ and $|U|$ be stochastic.

We will prove something slightly stronger than theorem 2:

Lemma 4.1. *Given an LSF-family $\mathcal{F}$, let $f, g : \{-1, 1\}^n \to \{0, 1\}$ be random functions such that $f^{-1}(1) = Q$ and $g^{-1}(1) = U$ for $Q, U \sim \mathcal{F}$. Assume further that $E_{f,g}[\sum_{|S|=k} \hat{f}(S)\hat{g}(S)] \geq 0$ for all $k \in [n]$, then any LSF data structure with $\rho_q = \rho_u = \rho$ must have*

$$\rho \geq \log\left(\frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}\right) \Big/ \log\left(\frac{w_2 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}\right).$$

In particular this bound holds when $\hat{f} = \hat{g}$ almost surely, since $\sum_{|S|=k} \hat{f}(S)^2$ is clearly non-negative. In the context of theorem 2 we have $w_q = w_u$ and $Q = U$ as $Q, U \sim \mathcal{F}$, so theorem 2 follows from lemma 4.1.

Proof. Let $\alpha = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$ and $\beta = \frac{w_2 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$, such that $p_1 = \sum_{S\subseteq[n]} \hat{f}(S)\hat{g}(S)\alpha^{|S|}$ and $p_2 = \sum_{S\subseteq[n]} \hat{f}(S)\hat{g}(S)\beta^{|S|}$.

By Hölder's inequality $\sum_S \hat{f}(S)\hat{g}(S) \leq \min\{\|f\|_1\|g\|_\infty, \|f\|_\infty\|g\|_1\} = \min\{p_q, p_u\}$. Let $p = E_{f,g}[\min\{p_u, p_q\}]$. By assumption $E_{f,g}[\sum_{|S|=k} \hat{f}(S)\hat{g}(S)] \geq 0$ for all k, so we have that $\sum_k \alpha^k E_{f,g}[\sum_{|S|=k} \hat{f}(S)\hat{g}(S)]/p$ is a weighted average over the α^k terms. As such we can use the

[9] This widens the range in which the bound is applicable – the O'Donnel bound is only asymptotic for $r \to 0$. However the values we obtain outside this range, when applied to Hamming space LSH, aren't sharp against the upper bounds.

power-means inequality:

$$(E_{f,g}[p_1]/p)^{1/\log\alpha} = \big(\sum_k (e^k)^{\log\alpha} E_{f,g}[\sum_{|S|=k} \hat{f}(S)\hat{g}(S)]/p\big)^{1/\log\alpha}$$
$$\leq \big(\sum_k (e^k)^{\log\beta} E_{f,g}[\sum_{|S|=k} \hat{f}(S)\hat{g}(S)]/p\big)^{1/\log\beta} = (E_{f,g}[p_2]/p)^{1/\log\beta}.$$

which implies by rearrangement

$$\rho = \frac{\log(E_{f,g}[p_1]/p)}{\log(E_{f,g}[p_2]/p)} \geq \frac{\log\alpha}{\log\beta}.$$

For ρ_q the inequality above follows from $\log(p/p_1)/\log(p/p_2)$ being increasing in p. For ρ_u it is simply increasing the denominator or decreasing the numerator. □

As noted the bound is sharp against our upper bound when w_u, w_q, w_1, w_2 are all small. Also notice that $\log\alpha/\log\beta \leq \frac{1-\alpha}{1+\alpha}\frac{1-\beta}{1+\beta}$ is a rather good approximation for α and β close to 1. Here the right hand side is the ρ value of Spherical LSH with the batch-normalization embedding discussed in section 3.2.

It would be interesting to try an extend this bound to get rid of the $\sum_{|S|=k} \hat{f}(S)\hat{g}(S) \geq 0$ assumption. We conjecture that this is the case, such that the bound holds even when $f \neq g$ and $w_q \neq w_u$.

Note that the lower bound becomes 0 when $w_2 \to w_q w_u$. In the next sections we will find a bound for exactly this case.

4.3 Hypercontractive Lower Bounds

For $x \in \{-1,1\}^d$ let $y \sim N_\sigma(x)$ be sampled by choosing $y_i \in \{-1,1\}$ for each coordinate $i \in [d]$ independently, such that with probability σ we set $y_i = x_i$ and otherwise we set y_i at random. The classic Bonami-Beckner (or Hypercontractive) inequality [39] says that for any $1 \leq p \leq q$ and $0 \leq \sigma \leq \sqrt{(p-1)/(q-1)}$, any boolean function $f : \{-1,1\}^d \to \mathbb{R}$ satisfies $\|T_\sigma f\|_q \leq \|f\|_p$ where $T_\sigma f(x) = E_{y\sim N_\sigma(x)}[f(y)]$.

This inequality has been used to show many lower bounds for locality sensitive data structures, see e.g. [35, 41, 20, 8]. Usually in a model like the following defined in [8] as a "list-of-points" data structure.

Definition 4 (List-of-points)**.** *Given some universes, Q, U, a similarity measure $S : Q \times U \to [0,1]$ and two thresholds $1 \geq s_1 > s_2 \geq 0$,*

1. *We fix (possibly random) sets $A_i \subseteq \{-1,1\}^d$, for $1 \leq i \leq m$; and with each possible query point $q \in \{-1,1\}^d$, we associate a (random) set of indices $I(q) \subseteq [m]$;*
2. *For a given dataset P, we maintain m lists of points $L_1, L_2, \ldots, L_m$, where $L_i = P \cap A_i$.*
3. *On query q, we scan through each list L_i for $i \in I(q)$ and check whether there exists some $p \in L_i$ with $S(q,p) \geq s_2$. If it exists, return p.*

The data structure succeeds, for a given $q \in Q, p \in P$ with $S(q,p) \geq s_1$, if there exists $i \in I(q)$ such that $p \in L_i$.

We use the same hard distribution as before. By Yao's principle we can assume the data structure is deterministic. For $i \in [m]$ we define $p_q^{(i)} = \Pr[i \in I(q)]$, $p_u^{(i)} = \Pr[y \in A_i]$, and $p_1^{(i)} \geq \Pr[i \in I(q), y \in A_i]$ for all q, y with $S(q,y) \geq s_1$.

Let r and s be constants, if

$$p_1^{(i)} \leq (p_q^{(i)})^{1/r} (p_u^{(i)})^{1/s} \quad \text{for all } i, \tag{7}$$

it was shown in [8] (lemma 7.2 and 7.3) that if "far points" are independent of the query, that is $\Pr[i \in I(q), y \in A_i] = p_q^{(i)} p_u^{(i)}$ for all q, i and $y \in P$, then any data structure that succeeds with constant probability must have

$$\rho_q \geq (1+\rho_u)(1-r) + r/s \tag{8}$$

where n^{ρ_q} is the expected query time and $n^{1+\rho_u}$ is the expected space consumption. In our case the "random" requirement corresponds to $w_2 = w_q w_u$.

Note that eq. (8) is equivalent to the requirement to the parametric inequalities

$$\rho_q \geq (1/r - 1)\alpha + 1/s \quad \text{or} \quad \rho_u \geq \alpha/r + 1/s - 1,$$

where $\alpha \geq 0$ which is the form we will use below. [10]

It was also shown in [8] that eq. (7) gives a cell-probe lower bound of $\Omega((n/w)^{\frac{r}{(r-1)s}})$ memory cells of size w, when the data structure is only allowed 1 probe. We won't go into details about this, just note that this matches eq. (8) at $\rho_q = 0$ for memory cells up to size $n^{o(1)}$.

4.3.1 Lower Bound 1

Recall theorem 1: Given $\alpha \geq 0$ and $r, s \geq 2$, let $u_q = \log \frac{1-w_q}{w_q}$, $u_u = \log \frac{1-w_u}{w_u}$, $\sigma = \frac{\sinh(u_q(1-1/r))}{\sinh(u_q/r)}$, and $\upsilon = \frac{\sinh(u_u(1-1/s))}{\sinh(u_u/s)}$. If r and s are such that $\sqrt{\sigma\upsilon} = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_q)w_u(1-w_u)}}$, then any LSF data structure must have

$$\rho_q \geq (1/r - 1)\alpha + 1/s \quad \text{or} \quad \rho_u \geq \alpha/r + 1/s - 1,$$

Proof. We will prove this theorem using the p-biased version of the hypercontractive inequality, which says:

Theorem 6 ([38] also [37] Theorem 10.18 and [47])**.** *Let (Ω, π) be a finite probability space, $|\Omega| \geq 2$, in which every outcome has probability at least $\lambda < 1/2$. Let $f \in L^2(\Omega, \pi)$. Then for any $q > 2$ and $\upsilon = \sqrt{\frac{\sinh(u/q)}{\sinh(u/q')}}$,*

$$\sum_{S \subseteq [n]} \upsilon^2 \hat{f}(S)^2 \leq \|f\|_{q'}$$

where $1/q + 1/q' = 1$ and $u = \log \frac{1-\lambda}{\lambda}$.

We can generalize this to two general functions, using Cauchy Schwartz:

$$p_1 = \sum_{S \subseteq [n]} \sqrt{\upsilon\sigma}^{|S|} \hat{f}(S)\hat{g}(S) \leq \sqrt{\sum_{S \subseteq [n]} \upsilon^{|S|} \hat{f}^2(S) \sum_{S \subseteq [n]} \sigma^{|S|} \hat{g}^2(S)} \leq \|f\|_r \|g\|_s = p_q^{1/r} p_u^{1/s}.$$

where $\upsilon = \frac{\sinh(u_q(1-1/r))}{\sinh(u_q/r)}$, $\sigma = \frac{\sinh(u_u(1-1/s))}{\sinh(u_u/s)}$, $u_q = \log \frac{1-w_q}{w_q}$, $u_u = \log \frac{1-w_u}{w_u}$ and $r, s \geq 2$.

Combined with the discussion around eq. (8) this gives the theorem. □

[10] This is indeed what we would have expected from the LSF-model, definition 2, since $\rho_q = \frac{\log p_1/p_q}{\log p_2/p_q} = \frac{\log p_q^{1/r} p_u^{1/s}/p_q}{\log p_q p_u/p_q} = (1/r-1)\alpha + 1/s$ and $\rho_u = \frac{\log p_1/p_u}{\log p_2/p_q} = \frac{\log p_q^{1/r} p_u^{1/s}/p_u}{\log p_q p_u/p_q} = \alpha/r + 1/s - 1$ for $\alpha = \frac{\log p_q}{\log p_u} \geq 0$.

Lower bound for corollary 1 We continue to prove a lower bound for corollary 1 in the case $w_q = w_u$, $w_2 = w_q w_u$. In the theorem, we set $\alpha = 1$, $r = s$ and $\sigma = \upsilon$, then the theorem asks us to set

$$r = \frac{2u_q}{\log \frac{e^{u_q}(e^{u_q}+\upsilon)}{1+e^{u_q}\upsilon}} = \frac{2\log\frac{1-w_q}{w_q}}{\log\frac{1-2w_q+w_1}{w_1}}$$

where we note that $r \geq 2$ since $w_1 < 1 - 2w_q + w_1$. Now $\rho_q, \rho_u \geq 1/r + 1/s - 1$ is exactly $\log\left(\frac{w_1}{1-2w_q+w_1}\frac{1-w_q}{w_q}\right) / \log\left(\frac{1-w_q}{w_q}\right)$, matching corollary 1 as we wanted.

Optimal choice of r and s The goal is to maximize $\alpha/r+1/s$ conditioned on $\sqrt{\sigma\upsilon} = \frac{w_1 - w_q w_u}{\sqrt{w_q(1-w_1)w_u(1-w_u)}}$ thus maximizing both ρ_q and ρ_u. To make things easier, we substitute $r \mapsto 1/(1-r')$ and $s \mapsto 1/(1-s')$. Then we have $\frac{d\upsilon}{dr'} = \frac{u_q \sinh(u_q)}{\sinh(u_q(1-r'))^2}$ and likewise for σ and s'. By Lagrange multipliers we get the two equations, for some $\lambda \in \mathbb{R}$: $\frac{\lambda\sigma u_q \sinh(u_q)}{\sinh(u_q(1-r'))^2} = \alpha$, $\frac{\lambda\upsilon u_u \sinh(u_u)}{\sinh(u_u(1-s'))^2} = 1$. Dividing through gives the condition:

$$\frac{\sigma u_q \sinh(u_q)\sinh(u_u s')\sinh(u_u(1-s'))}{\upsilon u_u \sinh(u_u)\sinh(u_q r')\sinh(u_q(1-r'))} = \alpha.$$

We can insert this in the theorem to get the optimal r, s for any α on the trade-off. Because of the $r, s \geq 2$ condition, this is not always possible to achieve, but when it is Figure 1 suggests that the lower bound is tight when this condition can be met and further $w_q = w_u$.

Wolff [47] has shown how to extend the p-biased hypercontractive inequality beyond $r, s \geq 2$. However his work is only asymptotic. From the plots it is also clear that for $w_q \neq w_u$ theorem 1 is not sharp. It thus seems evident that we need new methods. In the next section we will investigate a new two-function hypercontractive inequality for this purpose.

4.3.2 Lower Bound 2

We conjecture a new hypercontractive inequality:

Conjecture 2 (Two-Function p-Biased Hypercontractivity Inequality)**.** *For $0 < w_q w_u \leq w \leq w_q, w_u < 1$, Let $\psi : \{-1,1\}^2 \to [0,1]$ be the joint probability density function $\sim \begin{bmatrix} w & w_u - w \\ w_q - w & 1 - w_q - w_u + w \end{bmatrix}$.*

For any pair of boolean functions $f, g : \{-1,1\}^n \to \{0,1\}$ then

$$E_{x,y\sim\psi}[f(x)g(y)] \leq \|f\|_r \|g\|_s \quad (\text{if } w \geq w_q w_u)$$
$$E_{x,y\sim\psi}[f(x)g(y)] \geq \|f\|_r \|g\|_s \quad (\text{if } w \leq w_q w_u)$$

where $r = s = \log\frac{(1-w_q)(1-w_u)}{w_q w_u} / \log\frac{1-w_q-w_u+w}{w}$.

We reduce it to a simple two-variable inequality, from which conjecture 2 and conjecture 1 would follow. For this we will use the following inductive result by O'Donnell, which we have slightly generalized to support (x, y) from arbitrary shared distributions, rather than just ρ correlated. The proof in O'Donnell [37] goes through without changes.

Theorem 7 (Two-Function Hypercontractivity Induction Theorem [37])**.** *Assume that*

$$\mathop{\mathrm{E}}_{(x,y)\sim\pi}[f(x)g(y)] \leq \|f\|_r \|g\|_q$$

holds for every $f, g \in L^2(\Omega, \pi)$. Then the inequality also holds for every $f, g \in L^2(\Omega^n, \pi^n)$.

This means we just have to show a certain 'two point' inequality. That is, we would like the following to be true:

Lemma 4.2. *For $0 < w_q w_u \le w \le w_q, w_u < 1$, and any $f_{-1}, f_1, g_{-1}, g_1 \in \mathbb{R}$, then*

$$\begin{aligned} f_{-1}g_{-1}w + f_{-1}g_1(w_q - w) + f_1 g_{-1}(w_u - w) + f_1 g_1(1 - w_q - w_u + w) \\ \le (w_q f_{-1}^r + (1-w_q)f_1^r)^{1/r}(w_u g_{-1}^s + (1-w_u)g_1^s)^{1/s} \end{aligned}$$

for $r = s = \log \frac{(1-w_q)(1-w_u)}{w_q w_u} / \log \frac{1-w_q-w_u+w}{w}$.

If $w \le w_q w_u$ then the inequality goes the other direction.

Unfortunately we don't have a proof of this. However computer optimization suggests that it is true at least up to an error of 10^{-14}. Equality is achieved when $f_{-1}/f_1 = g_{-1}/g_1$ are either 1 or $(1 - w_q - w_u + w)/w$, and in these points the gradient match, which suggests the choice of r, s is sharp.

Dividing through, we may assume that $f(1)$ and $g(1)$ are both either 0 or 1. (If the values are negative, we can bound LSH by the positive versions. Rhs doesn't care.) If $g(1) = 0$ we just have to show

$$f(-1)g(-1)w + g(-1)(w_u - w) \le w_q^{1/r} f(-1) w_u^{1/s} g(-1)$$

which follows from the one-function Hypercontractive Inequality.

Otherwise we can focus on proving

$$\begin{aligned} xyw + x(w_q - w) + y(w_u - w) + (1 - w_q - w_u + w) \\ \le (w_q x^r + 1 - w_q)^{1/r}(w_u y^r + 1 - w_u)^{1/r} \end{aligned}$$

where we defined $x = f(-1)/f(1), y = g(-1)/g(1)$.

Assuming now conjecture 2, lower bound 2 (conjecture 1) follows from the discussion around eq. (8).

Setting $\alpha = \frac{D(w_u, 1-w_q)}{D(w_q, 1-w_u)}$ gives

$$\begin{aligned} \rho_q &\ge \frac{(1 - w_q - w_u)\log(\frac{1-w_q-w_u+w_1}{w_1}) - D(w_u, 1-w_q)}{D(w_q, 1-w_u)}, \\ \rho_u &\ge \frac{(1 - w_q - w_u)\log(\frac{1-w_q-w_u+w_1}{w_1}) - D(w_q, 1-w_u)}{D(w_q, 1-w_u)}, \end{aligned}$$

matching exactly corollary 1 for $w_2 = w_q w_u$ and all w_q, w_u, w_1.

Besides actually proving lemma 4.2, it would be nice to extend it to a complete spectrum of r, s values. The fact that we get a match in this specific point suggests that this may indeed be a fruitful path to showing optimality of the entire theorem 4.

5 Acknowledgements

I would like to thank Rasmus Pagh and Tobias Christiani on suggesting the problem discussed in this paper, as well as for reviews of old and new manuscripts. The majority of work on this paper was done while I was visiting Eric Price at University of Texas, and I would like to thank people there for encouragement and discussions on boolean functions. Finally I would like to thank Morgan Mingle, Morten Stöckel, John Kallaugher, Ninh Pham, Evangelos Kipouridis and everyone else who has given feedback on the manuscript.

6 Conclusion

We show new matching upper and lower bounds for Set Similarity in the symmetric setting, $w_q = w_u$. We also show strong evidence that our upper bound is optimal in the asymmetric setting, $w_q \neq w_u$, as well as in the time-space trade-offs. If the lower bounds can be extended, this would unify the approaches between sparse and vectors on the sphere, and finally allow grand old MinHash to retire (from data structures, we don't make any claims about sketching).

6.1 Open problems

More closed forms In particular theorem 4, but also the lower bounds, suffer from only being indirectly stated. It would be useful to have a closed form for how to set t_u and t_q for all values of w_q, w_u, w_1, w_2 - both for practical purposes and for showing properties about the trade-off.

More lower bounds Besides proving conjecture 1 it would be useful to extend it to the entire space/time trade-off. This would seemingly require new hypercontractive inequalities, something that may also be useful in other parts of boolean function analysis.

Handle small sets We currently assume that $w_q, w_u, w_1, w_2 \in [0, 1]$ are constants independent of $|U|$. For the purposes of finding the optimal space partition for GapSS this is not a big deal, but for practical applications of set similarity, supporting small sets would make supermajorities a lot more useful.

Algorithms for low dimension We know that LSF can break the LSH lower bounds when $d = O(\log n)$ [11]. It would be nice to have something similar for sets, even though universes that small will be pretty rare.

Data dependent As mentioned, the biggest break through in LSH over the last decade is probably data-dependent LSH. Naturally we will want to know how this can be extended to set data.

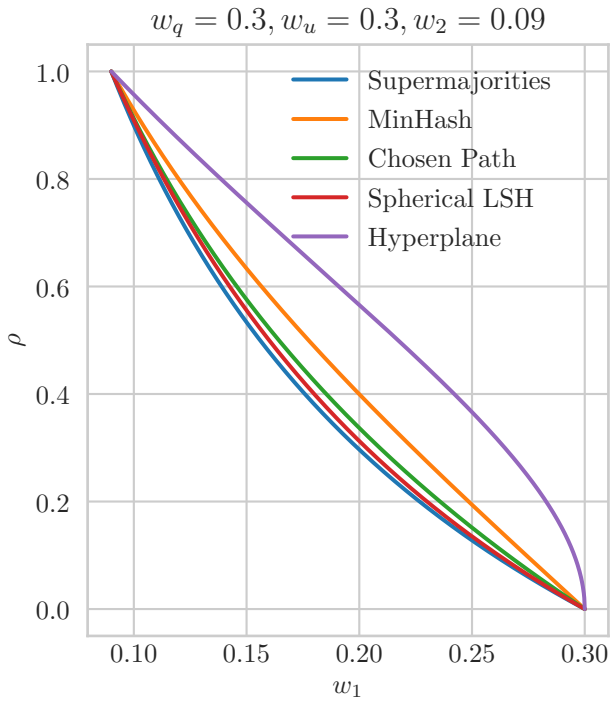
Sparse, non-binary data We now know that threshold functions do well on binary data and on the sphere. It is an exciting open problem to analyse how they do on sparse data on the sphere. This may be the most common type of data in practice.

New framework Valiant showed in [46] that the batch problem of nearest neighbours can be solved faster than permitted by LSH lower bounds. Finding a way to break out of the LSH framework and get similar performance for data structures is a great open problem for set similarity as well as nearest neighbours.

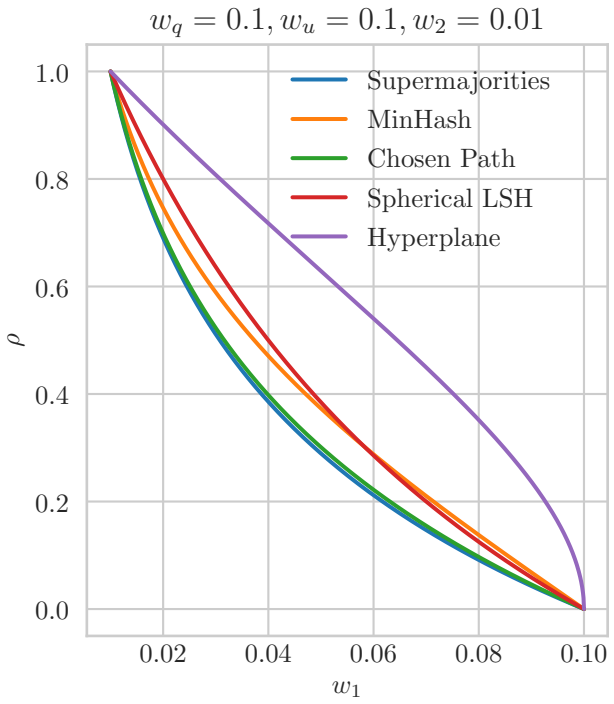
Sketching We have shown that supermajorities can shave large polynomial factors of space and query time in LSH. Can they be used to give similar gains in the field of sketching sets under various similarity measures?

7 Appendix

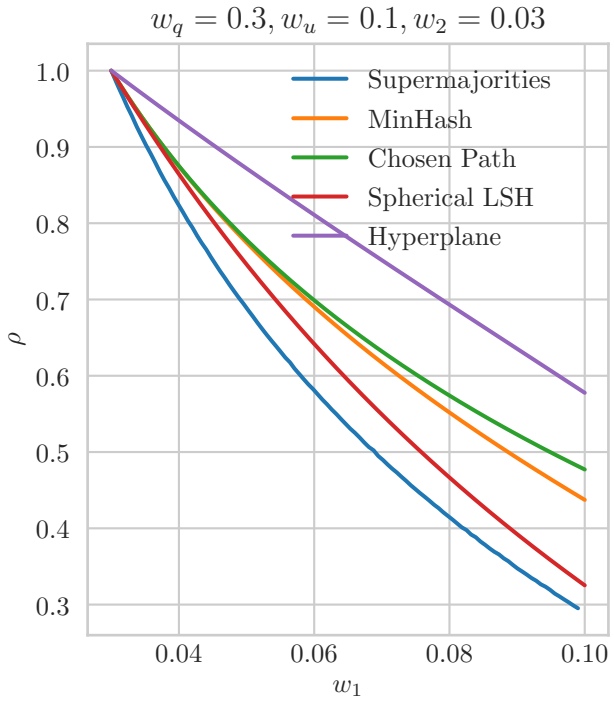
We provide more plots comparing different approaches to GapSS in fig. 3.



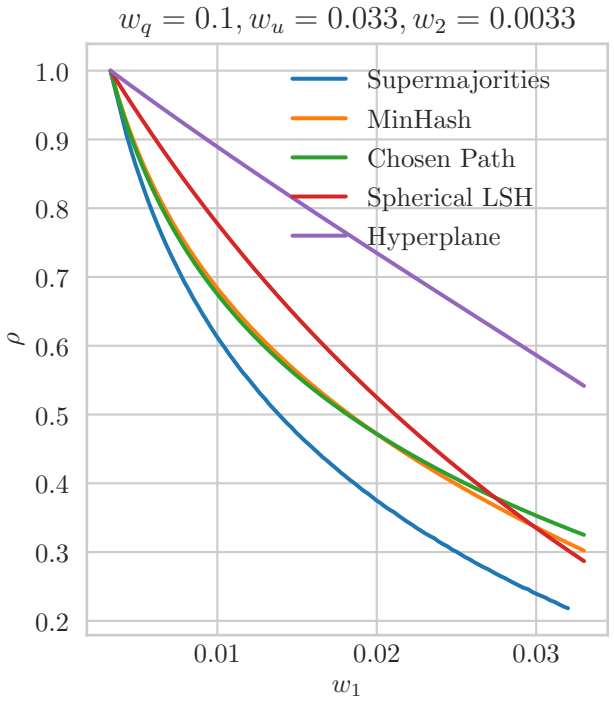
(a) Regular set sizes, relatively large. Note that Spherical LSH is proven optimal in this case, as the sets get large enough.



(b) Regular set sizes, relatively small. Note that Chosen Path is proven optimal in this case, as the sets get small enough.



(c) Irregular set sizes, relatively large.



(d) Irregular set sizes, relatively small.

Figure 3: Examples of ρ-values obtained from theorem 4 in the "balanced" case where query time equals update time, $n^{\rho+o(1)}$. The quantity ρ is plotted in various settings of (w_q, w_u, w_1, w_2)-GapSS, compared to that of other algorithms.

References

[1] Amir Abboud, Aviad Rubinstein, and Ryan Williams. Distributed pcp theorems for hardness of approximation in p. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 25–36. IEEE, 2017.

[2] Parag Agrawal, Arvind Arasu, and Raghav Kaushik. On indexing error-tolerant set containment. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 927–938. ACM, 2010.

[3] Thomas Dybdahl Ahle, Rasmus Pagh, Ilya Razenshteyn, and Francesco Silvestri. On the complexity of inner product similarity join. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 151–164. ACM, 2016.

[4] Josh Alman, Timothy M Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 467–476. IEEE, 2016.

[5] Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. In *Advances in Neural Information Processing Systems*, pages 1225–1233, 2015.

[6] Alexandr Andoni, Piotr Indyk, Huy L Nguyen, and Ilya Razenshteyn. Beyond locality-sensitive hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1018–1028. Society for Industrial and Applied Mathematics, 2014.

[7] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 47–66. Society for Industrial and Applied Mathematics, 2017.

[8] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 47–66. SIAM, 2017.

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

[10] Alexandr Andoni, Ilya Razenshteyn, and Negev Shekel Nosatzki. Lsh forest: Practical algorithms made theoretical. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 67–78. SIAM, 2017.

[11] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 10–24. SIAM, 2016.

[12] Andrei Z Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29. IEEE, 1997.

[13] Andrei Z Broder, Steven C Glassman, Mark S Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *Computer Networks and ISDN Systems*, 29(8-13):1157–1166, 1997.

[14] Timothy M Chan. Orthogonal range searching in moderate dimensions: kd trees and range trees strike back. In *33rd International Symposium on Computational Geometry (SoCG 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[15] Moses Charikar, Piotr Indyk, and Rina Panigrahy. New algorithms for subset query, partial match, orthogonal range searching, and related problems. In *International Colloquium on Automata, Languages, and Programming*, pages 451–462. Springer, 2002.

[16] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM Symposium on Theory of Computing*, pages 380–388. ACM, 2002.

[17] Lijie Chen and Ryan Williams. An equivalence class for orthogonal vectors. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 21–40. SIAM, 2019.

[18] Flavio Chierichetti and Ravi Kumar. Lsh-preserving functions and their applications. *Journal of the ACM (JACM)*, 62(5):33, 2015.

[19] Seung-Seok Choi, Sung-Hyuk Cha, and Charles C Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

[20] Tobias Christiani. A framework for similarity search with space-time tradeoffs using locality-sensitive filtering. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 31–46. SIAM, 2017.

[21] Tobias Christiani and Rasmus Pagh. Set similarity search beyond minhash. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1094–1107, 2017. URL: `https://doi.org/10.1145/3055399.3055443`, `doi:10.1145/3055399.3055443`.

[22] Richard Cole, Lee-Ad Gottlieb, and Moshe Lewenstein. Dictionary matching and indexing with errors and don't cares. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 91–100. ACM, 2004.

[23] Søren Dahlgaard, Mathias Bæk Tejs Knudsen, and Mikkel Thorup. Fast similarity sketching. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 663–671. IEEE, 2017.

[24] Ian H. Dinwoodie. Large deviations techniques and applications (amir dembo and ofer zeitouni). *SIAM Review*, 36(2):303–304, 1994. URL: `https://doi.org/10.1137/1036078`, `doi:10.1137/1036078`.

[25] Raul Castro Fernandez, Jisoo Min, Demitri Nava, and Samuel Madden. Lazo: A cardinality-based method for coupled estimation of jaccard similarity and containment. In *ICDE*, 2019.

[26] Philippe Flajolet, Éric Fusy, Olivier Gandouet, and Frédéric Meunier. Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm. In *Discrete Mathematics and Theoretical Computer Science*, pages 137–156. Discrete Mathematics and Theoretical Computer Science, 2007.

[27] Ashish Goel and Pankaj Gupta. Small subset queries and bloom filters using ternary associative memories, with applications. *ACM SIGMETRICS Performance Evaluation Review*, 38(1):143–154, 2010.

[28] Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory of computing*, 8(1):321–350, 2012.

[29] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

[30] Lianyin Jia, Lulu Zhang, Guoxian Yu, Jinguo You, Jiaman Ding, and Mengjuan Li. A survey on set similarity search and join. *International Journal of Performability Engineering*, 14(2), 2018.

[31] Michael Kapralov. Smooth tradeoffs between insert and query complexity in nearest neighbor search. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 329–342. ACM, 2015.

[32] Matti Karppa, Petteri Kaski, and Jukka Kohonen. A faster subquadratic algorithm for finding outlier correlations. *ACM Transactions on Algorithms (TALG)*, 14(3):31, 2018.

[33] Thijs Laarhoven. Tradeoffs for nearest neighbors on the sphere. *arXiv preprint arXiv:1511.07527*, 2015.

[34] Sergey Melnik and Hector Garcia-Molina. Adaptive algorithms for set containment joins. *ACM Transactions on Database Systems (TODS)*, 28(1):56–99, 2003.

[35] 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. ACM, 2006.

[36] Behnam Neyshabur and Nathan Srebro. On symmetric and asymmetric lshs for inner product search. In *International Conference on Machine Learning*, pages 1926–1934, 2015.

[37] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[38] Krzysztof Oleszkiewicz. On a nonsymmetric version of the khinchine-kahane inequality. In *Stochastic inequalities and applications*, pages 157–168. Springer, 2003.

[39] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Transactions on Computation Theory (TOCT)*, 6(1):5, 2014.

[40] Rina Panigrahy. Entropy based nearest neighbor search in high dimensions. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1186–1195. Society for Industrial and Applied Mathematics, 2006.

[41] Rina Panigrahy, Kunal Talwar, and Udi Wieder. A geometric approach to lower bounds for approximate near-neighbor search and partial match. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 414–423. IEEE, 2008.

[42] Karthikeyan Ramasamy, Jignesh M. Patel, Jeffrey F. Naughton, and Raghav Kaushik. Set containment joins: The good, the bad and the ugly. In *VLDB*, 2000.

[43] Ronald L Rivest. Partial-match retrieval algorithms. *SIAM Journal on Computing*, 5(1):19–50, 1976.

[44] Aviad Rubinstein. Hardness of approximate nearest neighbor search. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1260–1268. ACM, 2018.

[45] Anshumali Shrivastava and Ping Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *Advances in Neural Information Processing Systems*, pages 2321–2329, 2014.

[46] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *Journal of the ACM (JACM)*, 62(2):13, 2015.

[47] Paweł Wolff. Hypercontractivity of simple random variables. *Studia Mathematica*, 3(180):219–236, 2007.

[48] Xiao Yan, Jinfeng Li, Xinyan Dai, Hongzhi Chen, and James Cheng. Norm-ranging lsh for maximum inner product search. In *Advances in Neural Information Processing Systems*, pages 2956–2965, 2018.