A Framework for Similarity Search with Space-Time Tradeoffs using Locality-Sensitive Filtering

Tobias Christiani*
tobc@itu.dk
IT University of Copenhagen

Abstract

We present a framework for similarity search based on Locality-Sensitive Filtering (LSF), generalizing the Indyk-Motwani (STOC 1998) Locality-Sensitive Hashing (LSH) framework to support space-time tradeoffs. Given a family of filters, defined as a distribution over pairs of subsets of space that satisfies certain locality-sensitivity properties, we can construct a dynamic data structure that solves the approximate near neighbor problem in d-dimensional space 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 n denotes the number of points in the data structure. The space-time tradeoff is tied to the tradeoff between query time and update time (insertions/deletions), controlled by the exponents ρ_q, ρ_u that are determined by the filter family.

Locality-sensitive filtering was introduced by Becker et al. (SODA 2016) together with a framework yielding a single, balanced, tradeoff between query time and space, further relying on the assumption of an efficient oracle for the filter evaluation algorithm. We extend the LSF framework to support space-time tradeoffs and through a combination of existing techniques we remove the oracle assumption.

Laarhoven (arXiv 2015), building on Becker et al., introduced a family of filters with space-time tradeoffs for the high-dimensional unit sphere under inner product similarity and analyzed it for the important special case of random data. We show that a small modification to the family of filters gives a simpler analysis that we use, together with our framework, to provide guarantees for worst-case data. Through an application of Bochner's Theorem from harmonic analysis by Rahimi & Recht (NIPS 2007), we are able to extend our solution on the unit sphere to $\mathbb{R}^d$ under the class of similarity measures corresponding to real-valued characteristic functions. For the characteristic functions of s-stable distributions we obtain a solution to the (r, cr)-near neighbor problem in ℓ_s^d-spaces with query and update exponents $\rho_q = \frac{c^s(1+\lambda)^2}{(c^s+\lambda)^2}$ and $\rho_u = \frac{c^s(1-\lambda)^2}{(c^s+\lambda)^2}$ where $\lambda \in [-1, 1]$ is a tradeoff parameter. This result improves upon the space-time tradeoff of Kapralov (PODS 2015) and is shown to be optimal in the case of a balanced tradeoff, matching the LSH lower bound by O'Donnell et al. (ITCS 2011) and a similar LSF lower bound proposed in this paper. Finally, we show a lower bound for the space-time tradeoff on the unit sphere that matches Laarhoven's and our own upper bound in the case of random data.

*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

Let (X, D) denote a space over a set X equipped with a symmetric measure of dissimilarity D (a distance function in the case of metric spaces). We consider the (r, cr)-*near neighbor problem* first introduced by Minsky and Papert [38, p. 222] in the 1960's. A solution to the (r, cr)-near neighbor problem for a set P of n points in (X, D) takes the form of a data structure that supports the following operation: given a query point $\mathbf{x} \in X$, if there exists a data point $\mathbf{y} \in P$ such that $D(\mathbf{x}, \mathbf{y}) \leq r$ then report a data point $\mathbf{y}' \in P$ such that $D(\mathbf{x}, \mathbf{y}') \leq cr$. In some spaces it turns out to be convenient to work with a measure of similarity rather than dissimilarity. We use S to denote a symmetric measure of similarity and define the (α, β)-*similarity problem* to be the $(-\alpha, -\beta)$-near neighbor problem in $(X, -S)$.

A solution to the (r, cr)-near neighbor problem can be viewed as a fundamental building block that yields solutions to many other similarity search problems such as the c-approximate *nearest* neighbor problem [27, 24]. In particular, the (r, cr)-near neighbor problem is well-studied in ℓ_s^d-spaces where the data points lie in $\mathbb{R}^d$ and distances are measured by $D(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_s = (\sum_{i=1}^d |x_i - y_i|^s)^{1/s}$. Notable spaces include the Euclidean space $(\mathbb{R}^d, \|\cdot\|_2)$, Hamming space $(\{0,1\}^d, \|\cdot\|_1)$, and the d-dimensional unit sphere $\mathbb{S}^d = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\|_2 = 1\}$ under inner product similarity $S(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^d x_i y_i$.

Curse of dimensionality All known solutions to the (r, cr)-near neighbor problem for $c = 1$ (the exact near neighbor problem) either suffer from a space

usage that is exponential in d or a query time that is linear in n [24]. This phenomenon is known as the "curse of dimensionality" and has been observed both in theory and practice. For example, Alman and Williams [2] recently showed that the existence of an algorithm for determining whether a set of n points in d-dimensional Hamming space contains a pair of points that are exact near neighbors with a running time strongly subquadratic in n would refute the Strong Exponential Time Hypothesis (SETH) [57]. This result holds even when d is rather small, $d = O(\log n)$. From a practical point of view, Weber et al. [56] showed that the performance of many of the tree-based approaches to similarity search from the field of computational geometry [21] degrades rapidly to a linear scan as the dimensionality increases.

Approximation to the rescue If we allow an approximation factor of $c > 1$ then there exist solutions to the (r, cr)-near neighbor problem with query time that is strongly sublinear in n and space polynomial in n where both the space and time complexity of the solution depends only polynomially on d. Techniques for overcoming the curse of dimensionality through approximation were discovered independently by Kushilevitz et al. [31] and Indyk and Motwani [26]. The latter, classical work by Indyk and Motwani [26, 24] introduced a general framework for solving the (r, cr)-near neighbor problem known as Locality-Sensitive Hashing (LSH). The introduction of the LSH framework has inspired an extensive literature (see e.g. [5, 55] for surveys) that represents the state of the art in terms of solutions to the (r, cr)-near neighbor problem in high-dimensional spaces [26, 18, 20, 44, 4, 5, 3, 6, 30, 11, 14, 32].

Hashing and filtering frameworks The LSH framework and the more recent LSF framework introduced by Becker et al. [14] produce data structures that solve the (r, cr)-near neighbor problem with query and update time $dn^{\rho+o(1)}$ and space usage $dn + n^{1+\rho+o(1)}$. The LSH (LSF) framework takes as input a distribution over partitions (subsets) of space with the locality-sensitivity property that close points are more likely to be contained in the same part (subset) of a randomly sampled element from the distribution. The frameworks proceed by constructing a data structure that associates each point in space with a number of memory locations or "buckets" where data points are stored. During a query operation the buckets associated with the query point are searched by computing the distance to every data point in the bucket, returning the first suitable candidate. The set of memory locations associated with a particular point is independent of whether an update operation or a query operation is being performed. This symmetry between the query and update algorithm results in solutions to the near neighbor problem with a balanced space-time tradeoff. The exponent ρ is determined by the locality-sensitivity properties of the family of partitions/hash functions (LSH) or subsets/filters (LSF) and is typically upper bounded by an expression that depends only on the aproximation factor c. For example, Indyk and Motwani [26] gave a simple locality-sensitive family of hash functions for Hamming space with an exponent of $\rho \leq 1/c$. This exponent was later shown to be optimal by O'Donnell et al. [43] who gave a lower bound of $\rho \geq 1/c - o_d(1)$ in the setting where r and cr are small compared to d. The advantage of having a general framework for similarity search lies in the reduction of the (r, cr)-near neighbor problem to the, often simpler and easier to analyze, problem of finding a locality-sensitive family of hash functions or filters for the space of interest.

Space-time tradeoffs Space-time tradeoffs for solutions to the (r, cr)-near neighbor problem is an active line of research that can be motivated by practical applications where it is desirable to choose the tradeoff between query time and update time (space usage) that is best suited for the application and memory hierarchy at hand [44, 36, 3, 30, 32]. Existing solutions typically have query time $dn^{\rho_q+o(1)}$, update time (insertions/deletions) $dn^{\rho_u+o(1)}$, and use space $dn + n^{1+\rho_u+o(1)}$ where the query and update exponents ρ_q, ρ_u that control the space-time tradeoff depend on the approximation factor c and on a tradeoff parameter $\lambda \in [-1, 1]$. This paper combines a number of existing techniques [14, 32, 22] to provide a general framework for similarity search with space-time tradeoffs. The framework is used to show improved upper bounds on the space-time tradeoff in the well-studied setting of ℓ_s-spaces and the unit sphere under inner product similarity. Finally, we show a new lower bound on the space-time tradeoff for the unit sphere that matches an upper bound for random data on the unit sphere by Laarhoven [32]. We proceed by stating our contribution and briefly surveying the relevant literature in terms of frameworks, upper bounds, and lower bounds as well as some recent developments. See table Table 1 for an overview.

1.1 Contribution

Before stating our results we give a definition of locality-sensitive filtering that supports asymmetry in the framework query and update algorithm, yielding space-time tradeoffs.

DEFINITION 1. *Let (X, D) be a space and let $\mathcal{F}$ be a probability distribution over $\{(Q, U) \mid Q \subseteq X, U \subseteq X\}$. We say that $\mathcal{F}$ is $(r, cr, p_1, p_2, p_q, p_u)$-sensitive if for all points $\mathbf{x}, \mathbf{y} \in X$ and (Q, U) sampled randomly from $\mathcal{F}$ the following holds:*

Table 1: Overview of data-independent locality-sensitive hashing (LSH) and filtering (LSF) results

Reference	**Setting**	ρ_q	ρ_u
LSH [26, 24], LSF [14]	(X, D) or (X, S)	$\frac{\log(1/p)}{\log(1/q)}$	
Theorem 1.1		$\frac{\log(p_q/p_1)}{\log(p_q/p_2)}$	$\frac{\log(p_u/p_1)}{\log(p_q/p_2)}$
Cross-polytope LSH [6]	(α, β)-sim. in $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$	$\frac{1-\alpha}{1+\alpha}\Big/\frac{1-\beta}{1+\beta}$	
Spherical cap LSF [32]	$(\alpha, o_d(1))$-sim. in $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$	$\frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2}$	$\frac{(\alpha^\lambda-\alpha)^2}{1-\alpha^2}$
Theorem 1.2	(α, β)-sim. in $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$	$\frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2}\Big/\frac{(1-\alpha^\lambda\beta)^2}{1-\beta^2}$	$\frac{(\alpha^\lambda-\alpha)^2}{1-\alpha^2}\Big/\frac{(1-\alpha^\lambda\beta)^2}{1-\beta^2}$
Ball-carving LSH [4]	(r, cr)-nn. in ℓ_2^d	$1/c^2$	
Ball-search LSH* [30]		$\frac{c^2(1+\lambda)^2}{(c^2+\lambda)^2-c^2(1+\lambda^2)/2-\lambda^2}$	$\frac{c^2(1-\lambda)^2}{(c^2+\lambda)^2-c^2(1+\lambda^2)/2-\lambda^2}$
Theorem 1.3		$\frac{c^2(1+\lambda)^2}{(c^2+\lambda)^2}$	$\frac{c^2(1-\lambda)^2}{(c^2+\lambda)^2}$
Lower bound [43]	LSH in ℓ_2^d	$\geq 1/c^2$	
Theorem 1.4	LSF in ℓ_2^d	$\geq 1/c^2$	
Lower bound [39, 12]	LSH in $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$	$\geq \frac{1-\alpha}{1+\alpha}$	
Theorem 1.5, [9]	LSF in $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$	$\geq \frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2}$	$\geq \frac{(\alpha^\lambda-\alpha)^2}{1-\alpha^2}$

TABLE NOTES: Space-time tradeoffs for dynamic randomized solutions to similarity search problems in the LSH and LSF frameworks with query time $dn^{\rho_q+o(1)}$, update time $dn^{\rho_u+o(1)}+dn^{o(1)}$ and space usage $dn+n^{1+\rho_u+o(1)}$. Lower bounds are for the exponents ρ_q, ρ_u within their respective frameworks. Here $\varepsilon > 0$ denotes an arbitrary constant and $\lambda \in [-1, 1]$ controls the space-time tradeoff. We have hidden $o_n(1)$ terms in the upper bounds and $o_d(1)$ terms in the lower bounds.
*Assumes $c^2 \geq (1+\lambda)^2/2 + \lambda + \varepsilon$.

- *If* $D(\mathbf{x}, \mathbf{y}) \leq r$ *then* $\Pr[\mathbf{x} \in Q, \mathbf{y} \in U] \geq p_1$.
- *If* $D(\mathbf{x}, \mathbf{y}) > cr$ *then* $\Pr[\mathbf{x} \in Q, \mathbf{y} \in U] \leq p_2$.
- $\Pr[\mathbf{x} \in Q] \leq p_q$ *and* $\Pr[\mathbf{x} \in U] \leq p_u$.

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

Our main contribution is a general framework for similarity search with space-time tradeoffs that takes as input a locality-sensitive family of filters.

THEOREM 1.1. *Suppose we have access to a family of filters that is $(r, cr, p_1, p_2, p_q, p_u)$-sensitive. Then we can construct a fully dynamic data structure that solves the (r, cr)-near neighbor 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 $\rho_q = \frac{\log(p_q/p_1)}{\log(p_q/p_2)}$ and $\rho_u = \frac{\log(p_u/p_1)}{\log(p_q/p_2)}$.*

We give a worst-case analysis of a slightly modified version of Laarhoven's [32] filter family for the unit sphere and plug it into our framework to obtain the following theorem.

THEOREM 1.2. *For every choice of $0 \leq \beta < \alpha < 1$ and $\lambda \in [-1, 1]$ there exists a solution to the (α, β)-similarity problem in $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$ that satisfies the guarantees from Theorem 1.1 with exponents $\rho_q = \frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2}\Big/\frac{(1-\alpha^\lambda\beta)^2}{1-\beta^2}$ and $\rho_u = \frac{(\alpha^\lambda-\alpha)^2}{1-\alpha^2}\Big/\frac{(1-\alpha^\lambda\beta)^2}{1-\beta^2}$.*

We show how an elegant and powerful application of Bochner's Theorem [49] by Rahimi and Recht [48] allows us to extend the solution on the unit sphere to a large class of similarity measures, yielding as a special case solutions for ℓ_s-space.

THEOREM 1.3. *For every choice of $c \geq 1$, $s \in (0, 2]$, and $\lambda \in [-1, 1]$ there exists a solution to the (r, cr)-near neighbor problem in ℓ_s^d that satisfies the guarantees from Theorem 1.1 with exponents $\rho_q = \frac{c^s(1+\lambda)^2}{(c^s+\lambda)^2}$ and $\rho_u = \frac{c^s(1-\lambda)^2}{(c^s+\lambda)^2}$.*

This result improves upon the state of the art for every choice of asymmetric query/update exponents $\rho_q \neq \rho_u$ [44, 4, 3, 30]. We conjecture that this

tradeoff is optimal among the class of algorithms that *independently of the data* determine which locations in memory to probe during queries and updates. In the case of a balanced space-time tradeoff where we set $\rho_q = \rho_u$ our approach matches existing, optimal [43], data-independent solutions in ℓ_s-spaces [26, 20, 4, 40].

The LSF framework is very similar to the LSH framework, especially in the case where the filter family is symmetric ($Q = U$ for every filter in $\mathcal{F}$). In this setting we show that the LSH lower bound by O'Donnell applies to the LSF framework as well [43], confirming that the results of Theorem 1.3 are optimal when we set $\rho_q = \rho_u$.

THEOREM 1.4. (INFORMAL) *Every filter family that is symmetric and $(r, cr, p_1, p_2, p_q, p_u)$-sensitive in ℓ_s^d must have $\rho = \frac{\log(p_u/p_1)}{\log(p_q/p_2)} \geq 1/c^s - o_d(1)$ when $r = \omega_d(1)$ is chosen to be sufficiently small.*

Finally we show a lower bound on the space-time tradeoff that can be obtained in the LSF framework. Our lower bound suffers from two important restrictions. First the filter family must be regular, meaning that all query filters and all update filters are of the same size. Secondly, the size of the query and update filter cannot differ by too much.

THEOREM 1.5. (INFORMAL) *Every regular filter family that is $((1-\alpha)d/2, (1-\beta)d/2, p_1, p_2, p_q, p_u)$-sensitive in d-dimensional Hamming space with asymmetry controlled by $\lambda \in [-1, 1]$ cannot simultanously have that $\rho_q < \frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2} - o_d(1)$ and $\rho_u < \frac{(\alpha^\lambda - \alpha)^2}{1-\alpha^2} - o_d(1)$.*

Together our upper and lower bounds imply that the filter family of concentric balls in Hamming space is asymptotically optimal for random data.

Techniques The LSF framework in Theorem 1.1 relies on a careful combination of "powering" and "tensoring" techniques. For positive integers m and τ with $m \gg \tau$ the tensoring technique, a variant of which was introduced by Dubiner [22], allows us to simulate a collection of $\binom{m}{\tau}$ filters from a collection of m filters by considering the intersection of all τ-subsets of filters. Furthermore, given a point $\mathbf{x} \in X$ we can efficiently list the simulated filters that contain $\mathbf{x}$. This latter property is crucial as we typically need $\text{poly}(n)$ filters to split our data into sufficiently small buckets for the search to be efficient. The powering technique lets us amplify the locality-sensitivity properties of a filter family in the same way that powering is used in the LSH framework [26, 5, 43].

To obtain results for worst-case data on the unit sphere we analyze a filter family based on standard normal projections using the same techniques as Andoni et al. [6] together with existing tail bounds on bivariate Gaussians. The approximate kernel embedding technique by Rahimi and Recht [48] is used to extend the solution on the unit sphere to a large class of similarity measures, yielding Theorem 1.3 as a special case.

The lower bound in Theorem 1.4 relies on an argument of contradiction against the LSH lower bounds by O'Donnell [43] and uses a theoretical, inefficient, construction of a locality-sensitive family of hash functions from a locality-sensitive family of filters that is similar to the spherical LSH by Andoni et al. [7].

Finally, the space-time tradeoff lower bound from Theorem 1.5 is obtained through an application of an isoperimetric inequality by O'Donnell [41, Ch. 10] and is similar in spirit to the LSH lower bound by Motwani et al. [39].

1.2 Related work The LSH framework takes a distribution $\mathcal{H}$ over hash functions that partition space with the property that the probability of two points landing in the same partition is an increasing function of their similarity.

DEFINITION 2. *Let (X, D) be a space and let $\mathcal{H}$ be a probability distribution over functions $h\colon X \to R$. We say that $\mathcal{H}$ is (r, cr, p, q)-sensitive if for all points $\mathbf{x}, \mathbf{y} \in X$ and h sampled randomly from $\mathcal{H}$ the following holds:*

- *If $D(\mathbf{x}, \mathbf{y}) \leq r$ then $\Pr[h(\mathbf{x}) = h(\mathbf{y})] \geq p$.*
- *If $D(\mathbf{x}, \mathbf{y}) > cr$ then $\Pr[h(\mathbf{x}) = h(\mathbf{y})] \leq q$.*

The properties of $\mathcal{H}$ determines a parameter $\rho < 1$ that governs the space and time complexity of the solution to the (r, cr)-near neighbor problem.

THEOREM 1.6. (LSH FRAMEWORK [26, 24]) *Suppose we have access to a (r, cr, p, q)-sensitive hash family. Then we can construct a fully dynamic data structure that solves the (r, cr)-near neighbor problem with query time $dn^{\rho+o(1)}$, update time $dn^{\rho+o(1)}$, and with a space usage of $dn + n^{1+\rho+o(1)}$ where $\rho = \frac{\log(1/p)}{\log(1/q)}$.*

The LSF framework by Becker et al. [14] takes a symmetric $(r, cr, p_1, p_2, p_q, p_u)$-sensitive filter family $\mathcal{F}$ and produces a data structure that solves the (r, cr)-near neighbor problem with the same properties as the one produced by the LSH framework where instead we have $\rho = \frac{\log(p_q/p_1)}{\log(p_q/p_2)}$. In addition, the framework assumes access to an oracle that is able to efficiently list the relevant filters containing a point $\mathbf{x} \in X$ out of a large collection of filters. The LSF framework in this paper removes this assumption, showing how to construct an efficient oracle as part of the framework.

In terms of frameworks that support space-time tradeoffs, Panigrahy [44] developed a framework based on LSH that supports the two extremes of the space-time tradeoff. In the language of Theorem 1.1, Panigrahy's framework supports either setting $\rho_u = 0$ for a solution that uses near-linear space at the cost of a slower query time, or setting $\rho_q = 0$ for a solution with query time $n^{o(1)}$ at the cost of a higher space usage. To obtain near-linear space the framework stores every data point in $n^{o(1)}$ partitions induced by randomly sampled hash functions from a (r, cr, p, q)-sensitive LSH family $\mathcal{H}$. In comparison, the standard LSH framework from Theorem 1.6 uses n^{ρ} such partitions where ρ is determined by $\mathcal{H}$. For each partition induced by $h \in \mathcal{H}$ the query algorithm in Panigrahy's framework generates a number of random points $\mathbf{z}$ in a ball around the query point $\mathbf{x}$ and searches the parts of the partition $h(\mathbf{z})$ that they hash to. The query time is bounded by $n^{\hat{\rho}+o(1)}$ where $\hat{\rho} = \frac{I(h(\mathbf{z})|\mathbf{x},h)}{\log(1/q)}$ and $I(h(\mathbf{z})|\mathbf{x}, h)$ denotes conditional entropy, i.e. the query time is determined by how hard it is to guess where $\mathbf{z}$ hashes to given that we know $\mathbf{x}$ and h. Panigrahy's technique was used in a number of follow-up works that improve on solutions for specific spaces, but to our knowledge none of them state a general framework with space-time tradeoffs [36, 3, 30].

Upper bounds As is standard in the literature we state results in ℓ_s-spaces in terms of the properties of a solution to the (r, cr)-near neighbor problem. For results on the unit sphere under inner product similarity $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$ we instead use the (α, β)-similarity terminology, defined in the introduction, as we find it to be cleaner and more intuitive while aligning better with the analysis. The ℓ_s-spaces, particularly ℓ_1 and ℓ_2, as well as $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$ are some of most well-studied spaces for similarity search and are also widely used in practice [55]. Furthermore, fractional norms (ℓ_s for $s \neq 1, 2$) have been shown to perform better than the standard norms in certain use cases [1] which motivates finding efficient solutions to the near neighbor problem in general ℓ_s-space.

In the case of a balanced space-time tradeoff the best data-independent upper bound for the (r, cr)-near neighbor problem in ℓ_s^d are solutions with an LSH exponent of $\rho = 1/c^s$ for $0 < s \leq 2$. This result is obtained through a combination of techniques. For $0 < s \leq 1$ the LSH based on s-stable distributions by Datar et al. [20] can be used to obtain an exponent of $(1 + \varepsilon)/c^s$ for an arbitrarily small constant $\varepsilon > 0$. For $1 < s \leq 2$ the ball-carving LSH by Andoni and Indyk [4] for Euclidean space can be extended to ℓ_s using the technique described by Nguyen [40, Section 5.5]. Theorem 1.3 matches (and potentially improves in the case of $0 < s < 1$) these results with a single unified technique and analysis that we find to be simpler.

For space-time tradeoffs in Euclidean space (again extending to ℓ_s for $1 < s < 2$) Kapralov [30], improving on Panigrahy's results [44] in Euclidean space and using similar techniques, obtains a solution with query exponent $\rho_q = \frac{c^2(1+\lambda)^2}{(c^2+\lambda)^2 - c^2(1+\lambda^2)/2 - \lambda^2}$ and update exponent $\rho_u = \frac{c^2(1-\lambda)^2}{(c^2+\lambda)^2 - c^2(1+\lambda^2)/2 - \lambda^2}$ under the condition that $c^2 \geq (1+\lambda)^2/2 + \lambda + \varepsilon$ where $\varepsilon > 0$ is an arbitrary positive constant. Comparing to our Theorem 1.3 it is easy to see that we improve upon Kapralov's space-time tradeoff for all choices of c and λ. In addition, Theorem 1.3 represents the first solution to the (r, cr)-near neighbor problem in Euclidean space that for every choice of constant $c > 1$ obtains sublinear query time ($\rho_q < 1$) using only near-linear space ($\rho_u = 0$). Due to the restrictions on Kapralov's result he is only able to obtain sublinear query time for $c > \sqrt{3}$ when the space usage is restricted to be near-linear. It appears that our improvements can primarily be attributed to our techniques allowing a more direct analysis. Kapralov uses a variation of Panigrahy's LSH-based technique of, depending on the desired space-time tradeoff, either querying or updating additional memory locations around a point $\mathbf{x} \in X$ in the partition induced by $h \in \mathcal{H}$. For a query point $\mathbf{x}$ and a near neighbor $\mathbf{y}$ his argument for correctness is based on guaranteeing that both the query algorithm and update algorithm visit the part $h(\mathbf{z})$ where $\mathbf{z}$ is a point lying between $\mathbf{x}$ and $\mathbf{y}$, possibly leading to a loss of efficiency in the analysis. More details on the comparison of Theorem 1.3 to Kapralov's result can be found in Appendix E.

In terms of space-time tradeoffs on the unit sphere, Laarhoven [32] modifies a filter family introduced by Becker et al. [14] to support space-time tradeoffs, obtaining a solution for random data on the unit sphere (the (α, β)-similarity problem with $\beta = o_d(1)$) with query exponent $\rho_q = \frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2}$ and update exponent $\rho_u = \frac{(\alpha^{\lambda}-\alpha)^2}{1-\alpha^2}$. Theorem 1.2 extends this result to provide a solution to the (α, β)-similarity problem on the unit sphere for every choice of $0 \leq \beta < \alpha < 1$. This extension to worst case data is crucial for obtaining our results for ℓ_s-spaces in Theorem 1.3. We note that there exist other data-independent techniques (e.g. Valiant [54, Alg. 25]) for extending solutions on the unit sphere to ℓ_2, but they also require a solution for worst-case data on the unit sphere to work.

Lower bounds The performance of an LSH-based solution to the near neighbor problem in a given space that uses a (r, cr, p, q)-sensitive family of hash functions $\mathcal{H}$ is summarized by the value of the exponent $\rho = \frac{\log(1/p)}{\log(1/q)}$. It is therefore of interest to lower bound ρ

in terms of the approximation factor c. Motwani et al. [39] proved the first lower bound for LSH families in d-dimensional Hamming space. They show that for every choice of $c \geq 1$ then for some choice of r it must hold that $\rho \geq 0.462/c$ as d goes to infinity under the assumption that q is not too small ($q \geq 2^{-o(d)}$).

As part of an effort to show lower bounds for data-dependent locality-sensitive hashing, Andoni and Razenshteyn [13] strengthened the lower bound by Motwani et al. to $\rho \geq 1/(2c-1)$ in Hamming space. These lower bounds are initially shown in Hamming space and can then be extended to ℓ_s-space and the unit sphere by the fact that a solution in these spaces can be used to yield a solution in Hamming space, contradicting the lower bound if ρ is too small. Translated to (α, β)-similarity on the unit sphere, which is the primary setting for the lower bounds on LSF space-time tradeoffs in this paper, the lower bound by Andoni and Razenshteyn shows that an LSH on the unit sphere must have $\rho \geq \frac{1-\alpha}{1+\alpha}$ which is tight in the case of random data [6].

The lower bound uses properties of random walks over a partition of Hamming space: A random walk starting from a random point $\mathbf{x} \in \{0,1\}^d$ is likely to "walk out" of the the part identified by $h(\mathbf{x})$ in the partition induced by h. The space-time tradeoff lower bound in Theorem 1.5 relies on a similar argument that lower bounds the probability that a random walk starting from a subset Q ends up in another subset U, corresponding nicely to query and update filters in the LSF framework.

Using related techniques O'Donnell [43] showed tight LSH lower bounds for ℓ_s-space of $\rho \geq 1/c^s$. The work by Andoni et al. [8] and Panigrahy et al. [45, 46] gives cell probe lower bounds for the (r, cr)-near neighbor problem, showing that in Euclidean space a solution with a query complexity of t probes require space at least $n^{1+\Omega(1/tc^2)}$. For more details on these lower bounds and how they relate to the upper bounds on the unit sphere see [9, 32].

Data-dependent solutions The solutions to the (r, cr)-near neighbor problems considered in this paper are all *data-independent*. For the LSH and LSF frameworks this means that the choice of hash functions or filters used by the data structure, determining the mapping between points in space and the memory locations that are searched during the query and update algorithm, is made without knowledge of the data. Data-independent solutions to the (r, cr)-near neighbor problem for worst-case data have been the state of the art until recent breakthroughs by Andoni et al. [7] and Andoni and Razenshteyn [11] showing improved solutions to the (r, cr)-near neighbor problem in Euclidean space using *data-dependent* techniques. In this setting the solution obtained by Andoni and Razenshteyn has an exponent of $\rho = 1/(2c^2-1)$ compared to the optimal data-independent exponent of $\rho = 1/c^2$. Furthermore, they show that this exponent is optimal for data-dependent solutions in a restricted model [13].

Recent developments Recent work by Andoni et al. [10], done independently of and concurrently with this paper, shows that Laarhoven's upper bound for random data on the unit sphere can be combined with data-dependent techniques [11] to yield a space-time tradeoff in Euclidean space with ρ_u, ρ_q satisfying $c^2\sqrt{\rho_q} + (c-1)\sqrt{\rho_u} = \sqrt{2c^2-1}$. This improves the result of Theorem 1.3 and matches the lower bound in Theorem 1.5. In the same paper they also show a lower bound matching our lower bound in Theorem 1.5. Their lower bound is set in a more general model that captures both the LSH and LSF framework and they are able to remove some of the technical restrictions such as the filter family being regular that weaken the lower bound in this paper. In spite of these results we still believe that this paper presents an important contribution by providing a general and simple framework with space-time tradeoffs as well as improved data-independent solutions to nearest neighbor problems in ℓ_s-space and on the unit sphere. We would also like to point out the simplicity and power of using Rahimi and Recht's [48] result to extend solutions on the unit sphere to spaces with similarity measures corresponding to real-valued characteristic functions, further described in Appendix C.

2 A framework with space-time tradeoffs

We use a combination of powering and tensoring techniques to amplify the locality-sensitive properties of our initial filter family, and to simulate a large collection of filters that we can evaluate efficiently. We proceed by stating the relevant properties of these techniques which we then combine to yield our Theorem 1.1.

LEMMA 2.1. (POWERING) *Given a $(r, cr, p_1, p_2, p_q, p_u)$-sensitive filter family $\mathcal{F}$ for (X, D) and a positive integer κ define the family $\mathcal{F}^\kappa$ as follows: we sample a filter $F = (Q, U)$ from $\mathcal{F}^\kappa$ by sampling $(Q_1, U_1), \ldots, (Q_\kappa, U_\kappa)$ independently from $\mathcal{F}$ and setting $(Q, U) = (\bigcap_{i=1}^\kappa Q_i, \bigcap_{i=1}^\kappa U_i)$. The family $\mathcal{F}^\kappa$ is $(r, cr, p_1^\kappa, p_2^\kappa, p_q^\kappa, p_u^\kappa)$-sensitive for (X, D).*

Let $\mathbf{F}$ denote a collection (indexed family) of m filters and let $\mathbf{Q}$ and $\mathbf{U}$ denote the corresponding collections of query and update filters, that is, for $i \in \{1, \ldots, m\}$ we have that $\mathbf{F}_i = (\mathbf{Q}_i, \mathbf{U}_i)$. Given a positive integer $\tau \leq m$ (typically $\tau \ll m$) we define $\mathbf{F}^{\otimes\tau}$ to be the collection of filters formed by taking all the intersections of τ-combinations of filters from $\mathbf{F}$, that is,

for every $I \subseteq \{1, \ldots, m\}$ with $|I| = \tau$ we have that

$$\mathbf{F}_I^{\otimes\tau} = \left(\bigcap_{i\in I}\mathbf{Q}_i, \bigcap_{i\in I}\mathbf{U}_i\right) \tag{2.1}$$

The following properties of the tensoring technique will be used to provide correctness, running time, and space usage guarantees for the LSF data structure that will be introduced in the next subsection. We refer to the evaluation time of a collection of filters $\mathbf{F}$ as the time it takes, given a point $\mathbf{x} \in X$ to prepare a list of query filters $\mathbf{Q}(x) \subseteq \mathbf{Q}$ containing $\mathbf{x}$ and a list of update filters $\mathbf{U}(x) \subseteq \mathbf{U}$ containing $\mathbf{x}$ such that the next element of either list can be reported in constant time. We say that a pair of points $(\mathbf{x}, \mathbf{y})$ is contained in a filter (Q, U) if $\mathbf{x} \in Q$ and $\mathbf{y} \in U$.

LEMMA 2.2. (TENSORING) *Let $\mathcal{F}$ be a filter family that is $(r, cr, p_1, p_2, p_q, p_u)$-sensitive in (X, D). Let τ be a positive integer and let $\mathbf{F}$ denote a collection of $m = \lceil \tau/p_1 \rceil$ independently sampled filters from $\mathcal{F}$. Then the collection $\mathbf{F}^{\otimes\tau}$ of $\binom{m}{\tau}$ filters has the following properties:*

- *If $(\mathbf{x}, \mathbf{y})$ have distance at most r then with probability at least $1/2$ there exists a filter in $\mathbf{F}^{\otimes\tau}$ containing $(\mathbf{x}, \mathbf{y})$.*
- *If $(\mathbf{x}, \mathbf{y})$ have distance greater than cr then the expected number of filters in $\mathbf{F}^{\otimes\tau}$ containing $(\mathbf{x}, \mathbf{y})$ is at most $p_2^\tau \binom{m}{\tau}$.*
- *In expectation, a point $\mathbf{x}$ is contained in at most $p_q^\tau \binom{m}{\tau}$ query filters and at most $p_u^\tau \binom{m}{\tau}$ update filters in $\mathbf{F}^{\otimes\tau}$.*
- *The evaluation time and space complexity of $\mathbf{F}^{\otimes\tau}$ is dominated by the time it takes to evaluate and store m filters from $\mathcal{F}$.*

Proof. To prove the first property we note that there exists a filter in $\mathbf{F}^{\otimes\tau}$ containing $(\mathbf{x}, \mathbf{y})$ if at least τ filters in $\mathbf{F}$ contain $(\mathbf{x}, \mathbf{y})$. The binomial distribution has the property that the median is at least as great as the mean rounded down [28]. By the choice of m we have that the expected number of filters in $\mathbf{F}$ containing $(\mathbf{x}, \mathbf{y})$ is at least τ and the result follows. The second and third properties follow from the linearity of expectation and the fourth is trivial.

2.1 The LSF data structure We will introduce a dynamic data structure that solves the (r, cr)-near neighbor problem on a set of points $P \subseteq X$. The data structure has access to a $(r, cr, p_1, p_2, p_q, p_u)$-sensitive filter family $\mathcal{F}$ in the sense that it knows the parameters of the family and is able to sample, store, and evaluate filters from $\mathcal{F}$ in time $dn^{o(1)}$.

The data structure supports an initialization operation that initializes a collection of filters $\mathbf{F}$ where for every filter we maintain a (possibly empty) set of points from X. After initialization the data structure supports three operations: INSERT, DELETE, and QUERY. The INSERT (DELETE) operation takes as input a point $\mathbf{x} \in X$ and adds (removes) the point from the set of points associated with each update filter in $\mathbf{F}$ that contains $\mathbf{x}$. The QUERY operation takes as input a point $\mathbf{x} \in X$. For each query filter in $\mathbf{F}$ that contains $\mathbf{x}$ we proceed by computing the dissimilarity $D(\mathbf{x}, \mathbf{y})$ to every point $\mathbf{y}$ associated with the filter. If a point $\mathbf{y}$ satisfying $D(\mathbf{x}, \mathbf{y}) \leq cr$ is encountered, then $\mathbf{y}$ is returned and the query algorithm terminates. If no such point is found, the query algorithm returns a special symbol "$\varnothing$" and terminates.

The data structure will combine the powering and tensoring techniques in order to simulate the collection of filters $\mathbf{F}$ from two smaller collections: $\mathbf{F}_1$ consisting of m_1 filters from $\mathcal{F}^{\kappa_1}$ and $\mathbf{F}_2$ consisting of m_2 filters from $\mathcal{F}^{\kappa_2}$. The collection of simulated filters $\mathbf{F}$ is formed by taking all filters $(Q_1 \cap Q_2, U_1 \cap U_2)$ where (Q_1, U_1) is a member of $\mathbf{F}_1^{\otimes\tau}$ and (Q_2, U_2) is a member of $\mathbf{F}_2$. It is due to the integer constraints on the parameter τ in the tensoring technique and the parameter κ in the powering technique that we simulate our filters from two underlying collections instead of just one. This gives us more freedom to hit a target level of amplification of the simulated filters which in turn makes it possible for the framework to support efficient solutions for a wider range of parameters of LSF families.

The initialization operation takes $\mathcal{F}$ and parameters $m_1, \kappa_1, \tau, m_2, \kappa_2$ and samples and stores $\mathbf{F}_1$ and $\mathbf{F}_2$. The filter evaluation algorithm used by the insert, delete, and query operation takes a point $\mathbf{x} \in X$ and computes for $\mathbf{F}_1$ and $\mathbf{F}_2$, depending on the operation, the list of update or query filters containing $\mathbf{x}$. From these lists we are able to generate the list of filters in $\mathbf{F}$ containing $\mathbf{x}$.

Setting the parameters of the data structure to guarantee correctness while balancing the contribution to the query time from the filter evaluation algorithm, the number of filters containing the query point, and the number of distant points examined, we obtain a partially dynamic data structure that solves the (r, cr)-near neighbor problem with failure probability $\delta \leq 1/2 + 1/e$. Using a standard dynamization technique by Overmars and Leeuwen [42, Thm. 1] we obtain a fully dynamic data structure resulting in Theorem 1.1. The details of the proof have been deferred to Appendix A.

3 Gaussian filters on the unit sphere

In this section we show properties of a family of filters for the unit sphere $\mathbb{S}^d$ under inner product similarity.

Later we will show how to make use of this family to solve the near neighbor problem in other spaces, including ℓ_s for $0 < s \leq 2$.

LEMMA 3.1. *For every choice of $0 \leq \beta < \alpha < 1$, $\lambda \in [-1, 1]$, and $t > 0$ let $\mathcal{G}$ denote the family of filters defined as follows: we sample a filter (Q, U) from $\mathcal{G}$ by sampling $\mathbf{z} \sim \mathcal{N}^d(0, 1)$ and setting*

$$Q = \{\mathbf{x} \in \mathbb{R}^d \mid \langle \mathbf{x}, \mathbf{z} \rangle > \alpha^\lambda t\}, \tag{3.2}$$

$$U = \{\mathbf{x} \in \mathbb{R}^d \mid \langle \mathbf{x}, \mathbf{z} \rangle > t\}. \tag{3.3}$$

Then $\mathcal{G}$ is locality-sensitive on the unit sphere under inner product similarity with exponents

$$\rho_q \leq \left(\frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2} + \frac{\ln(2\pi(1+t/\alpha)^2)}{t^2/2} \right) \Big/ \frac{(1-\alpha^\lambda \beta)^2}{1-\beta^2},$$

$$\rho_u \leq \left(\frac{(\alpha^\lambda - \alpha)^2}{1-\alpha^2} + \frac{\ln(2\pi(1+t/\alpha)^2)}{t^2/2} \right) \Big/ \frac{(1-\alpha^\lambda \beta)^2}{1-\beta^2}.$$

Laarhoven's filter family [32] is identical to $\mathcal{G}$ except that he normalizes the projection vectors $\mathbf{z}$ to have unit length. The properties of $\mathcal{G}$ can easily be verified with a simple back-of-the-envelope analysis using two facts: First, for a standard normal random variable Z we have that $\Pr[Z > t] \approx e^{-t^2/2}$. Secondly, the invariance of Gaussian projections $\langle \mathbf{x}, \mathbf{z} \rangle$ to rotations, allowing us to analyze the projection of arbitrary points $\mathbf{x}, \mathbf{y} \in \mathbb{S}^d$ with inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \alpha$ in a two-dimensional setting $\mathbf{x} = (1, 0)$ and $\mathbf{y} = (\alpha, \sqrt{1-\alpha^2})$ without any loss of generality. The proof of Lemma 3.1 as well as the proof of Theorem 1.2 has been deferred to Appendix B.

4 Space-time tradeoffs under kernel similarity

In this section we will show how to combine the Gaussian filters for the unit sphere with kernel approximation techniques in order to solve the (α, β)-similarity problem over $(\mathbb{R}^d, S)$ for the class of similarity measures of the form $S(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$ where $k \colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a real-valued characteristic function [53]. For this class of functions there exists a feature map ψ into a (possibly infinite-dimensional) dot product space such that $k(\mathbf{x}, \mathbf{y}) = \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle$. Through an elegant combination of Bochner's Theorem and Euler's Theorem, detailed in Appendix C, Rahimi and Recht [48] show how to construct approximate feature maps, i.e., for every k we can construct a function v with the property that $\langle v(\mathbf{x}), v(\mathbf{y}) \rangle \approx \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle = k(\mathbf{x} - \mathbf{y})$. We state a variant of their result for a mapping onto the unit sphere.

LEMMA 4.1. *For every real-valued characteristic function k and every positive integer l there exists a family of functions $\mathcal{V} \subseteq \{v \mid v \colon \mathbb{R}^d \to \mathbb{S}^l\}$ such that for every $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\varepsilon > 0$ we have that*

$$\Pr_{v \sim \mathcal{V}}[|\langle v(\mathbf{x}), v(\mathbf{y}) \rangle - k(\mathbf{x}, \mathbf{y})| \geq \varepsilon] \leq e^{-\Omega(l\varepsilon^2)}. \tag{4.4}$$

Theorem C.3 in Appendix C shows that Theorem 1.2 holds with the space $(\mathbb{S}^d, \langle \cdot, \cdot \rangle)$ replaced by $(\mathbb{R}^d, k)$.

4.1 Tradeoffs in ℓ_s^d-space Consider the (r, cr)-near neighbor problem in ℓ_s^d for $0 < s \leq 2$. We solve this problem by first applying the approximate feature map from Lemma 4.1 for the characteristic function of a standard s-stable distribution [58], mapping the data onto the unit sphere, and then applying our solution from Theorem 1.2 to solve the appropriate (α, β)-similarity problem on the unit sphere. The characteristic functions of s-stable distributions take the following form:

LEMMA 4.2. (LÉVY [33]) *For every positive integer d and $0 < s \leq 2$ there exists a characteristic function $k \colon \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ of the form*

$$k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|_s^s}. \tag{4.5}$$

A result by Chambers et al. [17] shows how to sample efficiently from an s-stable distributions.

To sketch the proof of Theorem 1.3 we proceed by upper bounding the exponents ρ_q, ρ_u from Theorem 1.2 when applying Lemma 4.1 to get $\alpha \geq e^{-r^s} - \varepsilon$ and $\beta \leq e^{-c^s r^s} - \varepsilon$. We make use of the following standard fact (see e.g. [50]) that can be derived from the Taylor expansion of the exponential function: for $x \geq 0$ it holds that $1 - x \leq e^{-x} \leq 1 - x + x^2/2$. Scaling the data points such that $r^s = o(1)$ and inserting the above values of $\alpha \approx 1 - r^s$ and $\beta \approx 1 - c^s r^s$ into the expressions for ρ_q, ρ_u in Lemma 3.1 we can set parameters t and l such that Theorem 1.3 holds.

5 Lower bounds

We begin by stating the lower bound on the LSH exponent $\rho = \log(1/p)/\log(1/q)$ by O'Donnell et al. [43].

THEOREM 5.1. (O'DONNELL ET AL. [43]) *Fix $d \in \mathbb{N}$, $1 < c < \infty$, $0 < s < \infty$ and $0 < q < 1$. Then for a certain choice of $r = \omega_d(1)$ and under the assumption that $q \geq 2^{-o(d)}$ we have that every (r, cr, p, q)-sensitive family of hash functions for ℓ_s^d must satisfy*

$$\rho = \frac{\log(1/p)}{\log(1/q)} \geq \frac{1}{c^s} - o_d(1). \tag{5.6}$$

The following lemma shows how to use a filter family $\mathcal{F}$ to construct a hash family $\mathcal{H}$.

LEMMA 5.1. *Given a symmetric family of filters that is $(r, cr, p_1, p_2, p_q, p_u)$-sensitive in (X, D) we can construct a $(r, cr, p_1/(2p_q), p_2/p_q)$-sensitive family of hash functions in (X, D).*

Proof. Given the filter family $\mathcal{F}$ we sample a random function h from the hash family $\mathcal{H}$ taking an infinite sequence of independently sampled filters $(F_i)_{i=0}^{\infty}$ from $\mathcal{F}$ and setting $h(\mathbf{x}) = \min\{i \mid \mathbf{x} \in F_i\}$. The probability of collision is given by

$$\Pr_{h\sim\mathcal{H}}[h(\mathbf{x}) = h(\mathbf{y})] = \frac{\Pr_{F\sim\mathcal{F}}[\mathbf{x} \in F \wedge \mathbf{y} \in F]}{\Pr_{F\sim\mathcal{F}}[\mathbf{x} \in F \vee \mathbf{y} \in F]} \tag{5.7}$$

and the result follows from the properties of $\mathcal{F}$.

If the LSH family in Lemma 5.1 had $p = p_1/p_q$ and $q = p_2/p_q$ then the lower bound would follow immediately. We apply the powering technique from Lemma 2.1 to the underlying filter family in order make the factor 2 in $p_1/(2p_q)$ disappear in the statement of ρ as d tends to infinity.

THEOREM 1.4. *Every symmetric $(r, cr, p_1, p_2, p_q, p_u)$-sensitive filter family $\mathcal{F}$ for ℓ_s^d must satisfy the lower bound of Theorem 5.1 with $p = p_1/p_q$ and $q = p_2/p_q$.*

Proof. Given a family $\mathcal{F}$ that satisfies the requirements from Theorem 5.1 there exists an integer $\kappa = \omega_d(1)$ such the hash family $\mathcal{H}$ that results from applying Lemma 5.1 to the powered family $\mathcal{F}^\kappa$ also satisfies the requirements from Theorem 5.1. The constructed family $\mathcal{H}$ is (r, cr, p, q)-sensitive for $p = (1/2) \cdot (p_1/p_q)^\kappa$ and $q = (p_2/p_q)^\kappa$. By our choice of κ we have that $\log(1/p)/\log(1/q) = \log(p_q/p_1)/\log(p_q/p_2) + o_d(1)$ and the lower bound on $\log(1/p)/\log(1/q)$ from Theorem 5.1 applies.

5.1 Asymmetric lower bound The lower bound is based on an isoperimetric-type inequality that holds for randomly correlated points in Hamming space. We say that the pair of points $(\mathbf{x}, \mathbf{y})$ is α-correlated if $\mathbf{x}$ is a random point in $\{0,1\}^d$ and $\mathbf{y}$ is formed by taking $\mathbf{x}$ and independently flipping each bit with probability $(1-\alpha)/2$. We are now ready to state O'Donnell's generalized small-set expansion theorem. Notince the similarity to the value of p_1 for the Gaussian filter family described in Section 3 and Appendix B.

LEMMA 5.2. ([41, P. 285]) *For every $0 \le \alpha < 1$, $-1 \le \lambda \le 1$, and $Q, U \subseteq \{0,1\}^d$ satisfying that $|Q|/2^d = (|U|/2^d)^{\alpha^{2\lambda}}$ we have*

$$\Pr_{\substack{(\mathbf{x},\mathbf{y})\\ \alpha\text{-correlated}}}[\mathbf{x} \in Q, \mathbf{y} \in U] \le (|U|/2^d)^{\frac{1+\alpha^{2\lambda}-2\alpha^{1+\lambda}}{1-\alpha^2}}. \tag{5.8}$$

The argument for the lower bound assumes a regular $(r, cr, p_1, p_2, p_q, p_u)$-sensitive filter family $\mathcal{F}$ for Hamming space where we set $r = (1-\alpha)d/2$ and $cr = (1-\beta)d/2$ for some choice of $0 < \beta < \alpha < 1$. We then proceed by deriving constraints on p_1, p_2, p_q, p_u, and minimize ρ_q and ρ_u subject to those constrains. The proof of Theorem 1.5 is provided in Appendix D.

THEOREM 1.5. *Fix $0 < \beta < \alpha < 1$. Then for every regular $((1-\alpha)d/2, (1-\beta)d/2, p_1, p_2, p_q, p_u)$-sensitive filter family in d-dimensional Hamming space with and $|Q|/2^d = (|U|/2^d)^{\alpha^{2\lambda}}$ where λ satisfies $\alpha + 2\sqrt{\ln(d)/d} \le \alpha^\lambda \le 1/(\alpha - 2\sqrt{\ln(d)/d})$ it must hold that*

$$\rho_q = \frac{\log(p_q/p_1)}{\log(p_q/p_2)} \ge \frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2} - o_d(1),$$
$$\rho_u = \frac{\log(p_u/p_1)}{\log(p_q/p_2)} \ge \frac{(\alpha^\lambda - \alpha)^2}{1-\alpha^2} - o_d(1)$$

when p_q is set to minimize ρ_q and we assume that $|U|/2^d \ge 2^{-o_d(1)}$.

6 Open problems

An important open problem is to find simple and practical data-dependent solutions to the (r, cr)-near neighbor problem. Current solutions, the Gaussian filters in this paper included, suffer from $o(1)$ terms in the exponents that decrease very slowly in n. A lower bound for the unit sphere by Andoni et al. [6] indicates that this might be unavoidable.

Another interesting open problem is finding the shape of provably exactly optimal filters in different spaces. In the random data setting in Hamming space, this problem boils down to maximizing the number of pairs of points below a certain distance threshold that is contained in a subset of the space of a certain size. This is a fundamental problem in combinatorics that has been studied by among others [29], but a complete answer remains elusive. The LSH and LSF lower bounds [39, 43, 13], along with classical isoperimetric inequalities such as Harper's Theorem and more recent work summarized in the book by O'Donnell [41] hints that the answer is somewhere between a subcube and a generalized sphere.

A recent result by Chierichetti and Kumar [19] characterizes the set of transformations of LSH-able similarity measures as the set of probability-generating functions. This seems to have deep connections to result of this paper that uses characteristic functions that allow well-known kernel transformations. It seems possible that this paper can be viewed as a semi-explicit construction of their result, or that their result can described as an application of Bochner's Theorem.

Acknowledgment

I would like to thank Rasmus Pagh for suggesting the application of Rahimi & Recht's result [48] and the MinHash-like [16] connection between LSF and LSH used in Theorem 1.4. I would also like to thank Gregory Valiant and Udi Wieder for useful discussions about locality-sensitive filtering and the analysis of boolean functions during my stay at Stanford. Finally, I would like to thank the Scalable Similarity Search group at the IT University of Copenhagen for feedback during the writing process, and in particular Martin Aumüller for pointing out the importance of a general framework for locality-sensitive filtering with space-time tradeoffs.

A Framework

We state a version of Theorem 1.1 where the parameters of the filter family are allowed to depend on n.

THEOREM 1.1. *Suppose we have access to a filter family that is $(r, cr, p_1, p_2, p_q, p_u)$-sensitive. Then we can construct a fully dynamic data structure that solves the (r, cr)-near neighbor problem. Assume that $1/p_1$, $1/\log(p_q/p_2)$, and $\exp(\log(1/p_1)/\log(\min(p_q, p_u)/p_1))$ are $n^{o(1)}$, then the data structure has*

- *query time $dn^{\rho_q+o(1)}$,*
- *update time $n^{\rho_u+o(1)} + dn^{o(1)}$,*
- *space usage $n^{1+\rho_u+o(1)} + dn + dn^{o(1)}$*

where

$$\rho_q = \frac{\log p_q/p_1}{\log p_q/p_2}, \quad \rho_u = \frac{\log p_u/p_1}{\log p_q/p_2}. \tag{A.1}$$

To prove Theorem 1.1, we begin by setting the parameters mentioned in the description of the LSF data structure in Section 2.1.

$$\kappa_1 = \left\lceil \frac{\min(\rho_q, \rho_u)\log n}{\log(1/p_1)} \right\rceil \tag{A.2}$$

$$\tau = \left\lfloor \frac{\log n}{\kappa_1 \log(p_q/p_2)} \right\rfloor \leq \frac{\log(1/p_1)}{\log(\min(p_q, p_u)/p_1)} \tag{A.3}$$

$$m_1 = \lceil \tau/p_1^{\kappa_1} \rceil \tag{A.4}$$

$$\kappa_2 = \max(0, \lceil \log(n)/\log(p_q/p_2) \rceil - \tau\kappa_1) \tag{A.5}$$

$$m_2 = \lceil 1/p_1^{\kappa_2} \rceil \tag{A.6}$$

We will now briefly explain the reasoning behind the parameter settings. Begin by observing that the powering and tensoring techniques both amplify the filters from $\mathcal{F}$. Let $m = \binom{m_1}{\tau} \cdot m_2$ denote the number of simulated filters in our collection $\mathbf{F}$ and let $a = \tau\kappa_1 + \kappa_2$ be an integer denoting the number of times each filter has been amplified. Ignoring the time it takes to evaluate the filters, the query time is determined by the sum of the number of filters that contain a query point and the number of distant points associated with those filters that the query algorithm inspects. The expected number of activated filters is given by mp_q^a while the worst case expected number of distant points to be inspected by the query algorithm is given by nmp_2^a. Balancing the contribution to the query time from these two effects (ignoring the $O(d)$ factor from distance computations) results in a target value of $a = \lceil \log(n)/\log(p_q/p_2) \rceil$. Compared to having an oracle that is able to list the filters from a collection that contains a point, there is a small loss in efficiency from using the tensoring technique due to the increase in the number of filters required to guarantee correctness. The parameters of the LSF data structure are therefore set to minimize the use of tensoring such that the time spent evaluating our collection of filters roughly matches the minimum of the query and update time.

Consider the initialization operation of the LSF data structure with the parameters setting from above. We have that $\kappa_2 \leq \kappa_1$ implying that $m_2 = O(m_1)$. The initialization time and the space usage of the data structure prior to any insertions is dominated by the time and space used to sample and store the filters in $\mathbf{F}_1$. By the assumption that a filter from $\mathcal{F}$ can be sampled in $O(d)$ operations and stored using $O(d)$ words, we get a space and time bound on the initialization operation of

$$O(d\kappa_1 m_1) = O\left(dn^{\min(\rho_q, \rho_u)} \frac{p_1 \log(n)}{\log(p_q/p_2)}\right). \tag{A.7}$$

Importantly, this bound also holds for the running time of the filter evaluation algorithm, that is, the preprocessing time required for constant time generation of the next element in the list of filters in $\mathbf{F}$ containing a point. In the following analysis of the update and query time we will temporarily ignore the running time of the filter evaluation algorithm.

The expected time to insert or delete a point is dominated by the number of update filters in $\mathbf{F}$ that contains it. The probability that a particular update filter in $\mathbf{F}$ contains a point is given by p_u^a. Using a standard upper bound on the binomial coefficient we get that $m = O(e^\tau/p_1^a)$ resulting in an expected update time of

$$O(mp_u^a + d) = O(n^{\rho_u}(p_u/p_1)e^\tau + d). \tag{A.8}$$

In the worst case where every data point is at distance greater than cr from the query point and has collision probablity p_2 the expected query time can be upper bounded by

$$O(mp_q^a + dnmp_2^a) = O(n^{\rho_q} e^\tau (p_q/p_1 + d)). \tag{A.9}$$

With respect to the correctness of the query algorithm, if a near neighbor $\mathbf{y}$ to the query point $\mathbf{x}$ exists in P, then it is found by the query algorithm if $(\mathbf{x}, \mathbf{y})$ is contained in a filter in $\mathbf{F}_1^{\otimes\tau}$ as well as in a filter in $\mathbf{F}_2$. By Lemma 2.2 the first event happens with probability at least $1/2$ and by the choice of m_2, the second event happens with probability at least $1-(1-p_1^{\kappa_2})^{p_1^{-\kappa_2}} \geq 1 - 1/e$. From the independence between $\mathbf{F}_1$ and $\mathbf{F}_2$ we can upper bound the failure probability $\delta \leq (1/2)(1+1/e)$. This completes the proof of Theorem 1.1.

B Gaussian filters

In this section we upper and lower bound the probability mass in the tail of the bivariate standard normal distribution when the correlation between the two standard normals is at most β (upper bound) or at least α (lower bound). We make use of the following upper and lower bounds on the univariate standard normal as well as an upper bound for the multivariate case.

LEMMA B.1. (FOLLOWS SZAREK & WERNER [52]) *Let Z be a standard normal random variable. Then, for every $t \geq 0$ we have that*

$$\frac{1}{\sqrt{2\pi}}\frac{1}{t+1}e^{-t^2/2} \leq \Pr[Z \geq t] \leq \frac{1}{\sqrt{\pi}}\frac{1}{t+1}e^{-t^2/2}.$$

LEMMA B.2. (LU & LI [34]) *Let $\mathbf{z}$ be a d-dimensional vector of i.i.d. standard normal random variables and let $D \subset \mathbb{R}^d$ be a closed convex domain that does not contain the origin. Let Δ denote the Euclidean distance to the unique closest point in D, then we have that*

$$\Pr[\mathbf{z} \in D] \leq e^{-\Delta^2/2}. \tag{B.10}$$

LEMMA B.3. (TAIL UPPER BOUND) *For $\alpha, \lambda, t, \beta$ satisfying $0 < \alpha < 1$, $-1 \leq \lambda \leq 1$, $t > 0$, and $-1 < \beta < \alpha$ every pair of standard normal random variables (X, Y) with correlation $\beta' \leq \beta$ satisfies*

$$\Pr[X \geq t \wedge Y \geq \alpha^\lambda t] \leq e^{-\Delta^2/2} \tag{B.11}$$

where $\Delta^2 = (1 + \frac{(\alpha^\lambda - \beta)^2}{1-\beta^2})t^2$.

Proof. For $\beta' = -1$ the result is trivial. For values of β' in the range $-1 < \beta' \leq \beta$ we use the 2-stability of the normal distribution to analyze a tail bound for (X, Y) in terms of a Gaussian projection vector $\mathbf{z} = (Z_1, Z_2)$ applied to unit vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$. That is, we can define $X = \langle \mathbf{z}, \mathbf{x} \rangle$ and $Y = \langle \mathbf{z}, \mathbf{x} \rangle$ for some appropriate choice of $\mathbf{x}$ and $\mathbf{y}$. Without loss of generality we set $\mathbf{x} = (1, 0)$ and note that for $\mathbb{E}[XY] = \beta'$ we must have that $\mathbf{y} = (\beta', \sqrt{1-\beta'^2})$. If we consider the region of $\mathbb{R}^2$ where $\mathbf{z}$ satisfies $X \geq t \wedge Y \geq \alpha^\lambda t$ we get a closed domain D defined by $\mathbf{z} = (Z_1, Z_2)$ such that $Z_1 \geq t$ and $Z_2 \geq (\alpha^\lambda t - \beta' Z_1)/(\sqrt{1-\beta'^2})$. The squared Euclidean distance from the origin to the closest point in D at least Δ^2 as can be seen by the fact that Δ^2 decreasing in β. Combining this observation with Lemma B.2 we get the desired result.

LEMMA B.4. (TAIL LOWER BOUND) *For α, λ, t satisfying $0 < \alpha < 1$, $-1 \leq \lambda \leq 1$, and $t > 0$ every pair of standard normal random variables (X, Y) with correlation $\alpha' \geq \alpha$ satisfies*

$$\Pr[X \geq t \wedge Y \geq \alpha^\lambda t] \geq \frac{e^{-\Delta^2/2}}{2\pi(1+t/\alpha)^2} \tag{B.12}$$

where $\Delta^2 = (1 + \frac{(\alpha^\lambda - \alpha)^2}{1-\alpha^2})t^2$.

Proof. For $\alpha' = 1$ the result follows directly from Lemma B.1. For $\alpha' < 1$ we use the trick from the proof of Lemma B.3 and define $X = \langle \mathbf{z}, \mathbf{x} \rangle$ and $Y = \langle \mathbf{z}, \mathbf{x} \rangle$ where $\mathbf{x} = (1, 0)$ and $\mathbf{y} = (\alpha, \sqrt{1-\alpha^2})$ and $\mathbf{z} = (Z_1, Z_2)$ is a vector of two i.i.d. standard normal random variables. This allows us to rewrite the probability as follows:

$$\begin{aligned}
&\Pr[Z_1 \geq t \wedge \alpha Z_1 + \sqrt{1-\alpha^2} Z_2 \geq \alpha^\lambda t] \\
&= \Pr[Z_1 \geq t] \Pr[\alpha Z_1 + \sqrt{1-\alpha^2} Z_2 \geq \alpha^\lambda t \mid Z_1 \geq t] \\
&\geq \Pr[Z_1 \geq t] \Pr[\alpha t + \sqrt{1-\alpha^2} Z_2 \geq \alpha^\lambda t]
\end{aligned}$$

By the restrictions on α and λ we have that $(\alpha^\lambda - \alpha)t/\sqrt{1-\alpha^2} \leq t/\alpha$. The result follows from applying the lower bound from Lemma B.1 and noting that the bound is increasing in α.

B.1 Space-time tradeoffs on the unit sphere

Summarizing the bound from the previous section, the family $\mathcal{G}$ from Lemma 3.1 satisfies that

$$p_1 \geq \frac{e^{-(1+\frac{(\alpha^\lambda-\alpha)^2}{1-\alpha^2})t^2/2}}{2\pi(1+t/\alpha)^2} \tag{B.13}$$

$$p_2 \leq e^{-(1+\frac{(\alpha^\lambda-\beta)^2}{1-\beta^2})t^2/2} \tag{B.14}$$

$$p_q \leq e^{-\alpha^{2\lambda}t^2/2} \tag{B.15}$$

$$p_u \leq e^{-t^2/2}. \tag{B.16}$$

We combine the Gaussian filters with Theorem 1.1 to show that we can solve the (α, β)-similarity problem efficiently for the full range of space/time tradeoffs, even when α, β are allowed to depend on n, as long as the gap $\alpha - \beta$ is not too small.

THEOREM 1.2. *For every choice of $0 \leq \beta < \alpha < 1$ and $\lambda \in [-1, 1]$ we can construct a fully dynamic data structure that solves the (α, β)-similarity problem in*

$(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$. *Suppose that* $\alpha - \beta \geq (\ln n)^{-\zeta}$ *for some constant* $\zeta < 1/2$, *that satisfies the guarantees from Theorem 1.1 with exponents* $\rho_q = \frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2} \Big/ \frac{(1-\alpha^{\lambda}\beta)^2}{1-\beta^2}$ *and* $\rho_u = \frac{(\alpha^{\lambda}-\alpha)^2}{1-\alpha^2} \Big/ \frac{(1-\alpha^{\lambda}\beta)^2}{1-\beta^2}$.

Proof. Assuming that $\alpha - \beta \geq (\ln n)^{-\zeta}$ there exists a constant $\varepsilon > 0$ where by setting the parameter t of $\mathcal{G}$ such that $t^2/2 = \frac{1-\beta^2}{(1-\alpha^{\lambda}\beta)^2}(\ln n)^{\varepsilon}$ the family of filters satisfies the assumptions in Theorem 1.1 while guaranteeing that the second term in ρ_q and ρ_u from Lemma 3.1 are $o(1)$.

REMARK 1. *Theorem 1.2 aims for simplicity and generality while allowing* α *and* β *to depend on* n. *For specific values of* α, β, λ *it is easy to find better bounds on the probabilties (e.g. the bounds by Savage [50]) and to adjust* t *in Lemma 3.1 to avoid powering (setting* $\kappa_1 = 1, \kappa_2 = 0$*) in the LSF framework.*

C Approximate feature maps, characteristic functions, and Bochner's Theorem

We begin by defining what a characteristic function is and listing some properties that are useful for our application. More information about characteristic functions can be found in the books by Lukacs [35] and Ushakov [53].

LEMMA C.1. ([35, 53]) *Let* Z *denote a random variable with distribution function* μ. *Then the characteristic function* $k(\Delta)$ *of* Z *is defined as*

$$k(\Delta) = \int_{-\infty}^{\infty} \mu(t) e^{i\Delta t} dt \tag{C.17}$$

and it has the following properties:

- *A distribution function is symmetric if and only if its characteristic function is real and even.*
- *Every characteristic function* $k(\Delta)$ *is uniformly continuous, has* $k(0) = 1$, *and* $|k(\Delta)| \leq 1$ *for all real* Δ.
- *Suppose that* $k(\Delta)$ *denotes the characteristic function of an absolutely continuous distribution then* $\lim_{\Delta\to\infty} |k(\Delta)| = 0$.
- *Let* X *and* Y *be independent random variables with characteristic functions* k_X *and* k_Y. *Then the characteristic function of* $Z = (X, Y)$ *is given by* $k(x, y) = k_X(x)k_Y(y)$.

Bochner's Theorem reveals the relation between characteristic functions and the class of real-valued functions $k(\mathbf{x}, \mathbf{y})$ that admit a feature space representation $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$

THEOREM C.1. (BOCHNER'S THEOREM [49]) *A function* $k : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ *is positive definite if and only if it can be written on the form*

$$k(\mathbf{x}, \mathbf{y}) = \int_{\mathbb{R}^d} \mu(\mathbf{z}) e^{i\langle \mathbf{z}, \mathbf{x}-\mathbf{y} \rangle} d\mathbf{z} \tag{C.18}$$

where μ *is the probability density function of a symmetric distribution.*

Rahimi & Recht's [48] family of approximate feature maps $\mathcal{V}$ is constructed from Bochner's Theorem by making use of Euler's Theorem as follows:

$$\begin{aligned} k(\mathbf{x}, \mathbf{y}) &= \int_{\mathbb{R}^d} \mu(\mathbf{z}) e^{i\langle \mathbf{z}, \mathbf{x}-\mathbf{y} \rangle} d\mathbf{z} \\ &= \int_{\mathbb{R}^d} \mu(\mathbf{z})(\cos(\langle \mathbf{z}, \mathbf{x}-\mathbf{y} \rangle) + i \sin(\langle \mathbf{z}, \mathbf{x}-\mathbf{y} \rangle)) d\mathbf{z} \\ &= \mathop{\mathbb{E}}_{\mathbf{z}}[\cos(\langle \mathbf{z}, \mathbf{x}-\mathbf{y} \rangle)] \\ &= \mathop{\mathbb{E}}_{\mathbf{z},b}[\cos(\langle \mathbf{z}, \mathbf{x}-\mathbf{y} \rangle) + \cos(\langle \mathbf{z}, \mathbf{x} \rangle + \langle \mathbf{z}, \mathbf{y} \rangle + 2b)] \\ &= 2 \mathop{\mathbb{E}}_{\mathbf{z},b}[\cos(\langle \mathbf{z}, \mathbf{x} \rangle + b) \cdot \cos(\langle \mathbf{z}, \mathbf{y} \rangle + b)]. \end{aligned}$$

Where the third equality makes use of the fact that $k(\mathbf{x}, \mathbf{y})$ is real-valued to remove the complex part of the integral and the fifth equality uses that $2\cos(x)\cos(y) = \cos(x + y) + \cos(x - y)$.

Now that we have an approximate feature map onto the sphere for the class of shift-invariant kernels, we will take a closer look at what functions this class contains, and what their applications are for similarity search. Given an arbitrary similarity function, we would like to be able to determine whether it is indeed a characteristic function. Unfortunately, there are no known simple techniques for answering this question in general. However, the machine learning literature contains many applications of different shift-invariant kernels [51] and many common distributions have real characteristic functions (see Appendix B in [53] for a long list of examples). Characteristic functions are also well studied from a mathematical perspective [35, 53], and a number of different necessary and sufficient conditions are known. A classical result by Pólya [47] gives simple sufficient conditions for a function to be a characteristic function. Through the vectorization property from Lemma C.1, Pólya's conditions directly imply the existence of a large class of similarity measures on $\mathbb{R}^d$ that can fit into the above framework.

THEOREM C.2. (PÓLYA [47]) *Every even continuous function* $k : \mathbb{R} \to \mathbb{R}$ *satisfying the properties*

- $k(0) = 1$
- $\lim_{\Delta\to\infty} k(\Delta) = 0$

- $k(\Delta)$ *is convex for* $\Delta > 0$

is a characteristic function.

Based on the results of Section 4.1 one could hope for the existence of characteristic functions of the form $k(\Delta) = e^{-|\Delta|^s}$ for $s > 2$ but it is known that such functions cannot exist [15, Theorem D.8]. Furthermore, Marcinkiewicz [37] shows that a function of the form $k(\Delta) = \exp(-\operatorname{poly}(\Delta))$ cannot be a characteristic function if the degree of the polynomial is greater than two.

We state a more complete, constructive version of Lemma 4.1 as well as the proof here.

LEMMA C.2. *Let k be a real-valued characteristic function with associated distribution function μ and let l be a positive integer. Consider the family of functions $\mathcal{V} \subseteq \{v \mid v\colon \mathbb{R}^d \to \mathbb{S}^l\}$ where a randomly sampled function v is defined by, independently for $j = 1, \ldots, l$, sampling $\mathbf{z}$ from μ and b uniformly on $[0, 2\pi]$, letting $\hat{v}(\mathbf{x})_j = \sqrt{(2/l)}\cos(\langle \mathbf{z}, \mathbf{x}\rangle + b)$ and normalizing $v(\mathbf{x})_j = \frac{\hat{v}(\mathbf{x})}{\|\hat{v}(\mathbf{x})\|}$. The family $\mathcal{V}$ has the property that for every $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\varepsilon > 0$ we have that*

$$\Pr_{v\sim\mathcal{V}}[|\langle v(\mathbf{x}), v(\mathbf{y})\rangle - k(\mathbf{x}, \mathbf{y})| \geq \varepsilon] \leq 6e^{-l\varepsilon^2/128}. \tag{C.19}$$

Proof. Since $l \cdot \hat{v}(\mathbf{x})_j \hat{v}(\mathbf{y})_j$ is bounded between 2 and -2, and we have independence for different values of j, Hoeffding's inequality [25] can be applied to show that for every fixed pair of points $\mathbf{x}, \mathbf{y}$ and $\hat{\varepsilon} > 0$ it holds that

$$\Pr[|\langle \hat{v}(\mathbf{x}), \hat{v}(\mathbf{y})\rangle - k(\mathbf{x}, \mathbf{y})| \geq \hat{\varepsilon}] \leq 2e^{-l\hat{\varepsilon}^2/8}. \tag{C.20}$$

From the properties of characteristic functions we have that $k(\mathbf{x}, \mathbf{x}) = 1$ and $k(\mathbf{x}, \mathbf{y}) \leq 1$. The bound on the deviation of

$$\langle v(\mathbf{x}), v(\mathbf{y})\rangle = \frac{\langle \hat{v}(\mathbf{x}), \hat{v}(\mathbf{y})\rangle}{\sqrt{\langle \hat{v}(\mathbf{x}), \hat{v}(\mathbf{x})\rangle\langle \hat{v}(\mathbf{y}), \hat{v}(\mathbf{y})\rangle}} \tag{C.21}$$

from $k(\mathbf{x}, \mathbf{y})$ follows from setting $\hat{\varepsilon} = \varepsilon/4$ and using a union bound over the probabilities that the deviation of one of the inner products is too large.

Combining the approximate feature map onto the unit sphere with Theorem 1.2 we obtain the following:

THEOREM C.3. *Let $k\colon \mathbb{R}^d \to \mathbb{R}$ be a characteristic function and define the similarity measure $S(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$. Assume that we have access to samples from the distribution associated with k, then Theorem 1.2 holds with $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$ replaced by $(\mathbb{R}^d, S)$.*

Proof. According to Lemma C.2, we can set $l = n^{o(1)}$ to obtain a map $v\colon \mathbb{R}^d \to \mathbb{S}^l$ such that the the inner product on $\mathbb{S}^l$ preserves the pairwise similarity between $n^{O(1)}$ points with additive error $\varepsilon = o(1)$. This map has a space and time complexity of $O(dl) = dn^{o(1)}$. After applying v to the data we can solve the (α, β)-similarity problem on $(\mathbb{R}^d, k(\mathbf{x} - \mathbf{y}))$ by solving the $(\alpha - \varepsilon, \beta + \varepsilon)$-similarity problem on $(\mathbb{S}^d, \langle\cdot,\cdot\rangle)$. We can use Theorem 1.2 to construct a fully dynamic data structure for solving this problem, adjusting the parameter λ so that it continues to lie in the admissible range. The space and time complexities follow.

D Proof of Theorem 1.5

Consider $\rho_q = \frac{\log(p_q/p_1)}{\log(p_q/p_2)}$. Subject to the (implicit) LSF constraint that $p_q, p_u > p_1 > p_2 > 0$ we see that ρ_q is minimized by setting p_q, p_2 as small as possible and p_1 as large as possible. We will therefore derive lower bounds on p_q, p_2 and an upper bound on p_1. For every value of p_1 and p_2 we minimize ρ_q, ρ_u by choosing p_q as small as possible.

For a random point $\mathbf{x} \in \{0,1\}^d$ it must hold that $\Pr_{\mathcal{F}}[\mathbf{x} \in Q] = |Q|/2^d$. This implies the existence of a fixed point $\mathbf{y} \in \{0,1\}^d$ with the property that $\Pr_{\mathcal{F}}[\mathbf{y} \in Q] \geq |Q|/2^d$. A regular filter family must therefore satisfy that $p_q \geq |Q|/2^d$ and $p_u \geq |U|/2^d$. Let λ be defined as in Lemma 5.2 then by a similar argument we have that $p_2 \geq (U/2^d)^{1+\alpha^{2\lambda}}$.

In order to upper bound p_1 we make use of Lemma 5.2 together with the following lemma that follows directly from an application of Hoeffding's inequality [25].

LEMMA D.1. *For every $0 < \varepsilon < (1-\alpha)/2$ we have that*

$$\Pr_{\substack{(\mathbf{x},\mathbf{y})\\ \alpha+\varepsilon\text{-correlated}}}\left[\frac{1}{d}\sum_{i=1}^{d}(-1)^{\mathbf{x}_i}(-1)^{\mathbf{y}_i} \leq \alpha\right] \leq e^{-\varepsilon^2 d/2}.$$

In the following derivation, assume that α, ε satisfies $0 < \varepsilon < (1-\alpha)/2$, let $\mathbf{x}, \mathbf{y}$ denote randomly $(\alpha+\varepsilon)$-correlated vectors in $\{0,1\}^d$, and assume that $\alpha + \varepsilon \leq \alpha^\lambda \leq 1/(\alpha+\varepsilon)$, then

$$\begin{aligned}(|U|/2^d)^{\frac{1+\alpha^{2\lambda}-2\alpha^\lambda(\alpha+\varepsilon)}{1-(\alpha+\varepsilon)^2}} &\geq \Pr[\mathbf{x} \in Q, \mathbf{y} \in U]\\ &\geq \Pr[\mathbf{x} \in Q, \mathbf{y} \in U \mid \langle \mathbf{x}, \mathbf{y}\rangle \geq \alpha]\Pr[\langle \mathbf{x}, \mathbf{y}\rangle \geq \alpha]\\ &\geq p_1(1 - e^{-\varepsilon^2 d/2})\end{aligned}$$

Summarizing the bounds:

$$\begin{aligned}p_1 &\leq \frac{(|U|/2^d)^{\frac{1+\alpha^{2\lambda}-2\alpha^\lambda(\alpha+\varepsilon)}{1-\alpha^2}}}{1 - e^{-\varepsilon^2 d/2}}\\ p_2 &\geq (|U|/2^d)^{1+\alpha^{2\lambda}}\\ p_q &\geq |Q|/2^d\\ p_u &\geq |U|/2^d.\end{aligned}$$

When minimizing ρ_q we have that $\log(p_q/p_2) = -\log(|U|/2^d)$. Setting $\varepsilon = 2\sqrt{\ln(d)/d}$ results in $\log(1/p_1) \geq -\frac{1+\alpha^{2\lambda}-2\alpha^{\lambda}(\alpha+\varepsilon)}{1-\alpha^2}\log(|U|/2^d) - O(1/d^2)$. Putting things together:

$$\begin{aligned}\frac{\log(p_q/p_1)}{\log(p_q/p_2)} &\geq -\frac{\alpha^{2\lambda}\log(|U|/2^d)}{\log(|U|/2^d)} \\ &\quad + \frac{\frac{1+\alpha^{2\lambda}-2\alpha^{\lambda}(\alpha+\varepsilon)}{1-\alpha^2}\log(|U|/2^d) + O(1/d^2)}{\log(|U|/2^d)} \\ &= \frac{(1-\alpha^{1+\lambda})^2 - 2\alpha^{\lambda}\varepsilon}{1-\alpha^2} + \frac{O(1/d^2)}{\log(|U|/2^d)} \\ &= \frac{(1-\alpha^{1+\lambda})^2}{1-\alpha^2} - O(\sqrt{\log(d)/d}).\end{aligned}$$

The derivation of the lower bound for ρ_u is almost the same and the resulting expression is

$$\frac{\log(p_u/p_1)}{\log(p_q/p_2)} \geq \frac{(\alpha^{\lambda}-\alpha)^2}{1-\alpha^2} - O(\sqrt{\log(d)/d}). \tag{D.22}$$

E Comparison to Kapralov

Kapralov uses α to denote a parameter controlling the space-time tradeoff for his solution to the (r, cr)-near neighbor problem in Euclidean space. For every choice of tradeoff parameter $\alpha \in [0, 1]$, assuming that $c^2 \geq 3(1-\alpha)^2 - \alpha^2 + \varepsilon$ for arbitrarily small constant $\varepsilon > 0$, Kapralov [30] obtains query and update exponents

$$\rho_q = \frac{4(1-\alpha)^2}{c^2 + (1-\alpha)^2 - 3\alpha^2}, \tag{E.23}$$

$$\rho_u = \frac{4\alpha^2}{c^2 + (1-\alpha)^2 - 3\alpha^2}. \tag{E.24}$$

We convert Kapralov's notation to our own by setting $\lambda = 1 - 2\alpha$. To compare, Kapralov sets $\alpha = 0$ for near-linear space and we set $\lambda = 1$. We want to write Kapralov's exponents on the form

$$\rho_q = \frac{c^2(1+\lambda)^2}{(c^2+\lambda)^2 + x}, \quad \rho_u = \frac{c^2(1-\lambda)^2}{(c^2+\lambda)^2 + x} \tag{E.25}$$

for some x that we will proceed to derive. We have that $(1-\alpha)^2 = (1+\lambda)^2/4$ and $\alpha^2 = (1-\lambda)^2/4$. Multiplying the numerator and denominator in Kapralov's exponents by c^2 we can write Kapralov's exponents as

$$\rho_q = \frac{c^2(1+\lambda)^2}{c^4 + c^2(1+\lambda)^2/4 - 3c^2(1-\lambda)^2/4}, \tag{E.26}$$

$$\rho_u = \frac{c^2(1-\lambda)^2}{c^4 + c^2(1+\lambda)^2/4 - 3c^2(1-\lambda)^2/4}. \tag{E.27}$$

We have that

$$\begin{aligned}x &= c^4 + c^2(1+\lambda)^2/4 - 3c^2(1-\lambda)^2/4 - (c^2+\lambda)^2 \\ &= -c^2(1+\lambda^2)/2 - \lambda^2.\end{aligned}$$

For every choice of $\lambda \in [-1, 1]$, and under the assumption that $c^2 \geq (1+\lambda)^2/2 + \lambda + \varepsilon$ for an arbitrarily small constant $\varepsilon > 0$, this allows us to write Kapralov's exponents as

$$\rho_q = \frac{c^2(1+\lambda)^2}{(c^2+\lambda)^2 - c^2(1+\lambda^2)/2 - \lambda^2}, \tag{E.28}$$

$$\rho_u = \frac{c^2(1-\lambda)^2}{(c^2+\lambda)^2 - c^2(1+\lambda^2)/2 - \lambda^2}. \tag{E.29}$$

To compare Kapralov's result against our own for search in ℓ_s-spaces we consider the exponents from Theorem 1.3, ignoring additive $o(1)$ terms:

$$\rho_q = \frac{c^s(1+\lambda)^2}{(c^s+\lambda)^2}, \quad \rho_u = \frac{c^s(1-\lambda)^2}{(c^s+\lambda)^2}. \tag{E.30}$$

Setting $\lambda = 1$ we obtain a data structure that uses near-linear space and we get a query exponent $\rho_q = 16/25$ while Kapralov obtains an exponent of $\rho_q = 16/20$, ignoring $o(1)$ terms. At the other end of the tradeoff, setting $\lambda = -1$, we get a data structure with query time $n^{o(1)}$ and update exponent $\rho_u = 16/9$ while Kapralov gets an update exponent of $\rho_u = 4$, again ignoring additive $o(1)$ terms.

The assumption made by Kapralov that $c^2 \geq (1+\lambda)^2/2 + \lambda + \varepsilon$ means that in the case of a near-linear space data structure ($\lambda = 1$) sublinear query time can only be obtained for $c > \sqrt{3}$. In contrast, Theorem 1.3 gives sublinear query time for every constant $c > 1$.

F Details about dynamization and the model of computation

In order to obtain fully dynamic data structures we apply a powerful dynamization result of Overmars and Leeuwen [42] for decomposable searching problems. Their result allows us to turn a partially dynamic data structure into a fully dynamic data structure, supporting arbitrary sequences of queries and updates, at the cost of a constant factor in the space and running time guarantees. Suppose we have a partially dynamic data structure that solves the (r, cr)-near neighbor problem on a set of n points. By partially dynamic we mean that, after initialization on a set P of n points, the data structure supports $\Theta(n)$ updates without changing the query time by more than a constant factor. Let $T_q(n)$, $T_u(n)$, and $T_c(n)$ denote the query time, update time, and construction time of such a data structure containing n points. Then, by Theorem 1 of Overmars

and Leeuwen [42], there exists a fully dynamic version of the data structure with query time $O(T_q(n))$ and update time $O(T_u(n)+T_c(n)/n)$ that uses only a constant factor additional space. The data structures presented in this paper, as well as most related constructions from the literature, have the property that $T_c(n)/n = O(T_u(n))$, allowing us to go from a partially dynamic to a fully dynamic data structure "for free" in big O notation.

In terms of guaranteeing that the query operation solves the (r, cr)-near neighbor problem on the set of points P currently inserted into the data structure, we allow a constant failure probability $\delta < 1$, typically around $1/2$, and omit it from our statements. We make the standard assumption that the adversary does not have knowledge of the randomness used by the data structure. Say we have a data structure with constant failure probability and a bound on the expected space usage. Then, for every positive integer T we can create a collection of $O(\log T)$ independent repetitions of the data structure such that for every sequence of T operations it holds with high probability in T that the space usage will never exceed the expectation by more than a constant factor and no query will fail.

F.1 Model of computation We use the standard word RAM model as defined by Hagerup [23] with a word size of $\Theta(\log n)$ bits. Unless otherwise stated, we make the assumption that a point in (X, D) can be stored in d words and that the dissimilarity between two arbitrary points can be computed in d operations where d is a positive integer that corresponds to the dimension in the various well-studied settings mentioned in the main text. Furthermore, when describing framework-based solutions to the (r, cr)-near neighbor problem, we make the assumption that we can sample, evaluate, and represent elements from $\mathcal{F}$ and $\mathcal{H}$ with neglible error using space and time $dn^{o(1)}$.

Many of the LSH and LSF families rely on random samples from the standard normal distribution. We will ignore potential problems resulting from rounding due to the fact that our model only supports finite precision arithmetic. This approach is standard in the literature and can be justified by noting that the error introduced by rounding is neglible. Furthermore, there exists small pseudorandom standard normal distributions that support sampling using only few uniformly distributed bits as noted by Charikar [18]. In much of the related literature the model of computation is left unspecified and statements about the complexity of solutions to the (r, cr)-near neighbor problem are usually made with respect to particular operations such as the hash function computations, distance computations, etc., leaving out other details [26, 24].

References

[1] C. Aggarwal, D. A. Keim, and A. Hinneburg. On the surprising behaviour of distance metrics in high dimensional space. In *Proc. ICDT '01*, pages 420–434, 2001.

[2] J. Alman and R. Williams. Probabilistic polynomials and hamming nearest neighbors. In *Proc. FOCS '15*, pages 136–150, 2015.

[3] A. Andoni. *Nearest neighbor search: the old, the new, and the impossible*. PhD thesis, MIT, 2009.

[4] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proc. FOCS '06*, pages 459–468, 2006.

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

[6] A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, and L. Schmidt. Practical and optimal lsh for angular distance. In *Proc. NIPS '15*, pages 1225–1233, 2015.

[7] A. Andoni, P. Indyk, H. L. Nguyen, and I. P. Razenshteyn. Beyond locality-sensitive hashing. In *Proc. SODA '14*, pages 1018–1028, 2014.

[8] A. Andoni, P. Indyk, and M. Patrascu. On the optimality of the dimensionality reduction method. In *Proc. FOCS '06*, pages 449–458, 2006.

[9] A. Andoni, T. Laarhoven, I. P. Razenshteyn, and E. Waingarten. Lower bounds on time-space trade-offs for approximate near neighbors. *CoRR*, abs/1605.02701, 2016.

[10] A. Andoni, T. Laarhoven, I. P. Razenshteyn, and E. Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. *CoRR*, abs/1608.03580, 2016.

[11] A. Andoni and I. Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proc. STOC '15*, pages 793–801, 2015.

[12] A. Andoni and I. P. Razenshteyn. Tight lower bounds for data-dependent locality-sensitive hashing. *CoRR*, abs/1507.04299, 2015.

[13] A. Andoni and I. Razensteyn. Tight lower bounds for data-dependent locality-sensitive hashing. In *Proc. SoCG '16*, pages 9:1–9:11, 2016.

[14] A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proc. SODA '16*, pages 10–24, 2016.

[15] Y. Benyamini and J. Lindenstrauss. *Geometric nonlinear functional analysis*, volume 48. American Mathematical Soc., Providence, Rhode Island, 1998.

[16] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. In *Proc. STOC '98*, pages 327–336, 1998.

[17] J. M. Chambers, C. L. Mallows, and B. W. Stuck. A method for simulating stable random variables. *Jour. Am. Stat. Assoc.*, 71(354):340–344, 1976.

[18] M. Charikar. Similarity estimation techniques from

rounding algorithms. In *Proc. STOC '02*, pages 380–388, 2002.

[19] F. Chierichetti and R. Kumar. Lsh-preserving functions and their applications. *J. ACM*, 62(5):33, 2015.

[20] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. SOCG '04*, pages 253–262, 2004.

[21] M. de Berg, M. van Kreveld, M. Overmars, and O. C. Schwarzkopf. *Computational geometry*. Springer, Berlin, third edition, 2008.

[22] M. Dubiner. Bucketing coding and information theory for the statistical high-dimensional nearest-neighbor problem. *IEEE Trans. Inf. Theory*, 56(8):4166–4179, 2010.

[23] T. Hagerup. Sorting and searching on the word RAM. In *Proc. STACS '98*, pages 366–398, 1998.

[24] S. Har-Peled, P. Indyk, and R. Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory comp.*, 8(1):321–350, 2012.

[25] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Jour. Am. Stat. Assoc.*, 58(301):13–30, 1963.

[26] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. STOC '98*, pages 604–613, 1998.

[27] Piotr Indyk. Nearest neighbors in high-dimensional spaces. In *Handbook of Discrete and Computational Geometry, Second Edition.*, pages 877–892. Chapman and Hall/CRC, 2004.

[28] R. Kaas and J. M. Buhrman. Mean, median and mode in binomial distributions. *Statistica Neerlandica*, 34(1):13–18, 1980.

[29] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions (extended abstract). In *Proc. FOCS '88*, pages 68–80, 1988.

[30] M. Kapralov. Smooth tradeoffs between insert and query complexity in nearest neighbor search. In *Proc. PODS '15*, pages 329–342, 2015.

[31] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Comput.*, 30(2):457–474, 2000.

[32] T. Laarhoven. Tradeoffs for nearest neighbors on the sphere. *CoRR*, abs/1511.07527, 2015.

[33] P. Lévy. *Calcul des probabilités*, volume 9. Gauthier-Villars, Paris, 1925.

[34] D. Lu and W. V. Li. A note on multivariate gaussian estimates. *Journal of Mathematical Analysis and Applications*, 354(2):704–707, 2009.

[35] E. Lukacs. *Characteristic Functions*. Griffin, London, second edition, 1970.

[36] Q. Lv, W. Josephson, Z. Wang, M. Charikar, and K. Li. Multi-probe LSH: efficient indexing for high-dimensional similarity search. In *Proc. VLDB '07*, pages 950–961, 2007.

[37] J. Marcinkiewicz. Sur une propriete de la loi de gauss. *Mathematische Zeitschrift*, 44(1):612–618, 1939.

[38] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

[39] R. Motwani, A. Naor, and R. Panigrahy. Lower bounds on locality sensitive hashing. *SIAM J. Discrete Math.*, 21(4):930–935, 2007.

[40] H. L. Nguyen. *Algorithms for High Dimensional Data*. PhD thesis, Princeton, 2014.

[41] R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[42] M. H. Overmars and J. van Leeuwen. Worst-case optimal insertion and deletion methods for decomposable searching problems. *Information Processing Letters*, 12(4):168–173, 1981.

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

[44] R. Panigrahy. Entropy based nearest neighbor search in high dimensions. In *Proc. SODA '06*, pages 1186–1195, 2006.

[45] R. Panigrahy, K. Talwar, and U. Wieder. A geometric approach to lower bounds for approximate near-neighbor search and partial match. In *Proc. FOCS '08*.

[46] R. Panigrahy, K. Talwar, and U. Wieder. Lower bounds on near neighbor search via metric expansion. In *Proc. FOCS '10*, pages 805–814, 2010.

[47] G. Pólya. Remarks on characteristic functions. In *Proc. Berkeley Symposium on Mathematical Statistics and Probability 1945-1946*, pages 115–123, 1949.

[48] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proc. NIPS '07*, pages 1177–1184, 2007.

[49] W. Rudin. *Fourier Analysis on Groups*. Wiley, New York, 1990.

[50] I. R. Savage. Mill's ratio for multivariate normal distributions. *Jour. Res. NBS Math. Sci.*, 66(3):93–96, 1962.

[51] B. Schölkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, Massachusetts, 2002.

[52] S. J. Szarek and E. Werner. A nonsymmetric correlation inequality for gaussian measure. *Journal of multivariate analysis*, 68(2):193–211, 1999.

[53] N. G. Ushakov. *Selected Topics in Characteristic Functions*. VSP, Utrecht, The Netherlands, 1999.

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

[55] J. Wang, H. T. Shen, J. Song, and J. Ji. Hashing for similarity search: A survey. *CoRR*, abs/1408.2927, 2014.

[56] R. Weber, H. Schek, and B. Stephen. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proc. VLDB '98*, pages 194–205, 1998.

[57] Ryan Williams. A new algorithm for optimal constraint satisfaction and its implications. In *Proc. ICALP '04*, pages 1227–1237, 2004.

[58] V. M. Zolotarev. *One-dimensional stable distributions*, volume 65. American Mathematical Soc., 1986.