Data-Dependent Hashing via Nonlinear Spectral Gaps

Alexandr Andoni
Columbia University
United States
andoni@cs.columbia.edu

Assaf Naor
Princeton University
United States
naor@math.princeton.edu

Aleksandar Nikolov
University of Toronto
Canada
anikolov@cs.toronto.edu

Ilya Razenshteyn
Microsoft Research Redmond
United States
ilyaraz@microsoft.com

Erik Waingarten
Columbia University
United States
eaw@cs.columbia.edu

ABSTRACT

We establish a generic reduction from *nonlinear spectral gaps* of metric spaces to data-dependent Locality-Sensitive Hashing, yielding a new approach to the high-dimensional Approximate Near Neighbor Search problem (ANN) under various distance functions. Using this reduction, we obtain the following results:

For *general* d-dimensional normed spaces and n-point datasets, we obtain a *cell-probe* ANN data structure with approximation $O(\frac{\log d}{\varepsilon^2})$, space $d^{O(1)}n^{1+\varepsilon}$, and $d^{O(1)}n^{\varepsilon}$ cell probes per query, for any $\varepsilon > 0$. No non-trivial approximation was known before in this generality other than the $O(\sqrt{d})$ bound which follows from embedding a general norm into ℓ_2.

For ℓ_p and Schatten-p norms, we improve the data structure further, to obtain approximation $O(p)$ and sublinear query *time*. For ℓ_p, this improves upon the previous best approximation $2^{O(p)}$ (which required polynomial as opposed to near-linear in n space). For the Schatten-p norm, no non-trivial ANN data structure was known before this work.

Previous approaches to the ANN problem either exploit the low dimensionality of a metric, requiring space exponential in the dimension, or circumvent the curse of dimensionality by embedding a metric into a "tractable" space, such as ℓ_1. Our new generic reduction proceeds differently from both of these approaches using a novel partitioning method.

CCS CONCEPTS

• **Mathematics of computing → Dimensionality reduction**; • **Theory of computation → Random projections and metric embeddings**; **Computational geometry**; *Design and analysis of algorithms*;

KEYWORDS

Nearest neighbor search, nonlinear spectral gaps, randomized space partitions, locality-sensitive hashing

STOC'18, June 25–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5559-9/18/06…$15.00
https://doi.org/10.1145/3188745.3188846

ACM Reference Format:
Alexandr Andoni, Assaf Naor, Aleksandar Nikolov, Ilya Razenshteyn, and Erik Waingarten. 2018. Data-Dependent Hashing via Nonlinear Spectral Gaps. In *Proceedings of 50th Annual ACM SIGACT Symposium on the Theory of Computing (STOC'18).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3188745.3188846

1 INTRODUCTION

The *c-Approximate Near Neighbor Search* (c-ANN) problem is defined as follows. Given an n-point dataset $P \subset X$ lying in a metric space (X, d_X), we want to preprocess P to answer *approximate near neighbor queries* quickly. Namely, given a query point $q \in X$ such that there is a data point $p^* \in P$ with $d_X(q, p^*) \leq r$, the algorithm should return a data point $\widehat{p} \in X$ with $d_X(q, \widehat{p}) \leq cr$. We refer to $c > 1$ as the *approximation* and $r > 0$ as the *distance scale*; both parameters are known during the preprocessing. The main quantities to optimize are the *space* the data structure occupies and the *time* it takes to answer a query. In addition to being an indispensable tool for data analysis, ANN data structures have spawned two decades of theoretical developments (see, e.g., the surveys [4, 7] and the thesis [47] for an overview).

1.1 ANN for General Distances Functions

The best-studied metrics in the context of ANN are the Hamming/Manhattan (ℓ_1) and the Euclidean (ℓ_2) distances. Both ℓ_1 and ℓ_2 are very common in applications and admit efficient algorithms based on *hashing*: in particular, Locality-Sensitive Hashing (LSH) [3, 27] and its data-dependent counterparts [6, 9, 11]. Hashing-based algorithms for ANN over ℓ_1 and ℓ_2 have now been the subject of a long line of work, leading to a comprehensive understanding of the respective time–space trade-offs.

Beyond ℓ_1 and ℓ_2, the ANN landscape is much more mysterious despite having received significant attention (see Section 1.4 for an overview). In summary, we are still very far from having a general recipe for ANN data structures for *general* metrics with a non-trivial approximation. This state of affairs motivates the following broad question.

Problem 1. *For a given approximation $c > 1$, which metric spaces allow efficient ANN algorithms?*

An algorithm for general metrics is highly desirable both in theory and in practice. From the theoretical perspective, we are interested in a theory of ANN algorithms for a wide class of distance

functions. Such a theory would yield data structures (or impossibility results) for a variety of important distances for which we still do not know efficient ANN algorithms (e.g. the Earth Mover's Distance (EMD), the edit distance, generalized versions of the Hamming distance[1], etc). Perhaps even more tantalizing is the question of understanding what geometric properties of a metric space govern the hardness of ANN. In addition to the theoretical interest, in practice, one often needs to tune the distance function to the specifics of the application, and hence generic ANN algorithms are also preferred.

In this paper, we focus on the following important case of Problem 1, which was first raised in 2010 [2].

Problem 2. *Solve Problem 1 for d-dimensional* normed *spaces.*

Most metrics arising in applications are actually norms (e.g., the ℓ_p distances, matrix norms, the Earth Mover's Distance, etc.). Besides that, norms are geometrically nicer than general metrics, so there is more hope for a coherent theory (e.g., for the problems of *sketching* and *streaming* norms, see the general results of [8, 17], for ANN over general *symmetric* norms, see a recent result [10]).

1.2 Main Results

In this paper, we make progress towards resolving Problem 2. Our main contribution is a data structure for the $O(\log d)$-ANN problem over a general d-dimensional norm in the *cell-probe* model introduced by Yao [50]. Prior to this work, the only other ANN data structure for general norms achieved approximation $O(\sqrt{d})$ (see Section 1.4).

Theorem 1.1. *Let $0 < \varepsilon < 1$. Suppose that $(\mathbb{R}^d, \|\cdot\|)$ is a d-dimensional normed space. Then there exists a randomized data structure for $O\left(\frac{\log d}{\varepsilon^2}\right)$-ANN over $\|\cdot\|$ with the following parameters:*

- *The space used by the data structure is $n^{1+\varepsilon} \cdot d^{O(1)}$;*
- *The query procedure probes $n^{\varepsilon} \cdot d^{O(1)}$* words in memory, *where words consist of $O(\log n)$ bits*[2].

Let us emphasize that we do not claim any *time* bound on the query procedure. We only restrict the number of memory locations the data structure is allowed to probe (see Section 4 for a further discussion of the model). Nonetheless, we conjecture that one can in fact obtain a data structure for $O(\log d)$-ANN with sublinear *time* query complexity (as opposed to cell probe complexity only), provided a suitable oracle access to the norm.

Irrespective of the conjecture, our theorem can be thought of as a barrier for proving impossibility of efficient ANN data structures with approximation $O(\log d)$ for general norms. This is because *all* known unconditional data structure lower bounds proceed by proving a cell-probe lower bound [39]. Thus, a potential strong lower bound for the ANN problem would require a completely new approach to data structure lower bounds.

The main tool behind Theorem 1.1 is a new random partition for sets of points in a general normed space, and is of independent interest. In particular, we show how to convert an estimate on the nonlinear spectral gap of a metric space into a data-dependent Locality-Sensitive Hashing (LSH) family (see Section 1.3 for an overview).

Finally, our technique also gives a natural approach to designing data structures for *specific* metric spaces with better parameters, including *sublinear time*. Indeed, we instantiate our technique with the ℓ_p and Schatten-p norms, for which, with additional work, we obtain data structures with better approximations and sublinear time. For the ℓ_p norms, we obtain approximation $c = O(p)$, which improves exponentially over the approximation factor of $2^{O(p)}$ from [15, 43] (see Section 1.4).

Theorem 1.2. *Let $0 < \varepsilon < 1$ and $2 < p \le \infty$. There exists a randomized data structure for $O(p/\varepsilon)$-ANN over the ℓ_p norm with the following parameters:*

- *The space used by the data structure is $n^{1+\varepsilon} \cdot d^{O(1)}$;*
- *The query procedure takes* time *$n^{\varepsilon} \cdot d^{O(1)}$.*

Generalizing the theorem from above to Schatten-p norms, we obtain the first ANN data structures with a non-trivial approximation factor. Recall that the Schatten-p norm $\|\cdot\|_{S_p}$ of a matrix is the ℓ_p norm of the vector of its singular values[3]. The challenge of designing ANN under Schatten norms was posed in [2].

We state our Schatten-p results for regimes $1 \le p \le 2$ and $2 < p < \infty$ separately.

Theorem 1.3. *Let $0 < \varepsilon < 1$ and $1 \le p \le 2$. There exists a randomized data structure for c-ANN over the Schatten-p norm, where $c = O\left(\frac{1}{\varepsilon^{2/p}}\right)$ with the following parameters:*

- *The space used by the data structure is $n^{1+\varepsilon} \cdot d^{O(1)}$;*
- *The query procedure takes time $n^{\varepsilon} \cdot d^{O(1)}$.*

We now state the data structure for Schatten-p norms with $p > 2$. Compared to the ℓ_p algorithm from Theorem 1.2, the result for Schatten-p has worse dependence on the dimension for the space and query time. We note that for $p > \log d$, the norm $\|x\|_{S_p}$ is a constant factor from $\|x\|_{S_{\log d}}$; thus, it suffices to consider the cases when $2 < p \le \log d$.

Theorem 1.4. *Let $0 < \varepsilon < 1$ and $2 < p \le \infty$. There exists a randomized data structure for c-ANN over the Schatten-p norm, where $c = O(p/\varepsilon)$ with the following parameters:*

- *The space used by the data structure is $n^{1+\varepsilon} \cdot d^{O(p)}$;*
- *The query procedure takes time $n^{\varepsilon} \cdot d^{O(p)}$.*

See Section 1.4 for a more detailed exposition of how Theorem 1.3 and Theorem 1.4 relate to previously known results.

Let us note that the preprocessing procedures in all the new data structures are inefficient. Improving the preprocessing time is left as an interesting open problem.

1.3 Techniques

Nonlinear spectral gaps. At the conceptual level, the main contribution of the paper is a reduction from bounds on the *nonlinear spectral gap* to a data-dependent Locality-Sensitive Hashing (LSH)

[1] E.g., a metric of interest in applications is (X^d, ρ_{X^d}), where X is a metric itself, with the distance between vectors $x, y \in X^d$ defined as $\rho_{X^d}(x, y) = \sum_{i=1}^{d} d_X(x_i, y_i)$.

[2] We assume that all the coordinates of the dataset and query points as well as r can be stored in $O(\log n)$ bits.

[3] The Schatten-1 norm is known under the names of the nuclear or trace norm, the Schatten-2 norm is simply the Frobenius norm, and Schatten-∞ is known as the spectral or the operator norm.

family for a *general metric space*. Let $A = (a_{ij}) \in \mathbb{R}^{m\times m}$ be a symmetric doubly stochastic $m \times m$ matrix. Then, for a metric space (X, d_X) and $q \geq 1$ the nonlinear spectral gap $\gamma(A, d_X^q)$ is the smallest number for which the following holds. For every set of points $x_1, x_2, \ldots, x_m \in X$,

$$\frac{1}{m} \cdot \sum_{i=1}^{m}\sum_{j=1}^{m} d_X(x_i, x_j)^q \leq \gamma(A, d_X^q) \cdot \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij} \cdot d_X(x_i, x_j)^q.$$

For the ℓ_2 norm, $\gamma(A, \|\cdot\|_2^2) = \frac{1}{1-\lambda_2(A)}$, where $\lambda_2(A)$ is the second largest eigenvalue of A; i.e. in this case $\gamma(A, \|\cdot\|_2^2)$ is the inverse of the usual spectral gap of A. A systematic study of nonlinear spectral gaps of metric spaces was initiated in [37]. Similar inequalities can be found in earlier works (see, e.g., the introduction of [38] for a thorough literature review); we single out the reference [36] which is instrumental for our results. In the above-mentioned works, bounds on the nonlinear spectral gap were primarily used to show strong non-embeddability results.

We use the following recent result from [40] in order to build a cell-probe data structure for ANN over a general *norm*, as claimed in Theorem 1.1.

Theorem 1.5 ([40]). *For every norm $\|\cdot\|$ defined on $\mathbb{R}^d$, one has:*

$$\gamma(A, \|\cdot\|^2) = O\left(\frac{\log^2 d}{(1-\lambda_2(A))^2}\right).$$

We present a simplified proof of this theorem with a slight generalization to a weighted setting (which we need for the actual reduction) in Section 6.

At a high level, a strong enough upper bound on $\gamma(\cdot, d_X^q)$ in terms of $\gamma(\cdot, \|\cdot\|_2^2)$ gives a cell-probe data structure for ANN over a given metric space (X, d_X) using the reduction given in this paper. For the *time-efficient* data structures over ℓ_p and Schatten-p spaces (Theorem 1.2 and Theorem 1.4), we need the nonlinear spectral gap inequality in a strong *Rayleigh quotient* form. For the ℓ_p norms, such a stronger inequality was shown by Matoušek [36]. We adapt Matoušek's inequality to the weighted setting in the full version. For Schatten-p, the corresponding inequality is stated and proved in the full version. The new inequality is an extension of the Matoušek's inequality to the matrix setting using estimates from [48]. An additional twist compared to [36] is the need for a fixed-point statement similar to the Brouwer's theorem.

Data-dependent LSH. We now briefly describe how to utilize Theorem 1.5 to obtain a data-dependent LSH family for a general norm. Informally, for a given dataset, we would like to design a random partition of $\mathbb{R}^d$ that separates a query point from *far data points* often, while not separating a query point from *close data points* too often. With such a random partition, we can build the data structure as simply a collection of random decision trees. In each node, we sample a partition from the family, split the dataset among child nodes accordingly, and recurse on each child node. This connection has already been used in [9, 11, 24] (however, let us note that in [24] space partitions are used in a fundamentally different way; see the discussion in Section 1.5).

The construction of the data-dependent LSH incorporates three main ideas.

- We use the multiplicative weights update algorithm (MWU) [13] to reduce the problem of constructing a random partition to the problem of finding a deterministic partition that works on average with respect to a given distribution over points. This step is non-trivial since the resulting random partition must depend on the dataset fairly weakly so that a sample from it can be stored in poly(d) space. We end up using two levels of MWU, where the "outer" part is responsible for "guessing" the dataset iteratively, while the "inner" part finds the required random partition for the current guess.
- The problem of finding a deterministic partition can be seen as finding a sparse cut in an undirected graph embedded in $(\mathbb{R}^d, \|\cdot\|)$ so that the following conditions hold. First, we assume that the distance between the endpoints of every edge is at most 1. Additionally, we may assume that the distance between a typical pair of vertices is $\gg \log d$. It suffices to prove that this graph cannot be a spectral expander, since we may then employ Cheeger's inequality [19, 20] to obtain a sparse cut.
- Finally, Theorem 1.5 directly implies that expanders do not embed into $(\mathbb{R}^d, \|\cdot\|)$, so the above graph cannot be a spectral expander. In fact, if $A \in \mathbb{R}^{m\times m}$ is a normalized adjacency matrix of a graph and $x_1, x_2, \ldots, x_m \in \mathbb{R}^d$ are the points the vertices are mapped to, then the following holds. Since every edge has length at most 1,

$$\sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij} \cdot \|x_i - x_j\|^2 \leq \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij} = m. \tag{1}$$

  Since a typical pair of vertices is at distance $\gg \log d$ apart,

$$\frac{1}{m} \cdot \sum_{i=1}^{m}\sum_{j=1}^{m} \|x_i - x_j\|^2 \gg m \cdot \log^2 d. \tag{2}$$

  Combining (1) and (2) with Theorem 1.5, we get that $1 - \lambda_2(A) \ll 1$, which implies that the graph is not an expander.

Algorithmically, we construct a randomized space partition by combining the two-level MWU algorithm together with a spectral partitioning procedure. The new data-dependent LSH construction gives a generic approach to ANN, which departs substantially from the commonly-used embeddings technique.

Partitions of normed spaces. As mentioned briefly, Theorems 1.1, 1.2, 1.3, and 1.4 follow from new partitioning results for sets of points lying in normed spaces. The specific partitioning results are given in Sections 6 and the full version. Let us now state the partitioning results for ℓ_p spaces and for general normed spaces.

A *box* in $\mathbb{R}^d$ is an intersection of sets of the form $\{x \in \mathbb{R}^d \mid x_k \leq u\}$ or $\{x \in \mathbb{R}^d \mid x_k \geq u\}$, where $1 \leq k \leq d$ and $u \in \mathbb{R}$. In the full version, we obtain the following partitioning result for ℓ_p spaces.

Theorem 1.6. *Let $0 < \varepsilon < 1$, $2 < p < \infty$ and $R > 0$. Consider any dataset $P \subset \mathbb{R}^d$ of n points lying in $B_p(0, R) = \{x \in \mathbb{R}^d \mid \|x\|_p \leq R\}$. Either there is an ℓ_p-ball of radius $O(p/\varepsilon)$ containing $\Omega(n)$ points from P, or there exists a distribution $\mathcal{D}$ over* boxes *such that:*

(1) For every $u, v \in B_p(0, R)$ with $\|u - v\|_p \leq 1$, a random box $S \sim \mathcal{D}$ separates u and v with probability at most ε.

(2) *For every box S from the support of $\mathcal{D}$, the number of points in P lying in S is between $\Omega(n)$ and $\left(1-\Omega(1)\right)\cdot n$.*

Now let us state the partitioning result for general normed spaces, proved in Section 6.

Theorem 1.7. *Let $0<\varepsilon<1$, $X=(\mathbb{R}^d,\|\cdot\|_X)$ be a normed space and $0<R\le 2^{\mathrm{poly}(d)}$. There exists a collection C of measurable subsets of $B_X(0,R)=\{x\in\mathbb{R}^d \mid \|x\|_X\le R\}$ with $\log|C|\le \mathrm{poly}(d)$ such that the following holds. Consider any dataset $P\subset\mathbb{R}^d$ of n points lying in $B_X(0,R)$. Either there is an X-ball of radius $O\left(\frac{\log d}{\varepsilon^2}\right)$ containing $\Omega(n)$ points from P, or there exists a distribution $\mathcal{D}$ over the elements of C such that:*

(1) *For every $u,v\in B_X(0,R)$ with $\|u-v\|_X\le 1$, a random set $S\sim\mathcal{D}$ separates u and v with probability at most ε.*
(2) *For every set S from the support of $\mathcal{D}$, the number of points in P lying in S is between $\Omega(n)$ and $\left(1-\Omega(1)\right)\cdot n$.*

1.4 Related Work

Prior to our work, the quest for efficient ANN data structures in high-dimensional spaces beyond ℓ_1 and ℓ_2 has proceeded via embeddings. The idea is to embed the original space into an algorithmically tractable target space, for which one then builds a data structure. The common targets are ℓ_1 and ℓ_2 which can be handled with $O(1)$-approximation by [9], ℓ_∞ which can be handled with $O(\log\log d)$-approximation with [24], and ℓ_p-direct sums of these spaces, which can be handled with approximation $\mathrm{poly}(\log\log n)$ by [1, 5, 25, 26]. This approach gives the best known ANN data structure for a general norm with approximation $O(\sqrt{d})$ [14, 29]. It has also been successful for a $\mathrm{poly}(\log\log d)$-approximation for the Ulam metric [5], a $O(\log d)$-approximation for EMD [18, 28], a $2^{\widetilde{O}(\sqrt{\log d})}$-approximation for edit distance [46], and a $\mathrm{poly}(\log d)$-approximation for Frechét distance [25].

In a similar vein, the recent work [10] gives an ANN data structure for general *symmetric* norms with $\mathrm{poly}(\log\log n)$-approximation. It proceeds via a linear embedding of a d-dimensional symmetric norm into a $d^{O(1)}$-dimensional tractable universal space. However, the same paper shows that this approach fails for general norms.

For ANN under ℓ_p norms, constant factor approximations were known for $1\le p\le 2$ for near-linear space and sub-linear time [45]. The case when $p\ge 2$ is less clear. Prior to this work, the best algorithm for ℓ_p norms of [15, 43] achieved approximation $2^{O(p)}$ with polynomial space (as opposed to near-linear space) and poly-logarithmic query time. For large p, there is a better algorithm with approximation $O(\log\log d)$ [1, 5].

For ANN under Schatten-p norm, the previous best algorithm has polynomial in d approximation and follows from the relation between Schatten-p and ℓ_2 norms. An approximation $2^{O(p)}$ using polynomial space follows *implicitly* from a combination of the results from [15, 43] with the estimate from [48]. The related questions of *streaming*, *sketching* and *dimension reduction* of Schatten-p norms have been actively studied over the past few years [8, 32–35, 42].

For metrics with low intrinsic dimension, efficient ANN algorithms are known for *any metric space* [16, 21, 30, 31]. These results depend *exponentially* on the intrinsic dimension, and therefore the latter is assumed to be low. This is in contrast to this paper, where we do not make such assumptions, and focus on the *high-dimensional* regime (when $\omega(\log n)\le d\le n^{o(1)}$), where we cannot afford to have an exponential dependence on the dimension.

1.5 Lower Bounds

We complement our new algorithms with two impossibility results.

Limitation of efficient cuts. The reason that Theorem 1.1 is restricted to the cell-probe model is due to the inability to bound the time complexity of evaluating the random space partitions from Theorem 3.6 when working with general norms (even though we bound their space complexity). In constrast, for ℓ_p and Schatten-p norms, we manage to bound the time complexity and obtain *time-efficient* data structures. To explain this disparity, consider the following general scenario.

Let $G=(V,E)$ be a large graph embedded into an arbitrary normed space $(\mathbb{R}^d,\|\cdot\|)$ with edges between points at distance at most 1, and typical pair of vertices being well-separated. Following the discussion in Section 1.3, the graph G must have a sparse cut; however, the cut may not be induced by a "geometrically nice" subset of $\mathbb{R}^d$. During the algorithm from the proof of Theorem 1.1, graphs will have $d^{\Omega(d)}$ vertices, so we cannot afford to store the cut explicitly. Therefore, the query procedure re-computes the cuts on the fly. In order to achieve a time-efficient data structure for general norms, one would need to find geometrically nice cuts which can be evaluated efficiently.

For ℓ_p norms, we always find a sparse cut that is realized by a *coordinate cut* (that is, $\{v\in V \mid f(v)_k\le u\}$ for some $1\le k\le d$ and $u\in\mathbb{R}$). In our reduction we need to take intersections of cuts, which, in the case of coordinate cuts, are boxes, which are the main objects of Theorem 1.6. Thus, we store the boxes by storing the $2d$ values (lower and upper limits for each coordinate), and then we can easily evaluate on which side of a cut a given point lies. For Schatten-p norms, the argument is more delicate, but we are also able to store and compute cuts in an efficient manner.

In the full version, we show that it is not enough to consider a fixed family of cuts with small *description complexity* for general norms; these include coordinate cuts and hyperplane cuts. More generally, the result says that families of cuts used must be tailored to the particular normed space. We use a *random* norm construction similar to the one used by Gluskin in [22]. We note that this lower bound does not rule out *ball* cuts or other families of cuts that depend on the particular norm.

Optimality of data-dependent LSH. We show that for ℓ_p spaces, any *data-dependent LSH family* with sufficiently good parameters requires approximation $\Omega(\min\{p,\log d\})$,[4] thus our construction is optimal within the data-dependent LSH framework. To show this, we embed a large expander into ℓ_p using a result from [36]. We apply a similar argument to [12] to the embedded expander to show the desired lower bound. Thus, at least in some cases, embeddability of expanders captures the complexity of LSH *precisely*.

[4] Note that when $p>\log d$, ℓ_p is $O(1)$-close to $\ell_{\log d}$, so an $\Omega(p)$ lower bound when $1\le p\le\log d$ covers all interesting values of p.

This result should be contrasted with the $O(\log\log d)$-ANN data structure for ℓ_∞ from [24]. It also proceeds by certain[5] space partitions; the difference is that a dataset point is duplicated when inside some parts. This duplication allows the result of [24] to overcome the above-mentioned $\Omega(\log d)$ lower bound.

1.6 Open Problems

We state several natural open problems which seem approachable in light of the techniques developed in this paper.

- Can we get a *time-efficient* $O(\log d)$-ANN data structure for general norms? As mentioned in Section 1.5, randomized partitions from a family of "geometrically nice" cuts must be tailored to the norm of interest.
- Can we improve the approximation for general norms to $O(\log\log d)$ (even in the cell-probe model)? To accomplish this, we need to step out of the data-dependent LSH framework (see Section 1.5) to resemble the techniques from [24]. A related (perhaps easier) question is to obtain an $O(\log p)$-ANN data structure over the ℓ_p or Schatten-p norm.
- Can we make the preprocessing time *polynomial* in n and d, even for the ℓ_p case?
- For the *edit distance* defined on $\{0,1\}^d$, can we obtain a $(\log d)^{O(1)}$-ANN data structure by bounding the nonlinear spectral gap? The best known ANN data structure proceeds by embedding the metric into ℓ_1 with distortion $2^{\widetilde{O}(\sqrt{\log d})}$ [46].
- For the Earth Mover's Distance on $[d]^2$, can we obtain a $o(\log d)$-ANN data structure by bounding the nonlinear spectral gap? The best known ANN data structure (aside from the cell-probe data structure from Theorem 1.1) proceeds by embedding into ℓ_1 with distortion $O(\log d)$ [18, 28, 44].

1.7 Organization of the Paper

In Section 3, we show how to construct a data-dependent LSH family for a general *finite* metric space assuming a good enough bound on the spectral gap. We state this result in terms of a *cutting modulus* of a metric space, a quantity we introduce in Section 3.1. In Section 4, we show how to use this LSH family to construct a cell-probe ANN data structure for a finite metric. In order to handle general normed spaces defined over $\mathbb{R}^d$ (and not just finite metrics), we discretize the ambient space; the corresponding argument is standard and appears in Section 5. In Section 6, we show a minor generalization of Theorem 1.5, which bounds the spectral gap of a general norm. This allows us to give an upper bound on the cutting modulus of a normed space.

Using the results from Sections 4 and 5, we obtain a cell-probe data structure for $O(\log d)$-ANN, as claimed in Theorem 1.1. In the full version, we address the case of ℓ_p norms and prove Theorem 1.2, show a new spectral gap inequality for Schatten-p norms which implies Theorems 1.3 and 1.4, and show the two impossibility results discussed in Section 1.5.

[5] Deterministic.

2 PRELIMINARIES

We write χ_E as the indicator variable of event E. For any $m > 0$, we denote by $\Delta(m) \subset \mathbb{R}^{m\times m}$ the space of symmetric matrices $G = (g_{ij})$ with non-negative entries such that $\sum_{i=1}^m \sum_{j=1}^m g_{ij} = 1$. For $G \in \Delta(m)$, we denote the row sums as $\rho_G(i) = \sum_{j=1}^m g_{ij}$. The Laplacian of G is given by the $m \times m$ matrix

$$L_G = D - G,$$

and the normalized Laplacian of G is given by the $m \times m$ matrix

$$\mathcal{L}_G = I_m - D^{-1/2} G D^{-1/2},$$

where $D = \mathrm{diag}(\rho_G(1), \rho_G(2), \ldots, \rho_G(m))$ and I_m is the $m \times m$ identity matrix. We denote $0 = \lambda_1(\mathcal{L}_G) \le \lambda_2(\mathcal{L}_G) \le \ldots \le \lambda_m(\mathcal{L}_G)$ the eigenvalues of the normalized Laplacian of G, and $\nu_1(\mathcal{L}_G), \ldots, \nu_m(\mathcal{L}_G) \in \mathbb{R}^m$ be the corresponding eigenvectors. For a subset $S \subseteq [m]$, we write $\overline{S} = [m] \setminus S$ and $\rho_G(S) = \sum_{i\in S} \rho_G(i)$. We will frequently refer to sequences of m points in X, as the tuples $\boldsymbol{x} = (x_1, \ldots, x_m) \in X^m$. We will associate a subset $S \subset [m]$ with the corresponding subset of points $S_{\boldsymbol{x}} \subset X$ with $S_{\boldsymbol{x}} = \{x_i : i \in S\}$; and we often drop the subscript and refer to $S_{\boldsymbol{x}}$ as S when the sequence $\boldsymbol{x}$ is clear. In addition, for $S \subset X$, we write $S\colon X \to \{0,1\}$ for the map $S(x) = \chi_{\{x\in S\}}$. For some finite subset $P \subset X$ and $x \in X$, we let $S(x, P) = \{p \in P : S(x) = S(p)\}$.

For a fixed matrix $G \in \Delta(m)$ and $S \subset [m]$, the conductance of S with matrix G is given by:

$$\Phi_G(S) = \frac{\sum_{\substack{i\in S\\ j\notin S}} g_{ij}}{\min\left\{\rho_G(S), \rho_G(\overline{S})\right\}}.$$

Definition 2.1. For any $G \in \Delta(m)$, any metric space (X, d_X), and any $\boldsymbol{x} = (x_1, \ldots, x_m) \in X^m$, we define the Rayleigh quotient of $\boldsymbol{x}$ and G with respect to d_X^p by

$$\mathrm{R}(\boldsymbol{x}, G, d_X^p) = \frac{\sum_{i=1}^m \sum_{j=1}^m g_{ij} d_X(x_i, x_j)^p}{\sum_{i=1}^m \sum_{j=1}^m \rho_G(i)\rho_G(j) d_X(x_i, x_j)^p}.$$

Via a straight-forward calculation, we have that when the metric space is $\mathbb{R}$ with $d_X(x_i, x_j) = |x_i - x_j|$, if $x \in \mathbb{R}^m$ and $\sum_{i=1}^m \rho_G(i) x_i = 0$,

$$\mathrm{R}(x, G, |\cdot|^2) = \frac{\sum_{i=1}^m \sum_{j=1}^m g_{ij}|x_i - x_j|^2}{\sum_{i=1}^m \sum_{j=1}^m \rho_G(i)\rho_G(j)|x_i - x_j|^2} = \frac{x^T L_G x}{x^T d x}.$$

I.e. in this case $\mathrm{R}(x, G, |\cdot|^2)$ is the Rayleigh quotient $\frac{y^T \mathcal{L}_G y}{y^T y}$ for $y = D^{1/2}x$. Using this observation, we may state Cheeger's inequality with respect to $\mathrm{R}(x, G, |\cdot|^2)$.

Theorem 2.2 (Cheeger's Inequality, [19, 20], see also [49]). *For $x \in \mathbb{R}^m$ with $\sum_{i=1}^m \rho_G(i) x_i = 0$, there exists $t \in \mathbb{R}$ for which the set $S_t = \{i \in [m] : x_i < t\}$ satisfies:*

$$\Phi_G(S_t) \le \sqrt{\frac{\mathrm{R}(x, G, |\cdot|^2)}{2}}.$$

Letting $x = D^{-1/2}\nu_2(\mathcal{L}_G)$, there exists a subset $S \subset [m]$ which satisfies:

$$\Phi_G(S) \le \sqrt{2 \cdot \lambda_2(\mathcal{L}_G)}.$$

Remark 1 (Oracle access to a norm). *When working with a general normed space $(\mathbb{R}^d, \|\cdot\|_X)$, we assume oracle access to the function $\|\cdot\|: \mathbb{R}^d \to \mathbb{R}^{\geq 0}$. We also assume John's ellipsoid of $(\mathbb{R}^d, \|\cdot\|_X)$, i.e. the maximum volume centered ellipsoid in $\mathbb{R}^d$ contained in the unit ball of $\|\cdot\|_X$, is given by the d vectors in $\mathbb{R}^d$ specifying the ellipsoid.*

3 PARTITIONING GENERAL METRICS

In this section, we give a general approach for constructing LSH schemes for general metric spaces. Section 3.1 defines the cutting modulus of a metric space. At a high level, the cutting modulus captures the following property of a metric space (X, d_X): for any probability distribution on pairs of close points in X, either X contains a small ball with most of the mass (with respect to the marginal distribution), or there is a balanced partition of X which separates a small fraction of neighboring pairs.

The cutting modulus determines the approximation of the data structure and is an interface between the data structure description and nonlinear spectral gaps. We describe the data structure with cutting modulus as a parameter of the metric space, and we bound the cutting modulus of various metric spaces with bounds on the non-linear spectral gap.

3.1 Cutting Modulus of a Metric Space

We consider a metric space (X, d_X). The goal of this section is to define the cutting modulus of a metric space.

Definition 3.1. Fix some $G \in \Delta(m)$. We say $\boldsymbol{x} = (x_1, \ldots, x_m) \in X^m$ has a *β-dense ball* of radius R if there exists a point $c \in X$ such that $\rho_G(\{i \in [m] : x_i \in B_X(c, R)\}) \geq \beta$.

Definition 3.2. Let $\mathfrak{S}$ be family of subsets of the metric space X. We say that $G \in \Delta(m)$ has the *(R, ε)-ball-or-cut* property with respect to $\mathfrak{S}$ if for every m points $\boldsymbol{x} = (x_1, \ldots, x_m) \in X^m$ where $d_X(x_i, x_j) \leq 1$ if $g_{ij} > 0$, one of the two properties hold:

- Either $\boldsymbol{x}$ has a $\frac{1}{2}$-dense ball of radius R, or
- There exists a subset $S \in \mathfrak{S}$ such that $S_{\boldsymbol{x}} = \{i : x_i \in S\}$ satisfies $\Phi_G(S_{\boldsymbol{x}}) \leq \varepsilon$.

If $\mathfrak{S}$ contains all finite subsets of X, then we say that G has the *(R, ε)-ball-or-cut* property.

We may now formally define the notion of cutting modulus of a metric space.

Definition 3.3. We say that the *ε-cutting modulus* of a metric space (X, d_X) with respect to a family $\mathfrak{S}$ of subsets of X, $\Xi_{\mathfrak{S}}(X, \varepsilon)$, is the infimum over $R > 0$ such that for every $m \in \mathbb{N}$ every matrix $G \in \Delta(m)$ has (R, ε)-ball-or-cut property w.r.t. $\mathfrak{S}$.

If $\mathfrak{S}$ contains all finite subsets of X, we denote $\Xi_{\mathfrak{S}}(X, \varepsilon)$ simply by $\Xi(X, \varepsilon)$.

At a high level, the ε-cutting modulus of a metric space will govern the approximation ratio one may achieve with space $\mathrm{poly}(d) \cdot n^{1+O(\varepsilon)}$ and query time $\mathrm{poly}(d) \cdot n^{O(\varepsilon)}$. In particular, suppose the normed space (X, d_X) has $\Xi(X, \varepsilon) = R$. Consider any sequence of points $x_1, \ldots, x_m \in X$, and form a graph by connecting points lying at distance at most 1. The graph defines a normalized adjacency matrix $G \in \Delta(m)$ which has the (R, ε)-ball-or-cut property. If there exists a dense ball, then we know that a constant fraction of the points lie close to each other (within distance $2R$). Otherwise, there is a sparse cut of the points which does not cut many edges of G. Roughly speaking, the data-dependent LSH will be built by recursively applying this procedure, and using the multiplicative weights update rule in order to handle any possible distribution over datasets and queries. For our cell-probe algorithms we will allow $\mathfrak{S}$ to contain all finite subsets of X. However, our efficient data structures will use a restricted family $\mathfrak{S}$ which allows us to quickly determine which side of a cut a point lies on.

3.2 Partitioning Theorems

The goal of this section is to prove the main partitioning theorem. We consider a metric space (X, d_X) which consists of N points. Let $0 < \varepsilon < \varepsilon_0$ be a small positive parameter and $R = \Xi(X, \varepsilon)$.

We first define the notion of balanced collections of balls and cuts.

Definition 3.4. Let $\mathcal{S}$ be a collection of subsets $S_1, \ldots, S_m \subseteq X$. We say $\mathcal{S}$ is *ε-sparse* if for every two points $x, y \in X$ with $d_X(x, y) \leq 1$, at most an ε-fraction of subsets from $\mathcal{S}$ split x and y, i.e.,

$$\Pr_{i \sim [m]}[S_i(x) \neq S_i(y)] \leq \varepsilon.$$

Definition 3.5. Consider a dataset $P \subseteq X$ of n points. Let $\mathcal{S}$ be a collection of subsets $S_1, \ldots, S_m \subseteq X$. We say that $\mathcal{S}$ is *γ-balanced under P* if for any $S \in \mathcal{S}$ we have

$$(1 - \gamma)n \leq |S \cap P| \leq \gamma n.$$

These two notions of sparsity and balancedness will measure the quality of the data-dependent LSH. Intuitively, the data-dependent LSH is constructed by recursively partitioning the space with a random subset from a particular collection. We want the collection to be balanced, to ensure the algorithm makes progress, and sparse, to maintain a low probability of error. Lastly, we want collections of subsets which can be written succinctly; such a condition will ensure the querying algorithm can utilize the data-dependent LSH. We ensure our collection can be written succinctly by requiring there are not too many of them, and that the collections do not have too many sets.

We may now state the main partitioning theorem for general metric spaces.

Theorem 3.6. *Let $R = \Xi(X, \varepsilon)$ for some $\varepsilon \in (0, \frac{1}{4})$, and fix any $n \in \mathbb{N}$. There exists a collection C of subsets of X with $\log |C| = O(\log(N) \log(\log(N)/\varepsilon))$ such that for any dataset $P \subseteq X$ of n points,*

- *Either there exists a point $x_0 \in X$ with $|P \cap B_X(x_0, R)| \geq \frac{n}{50}$, or*
- *There exists a subcollection $\mathcal{S} \subseteq C$ of subsets of X such that:*
  - *$\mathcal{S}$ is 50ε-sparse,*
  - *$\mathcal{S}$ is $\frac{49}{50}$-balanced under P.*

Theorem 3.6 suggests a very natural data-dependent LSH. At each step of the algorithm, either we have a dense ball, or we have a collection of subsets with a distribution which decreases the size of the dataset and does not split the query from its dataset point too often. Note that the set C does not depend on P. This means the querying algorithm will know C, and needs to read

$O(\log(N)\log(\log(N)/\varepsilon)/\varepsilon)$ many bits from the data-structure in order to specify any particular set $S \in C$.

We now turn to proving Theorem 3.6. The proof is algorithmic and requires a few lemmas, which correspond to particular subroutines.

3.2.1 Partitioning with the (R,ε)-ball-or-cut property. Let $X = \{x_1,\ldots,x_N\}$ be the points of the metric space of size N. For the remainder of the section, let $G \in \Delta(N)$ be a fixed matrix with $g_{ij} > 0$ only if $d_X(x_i,x_j) \le 1$. We will frequently interchange between subsets $S \subseteq X$ and $S \subseteq [N]$ by associating $x_i \in X$ with $i \in [N]$. In addition, we frequently write $\overline{S} = [N] \setminus S$. The goal of this section is to use the (R,ε)-ball-or-cut property to give a subroutine which when given a matrix $G \in \Delta(N)$, outputs a dense ball with respect to G, or a particular subset of vertices which cuts few edges with respect to G.

Lemma 3.7. *Let $R = \Xi(X,\varepsilon)$ for some $\varepsilon \in (0,\frac{1}{4})$. Then there either exists a $\frac{1}{4}$-dense ball of radius R with respect to G, or there exists a subset $S \subseteq X$ where*

$$\frac{1}{3} \le \rho_G(S) \le \frac{3}{4} \qquad \text{and} \qquad \sum_{i\in S, j\notin S} g_{ij} \le 2\varepsilon.$$

Proof. We give an iterative procedure which begins with a set $S := \emptyset$, and at each step, either finds a dense ball of radius R, or adds some points to S while keeping $\rho_G(S) \le \frac{3}{4}$ and $\sum_{i\in S, j\notin S} g_{ij} \le 2\varepsilon$.

At the beginning of an iteration, assume $\rho_G(S) < \frac{1}{3}$. We repeat the following procedure:

(1) Consider the matrix $\widetilde{G} \in \Delta(|\overline{S}|)$ obtained by restricting G on the rows and columns corresponding to $\overline{S}$ and scaling the entries so they sum to 1. Note that we still have $g_{ij} > 0$ only if $d_X(x_i,x_j) \le 1$.
(2) The matrix $\widetilde{G}$ has the (R,ε)-ball-or-cut property, so either there exists a $\frac{1}{2}$-dense ball of radius R in $\overline{S}$ with respect to $\widetilde{G}$, or there exists a subset $\widetilde{S} \subset \overline{S}$ with $\Phi_{\widetilde{G}}(\widetilde{S}) \le \varepsilon$.
  (a) Suppose $\overline{S}$ has a $\frac{1}{2}$-dense ball of radius R with respect to $\widetilde{G}$. Then, that ball is $\frac{1}{4}$-dense with respect to G, since $\widetilde{G}$ was rescaled by at least $1 - \rho_G(S) - 2\varepsilon\rho_G(S) \ge \frac{1}{2}$.
  (b) Suppose $\widetilde{S} \subset \overline{S}$ is a subset with $\Phi_{\widetilde{G}}(\widetilde{S}) \le \varepsilon$, and assume, without loss of generality, that $\widetilde{S}$ has $0 < \rho_{\widetilde{G}}(\widetilde{S}) \le \frac{1}{2}$, since otherwise, we can switch $\widetilde{S}$ and $\overline{S} \setminus \widetilde{S}$. Then, let $S \leftarrow S \cup \widetilde{S}$.

The quantity $\rho_G(S)$ is monotonically increasing with the iterations, and the procedure terminates when $\rho_G(S) \ge \frac{1}{3}$. Thus, we just need to show that, as long as we do not return a $\frac{1}{4}$-dense ball with respect to G, we always have $\rho_G(S) \le \frac{3}{4}$ and $\Phi_G(S) \le 2\varepsilon$.

Consider the final iteration of the algorithm before S is returned; we have that $S \subset [N]$ satisfies $\rho_G(S) < \frac{1}{3}$ and $\rho_{\widetilde{G}}(\widetilde{S}) \le \frac{1}{2}$. Additionally, assume $\Phi_G(S) \le 2\varepsilon$ and $\Phi_{\widetilde{G}}(\widetilde{S}) \le \varepsilon$. Then,

$$\sum_{\substack{i\in S\cup\widetilde{S}\\ j\notin S\cup\widetilde{S}}} g_{ij} \le \sum_{\substack{i\in S\\ j\notin S}} g_{ij} + \sum_{\substack{i\in\widetilde{S}\\ j\notin S\cup\widetilde{S}}} g_{ij} \le 2\varepsilon\cdot\rho_G(S) + \varepsilon\cdot\rho_{\widetilde{G}}(\widetilde{S})$$
$$\le 2\varepsilon\left(\rho_G(S) + \rho_G(\overline{S})\right) = 2\varepsilon\cdot\rho_G(S\cup\widetilde{S}),$$

where we used the fact that $\rho_{\widetilde{G}}(\widetilde{S}) \le 2\rho_G(\widetilde{S})$, because the matrix $\widetilde{G}$ was normalized by a factor of at least $\frac{1}{2}$. Therefore, we have $\Phi_G(S\cup\widetilde{S}) \le 2\varepsilon$. Finally, note that:

$$\rho_G(S\cup\widetilde{S}) \le \rho_G(S) + \rho_G(\widetilde{S})$$
$$\le \rho_G(S) + \frac{1}{2}\left(1 - \rho_G(S) - \Phi_G(S)\right) + \Phi_G(S)$$
$$\le \frac{2}{3} + \frac{\varepsilon}{3} \le \frac{3}{4}.$$

□

3.2.2 Inner multiplicative weights update. The goal of this subsection is to use the partitioning procedure from Lemma 3.7 in order to either find a dense ball (with respect to a given distribution over X), or build a sparse collection of subsets. For the rest of the section, we let E be the set of unordered pairs of close points in X (at distance at most 1).

Lemma 3.8. *Let $R = \Xi(X,\varepsilon)$ for some $\varepsilon \in (0,\frac{1}{4})$, and let ν be a probability measure over points in X. Then, either there exists a ball B of radius R such that $\nu(B) \ge \frac{1}{6}$, or there exists a collection $\mathcal{S}$ of $O\left(\frac{\log N}{\varepsilon}\right)$ subsets $S \subseteq X$ such that:*

- *$\mathcal{S}$ is 50ε-sparse, and*
- *Every $S \in \mathcal{S}$ satisfies $\frac{1}{4} \le \nu(S) \le \frac{5}{6}$.*

Proof. We prove the lemma by giving an algorithm which produces the collection $\mathcal{S}$ via the multiplicative weights update algorithm. More specifically, we give an iterative procedure where for $t = 0,\ldots,O\left(\frac{\log N}{\varepsilon}\right)$, maintains at most N^2 weights, $w_t : E \to \mathbb{R}^{\ge 0}$. At each step, the procedure produces a matrix $G \in \Delta(N)$, checks the conditions of Lemma 3.7, and either outputs a dense ball or updates the weights w_{t+1}. Fix $\delta = \frac{1}{10}$. The procedure does the following:

(1) For $t = 0,\ldots,T = \left\lceil\frac{\log_2 N}{\varepsilon}\right\rceil$, maintain weights $w_t : E \to \mathbb{R}^{\ge 0}$, where initially, $w_0(x,y) = 1$ for all $(x,y) \in E$, and $\Psi_t = \sum_{(x,y)\in E} w_t(x,y)$. Start with $\mathcal{S} = \emptyset$.
(2) Let $G^{(t)} \in \Delta(N)$ be given by:

$$g_{ij}^{(t)} = \begin{cases} \delta\cdot\frac{w_t(x_i,x_j)}{2\Psi_t} & i\ne j, (x_i,x_j)\in E \\ 0 & i\ne j, (x_i,x_j)\notin E \\ (1-\delta)\nu(x_i) & i = j \end{cases},$$

and consider the possible outcomes of Lemma 3.7 with matrix $G^{(t)}$:
  (a) If there exists a $\frac{1}{4}$-dense ball B of radius R with respect to $G^{(t)}$, then

$$\frac{1}{4} \le \rho_{G^{(t)}}(B) = \sum_{i\in B}(1-\delta)\nu(i) + \sum_{i\in B}\sum_{j\ne i}\delta\frac{w_t(x_i,x_j)}{2\Psi_t}$$
$$\le (1-\delta)\nu(B) + \frac{\delta}{2}.$$

  Return B, since $\nu(B) \ge \frac{1}{6}$.
  (b) If there exists a subset $S^{(t)} \subset X$ with $\frac{1}{3} \le \rho_{G^{(t)}}(S^{(t)}) \le \frac{3}{4}$ and $\sum_{i\in S^{(t)}, j\notin S^{(t)}} g_{ij}^{(t)} \le 2\varepsilon$, then let $\mathcal{S} \leftarrow \mathcal{S}\cup\{S^{(t)}\}$ and for all $(x,y) \in E$, we let:

$$w_{t+1}(x,y) = w_t(x,y)\left(1 + \chi_{\{S^{(t)}(x)\ne S^{(t)}(y)\}}\right).$$

(3) After T iterations, if the procedure has not returned a ball, return $\mathcal{S}$.

It remains to show that if the procedure does not return a ball B, then the collection $\mathcal{S}$ is 50ε-sparse, and every $S^{(t)} \in \mathcal{S}$ satisfies $\frac{1}{4} \le \nu(S^{(t)}) \le \frac{5}{6}$. Note that $|\mathcal{S}| = O\left(\frac{\log N}{\varepsilon}\right)$ since $T = O\left(\frac{\log N}{\varepsilon}\right)$. In order to show that $\frac{1}{4} \le \nu(S) \le \frac{5}{6}$ for all $S^{(t)} \in \mathcal{S}$, note that, similarly to the case with B,

$$\frac{1}{3} \le \rho_{G^{(t)}}(S^{(t)}) \le \frac{\delta}{2} + (1-\delta)\nu(S^{(t)})$$

and

$$(1-\delta)\nu(S^{(t)}) \le \rho_{G^{(t)}}(S^{(t)}) \le \frac{3}{4},$$

where the claim follows since $\delta = \frac{1}{10}$. We now turn to showing that $\mathcal{S}$ is 50ε-sparse. On the one hand, we have:

$$\Psi_{t+1} = \sum_{(x,y)\in E} w_{t+1}(x,y) = \sum_{(x,y)\in E} w_t(x,y)\left(1 + \chi_{\{S^{(t)}(x)\neq S^{(t)}(y)\}}\right)$$
$$\le \Psi_t + \Psi_t \cdot \frac{2}{\delta} \sum_{i\in S^{(t)}, j\notin S^{(t)}} \delta \cdot \frac{w_t(x_i,x_j)}{2\Psi_t} \le \Psi_t\left(1 + \frac{4\varepsilon}{\delta}\right), \tag{3}$$

since $\delta \cdot \frac{w_t(x_i,x_j)}{2\Psi_t} = g_{ij}$ for every close pair (x_i, x_j), and

$$\sum_{i\in S^{(t)}, j\notin S^{(t)}} g_{ij} \le 2\varepsilon.$$

Thus,

$$\Psi_{T+1} \le \Psi_0\left(1 + \frac{4\varepsilon}{\delta}\right)^T \le N^2\left(1 + \frac{4\varepsilon}{\delta}\right)^T.$$

On the other hand, for each pair $(x,y) \in E$,

$$\Psi_{T+1} \ge 2^{p(x,y)\cdot T}, \tag{4}$$

where $p(x,y) = \Pr_{t\in[T]}[S^{(t)}(x) \neq S^{(t)}(y)]$. Combining (3) and (4), and taking logarithms, we have:

$$p(x,y) \le \frac{2\log_2 N}{T} + \log_2\left(1 + \frac{4\varepsilon}{\delta}\right)$$
$$\le \frac{2\log_2 N}{T} + \frac{4\varepsilon}{\delta} \le 2\varepsilon + 40\varepsilon \le 50\varepsilon.$$

□

3.2.3 Outer multiplicative weights update: proof of Theorem 3.6. The goal of this subsection is to prove Theorem 3.6. Similarly to Lemma 3.8, we use the multiplicative weights update rule to design an algorithm which incorporates (limited) information about the dataset P; in each update round, we call Lemma 3.8. We analyze this outer multiplicative weights update process using KL-divergence as a potential function. In particular, we use the following lemma, which is well known (see Theorem 2.4. in [13]), and has been used, for example, in the literature on differential privacy (Lemma IV.1. in [23]). We give the short proof here for completeness. Below, KL divergence will be defined with respect to the natural logarithm, i.e. for two measures μ and ν on X we have

$$D_{KL}(\mu\|\nu) = \sum_{x\in X} \mu(x)\ln\frac{\mu(x)}{\nu(x)}.$$

Lemma 3.9. *Let μ and ν be probability measures over X. For a subset $S \subseteq X$, let $\sigma = \mathrm{sign}(\mu(S) - \nu(S))$, and define a new probability measure ν' over X by*

$$\nu'(x) = \frac{\nu(x)e^{\eta\sigma S(x)}}{\sum_{y\in X}\nu(y)e^{\eta\sigma S(y)}}.$$

Then,

$$D_{KL}(\mu\|\nu') - D_{KL}(\mu\|\nu) \le -\eta|\mu(S) - \nu(S)| + \eta^2.$$

Proof. By the definition of KL-divergence we have

$$\begin{aligned} D_{KL}(\mu\|\nu') - D_{KL}(\mu\|\nu) &= \sum_{x\in X}\mu(x)\ln\frac{\nu(x)}{\nu'(x)} \\ &= \sum_{x\in X}\mu(x)\ln\frac{\sum_{y\in X}\nu(y)e^{\eta\sigma S(y)}}{e^{\eta\sigma S(x)}} \\ &= -\eta\sigma\mu(S) + \ln\sum_{y\in X}\nu(y)e^{\eta\sigma S(y)} \\ &\le -\eta\sigma\mu(S) \\ &\quad + \ln\sum_{y\in X}\nu(y)(1 + \eta\sigma S(y) + \eta^2 S(y)) \\ &= -\eta\sigma\mu(S) + \ln(1 + \eta\sigma\nu(S) + \eta^2\nu(S)) \\ &\le -\eta\sigma(\mu(S) - \nu(S)) + \eta^2 \\ &= -\eta|\mu(S) - \nu(S)| + \eta^2. \end{aligned}$$

The first inequality above follows from $e^z \le 1 + z + z^2$ for all $|z| \le 1$. The second inequality follows from $\ln(1+z) \le z$. □

In particular, notice that Lemma 3.9 implies that if $|\mu(S) - \nu(S)| > \alpha$, and we set $\eta = \frac{\alpha}{2}$, then the KL-divergence dicreases by at least $\frac{\alpha^2}{4}$.

Proof of Theorem 3.6. Similarly to Lemma 3.8, we give an iterative procedure where at each time step $t = 0, \ldots, T = O(\log N)$, we maintain N weights, $w_t : X \to \mathbb{R}^{\ge 0}$. At each step, the procedure produces a probability measure ν supported on points in X and uses Lemma 3.8 to get a collection of subsets of X. The procedure is defined as follows:

(1) For $t = 0, \ldots, T = 400\ln N$, maintain weights $w_t : X \to \mathbb{R}^{\ge 0}$, where initially, $w_0(x) = 1$ for all $x \in X$.
(2) Let $\nu^{(t)}$ be the probability measure supported on X given by $\nu^{(t)}(x) = \frac{w_t(x)}{\sum_{y\in X} w_t(x)}$. Consider the possible outcomes of Lemma 3.8 with measure $\nu^{(t)}$:
   (a) If there exists a ball $B^{(t)}$ of radius R such that $\nu^{(t)}(B) \ge \frac{1}{6}$ and $|P \cap B| \ge \frac{n}{50}$, then return $B = B^{(t)}$.
   (b) If there exists a ball $B^{(t)}$ of radius R such that $\nu^{(t)}(B) \ge \frac{1}{6}$ but $|P \cap B| < \frac{n}{50}$, then set
   $$w_{t+1}(x) = w_t(x)e^{-B^{(t)}(x)/20},$$
   and continue with the next iteration.
   (c) If there exists a collection $\mathcal{S}^{(t)}$ of subsets of X satisfying the conditions of Lemma 3.8, and $\frac{n}{25} \le |S \cap P| \le \frac{24n}{25}$ for all $S \in \mathcal{S}^{(t)}$, then return $\mathcal{S} = \mathcal{S}^{(t)}$.

(d) If there exists a collection $\mathcal{S}^{(t)}$ of subsets of X satisfying the conditions of Lemma 3.8, and for some $S \in \mathcal{S}^{(t)}$ we have $|S \cap P| < \frac{n}{25}$ or $|S \cap P| > \frac{24n}{25}$, then set $\sigma = \operatorname{sign}\left(\frac{|S\cap P|}{n} - \nu^{(t)}(S)\right)$, update the weights as

$$w_{t+1}(x) = w_t(x)e^{\sigma S(x)/20},$$

and continue with the next iteration.

Note that the procedure returns $B = B^{(t)}$ only if it is a ball of radius R that contains at least $\frac{n}{50}$ points, and it returns the collection $\mathcal{S} = \mathcal{S}^{(t)}$ only if it is 50ε-sparse and $\frac{49}{50}$-balanced. So, if the procedure returns B or $\mathcal{S}$, then we know it satisfies the condition of the theorem. Therefore, we just need to show that the procedure will return B or $\mathcal{S}$ in the first T iterations, and that $\mathcal{S}$ is a subcollection of a sufficiently small collection C. Since in each iteration we either return B or $\mathcal{S}$, or we update w_t, for the first claim it is enough to show that w_t is updated fewer than T times. We do so using KL-divergence as a potential function.

Let μ be the empirical distribution induced by the dataset, i.e. $\mu(x) = \frac{1}{n}$ for every $x \in P$ and $\mu(x) = 0$ for every $x \in X \setminus P$. At step 0, we have

$$D_{KL}(\mu\|\nu^{(0)}) = \ln N - H(\mu) \le \ln N, \tag{5}$$

where $H(\mu)$ is the Shannon entropy of μ, which is always non-negative. If we update w_t because there exists a ball $B^{(t)}$ with $\nu^{(t)}(B^{(t)}) \ge \frac{1}{6}$ but $\mu(B^{(t)}) = \frac{|P\cap B|}{n} < \frac{1}{50}$, then we have $|\mu(B^{(t)}) - \nu^{(t)}(B^{(t)})| > \frac{1}{6} - \frac{1}{50} > \frac{1}{10}$, so, by Lemma 3.9

$$D_{KL}(\mu\|\nu^{(t+1)}) < D_{KL}(\mu\|\nu^{(t)}) - \frac{1}{400}. \tag{6}$$

Similarly, if we update w_t because there exists a set $S \in \mathcal{S}^{(t)}$ with $\mu(S) = \frac{|S\cap P|}{n} < \frac{1}{25}$ or $\mu(S) > \frac{24}{25}$, then, by Lemma 3.8 we know that $\frac{1}{4} \le \nu^{(t)} \le \frac{5}{6}$, and, therefore,

$$|\mu(B) - \nu^{(t)}(B)| > \frac{24}{25} - \frac{5}{6} > \frac{1}{10}.$$

So, by Lemma 3.9, the inequality (6) holds in this case, too. By (5) and (6), and because KL-divergence is always non-negative, we have that w_t can be updated at most $400 \ln N \le T$ times. Therefore, after one of the T iterations the procedure will return either a ball B or a collection $\mathcal{S}$ satisfying the conditions of the theorem.

To finish the proof, we need to argue that $\mathcal{S}$ is a subcollection of a small collection C of subsets of X. Let M be the number of distinct collections $\mathcal{S}$ that the iterative procedure can return. Lemma 3.8 guarantees that, for any such collection, $|\mathcal{S}| = O(\log N/\varepsilon)$, and if we define C to be the union of all possible $\mathcal{S}$, then we have the bound $|C| = O(M \log(N/\varepsilon))$. To bound M, observe that $\mathcal{S}$ depends on the dataset P only to determine, for each $t = 1, \ldots, T$, whether the procedure has returned B or $\mathcal{S}$, or, otherwise, to determine the identity of a set $S \in \mathcal{S}^{(t)}$ such that $\mu(S) = \frac{|S\cap P|}{n} < \frac{1}{25}$ or $\mu(S) > \frac{24}{25}$, and the sign of $\mu(S) - \nu^{(t)}(S)$. Since $|\mathcal{S}^{(t)}| = O(\log N/\varepsilon)$, any set $S \in \mathcal{S}^{(t)}$ can be specified in $O(\log(\log(N)/\varepsilon))$ bits. Overall, $\mathcal{S}$ depends only on $O(T(1 + \log(\log(N)/\varepsilon))) = O(\log N \log(\log(N)/\varepsilon))$ bits from P, which gives the desired bound on M, and, therefore, on $\log |C|$. □

4 CELL-PROBE DATA STRUCTURE FOR GENERAL METRICS

Here, we describe a cell-probe data structure solving c-ANN for (X, d_X), where $|X| = N$. Along the way, we use Theorem 3.6 as the main tool.

We first define the cell-probe model (as used in Theorem 1.1). Given a dataset, the cell-probe algorithm is allowed unbounded preprocessing time and eventually stores some memory as a sequence of *cells* of $O(\log n)$ bits each. Then, given a query point, a cell-probe algorithm is allowed to probe some cells (possibly adaptively) to read the contents of a cell. The algorithm performs unbounded auxiliary computations and uses unbounded auxiliary memory. The complexity of a cell-probe algorithm is measured by the number of cells, or the space, the data structure uses, and the number of probes the algorithm makes during a query. We will assume that $\log \log N = O(\log n)$ and that any point in X can be specified using $O(\log N)$ cells.

The main theorem in this section is:

Theorem 4.1. *For any metric space X of size N, and $\alpha \in (0, \frac{1}{4})$, there exists a cell-probe data structure for $(2 \cdot \Xi(X, \Theta(\alpha)) + 1)$-ANN that uses $O(n^{1+\alpha} \cdot \log N)$ words of space and $O(n^{\alpha} \cdot \log n \cdot \log N)$ cell probes per query.*

While we do not measure time complexity in this section, we note the cell-probe algorithm described may be implemented with preprocessing time and query time which depend exponentially on the dimension.

In the rest of this section we fix $R = \Xi(X, \varepsilon)$ for a parameter $\varepsilon = \Theta(\alpha)$, to be determined later.

Preprocessing. Next we describe how to build the data structure (for the pseudocode, see Figure 1). Let $P \subset X$ be a dataset of n points. The data structure is a collection of independently generated random decision trees. Each node v of a tree stores the following fields:

- v.type: the type of the node;
- $v.P$: a subset of the dataset points;
- v.center: a point in X;
- $v.S$: $O(\log(N) \log(\log(N)/\varepsilon))$ bits used to indicate a set S in the collection C guaranteed by Theorem 3.6, defining a cut node;
- v.left and v.right: pointers to child nodes.

We keep a counter ℓ, which denotes the current level of the tree we are processing. Initially, $\ell = 0$, and it is incremented on each recursive call. Once ℓ reaches some threshold t (to be specified shortly), we store a leaf node v and save the points of the dataset which reached v in $v.P$. Thus the depth of the tree is bounded by t a priori.

(1) If there exists a point $x_0 \in X$ such that $|P \cap B_X(x_0, R)| \ge \frac{n}{50}$, we build a *ball node*. In this case, the ball node saves x_0 in v.center and $P \cap B_X(x_0, R)$ in $v.P$. We then recurse by building a data structure on $P \setminus B_X(x_0, R)$. (See ProcessBall in Figure 1).
(2) Otherwise, the second condition of Theorem 3.6 holds, and the set C guaranteed by the theorem contains a subcollection $\mathcal{S} \subseteq C$ of subsets of X which is 50ε sparse and $\frac{96}{100}$-balanced.

```
function Process(P, ℓ, v)
    if ℓ = t or |P| ≤ 100 then
        v.type ← "leaf."
        v.P ← P.
    else if ∃x_0 such that |P ∩ B_X(x_0, R)| ≥ |P|/50 then
        call ProcessBall(P, x_0, ℓ, v)
    else
        S ← MWU(P).
        v.mwu ← mwu
        sample S uniformly from S
        store bits necessary to identify S ∈ C in v.S
        ProcessCut(P, S, ℓ, v).

function MWU(P)
    S ⊆ C obtained from Theorem 3.6 with P.
    return S.

function ProcessBall(P, x, ℓ, v)
    v.type ← "ball."
    v.center ← x.
    v.P ← P ∩ B_X(x, R).
    Process(P \ B_X(x_0, R), ℓ + 1, v.left).

function ProcessCut(P, S, ℓ, v)
    v.type ← "cut."
    P_l = P ∩ S_i, P_r = P \ S.
    Process(P_l, ℓ + 1, v.left).
    Process(P_r, ℓ + 1, v.right).
    v.P ← ∅.
```

Figure 1: Pseudocode for constructing the data structure

```
function Query(q, v)
    if v.type = "leaf" then
        for p ∈ v.P do
            return p if d_X(q, p) ≤ 2R + 1.
        return ⊥.
    if v.type = "ball" then
        p ← QueryBall(q, v).
        return p if p ≠ ⊥.
    if v.type = "cut" then
        p ← QueryCut(q, v).
        return p if p ≠ ⊥.

function QueryBall(q, v)
    x_0 ← v.center.
    if d_X(x_0, q) ≤ R + 1 then
        return any p ∈ v.P.
    return Query(q, v.left).

function QueryCut(q, v)
    Identify S ∈ C from v.S
    if q ∈ S then
        return Query(q, v.left).
    return Query(q, v.right).
```

Figure 2: Pseudocode for querying the data structure

We sample a uniformly random $S \in \mathcal{S}$, and we build a *cut node* v. We store the $O(\log(N)\log(\log(N)/\varepsilon))$ bits necessary to identify S in $v.S$, and recursively create two child nodes, holding the points $P \cap S$ and $P \setminus S$. (See ProcessCut in Figure 1).

The final data structure consists of $k = O(n^\alpha)$ independent trees, rooted at the nodes $v_1, \ldots, v_k$, where the i-th tree was built by a call to $\textsc{Process}(P, 0, v_i)$.

Querying the Data Structure. We now specify how to query the data structure; the pseudocode is given in Figure 2. For each of the k trees in the data structure, we start the query procedure at the root of the tree, and proceed by cases, according to the type of node, as follows:

- *Leaf nodes*: If a query $q \in X$ queries a leaf node v, then the query scans $v.P$ and returns the first point which lies within distance $2R + 1$. If no such point is found, return $\perp$.
- *Ball nodes*: If a query $q \in X$ queries a ball node v, we test whether our query is close to the ball centered at v.center of radius R. In particular, if $d_X(q, v.\text{center}) \le R+1$ and $v.P \neq \emptyset$, we return an arbitrary $p \in v.P$. Otherwise, we recurse on the child node of v.
- *Cut nodes*: If a query $q \in X$ queries a cut node v, the querying algorithm runs the multiplicative weights algorithm, accessing the values stored in v.mwu. Once it determines the collection $\mathcal{S}$, the querying algorithm checks the index of the set $S_i \in \mathcal{S}$, which is stored in $v.S$. If $q \in S_i$, then the querying algorithm recurses on the left child, otherwise, it recurses on the right child of v.

We collect some simple facts about the data structure which we use later in the analysis.

Claim 1. *The following statements are true:*

- *The sets $v.P$ for nodes v partition the dataset P.*
- *If $\textsc{Query}(q, v)$ returns a point $p \in P$, then $d_X(p, q) \le 2R + 1$.*

Analysis.

It remains to set the parameters t and ε. We let $t = \left\lceil \frac{\log n}{\log(50/49)} \right\rceil$ and $\varepsilon = \left\lfloor \frac{\alpha \cdot \log(50/49)}{50} \right\rfloor$ in order to have $(1 - 50\varepsilon)^t \ge n^{-\alpha}$.

Consider a fixed dataset P, and let $q \in X$ be any query, which is promised to have a point $p \in P$ with $d_X(p, q) \le 1$. If there are multiple such points for q, we fix one arbitrarily. Let v be a node of the data structure built by a call to $\textsc{Process}(P_v, \ell, v)$ for some $P_v \subset P$ and $\ell < t$. We let $U = C(v, q)$ be the random variable (over the random choice of $S_i \in \mathcal{S}$ if v is a cut node) which specifies the child node followed by $\textsc{Query}(q, v)$, and $\perp$ if $\textsc{Query}(q, v)$ does not recurse down a child. We also consider the random variable P_U consisting of the dataset involved in the call $\textsc{Process}(P_U, \ell + 1, U)$ which builds the node U when $U \neq \perp$.

We first claim that for any node v of the data structure, if $p \in P_v$, then,

$$\Pr[p \in P_U \mid U \neq \perp] \ge 1 - 50\varepsilon. \tag{7}$$

To see this, first consider the case in which $\textsc{Process}(P_v, \ell, v)$ calls ProcessBall, and let x be the center of the ball. If $U \neq \perp$, then $\textsc{QueryBall}(q, v)$ did not return any point and $d_X(x, q) > R + 1$, so $p \notin B_X(x, R)$. Then $p \in P_v \setminus B_X(x, R) = P_U$ with probability 1. For the remaining case, when Process calls ProcessCut, we have:

$$\Pr_{S \sim \mathcal{S}}[p \in P_U] = \Pr_{S \sim \mathcal{S}}[S(p) = S(q)] \ge 1 - 50\varepsilon,$$

since $\mathcal{S}$ is guaranteed to be 50ε-sparse by Theorem 3.6.

By Claim 1, any point p' returned by $\textsc{Query}(q, v_i)$, where v_i is the root of one of the data structure trees, satisfies $d_X(p', q) \le 2R+1$. To prove correctness, it remains to argue that, with sufficiently high probability, at least one of the $\textsc{Query}(q, v_i)$ calls, for $i = 1, \ldots, k$, does in fact return a point. Fix some i between 1 and k, and define a random sequence $U_0, U_1, \ldots, U_s$ of nodes of the tree rooted at v_i by

$U_0 = v_i$ and $U_\ell = C(U_{\ell-1}, q)$; U_s is the first node in this sequence for which $C(U_s, q) = \perp$. Notice that $s \le t$. Clearly, $\textsc{Query}(q, v_i)$ will return a point if $p \in P_{U_s}$. By (7) and the choice of t, this happens with probability at least $(1-50\varepsilon)^s \ge (1-50\varepsilon)^t \ge n^{-\alpha}$. By picking the number k of trees in the data structure to be a sufficiently large multiple of n^α, we can guarantee that with large constant probability the data structure returns a point p' such that $d_X(p', q) \le 2R + 1$.

To finish the analysis, we need to bound the number of cells stored by the data structure, and the number of cell probes made by the query procedure. Each of the points stored in the leaves of each tree form a partition of the point set P, so each tree has at most n internal nodes. Each internal node stores $O(\log(N))$ cells, and all the leaves together use $O(n \log N)$ cells of space ($O(\log N)$ per point in P). Therefore, the total space used by the data structure is $O(n^{1+\alpha} \log N)$ cells.

The query procedure probes $O(\log N)$ cells at each internal node of a tree. The number of cells probed at a leaf node v is proportional to $O(|v.P| \cdot \log N)$. We claim that $v.P$ is bounded by a constant. Suppose that u is a child of a node v, and also that v was created by a call to $\textsc{Process}(P_v, \ell, v)$ and u by a call to $\textsc{Process}(P_u, \ell+1, u)$. Then, by the guarantees of Theorem 3.6, $|P_u| \le \frac{49}{50}|P_v|$, so the number of points that can reach a leaf of a tree is bounded by $n\left(\frac{49}{50}\right)^t$. By the choice of t, this number is bounded by a constant, as we claimed. Therefore, the total number of cells probed by the query procedure is $O(kt \log N) = O(n^\alpha \log n \log N)$. This completes the proof of Theorem 4.1.

5 DISCRETIZING THE SPACE

Let $\|\cdot\|$ be a norm on $\mathbb{R}^d$ with unit ball K. Let $\mathcal{E}$ be the John Ellipsoid of K, i.e. the largest volume ellipsoid contained inside K. By John's theorem [29],

$$\mathcal{E} \subset K \subset \sqrt{d} \cdot \mathcal{E}.$$

We let $C \supset \mathcal{E}$ be the *smallest rotated box* (with side-length 2 in $\|\cdot\|$) containing $\mathcal{E}$. More formally, consider the affine transform $F\colon \mathbb{R}^d \to \mathbb{R}^d$ which maps B_2^d (the unit ball of $\|\cdot\|_2$) to $\mathcal{E}$. Then $C = F(B_\infty^d)$. Note that the collection

$$\mathcal{H}_s = \{F(2s \cdot x) + s \cdot C \subset \mathbb{R}^d : x \in \mathbb{Z}^d\},$$

partitions $\mathbb{R}^d$ into disjoint translated copies of C with side-length $2s$.

In this section, we reduce the problem of c-ANN for $\|\cdot\|$ over $\mathbb{R}^d$ to the problem of c-ANN for $\|\cdot\|$ over a finite set of points. We first reduce to the case when the dataset and query are bounded by a high-dimensional box, then we will show how to discretize the boxes in order to reduce to a finite set of points.

Lemma 5.1. *Let A be a data structure solving c-ANN for $\|\cdot\|$ over $s \cdot C$ where $s = O(d)$ with success probability $\frac{9}{10}$, query time $T(n)$ and space $S(n) = \Omega(dn)$. Then there exists a data structure A' solving c-ANN for $\|\cdot\|$ over $\mathbb{R}^d$ which solves the problem with probability $\frac{8}{10}$, query time $T(n) + O(d)$ and space $S(n) + O(dn)$.*

Proof. The data structure A', upon receiving the dataset P, proceeds in the following way:

- Partition the space by a randomly shifted $s \cdot C$ where $s = 5d$ (with respect to $\|\cdot\|$). More formally, we sample $y \sim [0, 2s]^d$ and consider the collection:
$$\mathcal{H}_{s,y} = \{F(y) + H \subset \mathbb{R}^d : H \in \mathcal{H}_s\}.$$
- For each $H \in \mathcal{H}_{s,y}$, we take the dataset $P \cap H$ falling inside this location, translate the dataset by the center of H and invoke the data structure A on the translated points of $P \cap H$.

On a query q, we identify the location $q \in H \in \mathcal{H}_{s,y}$. We translate the query by the center of H, and query the corresponding data structure holding $P \cap H$.

We say that two points $p, q \in \mathbb{R}^d$ are split if they lie in different cells of the partition $\mathcal{H}_{s,y}$. For any p and q with $\|p-q\| \le 1$, we have

$$\Pr[p \text{ and } q \text{ split}] \le \frac{d \cdot \|p-q\|}{2s} \le \frac{1}{10}$$

where we used the fact that after the affine transform F which maps $e_1, \ldots, e_d$ to the major axes of $\mathcal{E}$, we have the probability that we split points p and q is at most

$$\begin{aligned}\frac{1}{2s}\sum_{i=1}^{d} |(F^{-1}(p-q))_i| &= \frac{1}{s}\left\|F^{-1}(p) - F^{-1}(q)\right\|_1 \\ &\le \frac{\sqrt{d}}{2s}\left\|F^{-1}(p) - F^{-1}(q)\right\|_2 \\ &= \frac{\sqrt{d}}{2s}\|p-q\|_{\mathcal{E}} \le \frac{\|p-q\|}{2s}.\end{aligned}$$

Thus, with probability $\frac{9}{10}$, the query point and the dataset point fall in the same grid location. The query time of $T(n) + O(d)$ is immediate, and the space $S(n) + O(dn)$ follows from the fact that we must store a hash of the non-empty values of $P \cap H$ where $H \in \mathcal{H}_{s,y}$, as well as y, as well as the fact that $S(n) = \Omega(n)$. □

We now proceed to the second step where we reduce to the case the dataset and query lie within a fixed set of points. We let X be a greedily constructed γ-net of $s \cdot C$ (where distances are measured with respect to $\|\cdot\|$). Let $(X, \|\cdot\|)$ be the metric space obtained by restricting the norm to T.

A standard volume argument gives the following fact.

Fact 1. *We have that $|X| \le \exp(O(d \log(d/\gamma)))$.*

Since X is a γ-net, we may identify points with their closest neighbor in X. The following lemma is immediate, and finishes the reduction.

Lemma 5.2. *Let A be a data structure solving c-ANN for $(X, \|\cdot\|)$ with success probability $\frac{9}{10}$, time $T(n)$ and space $S(n)$. There exists a data structure A' solving $c \cdot (\frac{1+2\gamma}{1-2\gamma})$-ANN for $\|\cdot\|$ over $s \cdot C$ with success probability $\frac{9}{10}$ in time $T(n) + O(d)$ and space $S(n)$.*

6 BOUNDING THE CUTTING MODULUS OF A NORMED SPACE

For $G = (g_{ij}) \in \Delta(m)$, we denote the diagonal $m \times m$ matrix $D = \mathrm{diag}(\rho_G(1), \rho_G(2), \ldots, \rho_G(m))$. We set $A = (a_{ij}) = D^{-1/2} G D^{-1/2}$, so that $a_{ij} = \frac{g_{ij}}{\sqrt{\rho_G(i)\rho_G(j)}}$ and $\mathcal{L}_G = I - A$. For a metric space

$(\mathcal{M}, d_{\mathcal{M}})$ and $q > 0$, we define (the inverse of) the nonlinear spectral gap $\gamma(G, d_{\mathcal{M}}^q)$ to be the infimum over $\gamma > 0$ such that for every $u_1, u_2, \ldots, u_m \in \mathcal{M}$, one has:

$$\sum_{i,j=1}^m \rho_G(i)\rho_G(j) \cdot d_{\mathcal{M}}(u_i, u_j)^q \le \gamma \sum_{i,j=1}^m g_{ij} \cdot d_{\mathcal{M}}(u_i, u_j)^q.$$

Note that this definition agrees with the one from the Introduction if G is (a multiple of) a doubly-stochastic matrix.

In this section, we show that for every d-dimensional normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$ and every $0 < \varepsilon < 1/2$, one has $\Xi(X, \varepsilon) \lesssim \frac{\log d}{\varepsilon^2}$.[6] This bound easily follows (see Theorem 6.4) from a slight extension of Theorem 1.5 to the case when A is not necessarily doubly stochastic. This extension can be obtained by examining the proof from [40], but instead we present a new, shorter and more elementary argument, which constitutes the bulk of the present section (for a slightly different exposition of the same argument, see [41]).

Recall that for normed spaces $X = (\mathbb{R}^d, \|\cdot\|_X)$ and $Y = (\mathbb{R}^d, \|\cdot\|_Y)$ and a linear map $T : \mathbb{R}^d \to \mathbb{R}^d$, the operator norm $\|T\|_{X\to Y}$ is defined as follows: $\|T\|_{X\to Y} = \sup_{\|x\|_X=1} \|Tx\|_Y$. The Banach–Mazur distance $d_{\mathrm{BM}}(X, Y)$ between X and Y is defined as follows: $d_{\mathrm{BM}}(X, Y) = \inf_{T:\, \mathbb{R}^d \to \mathbb{R}^d} \|T\|_{X\to Y} \cdot \|T^{-1}\|_{Y\to X}$. By John's theorem, one always has: $d_{\mathrm{BM}}(X, \ell_2^d) \le \sqrt{d}$.

Theorem 6.1. *For every normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$ and every $G = (g_{ij}) \in \Delta(m)$, one has $\gamma(G, \|\cdot\|_X^2) \lesssim \left(\frac{1+\log \mathrm{d}}{\lambda_2(\mathcal{L}_G)}\right)^2$, where $\mathrm{d} = d_{\mathrm{BM}}(X, \ell_2^d) \le \sqrt{d}$. In particular, one always has: $\gamma(G, \|\cdot\|_X^2) \lesssim \left(\frac{\log d}{\lambda_2(\mathcal{L}_G)}\right)^2$.*

Let $V \subset (\mathbb{R}^d)^m$ be the following codimension-1 subspace:

$$V = \left\{ (v_1, v_2, \ldots, v_m) \in (\mathbb{R}^d)^m \;\middle|\; \sum_{i=1}^m \sqrt{\rho_G(i)} \cdot v_i = 0 \right\}.$$

We denote $V_X = (V, \|\cdot\|_{V_X})$ the normed space where for $\boldsymbol{v} = (v_1, v_2, \ldots, v_m) \in V$, the norm is given by $\|\boldsymbol{v}\|_{V_X} = \sqrt{\sum_{i=1}^m \|v_i\|_X^2}$. Denote $\mathcal{A} : V \to V$ the following linear map: $(\mathcal{A}\boldsymbol{v})_i = \sum_{j=1}^m a_{ij} v_j = \sum_{j=1}^m \frac{g_{ij} v_j}{\sqrt{\rho_G(i)\rho_G(j)}}$. In words, $\mathcal{A}$ acts on a tuple of d-dimensional vectors the same way as $A = D^{-1/2} G D^{-1/2}$ acts on a tuple of scalars. It is immediate to check that the image of $\mathcal{A}$ indeed lies in V; this follows from the fact $\left(\sqrt{\rho_G(1)}, \sqrt{\rho_G(2)}, \ldots, \sqrt{\rho_G(m)}\right)$ is an eigenvector of A. Let $\mathcal{I} : V \to V$ be the identity map.

Let us show that Theorem 6.1 readily follows from the following lemma.

Lemma 6.2. *One has: $\left\|(\mathcal{I} - \mathcal{A})^{-1}\right\|_{V_X \to V_X} \lesssim \frac{1+\log \mathrm{d}}{\lambda_2(\mathcal{L}_G)}$.*

Proof of the implication "Lemma 6.2 ⇒ Theorem 6.1". An immediate reformulation of Lemma 6.2 is that for every $v_i \in \mathbb{R}^d$ such that $\sum_{i=1}^m \sqrt{\rho_G(i)} \cdot v_i = 0$, one has:

$$\sum_{i=1}^m \|v_i\|_X^2 \lesssim \left(\frac{1+\log \mathrm{d}}{\lambda_2(\mathcal{L})}\right)^2 \cdot \sum_{i=1}^m \left\| v_i - \sum_{j=1}^m \frac{g_{ij} v_j}{\sqrt{\rho_G(i)\rho_G(j)}} \right\|_X^2. \quad (8)$$

[6] Here the notation $a \lesssim b$ means that there exists a constant C, independent of all other parameters, such that $a \le Cb$.

Our goal is to show that for every $u_1, u_2, \ldots, u_m \in \mathbb{R}^d$, one has:

$$\sum_{i,j=1}^m \rho_G(i)\rho_G(j) \cdot \|u_i - u_j\|_X^2 \lesssim \left(\frac{1+\log \mathrm{d}}{\lambda_2(\mathcal{L}_G)}\right)^2 \cdot \sum_{i,j=1}^m g_{ij} \cdot \|u_i - u_j\|_X^2. \quad (9)$$

Without loss of generality, we can assume that $\sum_{i=1}^m \rho_G(i) \cdot u_i = 0$. We set $v_i = \sqrt{\rho_G(i)} \cdot u_i$. Hence, $\sum_{i=1}^m \sqrt{\rho_G(i)} \cdot v_i = 0$ and (8) applies. On the one hand, one has:

$$\begin{aligned}
&\sum_{i,j=1}^m \rho_G(i)\rho_G(j) \cdot \|u_i - u_j\|_X^2 \\
&\le \sum_{i,j=1}^m \rho_G(i)\rho_G(j) \cdot \left(\|u_i\|_X + \|u_j\|_X\right)^2 \\
&\le 2 \sum_{i,j=1}^m \rho_G(i)\rho_G(j) \cdot \left(\|u_i\|_X^2 + \|u_j\|_X^2\right) \\
&= 4 \sum_{i=1}^m \rho_G(i) \cdot \|u_i\|_X^2 = 4 \sum_{i=1}^m \|v_i\|_X^2. \quad (10)
\end{aligned}$$

On the other hand, one has:

$$\begin{aligned}
&\sum_{i=1}^m \left\| v_i - \sum_{j=1}^m \frac{g_{ij} v_j}{\sqrt{\rho_G(i)\rho_G(j)}} \right\|_X^2 \\
&= \sum_{i=1}^m \left\| \sum_{j=1}^m \frac{g_{ij}}{\sqrt{\rho_G(i)}} \cdot \left(\frac{v_i}{\sqrt{\rho_G(i)}} - \frac{v_j}{\sqrt{\rho_G(j)}} \right) \right\|_X^2 \\
&= \sum_{i=1}^m \left\| \sum_{j=1}^m \frac{g_{ij}}{\sqrt{\rho_G(i)}} \cdot (u_i - u_j) \right\|_X^2 \\
&\le \sum_{i=1}^m \left(\sum_{j=1}^m \frac{g_{ij}}{\sqrt{\rho_G(i)}} \cdot \|u_i - u_j\|_X \right)^2 \\
&\le \sum_{i,j=1}^m g_{ij} \cdot \|u_i - u_j\|_X^2, \quad (11)
\end{aligned}$$

where the third step is due to the triangle inequality, and the fourth step is due to Jensen's inequality. Combining (8), (10) and (11), we obtain (9). □

Now let us show the proof of Lemma 6.2. For this we will need to relate the geometry of X and the Euclidean geometry. Let $H = (\mathbb{R}^d, \|\cdot\|_H)$ be a Hilbert space such that for every $v \in \mathbb{R}^d$, one has $\|v\|_H \le \|v\|_X \le \mathrm{d} \cdot \|v\|_H$. We define the normed space $V_H = (V, \|\cdot\|_{V_H})$ similarly to V_X: the norm $\|\boldsymbol{v}\|_{V_H}$ for $\boldsymbol{v} = (v_1, v_2, \ldots, v_m) \in V$ is defined as follows: $\|\boldsymbol{v}\|_{V_H} = \sqrt{\sum_{i=1}^m \|v_i\|_H^2}$. Clearly, for every $\boldsymbol{v} \in V$, one has:

$$\|\boldsymbol{v}\|_{V_H} \le \|\boldsymbol{v}\|_{V_X} \le \mathrm{d} \cdot \|\boldsymbol{v}\|_{V_H}. \quad (12)$$

Finally, we define $\widetilde{A} = \frac{A+I}{2}$ and $\widetilde{\mathcal{A}} = \frac{\mathcal{A}+\mathcal{I}}{2}$. Let us observe that $\left\|(\mathcal{I} - \mathcal{A})^{-1}\right\|_{V_X \to V_X} \lesssim \left\|(\mathcal{I} - \widetilde{\mathcal{A}})^{-1}\right\|_{V_X \to V_X}$, thus it is enough to show that

$$\left\|(\mathcal{I} - \widetilde{\mathcal{A}})^{-1}\right\|_{V_X \to V_X} \lesssim \frac{1+\log \mathrm{d}}{\lambda_2(\mathcal{L}_G)}. \quad (13)$$

One can see that (13) is an immediate corollary of the following three statements together with (12). Let us note that Lemma 6.3 is the place, where the logarithmic dependence on d shows up.

Claim 2. *One has* $\|\widetilde{\mathcal{A}}\|_{V_X \to V_X} \le 1$.

Claim 3. *One has* $\|\widetilde{\mathcal{A}}\|_{V_H \to V_H} \le 1 - \frac{\lambda_2(\mathcal{L}_G)}{2}$.

Lemma 6.3. *Let* $\|\cdot\|_P$ *and* $\|\cdot\|_Q$ *be two norms on* $\mathbb{R}^{d'}$ *such that for some* $\Phi \ge 1$ *for every* $u \in \mathbb{R}^{d'}$ *one has* $\|u\|_Q \le \|u\|_P \le \Phi \cdot \|u\|_Q$. *Suppose that* $T\colon \mathbb{R}^{d'} \to \mathbb{R}^{d'}$ *is a linear map such that* $\|T\|_{P\to P} \le 1$ *and* $\|T\|_{Q\to Q} \le 1-\varepsilon$ *for some* $0 < \varepsilon < 1$. *Then,* $\|(I-T)^{-1}\|_{P\to P} \lesssim \frac{1+\log\Phi}{\varepsilon}$.

Proof of Claim 2. One has for every $\boldsymbol{v} = (v_1, v_2, \ldots, v_m) \in V$:

$$\begin{aligned}\|\mathcal{A}\boldsymbol{v}\|_{V_X}^2 &= \sum_{i=1}^m \|(\mathcal{A}\boldsymbol{v})_i\|_X^2 = \sum_{i=1}^m \left\|\sum_{j=1}^m \frac{g_{ij} v_j}{\sqrt{\rho_G(i)\rho_G(j)}}\right\|_X^2 \\ &\le \sum_{i=1}^m \left(\sum_{j=1}^m \frac{g_{ij}}{\rho_G(i)} \left\|\sqrt{\frac{\rho_G(i)}{\rho_G(j)}}\cdot v_j\right\|_X\right)^2 \\ &\le \sum_{i=1}^m \sum_{j=1}^m \frac{g_{ij}}{\rho_G(i)} \left\|\sqrt{\frac{\rho_G(i)}{\rho_G(j)}}\cdot v_j\right\|_X^2 \\ &= \sum_{i=1}^m \sum_{j=1}^m \frac{g_{ij}}{\rho_G(j)} \|v_j\|_X^2 = \sum_{j=1}^m \|v_j\|_X^2 = \|\boldsymbol{v}\|_{V_X}^2,\end{aligned}$$

where the third step is by the triangle inequality, and the fourth step is by the Jensen's inequality. Hence, $\|\mathcal{A}\|_{V_X\to V_X} \le 1$. But this implies that $\|\widetilde{\mathcal{A}}\|_{V_X\to V_X} \le 1$ as well. □

Proof of Claim 3. Let us first observe that for every $u \in \mathbb{R}^m$ such that $\sum_{i=1}^m \sqrt{\rho_G(i)}\cdot u_i = 0$, one has:

$$\|\widetilde{A}u\|_2 \le \left(1 - \frac{\lambda_2(\mathcal{L}_G)}{2}\right)\cdot \|u\|_2, \tag{14}$$

since $\widetilde{A}$ is positive semidefinite, the largest eigenvalue is 1, the corresponding eigenvector is $(\sqrt{\rho_G(i)})_{i=1}^m$, and the second largest eigenvalue is $1 - \lambda_2(\mathcal{L}_G)/2$.

The desired inequality reduces to (14) as follows. Since H is a Hilbert space, there exists an orthogonal basis $e_1, e_2, \ldots, e_d \in \mathbb{R}^d$ such that for every $u \in \mathbb{R}^m$, one has $\|u\|_H^2 = \sum_{i=1}^m \langle u, e_i\rangle^2$. For $1 \le i \le d$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_m) \in V$, define $\pi_i(\boldsymbol{v}) = (\langle v_1, e_i\rangle, \ldots, \langle v_m, e_i\rangle) \in \mathbb{R}^m$. Then, $\|\boldsymbol{v}\|_{V_H}^2 = \sum_{i=1}^d \|\pi_i(\boldsymbol{v})\|_2^2$. One has:

$$\begin{aligned}\|\widetilde{\mathcal{A}}\boldsymbol{v}\|_{V_H}^2 &= \sum_{i=1}^d \|\pi_i(\widetilde{\mathcal{A}}\boldsymbol{v})\|_2^2 = \sum_{i=1}^d \|\widetilde{A}\pi_i(\boldsymbol{v})\|_2^2 \\ &\le \left(1 - \frac{\lambda_2(\mathcal{L}_G)}{2}\right)^2 \sum_{i=1}^d \|\pi_i(\boldsymbol{v})\|_2^2 \\ &= \left(1 - \frac{\lambda_2(\mathcal{L}_G)}{2}\right)^2 \|\boldsymbol{v}\|_{V_H}^2.\end{aligned}$$

□

Proof of Lemma 6.3. For every $k \ge 1$, one has $\|T^k\|_{P\to P} \le \Phi\cdot(1-\varepsilon)^k$. Thus, we can choose $k^* \lesssim \frac{1+\log\Phi}{\varepsilon}$ such that $\|T^{k^*}\|_{P\to P} \le 1/2$. Finally, we have:

$$\begin{aligned}\|(I-T)^{-1}\|_{P\to P} &\le \sum_{k=0}^\infty \|T^k\|_{P\to P} \le k^* \cdot \sum_{i=0}^\infty \|T^{ik^*}\|_{P\to P} \\ &\le k^* \cdot \sum_{i=0}^\infty (1/2)^i \lesssim k^* \lesssim \frac{1+\log\Phi}{\varepsilon}\end{aligned}$$

as desired. □

Theorem 6.4. *For every normed space* $X = (\mathbb{R}^d, \|\cdot\|_X)$ *with* $d_{\mathrm{BM}}(X, \ell_2^d) = \mathrm{d} \le \sqrt{d}$, *and every* $0 < \varepsilon < 1/2$, *one has:* $\Xi(X,\varepsilon) \lesssim \frac{1+\log \mathrm{d}}{\varepsilon^2}$. *In particular, one always has:* $\Xi(X,\varepsilon) \lesssim \frac{\log d}{\varepsilon^2}$.

Proof. Let $R > 0$ be a parameter to be fixed later. Let $G \in \Delta(m)$ and let $\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \in X^m$ be such that $\|x_i - x_j\|_X \le 1$ if $g_{ij} > 0$. Suppose that $\boldsymbol{x}$ has no 1/2-dense ball of radius R. Then,

$$\sum_{i,j=1}^m \rho_G(i)\rho_G(j)\cdot\|x_i - x_j\|_X^2 \ge \frac{R^2}{2}. \tag{15}$$

On the other hand, we have:

$$\sum_{i,j=1}^m g_{ij}\cdot\|x_i - x_j\|_X^2 \le 1, \tag{16}$$

since $\|x_i - x_j\|_X^2 \le 1$ whenever $g_{ij} > 0$. Thus, combining (15), (16) and Theorem 6.1, we get: $\lambda_2(\mathcal{L}_G) \lesssim \frac{1+\log \mathrm{d}}{R}$. Thus, by setting $R \lesssim \frac{1+\log \mathrm{d}}{\varepsilon^2}$ and using Cheeger's inequality (Theorem 2.2), we conclude that G has a cut with conductance at most ε. □

Note that Theorems 3.6 and 6.4, together with a standard discretization argument imply Theorem 1.7. Indeed, given a norm $\|\cdot\|_X$ on $\mathbb{R}^d$, and a radius R so that $\log R$ is polynomial in d, we can greedily find N points $x_1, \ldots x_N$ so that the balls $B_X(x_1,\gamma)$, …, $B_X(x_N,\gamma)$ cover $B_X(0,R)$, and $\log N = O(d\log(R/\gamma))$. We can then use Theorem 3.6 with the metric space of size N induced on $\{x_1,\ldots,x_N\}$ and the cutting modulus bound given in Theorem 6.4. We identify any set S in the collection C guaranteed by Theorem 3.6 with the union of the balls $B_X(x_i,\gamma)$ that cover the elements of S. It is easy to verify that any two points $u, v \in B_X(0,R)$ that lie at a distance at most $1-2\gamma$ apart are separated by a uniformly random set in the subcollection $\mathcal{S}$ with probability at most 50ε. The guarantee of Theorem 1.7 then follows by a simple rescaling.

ACKNOWLEDGMENTS

We thank Richard Peng and Tselil Schramm for useful discussions.

The first-named author was supported in part by the Simons Foundation (#491119), NSF grants CCF-1617955, CCF-1740833, and Google Research Award. The second-named author was supported in part by the NSF grant CCF-1412958, the Packard Foundation and the Simons Foundation. The third-named author was supported in part by the NSF grant CCF-1740425 and NSERC Discovery Grant. The fourth-named author was supported in part by the Simons Junior Fellowship. The fifth-named author was supported in part

by the NSF grants CCF-1563155, CCF-1420349, CCF-1149257, CCF-1423100), and the NSF Graduate Research Fellowship (DGE-16-44869).

The work was done in part while the third-named author was visiting Simons Institute for the Theory of Computing and while the fourth-named author was a graduate student at MIT and a postdoc at Columbia University. This work was was carried out under the auspices of the Simons Algorithms and Geometry (A&G) Think Tank.

REFERENCES

[1] Alexandr Andoni. 2009. *Nearest Neighbor Search: the Old, the New, and the Impossible.* Ph.D. Dissertation. MIT.
[2] Alexandr Andoni. 2010. Nearest Neighbor Search in high-dimensional spaces. (2010). Invited talk at the Workshop on Barriers in Computational Complexity II, http://www.mit.edu/~andoni/nns-barriers.pdf.
[3] Alexandr Andoni and Piotr Indyk. 2006. Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2006)*. 459–468.
[4] Alexandr Andoni and Piotr Indyk. 2017. Nearest Neighbors in High-Dimensional Spaces. In *Handbook of Discrete and Computational Geometry*, Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth (Eds.). CRC Press LLC, 1133–1153.
[5] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. 2009. Overcoming the ℓ_1 Non-Embeddability Barrier: Algorithms for Product Metrics. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms (SODA '2009)*. 865–874.
[6] Alexandr Andoni, Piotr Indyk, Huy L. Nguyen, and Ilya Razenshteyn. 2014. Beyond Locality-Sensitive Hashing. In *Proceedings of the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA '2014)*. 1018–1028. Available as arXiv:1306.1547.
[7] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. 2018. Approximate Nearest Neighbor Search in High Dimensions. In *Proceedings of ICM 2018 (to appear)*.
[8] Alexandr Andoni, Robert Krauthgamer, and Ilya Razenshteyn. 2015. Sketching and Embedding are Equivalent for Norms. In *Proceedings of the 47th ACM Symposium on the Theory of Computing (STOC '2015)*. 479–488. Available as arXiv:1411.2577.
[9] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. 2017. Optimal Hashing-based Time–Space Trade-offs for Approximate Near Neighbors. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA '2017)*. 47–66. Available as arXiv:1608.03580.
[10] Alexandr Andoni, Huy L. Nguyen, Aleksandar Nikolov, Ilya Razenshteyn, and Erik Waingarten. 2017. Approximate Near Neighbors for General Symmetric Norms. In *Proceedings of the 49th ACM Symposium on the Theory of Computing (STOC '2017)*. 902–913. Available as arXiv:1611.06222.
[11] Alexandr Andoni and Ilya Razenshteyn. 2015. Optimal Data-Dependent Hashing for Approximate Near Neighbors. In *Proceedings of the 47th ACM Symposium on the Theory of Computing (STOC '2015)*. 793–801. Available as arXiv:1501.01062.
[12] Alexandr Andoni and Ilya Razenshteyn. 2016. Tight Lower Bounds for Data-Dependent Locality-Sensitive Hashing. In *Proceedings of the 32nd International Symposium on Computational Geometry (SoCG '2016)*. 9:1–9:11. Available as arXiv:1507.04299.
[13] Sanjeev Arora, Elad Hazan, and Satyen Kale. 2012. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing* 8, 1 (2012), 121–164.
[14] Keith Ball. 1997. *An Elementary Introduction to Modern Convex Geometry.* MSRI Publications, Vol. 31. Cambridge University Press.
[15] Yair Bartal and Lee-Ad Gottlieb. 2015. Approximate Nearest Neighbor Search for ℓ_p-Spaces ($2 < p < \infty$) via Embeddings. (2015). Available as arXiv:1512.01775.
[16] Alina Beygelzimer, Sham Kakade, and John Langford. 2006. Cover Trees for Nearest Neighbor. In *Proceedings of the 23rd International Conference on Machine Learning (ICML '2006)*. 97–104.
[17] Jaroslaw Blasiok, Vladimir Braverman, Stephen R. Chestnut, Robert Krauthgamer, and Lin F. Yang. 2017. Streaming Symmetric Norms via Measure Concentration. In *Proceedings of the 49th ACM Symposium on the Theory of Computing (STOC '2017)*. Available as arXiv:1511.01111.
[18] Moses Charikar. 2002. Similarity Estimation Techniques from Rounding Algorithms. In *Proceedings of the 34th ACM Symposium on the Theory of Computing (STOC '2002)*. 380–388.
[19] Jeff Cheeger. 1969. A Lower Bound for the Smallest Eigenvalue of the Laplacian. In *Proceedings of the Princeton conference in honor of Professor S. Bochner*. 195–199.
[20] F. R. K. Chung. 1996. Laplacians of graphs and Cheeger's inequalities. In *Combinatorics, Paul Erdős is eighty, Vol. 2 (Keszthely, 1993)*. Bolyai Soc. Math. Stud., Vol. 2. János Bolyai Math. Soc., Budapest, 157–172.
[21] Kenneth L. Clarkson. 1999. Nearest Neighbor Queries in Metric Spaces. *Discrete and Computational Geometry* 22, 1 (1999), 63–93.
[22] Efim D. Gluskin. 1981. Diameter of the Minkowski Compactum is Approximately Equal to n. *Functional Analysis and Its Applications* 15, 1 (1981), 57–58.
[23] Moritz Hardt and Guy N. Rothblum. 2010. A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*. IEEE Computer Society, 61–70. https://doi.org/10.1109/FOCS.2010.85
[24] Piotr Indyk. 2001. On Approximate Nearest Neighbors under ℓ_∞ Norm. *J. Comput. System Sci.* 63, 4 (2001), 627–638.
[25] Piotr Indyk. 2002. Approximate Nearest Neighbor Algorithms for Fréchet Distance via Product Metrics. In *Proceedings of the 18th ACM Symposium on Computational Geometry (SoCG '2002)*. 102–106.
[26] Piotr Indyk. 2004. Approximate Nearest Neighbor under Edit Distance via Product Metrics. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA '2004)*. 646–650.
[27] Piotr Indyk and Rajeev Motwani. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the 30th ACM Symposium on the Theory of Computing (STOC '1998)*. 604–613.
[28] Piotr Indyk and Nitin Thaper. 2003. Fast Color Image Retrieval via Embeddings. (2003). Workshop on Statistical and Computational Theories of Vision (at ICCV).
[29] Fritz John. 1948. Extremum Problems with Inequalities as Subsidiary Conditions. In *Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948*. Interscience Publishers, Inc., New York, N. Y., 187–204.
[30] David R. Karger and Matthias Ruhl. 2002. Finding Nearest Neighbors in Growth-Restricted Metrics. In *Proceedings of the 34th ACM Symposium on the Theory of Computing (STOC '2002)*. 741–750.
[31] Robert Krauthgamer and James R. Lee. 2004. Navigating Nets: Simple Algorithms for Proximity Search. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA '2004)*. 798–807.
[32] Yi Li, Huy L. Nguyên, and David P. Woodruff. 2014. On Sketching Matrix Norms and the Top Singular Vector. In *Proceedings of the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA '2014)*. 1562–1581.
[33] Yi Li and David P. Woodruff. 2016. On Approximating Functions of the Singular Values in a Stream. In *Proceedings of the 48th ACM Symposium on the Theory of Computing (STOC '2016)*. 726–739.
[34] Yi Li and David P. Woodruff. 2016. Tight Bounds for Sketching the Operator Norm, Schatten Norms, and Subspace Embeddings. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 19th International Workshop, APPROX '2016, and 20th International Workshop, RANDOM '2016*. 39:1–39:11.
[35] Yi Li and David P. Woodruff. 2017. Embeddings of Schatten Norms with Applications to Data Streams. In *Proceedings of the 44th International Colloquium on Automata, Languages and Programming (ICALP '2017)*. 60:1–60:14.
[36] Jiří Matoušek. 1997. On Embedding Expanders into ℓ_p Spaces. *Israel Journal of Mathematics* 102 (1997), 189–197.
[37] Manor Mendel and Assaf Naor. 2014. Nonlinear Spectral Calculus and Super-Expanders. *Publications Mathématiques de l'IHÉS* 119, 1 (2014), 1–95.
[38] Manor Mendel and Assaf Naor. 2015. Expanders with Respect to Hadamard Spaces and Random Graphs. *Duke Mathematical Journal* 164, 8 (2015), 1471–1548.
[39] Peter Bro Miltersen. 1999. Cell Probe Complexity – a Survey. In *Advances in Data Structures*.
[40] Assaf Naor. 2017. A Spectral Gap Precludes Low-Dimensional Embeddings. In *Proceedings of the 33rd International Symposium on Computational Geometry (SoCG '2017)*. 50:1–50:16.
[41] Assaf Naor. 2018. Metric Dimension Reduction: a Snapshot of the Ribe Program. In *Proceedings of ICM 2018 (to appear)*.
[42] Assaf Naor, Gilles Pisier, and Gideon Schechtman. 2018. Impossibility of Dimension Reduction in the Nuclear Norm. In *Proceedings of the 29th ACM-SIAM Symposium on Discrete Algorithms (SODA '2018)*.
[43] Assaf Naor and Yuval Rabani. 2006. On Approximate Nearest Neighbor Search in ℓ_p, $p > 2$. (2006). Manuscript, available on request.
[44] Assaf Naor and Gideon Schechtman. 2007. Planar Earthmover is not in L_1. *SIAM J. Comput.* 37, 3 (2007), 804–826.
[45] Huy L. Nguyên. 2014. *Algorithms for High Dimensional Data.* Ph.D. Dissertation. Princeton University. Available as http://arks.princeton.edu/ark:/88435/dsp01b8515q61f.
[46] Rafail Ostrovsky and Yuval Rabani. 2007. Low Distortion Embedding for Edit Distance. *J. ACM* 54, 5 (2007), 23:1–23:16.
[47] Ilya Razenshteyn. 2017. *High-Dimensional Similarity Search and Sketching: Algorithms and Hardness.* Ph.D. Dissertation. Massachusetts Institute of Technology.
[48] Éric Ricard. 2015. Hölder Estimates for the Noncommutative Mazur Map. *Archiv der Mathematik* 104, 1 (2015), 37–45.
[49] Daniel A. Spielman. 2015. Conductance, the Normalized Laplacian, and Cheeger's Inequality. Lecture Notes. http://www.cs.yale.edu/homes/spielman/561/lect06-15.pdf
[50] Andrew Chi-Chih Yao. 1981. Should Tables be Sorted? *J. ACM* 28, 3 (1981), 615–628.