Lower Bounds on Near Neighbor Search via Metric Expansion

Rina Panigrahy
Microsoft Research
Email: rina@microsoft.com

Kunal Talwar
Microsoft Research
Email: kunal@microsoft.com

Udi Wieder
Microsoft Research
Email: uwieder@microsoft.com

***Abstract*—In this paper we show how the complexity of performing nearest neighbor (NNS) search on a metric space is related to the expansion of the metric space. Given a metric space we look at the graph obtained by connecting every pair of points within a certain distance r . We then look at various notions of expansion in this graph relating them to the cell probe complexity of NNS for randomized and deterministic, exact and approximate algorithms. For example if the graph has node expansion Φ then we show that any deterministic t-probe data structure for n points must use space S where $(St/n)^t > \Phi$. We show similar results for randomized algorithms as well. These relationships can be used to derive most of the known lower bounds in the well known metric spaces such as l_1, l_2, l_∞, and some new ones, by simply computing their expansion. In the process, we strengthen and generalize our previous results [18]. Additionally, we unify the approach in [18] and the communication complexity based approach. Our work reduces the problem of proving cell probe lower bounds of near neighbor search to computing the appropriate expansion parameter.**

In our results, as in all previous results, the dependence on t is weak; that is, the bound drops exponentially in t. We show a much stronger (tight) time-space tradeoff for the class of *dynamic low contention* data structures. These are data structures that supports updates in the data set and that do not look up any single cell too often. A full version of the paper could be found in [19].

I. Introduction

In the Nearest Neighbor Problem we are given a data set of n points $x_1, ..., x_n$ lying in a metric space V. The goal is to preprocess the data set into a data structure such that when given a query point $y \in V$, it is possible to recover the data set point which is closest to y by querying the data structure at most t times. The goal is to keep both the querying time t and the data structure space m as small as possible. Nearest Neighbor Search is a fundamental problem in data structures with numerous applications to web algorithms, computational biology, information retrieval, machine learning, etc. As such it has been researched extensively.

The time space tradeoff of known solutions crucially depend upon the underlying metric space. Natural metric spaces include the spaces $\Re^d$ equipped with the ℓ_1 or ℓ_2 distance, but other metrics such as ℓ_∞, edit distance and earth mover distance may also be useful. The known upper bounds exhibit the 'curse of dimensionality':for d dimensional spaces either the space or time complexity is exponential in d. More efficient solutions are known when considering approximations e.g. [11], [13], [10], [2], however, in general these algorithms still demonstrate a relatively high complexity.

There is a substantial body of work on lower bounds covering various metric spaces and parameter settings; we discuss the known bounds in Section I-C. Traditionally, cell probe lower bounds for data structures have been shown using communication complexity arguments [15]. Pătraşcu and Thorup [20] use a direct sum theorem along with the richness technique to obtain lower bounds for deterministic algorithms. Andoni, Indyk and Pătraşcu [3] showed randomized lower bounds using communication complexity lower bounds for Lopsided Set Disjointness. In a previous work [18], the authors used a more direct geometric argument to show lower bounds for randomized algorithms for the search version of the problem.

In this work we strengthen and significantly generalize our previous results. We give a common framework that unifies almost all known cell probe lower bounds for near neighbor search. At one extreme, it gives us the communication complexity lower bounds, and implies e.g. the result of [3]. At the other extreme, we get direct data structure lower bounds leading to a strengthening to the decision problem of our results in [18]. Our work in fact shows that all near neighbor lower bounds follow from basic expansion properties of the metric space. Vertex expansion translates to lower bounds for deterministic data structures. Edge expansion can be translated to lower bounds for randomized data structures, and this lets us strengthen [18]. It also allows us to prove lower bounds when the dimension is $o(\log n)$. We also identify a new (to our knowledge) graph parameter that interpolates between vertex and edge expansion, that we call *robust expansion*. We show that robust expansions suffices to prove NNS lower bounds. Additionally, for random inputs in highly symmetric metrics, robust expansion also translates to upper bounds in the cell probe model, that match our lower bounds for constant t. Thus, in many cases, NNS lower bounds *imply* isoperimetric inequalities. Finally, we present a natural conjecture regarding the complexity of approximate near neighbor search and show tight bounds for dynamic low contention data structures.

A. Basic Definitions

The *Near Neighbor Problem* is parameterized by a number r. As in the Nearest Neighbor Search Problem the input to

the preprocessing phase is a data set of n points in a metric space. Given a query point y the goal is to determine whether the data set contains a point of distance at most r from y. In the approximation version (ANNS) the preprocessing phase receives as input also an approximation ratio c. Given a query point y the goal is to differentiate between the case where the closest data set point is of distance at most r from y, to the case where the closest data set point is of distance at least cr from y. Clearly a lower bound for these problems holds also for nearest neighbor search.

We prove lower bounds for a generalization we call Graphical Neighbor Search (GNS) which we define shortly. We then show that lower bounds on GNS imply ANNS lower bounds. In the GNS problem we are given an undirected bipartite graph $G = (U, V, E)$ where the data set comes from U and queries come from V. For a node u the set $N(u)$ denotes its neighbors in G. In the preprocessing phase we are given a set of pairs $(x_1, b_1), \ldots, (x_n, b_n)$ where x_i is a vertex in U and $b_i \in \{0, 1\}$. The goal is to build a data structure such that given a node $y \in V$, if there is a unique i such that $y \in N(x_i)$ then it is possible to query the data structure t times and output b_i. If there is no such i or it is not unique any output is considered correct.

We observe that ANNS reduces to GNS when assuming a query point is at distance at most r from some x_i and a least cr from all other x_j. In this case we have the nodes of U and V correspond to the points in the metric space, and the set of edges consists of all pairs of nodes at distance at most r. A formal reduction is proven in Section IV where we also show that average instances of ANNS translate to average instances of GNS for which our lower bounds hold. The bounds we show depend only on the expansion properties of G. We need the following definitions:

Definition 1 (Vertex expansion)**.** *Let μ be a probability measure over U and ν be a probability measure over V. The $\delta-$vertex expansion of the graph with respect to μ, ν is defined as*

$$\Phi_v(\delta) := \min_{A \subset V, \nu(A) \leq \delta} \frac{\mu(N(A))}{\nu(A)}.$$

The vertex-expansion Φ_v is defined as the largest k such that for all $\delta \leq \frac{1}{2k}$, $\Phi_v(\delta) \geq k$.

Let $A \subset V$, $B \subset U$ and $\delta = \nu(A)$. Observe that if $E(A, B) = E(A, U)$ then $\mu(B) \geq \Phi_v(\delta)\nu(A)$. In other words $\Phi_v(\delta)$ bounds the measure of the sets that cover all the edges incident on a set of measure δ. The notion of *robust expansion* relaxes this by requiring B to cover at least a γ-fraction of the edges incident on A. This idea is captured in the definition below. For simplicity we assume that $V = U$ and that μ and ν are the uniform distribution and that G is regular. A more subtle definition which takes into account other measures is presented in Section III.

Definition 2 (Robust expansion)**.** *G has robust-expansion $\Phi_r(\delta, \gamma)$ if $\forall A, B \subseteq V$ satisfying $|A| \leq \delta|V|, |B| \leq \Phi(\delta, \gamma)|A|$, it is the case that $\frac{|E(A,B]|}{|E(A,V)|} \leq \gamma$. Note that $\Phi_r(\delta, 1) = \Phi_v(\delta)$.*

B. Our Contributions

1) Bounds for Deterministic Algorithms: In this section we require that the algorithm always output the correct answer. We show time space tradeoffs based on the vertex expansion properties of G. Our lower bounds are in the average case. Given a distribution μ over U, a data set is built by sampling n data set points independently from μ.

Note that in order for the problem to be interesting we must have that $N(x_i)$ and $N(x_j)$ are likely to be disjoint. We thus have the following definition:

Definition 3. *A distribution μ over U is said to be* strongly independent *for G if*

$$\Pr_{\substack{x \sim \mu \\ z \sim \mu}} \{N(x) \cap N(z) \neq \emptyset\} \leq 1/100n^2.$$

Note that if μ is strongly independent and $x_1, \ldots, x_n$ are sampled independently by μ then with probability at least 0.99 $N(x_i) \cap N(x_j) = \emptyset$ for all $i \neq j$. In the following m denotes the number of cells in the data structure and w denotes the word size in bits, t is the number of cell probes used by the algorithm.

Theorem 4. *For a given G, let μ, ν be probability measures such that μ is strongly independent, and the vertex expansion with respect to μ, ν is $\Phi_v(\cdot)$. Then any deterministic algorithm solving GNS must satisfy the following inequalities*

$$\left(\frac{mwt}{n}\right)^t \geq \Phi_v \tag{1}$$

$$\frac{m^t t w}{n} \geq \Phi_v(1/m^t) \tag{2}$$

These theorems, combined with known isoperimetric inequalities yield most known cell probe lower bounds for near neighbor problems, and generalize them to general expanding metrics. To see this consider for example the $d-$dimensional hypercube equipped with the Hamming distance. It is shown in [20], [14] that any deterministic solution for ANNS with approximation $1/\epsilon$ must satisfy $t \geq d\epsilon^3/\log(mwd/n)$. This bound can be slightly improved by creating the following GNS instance: Let U and V both equal the set of nodes of the hypercube, and let $E = \{(u, v) \ : \ |u - v|_1 \leq \epsilon d\}$. Let μ and ν be the uniform distribution. Chernoff bounds implies that for $d = \Omega(\log n)$, $|u - v|_1 \geq 0.49d$ with overwhelming probability, so (G, μ) is a strongly independent instance. A lower bound on this instance of GNS implies a lower bound on ANNS with approximation $1/\epsilon$.

Now we use known isoperimetric properties: Harper's theorem (see e.g. [6]) implies that there is a constant $a > 1$ such that $\Phi_v \geq a^{\epsilon^2 d}$. Plugging this in (1) we have that $t \geq d\epsilon^2 \log a/\log(mwd/n)$. In Section IV we discuss how to apply these theorems in greater length.

2) Bounds for Randomized Algorithms: Assume that G is regular. Let x and z be vertices drawn uniformly at random, and y be a random neighbor of x. We say G has the property of being *weakly independent* if $\Pr[y \in N(z)] \leq \gamma/n$ for a small enough constant γ.

Theorem 5. *There exists an absolute constant γ such that the following holds. Any randomized algorithm for a weakly independent instance of GNS which is correct with probability at least half (where the probability is taken over the sampling of the input and the algorithm), satisfies the following inequalities:*

$$(\frac{mwt^4}{n})^{2t} \geq \Phi_r(\frac{1}{m}, \frac{\gamma}{t}) \quad (3)$$

$$\frac{m^t w}{n} \geq \Phi_r(\frac{1}{m^t}, \frac{\gamma}{t}) \quad (4)$$

As an example, we show in Section IV that for the Hypercube with $E = \{(u, v) : |u - v|_1 \leq (\frac{1}{2} - \epsilon)d\}$, the robust expansion $\Phi_r(\frac{1}{m^t}, o(1)) \geq \frac{1}{m^{t(1-4\epsilon^2)}}$. For $d = \Omega(\log n/\epsilon^2)$, the weak independence property is easy to verify. Plugging this into Equations 4, we conclude that $m^{4t\epsilon^2} w \geq n$ so that $m \geq (\frac{n}{w})^{\frac{1}{4t\epsilon^2}}$. This result was previously shown by [3] for slightly larger d.

Our framework suggests a natural conjecture on the complexity of approximate near neighbor problems.

Conjecture 6. *Any randomized t-probe datastructure for a weakly independent GNS instance must satisfy $\frac{mw}{n} t \geq \Phi_r(\frac{1}{m}, \frac{1}{2t})^{\Omega(1)}$.*

We point out that for some interesting metric spaces such as the Hamming cube and Euclidean space, the known upper bound matches the lower bound in the conjecture for a wide range of parameters. We next present some evidence in support of this conjecture.

3) An Upper Bound: There are cases where the bounds above are known to be tight when $t = O(1)$. We show that this is no coincidence: In Section V we show that if G is symmetric, there is an algorithm in the cell probe model that solves random instances of GNS using space that matches the lower bound in equation (4) for $t = 1$.

4) Dynamic Data Structure: In the dynamic version of the problem we want the data structure to support the operation of inserting and deleting a point in the data-set. Let t_U denote the update time. A weaker version of the conjecture is the following:

Conjecture 7. *For any dynamic randomized t-probe datastructure for weakly independent GNS on n points, it holds that $t_U t \geq \Phi_r(\frac{1}{nt_U}, \frac{1}{2t})^{\Omega(1)}$*

To see why this conjecture follows from the stronger one, observe that a data structure with update time t_U uses space $mw \leq nt_U$ after n inserts. We show that this weaker conjecture holds for a restricted family of algorithms which we call *low contention*; i.e., on those where no memory location of the data structure is accessed by too many query points (see Section VI for a formal definition). While this may seem like a severe limitation, we remark that known LSH data structures, and our upper bound in Section V, are in fact dynamic and low contention under our definition.

We show that

Theorem 8. *For any low contention, dynamic t-probe datastructure for GNS on n points, the update time is at least $\Omega\left(\Phi_r(\tau, \frac{1}{4t^2})/32t^4\right)$.*

Plugging in the expansion of the hypercube, we see that for a wide range of parameters Locality Sensitive Hashing is *optimal* for the class of the low contention dynamic data structures over the hypercube.

C. Related Work

Most previous papers are concerned with the Hamming distance over the d-dimensional hypercube. The cases of exact or deterministic algorithms were handled in a series of papers [8], [7],[14], [4]. These lower bounds hold for any polynomial space. In contrast the known upper bounds are both approximate and randomized, and with polynomial space can retrieve the output with one query. Chakrabarti and Regev [9] allow for both randomization and approximation, with polynomial space and show a tight bound for the *nearest* neighbor problem. Pătraşcu and Thorup[20] showed lower bounds on the query time of near neighbor problems with a stronger space restriction (near linear space), although their bound holds for deterministic or exact algorithms. The metric ℓ_∞ is considered in an intriguing paper by Andoni *et al.* [1] who prove a lower bound for deterministic algorithms. The paper uses the richness lemma though the crux of the proof is an interesting isoperimetric bound on ℓ_∞ for a carefully chosen measure.

We are aware of only two papers which prove time-space lower bounds for near neighbor problems where both randomization and approximation are allowed.

Andoni, Indyk and Pătraşcu [3] show that for small $\epsilon > 0$, any $O(1)$-probe algorithm for $(1+\epsilon)$-approximate near neighbor problem must use space $n^{\Omega(\frac{1}{\epsilon^2})}$. This bound is tight for small enough $\epsilon > 0$ [3]. Panigrahy *et al.* [18] show that space $n^{1+\Omega(\frac{1}{\epsilon t})}$ is needed for any algorithm with t queries and ϵ approximation, for the search version of the problem. This bound is tight for constant t.

With the exception of [18] all previous bounds were proven using communication complexity framework [15], and in particular the richness lemma.

a) Comparison to [18]:: While there is some overlap in the techniques between this work and [18], the current work is much more general, and stronger even for the special case (our lower bound now applies to the decision version of NNS). We show that expansion may serve as a single explanation that unifies all previous results, and also gives a simple recipe to prove lower bounds for other metrics. While [18] essentially contained a version of the lower bound (3) with the edge expansion, we are now able to additionally show (3). Additionally, we can use vertex expansion to show lower

bounds for deterministic data structures. Moreover, we show that the randomized lower bounds hold under the much weaker notion of robust expansion. As we discuss in Section I-E, this strengthening is provably needed for deriving the right lower bound for the $(1+\epsilon)$-approximation range for the Hypercube. We remark that both (2) and (4) hold for communication protocol. While we do not know if (1) and (3) hold for communication protocols, our proofs do shed some light on how the two approaches differ, and make clearer how the data structure is used in proving our lower bound.

b) Restricted Models:: Higher lower bounds may be achieved when considering models which are more restricted than the cell probe model. Beame and Vee [5] investigate branching programs. Krauthgamer and Lee [12] show tight upper and lower bounds for the 'black box model' where the algorithm is only allowed to query distances between points of the data set. They show that in this case the complexity of NNS is determined by the intrinsic doubling dimension of the data-set. Motwani, Naor and Panigrahy [16] prove an LSH lower bound for ℓ_1, which has recently been strengthened to the tight bound by O'Donnell, Wu and Zhou [17].

D. Notation and Preliminaries

A data structure for Graph neighbor search is defined as follows. Given a database of n points $x_1, \ldots, x_n \in U$, and $b_1, \ldots, b_n \in \{0,1\}$ the preprocessing algorithm computes a set of t tables $T_1, \ldots, T_t$, where each table stores m words of w bits each. We often call each such word a *cell* of the table. In practice there is only one table, but for notational convenience and with out loss of generality we let the data structure construct a different table for each query.

The query algorithm is specified by t lookup functions $F_1, \ldots, F_t$, where F_i takes in the query point y and $(i-1)$ words of w bits each, and outputs an integer in $[m]$, and function $F_* : V \times (2^w)^t \to \{0,1\}$. On a query y, the data structure looks up $c_1 = T_1[F_1(y)]$, $c_2 = T_2[F_2(y, c_1)], \ldots, c_t = T_t[F_t(y, c_1, \ldots, c_{t-1})]$. Finally it computes $F_*(y, c_1, \ldots, c_t)$. Note that the lookup functions, F_i's and F_* are fixed independent of the database, only the tables $T_1, \ldots, T_t$ can depend on $x_1, \ldots, x_t, b_1, \ldots, b_t$. We say the algorithm is *non adaptive* if the lookup functions are independent of the content of the tables, i.e. of the c values.

E. Overview of Techniques

The core idea behind our approach is quite simple. We demonstrate it by showing a simple argument that the vertex expansion of G provides a lower bound on the space of 1-probe data structures for deterministic algorithms. By the definition of vertex expansion, every set of $|V|/\Phi_v$ nodes is incident to at least half of the nodes of G. Let L be a uniformly random sample of a $1/\Phi_v$ fraction of the cells of the table T, and let Q be the set of nodes in V for which the algorithm probes a cell in L. Clearly Q is expected to contain a $1/\Phi_v$ fraction of the nodes in G. Now consider a sample data set $(x_1, b_1), \ldots, (x_n, b_n)$ where $x_1, \ldots, x_n$ are randomly sampled nodes in the graph and $b_1, \ldots, b_n$ are random bits. With overwhelming probability at least a quarter of the x_i's have a neighbor in the set Q, and thus the random bits associated with these points should be retrievable from the contents of L alone. We conclude that the total number of bits in L is at least $n/4$ and thus the space of the data structure is at least $n\Phi_v/4$ bits.

This basic sampling approach for 1-probe data structures can be extended to t-probe data structures in two different ways.

Cell Sampling: Here we sample a $\Phi_v^{-\frac{1}{t}}$ fraction of the cells in each table. Thus a $1/\Phi_v$ fraction of V is expected to access only the sampled cells. This immediately gives bound (1) for non-adaptive algorithms.

Path Sampling: Here we sample a *path* as follows: we pick a cell randomly from the first table so that a $\frac{1}{m}$ fraction of the vertices Q_1 lookup this cell. Then we sample a cell from the second table in such a way that a $\frac{1}{m}$ fraction of Q_1 looks up this cell in the second read, and so on. This immediately leads to the lower bound in (2) for non-adaptive algorithms.

We remark that the path sampling approach actually leads to communication complexity lower bounds for the 2-player version of the problem where Alice has the query point and Bob has the database. Any t-probe data structure with m cells of w words each implies the existence of a t-round communication protocol where Alice sends $\log m$ bits, and Bob sends w bits, in each round. A communication protocol has more freedom however; unlike in a data structure, where the same table T_2 is used to answer any second query, in a communication protocol, the message Bob sends in the second round may depend not just on the second message from Alice, but also on the first. Path sampling can be immediately translated to a "transcript sampling" technique and thus gives lower bounds for communication protocols. There is no similarly obvious translation for cell sampling.

We can extend these ideas and provide lower bounds for adaptive algorithms by observing the following two facts. Firstly, for a fixed data structure, the probability over a random data set that the data structure succeeds is exponentially small in n. On the other hand, the number of bits read by the sampling procedures above is sublinear, thus the number of all possible non-adaptive algorithms is sub exponential. Informally, this allows us to do a union bound over all possible values of the bits read.

In randomized algorithms not all points in $N(x)$ are good query points for x. In particular, the specific query point that queries the cells that are sampled may be a point on which the algorithm errs. The notion of *shattering* plays a major role in extending the bound for this case: Given any fixed partitioning $A_1, \ldots, A_m$ of V such that each set is of cardinality $O(|V|/m)$, a randomly chosen x has (with high probability) the property that $\max_i |N(x) \cap A_i|$ is at most $|N(x)|/K$, for a K that depends on the edge expansion. In other words, $N(x)$ is *shattered* by the partitioning. Given that the lookup algorithm is correct for a large fraction of $N(x)$,

shattering suffices to show that the algorithm still gives the right answer for a *majority* of the points in $N(x)$ which can be looked up from the cell sample (or the path sample).

In order to prove lower bounds for randomized adaptive algorithms we need to combine the ideas outlined in the two previous paragraphs, which requires more work. Intuitively, for every x such the $N(x)$ is shattered, and for any fixed subset $N'(x)$ on which the algorithm succeeds, the sampling is very likely to recover the correct answer. Moreover, for every collection of bits read, almost all points shatter. While it would be tempting to use a union bound at this point, that does not quite work. Informally, there are dependencies everywhere: the part of $N(x_i)$ that the algorithm gets right depends on all the other x_j's, the bits that are read depend on the sampled cells, etc. The proof carefully defines a notion of shattering that depends only on the x's and not on $N'(x)$'s and argues (over the randomness in picking the x_i's) that most points get shattered. Separately, we argue that for a point that gets shattered, and for any fixed $N'(x)$, the majority answer is correct with high probability (over the sampling procedure alone).

The notion of edge expansion does not quite suffice: for the hypercube when $r = (\frac{1}{2} - \epsilon)d$, for fixed partitioning $A_1, \ldots, A_m$ of V into cells of size $|V|/m$, the largest $|N(x) \cap A_i|/|N(x)|$ is likely to be quite large ($\approx \frac{1}{m^\epsilon}$), whereas we would need it to be $\frac{1}{m^{\epsilon^2}}$ to get the correct bound. The definition of robust expansion Φ_r comes to our rescue here. We can show that while the largest $|N(x) \cap A_i|$ is usually large, the large pieces account for a very small fraction of $N(x)$. In fact, after removing a vanishingly small fraction of $|N(x)|$, every other piece is only about $\frac{1}{m^{\epsilon^2}}$. We show that our lower bound proofs are robust enough to handle this weaker notion of shattering.

While our techniques do not improve the dependency on the query time t, they overcome some of the inherent obstacles in the richness method, so for instance, strengthening the isoperimetric bound of ℓ_∞ would imply that the bound in [1] extends to randomized algorithms as well.

II. Deterministic Lower bounds

In this section we prove Theorem 4. The analysis of deterministic algorithms involves node expansion and does not require shattering. It allows us to demonstrate the techniques of cell sampling and path sampling in a simple setting.

A. Cell Sampling

The following theorem is a restatement of inequality (1)

Theorem 9. *Let μ be such that (G, μ) is a strongly independent instance, and Φ_v be the vertex expansion with respect to μ, ν. Then any deterministic algorithm solving GNS must satisfy $(\frac{8mwt}{n})^t \geq \Phi_v$.*

Proof: Recall that T_i represents a table with m cells, from which the $i'th$ query reads, and $F_i : V \to T_i$ denotes the i'th lookup function. We will state a procedure that obtains a set of at most $tm/\Phi_v^{1/t}$ cells so that at least a $1/\Phi_v$ fraction of the query points only access these cells. We call this procedure *cell sampling*. Note that here this procedure is entirely deterministic. A probabilistic variant is used in the next section.

Cell sampling procedure: The cells are obtained iteratively in t phases, each phase corresponds to a query of the table. In each phase at most $m/\Phi_v^{1/t}$ cells are chosen. The first lookup function F_1 induces a partition over V. The set L_1 is chosen to be the $m/\Phi_v^{1/t}$ cells in T_1 that maximize $\nu(F_1^{-1}(L_1))$. In other words, the first lookup function partitions V according to its image in T_1. We choose L_1 to be the cells corresponding to the $m/\Phi_v^{1/t}$ largest partitions, as measured by ν. Set $Q_1 \subset V$ to be those vertices; i.e. $F_1^{-1}(L_1)$. The selection process continues iteratively in a similar manner. Let L_i denote the set of cells obtained in the i'th phase and let $Q_i \subseteq V$ denote the set of vertices that (if given as a query) access only cells in $L_1, \ldots, L_i$ In the $(i+1)$'th phase we consider F_{i+1} and set L_{i+1} to be the $m/\Phi_v^{1/t}$ cells with highest measure, where we restrict ν to Q_i. In other words, when measuring $F_i^{-1}(L_{i+1})$ we assign a measure of 0 for vertices outside Q_i. It is easy to inductively argue that $\nu(Q_i) \geq \Phi_v^{-i/t}$, so that $\nu(Q_t) \geq 1/\Phi_v$ and thus $\mu(N(Q_t)) \geq \frac{1}{2}$.

Intuitively, the set of cells in $L_1, \ldots, L_t$ encode half of the b_i bits and therefore must contain $\Omega(n)$ bits, which would imply the lower bound. Of course, $L_1, \ldots, L_t$ depends upon the content of the table which depends upon the points in the data set. These dependencies can be handled by using a union bound over all the possible values of $L - 1, ..., L_t$: To see this, fix the values written in the cells $L_1, \ldots, L_t$ to some string ω and sample the n data set points from U independently according to μ. Let A_ω denote the event that when the value of the cells $L_1, \ldots, L_t$ is ω, an algorithm reading ω succeeds in guessing the b_i bits for the data set points that fall in $N(Q_t)$. Note that Q_t depends only upon ω and that since the procedure of obtaining L_t is deterministic, the locations of the cells obtained also depends only on ω. Vertex expansion implies that $\mu(N(Q_t)) \geq \frac{1}{2}$. Also, since μ is strongly independent and the algorithm is assumed to be correct, there is a ω for which it is correct, i.e. $\Pr[\cup_\omega A_\omega] \geq \frac{1}{2}$. By Chernoff's bound, the probability that less than $n/8$ points fall in $N(Q_t)$ is at most $2^{-n/8}$. Note that the b_i's are chosen independently, therefore, for a fixed table, if $n/8$ points indeed fall in $N(Q_t)$, then the probability that the sampled b_i's match the output of the algorithms is $2^{-n/8}$. We conclude that $\Pr[A_\omega] \leq 2^{-n/8} + 2^{-n/8}$. Now, let $K = 1/\Phi_v^{1/t}$. There are 2^{Kmtw} ways of choosing ω, so we must have $2^{1-n/8}2^{Kmtw} \geq \frac{1}{2}$. We conclude that $Kmtw \geq n/8$ which implies the theorem. ∎

B. Path Sampling

We now prove Inequality (2).

Theorem 10. *Let μ be strongly independent, and Φ_v be the vertex expansion with respect to μ, ν, then any data structure with a deterministic querying algorithm must satisfy $\frac{9m^t wt}{n} \geq \Phi_v(1/m^t)$.*

Proof: The proof is similar to that of Theorem 9 with a different choice of parameters. We present it here separately because the two approaches diverge in the next section when we deal with the randomized case. We use the cell sampling technique to select a set of cells from the tables, only this time, each phase we select *a single* cell from the table (as opposed to selecting $m\Phi_v^{-1/t}$ cells in Theorem 9). We call the approach of sampling a single cell from each table *path sampling* because we sample a single possible "query path" along the t tables. We also observe that lower bounds based on this approach imply communication complexity lower bounds.

Now the contents of $L_1, \ldots, L_t$ are tw bits, and $\nu(Q_t) \geq m^{-t}$ so that $\mu(N(Q_t)) \geq \Phi_v(m^{-t})m^{-t}$. When fixing the bits of L_t to be the string ω, the expected number of data set points that fall in $N(Q_t)$ is at least $n\Phi_v(m^{-t})m^{-t}$. Define A_ω as before and recall that we have $\Pr[\cup A_\omega] \geq \frac{1}{2}$. Let Z_ω be the number of data set points falling in $N(Q_t)$. Chernoff's bound implies that $\Pr[Z_\omega \leq \frac{1}{2}n\Phi_v(m^{-t})m^{-t}] \leq 2^{-\Phi_v(m^{-t})m^{-t}/8}$. Since the string ω now encodes Z_ω random bits, $\Pr[A_\omega] \leq 2^{-\Phi_v(m^{-t})m^{-t}/8} + 2^{-\Phi_v(m^{-t})m^{-t}/2}$. There are 2^{tw} ways of choosing ω, so we have $2^{tw} \cdot 2^{-\Phi_v(m^{-t})m^{-t}/8} \geq \frac{1}{2}$ which implies the theorem. ∎

III. Randomized Lower bounds

A. Preliminaries

To prove lower bounds for randomized data structure, we will use Yao's minimax theorem, and instead show a distribution over instances such that for some constant $\delta > 0$, any deterministic t-probe data structure that succeeds with probability $(1-\delta)$ needs large space.

We consider the following randomized version of the *Graph Neighbor Search (GNS)* problem on a bipartite graph $G = (U, V, E)$. We are given a set of n tuples $(x_1, b_1), \ldots, (x_n, b_n)$, where $x_i \in U$ and $b_i \in \{0, 1\}$ to preprocess into a data structure. Then given a query $y \in V$, the query algorithm makes t probes into the data structure, and is expected to return b_i if x_i is the unique neighbor in G of y in $\{x_1, \ldots, x_n\}$. If there is no unique neighbor, any output is considered valid.

Let e be a probability distribution over E. Let $\mu(u) = e(u, V) = \sum_{v \in V} e(u, v)$ be the induced distribution on U, and let $\nu(v) = e(U, v)$ be the induced distribution on V. For $x \in U$, we denote by ν_x the conditional distribution of the endpoints in V of edges incident on u, i.e. $\nu_x(y) = e(x, y)/e(x, V)$. We observe that (G, e) define a distribution over instances of GNS as follows. We select n points $x_1, \ldots, x_n$ independently from the distribution μ uniformly at random and pick $b_1, \ldots, b_n$ independently and uniformly from $\{0, 1\}$. This defines the database distribution. To generate the query, we pick an $i \in [n]$ uniformly at random, and sample y independently from ν_{x_i}.

Definition 11. *We say the tuple (G, e) satisfies* γ-weak independence (WI) *if* $\Pr_{x,z\sim\mu, y\sim\nu_x}[(y, z) \in E] \leq \frac{\gamma}{n}$.

In other words, WI ensures that with probability $(1-\gamma)$, for the instance generated as above, x is indeed the unique neighbor in G of y in $\{x_1, \ldots, x_n\}$.

We next define the notion of expansion that we use. Recall that the vertex expansion of a set $A \subseteq V$ in an unweighted graph G is the ratio $\frac{|B|}{|A|}$, where $B = N(A)$ is the smallest set such that all edges incident on A are captured by B, i.e. $|E(B, A)| = |E(U, A)|$. A relaxation of this definition, which we call γ-robust expansion, is the ratio $\frac{|B|}{|A|}$ where B is now the smallest set that captures a γ fraction of the edges incident on A, i.e. $e(B, A) \geq \gamma e(U, A)$. The following definition generalizes this notion to weighted bipartite graphs.

Definition 12. *[Robust Expansion] The γ-robust expansion of a set $A \subseteq V$ is defined as $\phi_r(A, \gamma) \stackrel{def}{=} \min_{B\subseteq U: e(B,A)\geq\gamma e(U,A)} \mu(B)/\nu(A)$.*

Let w be an auxiliary weight function on U with $\sum_{u\in U} w(u) = 1$. The γ-robust expansion with respect to w is defined as $\phi_r^w(A, \gamma) \stackrel{def}{=} \min_{B\subseteq U: e(B,A)\geq\gamma e(U,A)} w(B)/\nu(A)$.

We say that (G, e) has (β, γ)-robust expansion $\phi_r^w = \phi_r^w(\beta, \gamma)$ at least K if for every subset $A \subseteq V$ such that $\nu(A) \leq \beta$, we have $\phi_r^w(A, \gamma) \geq K$.

For intuition, consider the setting where $G = (U, V, E)$ is derived naturally from an undirected graph $H = (V_H, E_H)$ by making two copies of V_H and for each edge $(u, v) \in E_H$, placing the edges (u_1, v_2) and (v_1, u_2). Formally, $U_G = V_H \times \{1\}, V_G = V_H \times \{2\}$, and $E_H = \{((u, 1), (v, 2)) \in U_G \times V_G : (u, v) \in E_H\}$. Then for any set $A \subseteq V$, we have $\phi_r^w(A, 1) = w(N(A))/\nu(A)$, which is the vertex expansion of A in H under ν, for $w = \nu$. Similarly, if a set A has conductance $e(A, A^c)/e(A, V_H)$ at most $1-\gamma$, then $e(A, A) \geq \gamma e(A, V)$ so that $\phi_r^w(A, \gamma) \leq w(A)/\nu(A)$. A similar correspondence holds for directed graphs.

B. Main Result

The main result of this section is the following.

Theorem 13. *There exists an absolute constant γ such that the following holds. Let (G, e) satisfy γ-weak independence (WI). Then, any deterministic t-probe data structure for the distribution over GNS instances defined by (G, e) that succeeds with probability $(1-\gamma)$ must satisfy*

$$(\frac{mwt^4}{n})^{2t} \geq \Phi_r^w(\frac{1}{m}, \frac{\gamma}{t}) \tag{5}$$

$$\frac{m^t w}{n} \geq \Phi_r^w(\frac{1}{m^t}, \frac{\gamma}{t}) \tag{6}$$

where w is an arbitrary auxiliary weight function, and t is $o(n^{\frac{1}{4}})$.

The theorem's proof can be found in the full version of the paper [19].

IV. Applications

We show how lower bounds on GNS imply lower bounds for ANNS. We stress that these bounds hold for the *average case* where the n data-set points are sampled randomly from a distribution over $|V|$. Thus, if with high probability the distance between all pairs of points in the data set is at least cr, then the bounds above hold also for the approximate nearest

neighbor within factor c. We remark that while bounds for ℓ_2 are implied by the bounds for ℓ_1, our method allows us to compute those bounds directly and thus obtain an lower bound on a natural *average* case.

A. GNS to ANNS

In the decisional version of the (c, r)-ANNS problem we have a metric space $\mathcal{M}$ and parameters c and r. We preprocess n points $x_1, ..., x_n$ into a data structure. When given a query point y the goal is to distinguish between the case where $d(x_i, y) \leq r$ for some $i \in [n]$, and the case where for all i $d(x_i, y) \geq cr$. The query algorithm is required to output 1 in the former case, 0 in the latter case, and may report anything if neither of the two cases hold.

We show that if we have appropriate distributions over $\mathcal{M}$, we can derive lower bounds for (c, r)-ANNS by simply computing the relevant expansion parameter.

Theorem 14 (GNS to ANNS Deterministic)**.** *Let $c \geq 1$ and let μ be a distribution over a metric $\mathcal{M} = (V, d)$ satisfying:*

$$(c\text{-Strong Independence}) \quad \Pr_{x,z\sim\mu}[d(x, z) \leq (c + 1)r] \leq \frac{1}{100n^2}.$$

Let $G_r = (V, \{(u, v) : d(u, v) \leq r\})$. Then GNS on (G_r, μ) reduces to (c, r)-approximate GNS on $\mathcal{M}$.

Proof: Given a GNS instance $(x_1, b_1), \ldots, (x_n, b_n)$, we consider the dataset $D_1 = \{x_i : b_i = 1\}$ as our input for the (c, r)-ANNS problem. It is easy to see that when $x_i \sim \mu$ and $b_i \sim \{0, 1\}$, this set D_1 is a uniformly random dataset from μ. The c-Strong independence implies strong independence for the GNS instance. Whenever $d(x_i, x_j) > (c + 1)r$, and $d(x_i, y) \leq r$, we have $d(x_j, y) > cr$ so that for the c-approximate NNS instance, the answer is 1 if and only x_i is in D_1, i.e. if and only if $b_i = 1$. The claim follows. ∎

Thus to prove deterministic data structure lower bounds for c-approximate NNS, it suffices to exhibit r and a distribution μ which satisfies c-strong independence and has large expansion.

Similarly

Theorem 15 (GNS to ANNS Randomized)**.** *Let $c \geq 1$ and let e be a distribution over pairs of points in a metric $\mathcal{M} = (V, d)$. Let $\mu(x) = e(x, V)$ and $\nu(y) = e(V, y)$. Suppose that for small enough γ*

$$(c\text{-Weak Independence}) \qquad \Pr_{y\sim\nu,z\sim\mu}[d(y, z) \leq cr] \leq \frac{\gamma}{n},$$

and

$$\Pr_{(x,y)\sim e}[d(x, y) \leq r] \geq 1 - \gamma.$$

Then GNS on (G, e) reduces to (c, r)-approximate GNS on $\mathcal{M}$.

Proof: As before, given a GNS instance $(x_1, b_1), \ldots, (x_n, b_n)$, we consider the dataset $D_1 = \{x_i : b_i = 1\}$ as our input for the (c, r)-ANNS problem. It is easy to see that when $x_i \sim \mu$ and $b_i \sim \{0, 1\}$, this set D_1 is a uniformly random dataset from μ. The properties above imply weak independence for the GNS instance. Finally, except with small probability, we have $d(x_j, y) > cr$ for all $j \neq i$, so that for the c-approximate NNS instance, the answer is 1 if and only x_i is in D_1, i.e. if and only if $b_i = 1$. The claim follows. ∎

Thus to prove randomized data structure lower bounds for c-approximate NNS, it suffices to exhibit r and a distribution e which satisfies the above properties and has large expansion.

B. Computing Expansion

Next we bound the expansion of the hypercube for appropriate distributions, which would imply the claimed lower bounds for the Hypercube. The vertex expansion result for ℓ_∞ in [1] will lead to the cell probe lower bounds for ℓ_∞ proved in their work.

We will set μ to be the uniform distribution over the hypercube. For a set $A \subseteq H$, we let $a = \mu(A)$. Observe that if we take n uniformly random points from a d dimensional hypercube then with high probability all pairs of points are at least $d/2 - O(\sqrt{d \log n})$ apart. Using bounds for the expansion for the (G_r corresponding to the) d-dimensional hypercube, we will derive lower bounds for the near neighbor problem on the hypercube.

The following lemma is proved in [6]

Lemma 16 (Vertex Expansion of Hypercube)**.** *Let $H = \{0, 1\}^d$ be the boolean hypercube, and let G_r be the graph with the edge set $E_r = \{(u, v) : |u - v|_1 \leq r\}$. Let $h_i = \frac{1}{2^d} \sum_{j=0}^{i} \binom{d}{j}$. Then the vertex expansion $\Phi_v(h_i) \geq h_{i+r}$*

Setting $r = d/3$ we see that the node expansion $\Phi_v \geq h_{d/2}/h_{d/2-d/3} = 2^{\Omega(d)}$. From theorem 4 we get $(mwt/n)^2 \geq 2^{\Omega(d)}$ or $t \geq d/log(mwd/n)$. This gives us the deterministic lower bound for $2-$approximation in ℓ_1 norm. Setting $r = \epsilon d/2$ gives us an a lower bound for $O(\epsilon)-$ approximation. For this value of r, $\Phi_v \geq h_{d/2}/h_{(1-\epsilon)d/2} = 2^{\Omega(\epsilon^2 d)}$. So we get the bound of $t \geq \epsilon^2 d/log(mwd/n)$.

For randomized lower bounds, we will use the distribution defined by the noise operator T_ρ where $\rho = (1 - \frac{2r}{d})$. I.e. to sample from e, we sample x from the uniform distribution μ and sample y by flipping each bit of x independently with probability $\frac{(1-\rho)}{2} = \frac{r}{d}$. It is easy to check that for any $r \leq (\frac{1}{2} - \epsilon)d$ and d being $\Omega(\log n/\epsilon^2)$, we indeed have $\Pr_{(x,y)\sim e}[d(x, y) \leq (1 + \frac{\epsilon}{10})r] \geq 1 - \frac{\gamma}{n}$. Moreover, since $\mu = \nu$, by the discussion above, $\Pr_{y\in\nu,z\in\mu}[d(y, z) \leq \frac{d}{2} - O(\sqrt{d \log n})] \leq \frac{\gamma}{n^2}$. Thus it remains to compute the expansion for appropriate r.

It will be convenient to work with the edge expansion.

Definition 17. *We define the edge expansion $\Phi_e(\delta)$ for a (G, e) as $\Phi_e(\delta) = min_{\mu(A)\leq\delta} \frac{e(A,\bar{V})}{e(A,A)}$. Thus for any set of size measure δ, at most $\frac{1}{\Phi_e(\delta)}$ mass of edges incident on A stay within A.*

Observation 18. *For any (G, e), if μ is uniform, then $\Phi_r(\delta, \gamma) = \Omega(\gamma\Phi_e(2\delta))$*

Proof: First we will argue that for any sets A and B where $\mu(A) = \mu(B) \leq \delta$, $e(A, B) \leq 2e(A, V)/\Phi_e(2\delta)$. To see this

note that $e(A,B) \leq e(A\cup B, A\cup B) \leq \frac{1}{\Phi_e(\mu(A\cup B))} e(A\cup B, V) = \frac{1}{\Phi_e(2\delta)} 2e(A,V)$

Now consider any set A of measure at most δ, and let B by any other set of measure $\delta\gamma\Phi_e(2\delta)/2$. We wish to argue that $e(A,B) \leq \gamma e(A,V)$ which would imply the claim.

Let $B_1, \ldots, B_k$ be a partition of B into $k = \lceil \frac{\delta\gamma\Phi_e(2\delta)}{2\delta} \rceil$ pieces of measure δ each. $e(A,B) \leq \sum_i e(A,B_i) \leq \frac{k}{\frac{1}{\Phi_e(\mu(A\cup B))}} \leq \gamma$. The claim follows. ∎

Lemma 19 (Edge expansion of Hypercube)**.** *Let $H = \{0,1\}^d$ be the boolean hypercube, and (G,e) be as above for $r < \frac{d}{4}$. Then the edge expansion $\Phi_e(a) \geq a^{-\Omega(r/d)}$.*

Proof: For sets A, B, it is easy to see that $e(A,B) = \langle T_\rho \mathbf{1}_A, \mathbf{1}_B\rangle$. But by the Hypercontractive inequality,

$$\langle T_\rho \mathbf{1}_A, \mathbf{1}_A\rangle = \langle T_{\sqrt{\rho}}\mathbf{1}_A, T_{\sqrt{\rho}}\mathbf{1}_A\rangle = \|T_{\sqrt{\rho}}\mathbf{1}_A\|_2^2 \leq |1_A|_{1+\rho}^2 = a^{\frac{2}{1+\rho}}.$$

Also $e(A,V) = a$. The claim follows by substituting the value of ρ. ∎

Setting $r = \epsilon d/2$, we get that for constant γ, $\Phi_r(a,\gamma) \geq a^{-\Omega(\epsilon)}$. From theorem 5 (first inequality) it follows that $(mw/n)^t \geq m^{-\Omega(\epsilon)}$ implying $m \geq (n/w)^{1+\Omega(\epsilon/t)}$.

Lemma 20 (Robust expansion of Hypercube)**.** *Let $H = \{0,1\}^d$ be the boolean hypercube, and let (G,e) be as above for $r = (\frac{1}{2} - \epsilon)d$ Then for any sets A and B such that $\mu(B) \leq \mu(A)^{4(1-\frac{2r}{d})^2}$, $e(A,B) \leq \mu(A)^{(1-\frac{2r}{d})^2} de(A,V)$.*

Proof: As in the proof of lemma 19, we use the hypercontractive inequality.

$$\begin{aligned} e(A,B) &= \langle T_\rho \mathbf{1}_A, \mathbf{1}_B\rangle \\ &\leq \|T_\rho \mathbf{1}_A\|_2 \|\mathbf{1}_B\|_2 \\ &\leq \|\mathbf{1}_A\|_{1+\rho^2} \|\mathbf{1}_B\|_2 \\ &= a^{\frac{1}{1+\rho^2}} b^{\frac{1}{2}} \\ &\leq a^{1-\rho^2} a^{2\rho^2} \\ &= a^{\rho^2} e(A,V). \end{aligned}$$

The claim follows. ∎

Corollary 21. *For any $\beta \geq 0$, and $r \leq \frac{d}{2}$, setting $\rho = 1 - \frac{2r}{d}$, $\Phi_{av}(\beta, \beta^{\rho^2}) \geq \beta^{1-4\rho^2}$ for G_r as above.*

Setting $r = (1-\epsilon)d/2$, we see that from theorem 5 that any randomized algorithm must use space $m \geq (\frac{n}{wt^5})^{\frac{4}{t\epsilon^2}}$.

The vertex expansion result for ℓ_∞ has already been computed in the work by [1] for proving cell probe lower bounds for ℓ_∞. They consider the d-dimensional grid $\{0,1,\ldots,m\}^d$ with a non uniform measure defined as follows. The measure π over $\{0,1,\ldots,m\}$ is defined by $\pi(i) = 2^{-(2\rho)^i}$ for all $i > 0$ and $\pi(0) = 1 - \sum_{i>0}\pi(i)$. One then defines $\mu_d(x_1,x_2,\ldots,x_d) = \pi(x_1)\cdot\pi(x_2)\ldots\pi(x_d)$. For this measure μ_d the following expansion theorem was shown in [1].

Lemma 22. *[1] $\mu_d(N(A)) \geq \mu_d(A)^{1/\rho}$*

Now it is easy to show that if n random points are chosen from the measure μ_d then every point is at least $g = \log_{2\rho}(\frac{\epsilon}{4}\log d)$ away from the origin under the ℓ_∞ norm. This is because the probability that a certain coordinate of a point is at least $\Omega(g)$ is at least $1/d^\epsilon$. So probability that no coordinate is more than g is at most $(1-1/d^{\epsilon/4})^d \leq e^{-d^{1-\epsilon/4}}$. For $d = \Omega(\log^{1+\epsilon} n)$ this is at most $1/n^{\Omega(1)}$. So all points will be at least distance g from the origin. In fact this argument also easily shows that they are in g distance from each other. So setting $r = 1$ gives us a lower bound for $(g-1,r)-$ANNS for ℓ_∞. From Theorem 4 it follows that to get a $O(\log_\rho \log d)$ approximation for NNS on ℓ_∞ the amount of space required is at least $m \geq (\frac{n}{wt})^{\rho/t}$.

C. Lower bounds for ANNS in $\Re^d$

We next show that this approach leads to non-trivial lower bounds for 2-approximate NNS in ℓ_2^d when $d = \Omega(\frac{\log n}{\log\log n})$. Let μ be the measure on $\Re^d$ defined by drawing each coordinate independently from the standard normal distribution $N(0,1)$. To generate y given x, we will use Gaussian noise; i.e. for each coordinate i, we set $y_i = \rho x_i + \sqrt{1-\rho^2}N(0,1)$ for an appropriate parameter ρ. The Ornstein-Uhlenbeck operator U_ρ is defined as $(U_\rho f)(x) = \mathbb{E}_{z\in\mu}[f(\rho x + \sqrt{1-\rho^2}z]$. It is known to be hypercontractive, i.e. for any $f : L_2(\mu) \to L_2(\mu)$, we have $\|U_\rho f\|_2 \leq \|f\|_{1+\rho^2}$.

With some foresight we set $\rho = \sqrt{1 - \frac{d}{Cn^{\frac{2}{d}}}}$. Clearly when y is derived by applying noise to x (i.e. $y = \rho x + \sqrt{1-\rho^2}Z$ for an independent sample Z from μ), then $\|x-y\|_2 \leq 2\sqrt{1-\rho^2} \leq \frac{\sqrt{d}}{2cn^{\frac{1}{d}}}$ with probability $(1-\gamma)$, for a suitably large constant C. Moreover, when x' is drawn independently from μ, $\frac{1}{\sqrt{2}}(x'-y)$ is easily seen to be distributed identically to μ so that its ℓ_2 norm has the χ^2 distribution. A simple computation shows that $\|x'-y\|_2$ is at least $\frac{\sqrt{d}}{cn^{\frac{1}{d}}}$ with probability $(1-\frac{\gamma}{n})$. Thus we have weak independence.

Finally, the expansion computation is similar to Lemma 19:

$$e(A,A) = \langle U_\rho \mathbf{1}_A, \mathbf{1}_A\rangle = \langle U_{\sqrt{\rho}}\mathbf{1}_A, U_{\sqrt{\rho}}\mathbf{1}_A\rangle = \|U_{\sqrt{\rho}}\mathbf{1}_A\|_2^2 \leq |1_A|_{1+\rho}^2 = a^{\frac{2}{1+\rho}}.$$

Thus $\mu(A)/e(A,A) \geq a^{-\frac{1-\rho}{1+\rho}} = a^{-\frac{1-\rho^2}{(1+\rho)^2}}$. Substituting the value of ρ, we conclude that for large enough constant K, when $\frac{K\log n}{\log\log n} \leq d \leq \log n$, $\phi_e(\frac{1}{n}) \geq 2^{\Omega(\sqrt{d})}$. Thus any t-probe data structure needs space at least $n2^{\Omega(\sqrt{d}/t)}$.

We make two remarks. Firstly, doing the natural rounding of the real numbers involved implies similar lower bounds for $\mathbb{Z}^d$. Secondly, we observe that choosing a measure different than Gaussian may potentially allow us to prove stronger lower bounds, in particular for dimensions smaller than $\log n/\log\log n$.

V. A Matching Upper Bound

We already know that the lower bound is tight in many specific cases. Here we show that the tightness holds more

generally, for highly symmetric graphs. For such graphs, we show that the notion of robust expansion correctly captures the complexity of GNS for the regime with a constant number of queries.

Let G be an undirected Cayley graph[1], and assume that it has the weak independence property for the uniform distribution. Let m be such that $m = n\Phi_r(\frac{1}{m}, \gamma)$ (denoted by Φ for brevity) where $\gamma \geq \frac{3}{4}$. Below we describe a data structure with m cells and word size $O(\log n)$ which can solve GNS in a single query with constant probability for the hard distribution of inputs. This matches the lower bound of Theorem 5 for the case $t = 1$.

Theorem 23. *Let G be an undirected Cayley graph that has the weak independence property for the uniform distribution. Let m be such that $m = n\Phi_r(\frac{1}{m}, \gamma)$ (denoted by Φ for brevity) where $\gamma \geq \frac{3}{4}$. Then there is a 1-probe data structure that uses m words of $w = \log |G|$ bits each that succeeds with constant probability when the data-set points $x_1, \ldots, x_n$ are drawn randomly and independently, and the query point is a random neighbor of a random x_i.*

We observe that this is the distribution for which we show the lower bound.

Proof: The main idea is to use the low expanding sets in order to construct something similar to a Locality Sensitive Hashing solution. We stress that the upper bound is in the cell probe model, which allows us to ignore the (practically very important) issue of actually computing the LSH efficiently.

Let $A \subset V$ be a set of measure $1/m$ for which the robust expansion is Φ and $m = n\Phi$. By the definition of robust expansion, we know that there is a set B of measure Φ/m such that $|E(A, B)| \geq \gamma |E(A, V)|$. We take m random translations of A and B, denoted by $A_1, \ldots, A_m$ and $B_1, \ldots, B_m$, formally, we sample uniformly m elements $a_1, ..., a_m$ from the group underlying the Cayley graph, and set $A_i = \{u : u = a_i + v, v \in A\}$ and similarly $B_i = \{u : u = a_i + v, v \in B\}$. The translation by a_i is an automorphism that maps A to A_i and B to B_i so for each i $|E(A_i, B_i)| \geq \gamma |E(A_i, V)|$.

We construct a table T with m cells as follows: Given a data set point x, we check for each $i \leq m$ whether $x \in B_i$, and if so we place x in T_i. Note that the measure of each B_i is Φ/m so that the expected number of data set points x_j that fall in B_i is $n\Phi/m$ which is 1. For random data sets, most B_i's will contain at most (say) 10 data-set points. In order to keep the word size small we store at most 10 data set points in each table cell T_i, and assuming that $O(\log n)$ bits suffice to represent a data-set point, we have $w = O(\log n)$.

Now, given a query point y we find an i for which $y \in A_i$ and output the data-set point in $T[i]$ which is closest to y. Note that with constant probability, such an i exists and is unique.

Recall that the data set $x_1, \ldots, x_n$ is obtained by sampling n points uniformly and independently from V. Further, we assume that this distribution is weakly independent; i.e. if x and y are random nodes and z is a random neighbor of x, then $\Pr[z \in N(y)] \leq 1/100n$. Further, the query point is obtained by sampling a random neighbor of a random data set point. Assume that the correct answer is x. Now if $y \in A_i$ (which happens with a constant probability), then the lookup succeeds if $x \in B_i$ and there were less than 10 data set points in B_i. The first event occurs with probability $\gamma \geq \frac{3}{4}$ and the second event occurs independently with probability at least $\frac{3}{4}$ as well. We conclude that the data structure succeeds with constant probability. ∎

[1]We need the graph to be highly symmetric and the symmetries of Cayley graphs are convenient for the claims we need.

VI. Low Contention Dynamic Data Structures

Let Q_l denote the set of queries that read cell $T[l]$ from the table.

Definition 24. *A data structure is said to have contention τ if $\nu(Q_l) \leq \tau$ for all l.*

Say we insert a new point x. After the insertion there would be a subset of $N'(x) \subseteq N(x)$ such that the query algorithm outputs the correct answer for any $y \in N'(x)$. We say the insertion x is successful if the measure of this set under ν_x is at least $1 - \gamma$.

We remark that we look at contention only for the part of the data structure that depends on the database; accesses to any randomness are free.

In the full version of the paper[19] we prove Theorem 8, which states that in such data structures, the update time is at least $\Omega(\Phi_r(\tau, O(\frac{1}{t^2}))/t^4)$.

References

[1] A. Andoni, D. Croitoru, and M. Patrascu. Hardness of nearest neighbor under L-infinity. In FOCS, pages 424–433, 2008.

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

[3] A. Andoni, P. Indyk, and M. Patrascu. On the optimality of the dimensionality reduction method. In FOCS, pages 449–458, Washington, DC, USA, 2006. IEEE Computer Society.

[4] O. Barkol and Y. Rabani. Tighter lower bounds for nearest neighbor search and related problems in the cell probe model. *J. Comput. Syst. Sci.*, 64(4):873–896, 2002.

[5] P. Beame and E. Vee. Time-space tradeoffs, multiparty communication complexity, and nearest-neighbor problems. In STOC, pages 688–697, New York, NY, USA, 2002. ACM.

[6] B. Bollobas. *Combinatorics*. Cambridge University Press, 1986.

[7] A. Borodin, R. Ostrovsky, and Y. Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. In STOC, pages 312–321, New York, NY, USA, 1999. ACM.

[8] A. Chakrabarti, B. Chazelle, B. Gum, and A. Lvov. A lower bound on the complexity of approximate nearest-neighbor searching on the hamming cube. In STOC, pages 305–311, New York, NY, USA, 1999. ACM.

[9] A. Chakrabarti and O. Regev. An optimal randomised cell probe lower bound for approximate nearest neighbour searching. In FOCS, pages 473–482, 2004.

[10] P. Indyk. On approximate nearest neighbors under ℓ_∞ norm. *J. Comput. Syst. Sci*, 63(4):627–638, 2001.

[11] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In STOC, pages 604–613, 1998.

[12] R. Krauthgamer and J. R. Lee. The black-box complexity of nearest-neighbor search. *Theor. Comput. Sci.*, 348(2-3):262–276, 2005.

[13] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In STOC, pages 614–623, New York, NY, USA, 1998. ACM.

[14] D. Liu. A strong lower bound for approximate nearest neighbor searching. *Inf. Process. Lett.*, 92(1):23–29, 2004.

[15] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998.

[16] R. Motwani, A. Naor, and R. Panigrahy. Lower bounds on locality sensitive hashing. In SCG, pages 154–157, New York, NY, USA, 2006. ACM.

[17] R. O'Donnell, Y. Wu, and Y. Zhou. Optimal lower bounds for locality sensitive hashing (except when q is tiny). Manuscript, 2009.

[18] R. Panigrahy, K. Talwar, and U. Wieder. A geometric approach to lower bounds for approximate near-neighbor search and partial match. In FOCS, pages 414–423, 2008.

[19] R. Panigrahy, K. Talwar, and U. Wieder. Lower Bounds on Near Neighbor Search via Metric Expansion. arXiv:1005.0418v1 [cs.DS], 2010.

[20] M. Pătraşcu and M. Thorup. Higher lower bounds for near-neighbor and further rich problems. In FOCS, pages 646–654, 2006.