Hardness of Nearest Neighbor under L-infinity

Alexandr Andoni
MIT

Dorian Croitoru
MIT

Mihai Pătraşcu
MIT

Abstract

Recent years have seen a significant increase in our understanding of high-dimensional nearest neighbor search (NNS) for distances like the ℓ_1 and ℓ_2 norms. By contrast, our understanding of the ℓ_∞ norm is now where it was (exactly) 10 years ago.

In FOCS'98, Indyk proved the following unorthodox result: there is a data structure (in fact, a decision tree) of size $O(n^\rho)$, for any $\rho > 1$, which achieves approximation $O(\log_\rho \log d)$ for NNS in the d-dimensional ℓ_∞ metric.

In this paper, we provide results that indicate that Indyk's unconventional bound might in fact be optimal. Specifically, we show a lower bound for the asymmetric communication complexity of NNS under ℓ_∞, which proves that this space/approximation trade-off is optimal for decision trees and for data structures with constant cell-probe complexity.

1 Introduction

Nearest neighbor search (NNS) is the problem of preprocessing a collection of n points, such that we can quickly find the point closest to a given query point. This is a key algorithmic problem arising in several areas such as data compression, databases and data mining, information retrieval, image and video databases, machine learning, pattern recognition, statistics and data analysis.

The most natural examples of spaces for which NNS can be defined are the ℓ_p^d spaces, denoting the space $\Re^d$ endowed with the distance $\|x-y\|_p = \left(\sum_{i=1}^d |x_i - y_i|^p\right)^{1/p}$. Significant attention has been devoted to NNS in these spaces, with many deep results surfacing for the Euclidean norm, ℓ_2, and the Manhattan norm, ℓ_1. We refer the reader to surveys in [SDI06, Sam06].

The ℓ_∞ metric is the odd-man out in this research direction. The structure of this very natural space, dictated by the max operator $\|x-y\|_\infty = \max_{i=1}^d |x_i - y_i|$, is intriguingly different from the other ℓ_p norms, and remains much less understood.

In fact, there is precisely one worst-case[1] result for NNS under ℓ_∞. In FOCS'98, Indyk [Ind01b] achieves an NNS algorithm for d-dimensional ℓ_∞ with approximation $4\lceil \log_\rho \log 4d \rceil + 1$, which requires space $dn^\rho \log^{O(1)} n$ and $d \cdot \log^{O(1)} n$ query time, for any $\rho > 1$. For 3-approximation, Indyk also gives a $n^{\log d + 1}$ storage algorithm. In the important regime of polynomial space, the algorithm achieves an uncommon approximation factor of $O(\log \log d)$.

Applications of ℓ_∞. For some applications, especially when coordinates are rather heterogeneous, ℓ_∞ may be a natural choice for a similarity metric. If the features represented by coordinates are hard to relate, it is hard to add up their differences numerically, in the sense of ℓ_1 or ℓ_2 (the "comparing apples to oranges" phenomenon). One popular proposal is to convert each coordinate to rank space, and use the maximum rank difference as an indication of similarity. See for example [Fag96, Fag98].

However, the most compelling reasons for studying ℓ_∞ are extroverted, stemming from its importance in a theoretical understanding of other problems. For example, many NNS problems under various metrics have been reduced to NNS under ℓ_∞ via *embeddings* (maps that preserve distances up to some distortion). A well-known result of Matoušek states that *any* metric on n points can be embedded into ℓ_∞ with dimension $d = O(cn^{1/c} \log n)$ and distortion $2c-1$ [Mat96]. In particular, if we allow dimension n, the embedding can be isometric (no distortion). Of course, this general guarantee on the dimension is too high for many applications, but it suggests that ℓ_∞ is a very good target space for trying to embed some particular metric more efficiently.

Early embeddings into ℓ_∞ with interesting dimension included various results for Hausdorff metrics [FCI99], embedding tree metrics into dimension $O(\log n)$ [LLR94], and planar graphs metrics into dimension $O(\log n)$ [KLMN05] (improving over [Rao99]).

More recently, embeddings have been found into generalizations of ℓ_∞, namely product spaces. For a metric $\mathcal{M}$,

[1] Heuristic methods have also been devised. On the theoretical side, [AHL01] analyze a brute-force algorithm for the uniform input distribution, showing a bound of $\Theta(nd/\lg n)$.

the *max-product* over k copies of $\mathcal{M}$ is the space $\mathcal{M}^k$ with the distance function $d_{\infty,\mathcal{M}}(x,y) = \max_{i=1}^{k} d_{\mathcal{M}}(x_i, y_i)$, where $x, y \in \mathcal{M}^k$.

Indyk [Ind02] has extended his original NNS algorithm from ℓ_∞ to max-product spaces. Since max-product spaces are a generalization of ℓ_∞, our lower bound carries over trivially.

Using this algorithm for max-product spaces, Indyk [Ind02] obtained an NNS algorithm for the Frechet metric. In current research, it was shown [AIK08] that algorithms for max-product spaces yield (via a detour through *sum-product* spaces) interesting upper bounds for the Ulam metric and the Earth-Mover Distance. By embedding these metrics into (iterated) sum-products, one can achieve approximations that are provably smaller than the best possible embeddings into ℓ_1 or ℓ_2.

Thus, the bottleneck in some of the best current algorithms for Frechet, Ulam and EMD metrics is the ℓ_∞ metric. In particular, one obtains the same unusual log-logarithmic approximation for polynomial space. Our lower bound is a very timely indication that, if further improvement for these metrics is possible, it has to avoid max-product spaces.

1.1 Lower Bounds

The unusual structure of ℓ_∞ NNS (as evidenced by an uncommon approximation result) and the current developments leading to interesting applications plead for a better understanding of the problem.

The decade that has passed without an improvement seems to suggest a lower bound is needed. However, the only existing lower bound is a simple reduction of [Ind01b] that shows ℓ_∞ with approximation better than 3 is as hard as the partial match problem; see [JKKR04, Pătt08] for partial match lower bounds. This does not explain the most interesting feature of the problem, namely the space/approximation trade-off. It also leaves open the most interesting possibility: a constant factor approximation with polynomial space. (It appears, in fact, that researchers were optimistic about such an upper bound being achievable [Ind01a].)

Communication complexity. As with all known lower bounds for data structures with large space, we approach the problem via *asymmetric communication complexity*. We consider a setting where Alice holds the query point and Bob holds the set D on n database points. Assuming for convenience that the space is discretized, we arrive at the following formal definition:

Definition 1 (c-NNS). *Alice is given a "query point" $q \in \{-m, \dots m\}^d$, and Bob is given the "dataset" $D \subset \{-m, \dots m\}^d$ of size n. Then the c-NNS problem is a promise problem in which the two players must:*

- *output 1 if there exists some $p \in D$ such that $\|q - p\|_\infty \le 1$;*
- *output 0 if, for all $p \in D$, we have that $\|q - p\|_\infty \ge c$.*

We show the following lower bound on the communication complexity of this problem, which is asymptotically optimal by Indyk's algorithm [Ind01b].

Theorem 2. *Fix $\delta, \epsilon > 0$. Consider a dimension d satisfying $\Omega(\log^{1+\epsilon} n) \le d \le o(n)$, and an approximation ratio c satisfying $3 < c \le O(\log\log d)$. Further define $\rho = \frac{1}{2}(\frac{\epsilon}{4}\log d)^{1/c} > 10$.*

In a deterministic protocol solving c-NNS, either Alice sends $a = \Omega(\delta\rho\log n)$ bits, or Bob sends $b = \Omega(n^{1-\delta})$.

Data structures. Asymmetric communication lower bounds imply cell-probe lower bounds for data structures by constructing the natural communication protocol in which Alice sends a cell address in each round, and Bob replies with the cell contents. Thus by a standard analysis of [MNSW98], our communication lower bound implies:

Corollary 3. *Consider any cell-probe data structure solving d-dimensional near-neighbor search under ℓ_∞ with approximation $c = O(\log_\rho \log d)$. If the word size is $w = n^{1-\delta}$ for some $\delta > 0$, the data structure requires space $n^{\Omega(\rho/t)}$ for cell-probe complexity t.*

As with all large-space lower bounds known to date, this bound is primarily interesting for constant query time, and degrades exponentially with t. We expect this dependence on t to be far from optimal, but proving a tight lower bound for superconstant t is well beyond the reach of current techniques.

By another standard reduction to the decision tree model (see [KN97] and Appendix A), we have the following:

Corollary 4. *Let $\delta > 0$ be arbitrary. A decision tree of depth $n^{1-2\delta}$ and node size n^δ that solves d-dimensional near-neighbor search under ℓ_∞ with approximation $c = O(\log_\rho \log d)$, must have size $n^{\Omega(\rho)}$.*

Unlike cell-probe complexity, where the bound degrades quickly with the query time, the lower bound for decision trees holds even for extremely high running time (depth) of $n^{1-\delta}$. A decision tree with depth n and predicate size $O(d \log M)$ is trivial: simply test all database points.

Indyk's result [Ind01b] is a decision tree with depth $d \cdot \log^{O(1)} n$ and predicate size $O(\log(n+M))$, which achieves the same trade-off between approximation and space. Thus, we show that Indyk's interesting trade-off is optimal, at least in the decision tree model. In particular, for polynomial space, the approximation factor of $\Theta(\lg\lg d)$ is intrinsic to NNS under ℓ_∞.

Technical discussion. Perhaps the most innovative component of our lower bound is the conceptual step of understanding why this dependence on the approximation "should" be optimal. In Section 2, we recast Indyk's upper bound idea in an information-theoretic framework that explains the behavior of the algorithm more clearly.

This understanding suggests a heavily biased distribution over the database points, which ellicits the worst behavior. On each coordinate, the probability of some value x decays doubly-exponentially with x, more precisely as $2^{-(2\rho)^x}$. All d dimensions are independent and identically distributed.

By a standard analysis in communication complexity, Alice's communication will fix the query to be in a set S whose measure in our probability space is bounded from below. Technically, the crucial step is to determine the probability that some point in the database lies in the neighborhood of S. The neighborhood $N(S)$ is the Minkowski sum of the set with the ℓ_∞ ball $[-1, 1]^d$. In other words, $N(S)$ is the set of points that *could* be a nearest neighbor. To find an instance for which the algorithm makes a mistake, we must prove a lower bound on the measure of the neighborhood $N(S)$, showing that a point will fall in the neighborhood with good probability.

The crux of the lower bound is not in the analysis of the communication protocol (which is standard), but in proving a lower bound for $N(S)$, i.e. in proving an isoperimetric inequality. Of course, the initial conceptual step of defining an appropriate biased distribution was the key to obtaining the isopermetric inequality that we need. The proof is rather non-standard for an isoperimetric inequality, because we are dealing with a very particular measure on a very particular space. Fortunately, a few mathematical tricks save the proof from being too technical.

The easy steps in communication complexity are described in Section 3 (roughly one page). The isoperimetric inequality is shown in Section 4.

Randomized lower bounds. As explained above, our lower bound uses distributions on the input rather pervasively, but still, it only works for deterministic protocols. (Fortunately, the upper bound is also deterministic. . .)

It would be a nice technical development to also show this lower bound for a bounded-error protocol. Unfortunately, this seems beyond the scope of existing techniques. The trouble is that all analyses of asymmetric communication games have been unable to employ non-product distributions.

In Section 5, we show the following interesting factlet: it is not possible to prove asymmetric communication lower bounds over product distributions, for the NNS problem with approximation $c > 3$. Thus, a randomized lower bound would need to develop new tools in communication complexity.

2 Review of Indyk's Upper Bound

Decision trees. Due to the decomposability of ℓ_∞ as a maximum over coordinates, a natural idea is to solve NNS by a decision tree in which every node is a coordinate comparison. A node v is reached for some set $Q_v \subseteq \{-m, \dots, +m\}^d$ of queries. If the node compares coordinate $i \in [d]$ with a "separator" x, its two children will be reached for queries in $Q_\ell = Q_v \cap \{q \mid q_i < x\}$, respectively in $Q_r = Q_v \cap \{q \mid q_i > x\}$ (assume x is non-integral to avoid ties).

Define $[x, y]_i = \{p \mid p_i \in [x, y]\}$. Then, $Q_\ell = Q_v \cap [-\infty, x]_i$ and $Q_r = Q_v \cap [x, \infty]_i$.

If the query is known to lie in some Q_v, the set of database points that could still be a near neighbor is $N_v = D \cap (Q_v + [-1, 1]^d)$, i.e. the points inside the Minkowski sum of the query set with the ℓ_∞ "ball" of radius one. For our example node comparing coordinate $i \in [d]$ with x, the children nodes have $N_\ell = N_v \cap [-\infty, x + 1]_i$, respectively $N_r = N_v \cap [x - 1, +\infty]_i$.

Observe that $N_\ell \cap N_r = N_v \cap [x - 1, x + 1]_i$. In some sense, the database points in this slab are being "replicated," since both the left and right subtrees must consider them as potential near neighbors. This recursive replication of database points is the cause of superlinear space. The contribution of Indyk [Ind01b] is an intriguing scheme for choosing a separator that guarantees a good bound on this recursive growth.

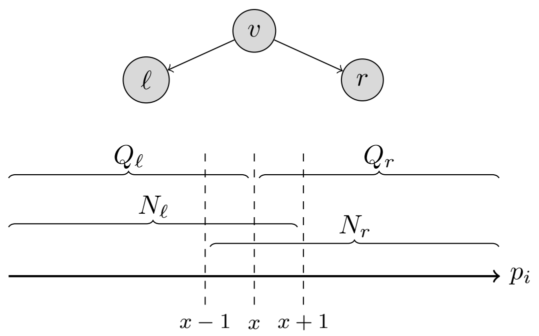

Figure 1. A separator x on coordinate i.

Information progress. Our first goal is to get a handle on the growth of the decision tree, as database points are replicated recursively. Imagine, for now, that queries come from some distribution μ. The reader who enjoys worst-case algorithms need not worry: μ is just an analysis gimmick, and the algorithm will be deterministic.

We can easily bound the tree size in terms of the measure of the smallest Q_v ever reached: there can be at most $1/\min_v \Pr_\mu[Q_v]$ distinct leaves in the decision tree, since different leaves are reached for disjoint Q_v's. Let $I_Q(v) = \log_2 \frac{1}{\Pr_\mu[Q_v]}$; this can be understood as the information learned about the query, when computation reaches node v. We can now rewrite the space bound as $O\big(2^{\max_v I_Q(v)}\big)$.

Another quantity that can track the behavior of the decision tree is $H_N(v) = \log_2 |N_v|$. Essentially, this is the "entropy" of the identity of the near neighbor, assuming that one exists.

At the root λ, we have $I_Q(\lambda) = 0$ and $H_N(\lambda) = \lg n$. Decision nodes must reduce the entropy of the near neighbor until H_N reaches zero ($|N_v| = 1$). Then, the algorithm can simply read the single remaining candidate, and test whether it is a near neighbor of the query. Unfortunately, decision nodes also increase I_Q along the way, increasing the space bound. The key to the algorithm is to balance this tension between reducing the entropy of the answer, H_D, and not increasing the information about the query, I_Q, too much.

In this information-theoretic view, Indyk's algorithm shows that we can (essentially) always find a separator that decreases H_N by some δ but does not increase I_Q by more than $\rho \cdot \delta$. Thus, H_D can be pushed from $\lg n$ down to 0, without ever increasing I_Q by more than $\rho \lg n$. That is, space $O(n^\rho)$ is achieved.

Searching for separators. At the root λ, we let $i \in [d]$ be an arbitrary coordinate, and search for a good separator x on that coordinate. Let π be the frequency distribution (the empirical probability distribution) of the projection on coordinate i of all points in the database. To simplify expressions, let $\pi(x : y) = \sum_{j=x}^{y} \pi(j)$.

If x is chosen as a separator at the root, the entropy of the near neighbor in the two child nodes is reduced by:

$$\begin{aligned} H_N(\lambda) - H_N(\ell) &= \log_2 \tfrac{|N_\lambda|}{|N_\ell|} \\ &= \log_2 \tfrac{|D|}{|D \cap [-\infty, x+1]_i|} = \log_2 \tfrac{1}{\pi(-\infty : x+1)} \\ H_N(\lambda) - H_N(r) &= \log_2 \tfrac{1}{\pi(x-1 : \infty)} \end{aligned}$$

Remember that we have not yet defined μ, the assumed probability distribution on the query. From the point of view of the root, it only matters what probability μ assigns to Q_ℓ and Q_r. Let us reason, heuristically, about what assignments are needed for these probabilities in order to generate difficult problem instances. If we understand the most difficult instance, we can use that setting of probabilities to obtain an upper bound for all instances.

First, it seems that in a hard instance, the query needs to be close to some database point (at least with decent probability). In our search for a worst case, let's just assume that the query is always planted in the neighborhood of a database point; the problem remains to find this near neighbor.

Assume by symmetry that $H_N(\ell) \geq H_N(r)$, i.e. the right side is smaller. Under our heuristic assumption that the query is planted next to a random database point, we can lower bound $\Pr_\mu[Q_r] \geq \pi(x+1, \infty)$. Indeed, whenever the query is planted next to a point in $[x+1, \infty]_i$, it cannot escape from $Q_r = [x, \infty]_i$. Remember that our space guarantee blows up when the information about Q_v increases quickly (i.e. the probability of Q_v decreases). Thus, the worst case seems to be when $\Pr_\mu[Q_r]$ is as low as possible, namely equal to the lower bound.

Thus, we have convinced ourselves that it's reasonable to define μ such that:

$$\Pr_\mu[Q_\ell] = \pi(-\infty : x+1); \qquad \Pr_\mu[Q_r] = \pi(x+1, \infty) \quad (1)$$

We apply the similar condition at all nodes of the decision tree. Note that there exists a μ safisfying all these conditions: the space of queries is partitioned recursively between the left and right subtrees, so defining the probability of the left and right subspace at all nodes is a definition of μ (but note that μ with these properties need not be unique).

From (1), we can compute the information revealed about the query:

$$\begin{aligned} I_Q(\ell) - I_Q(\lambda) &= \log_2 \tfrac{\Pr[Q_\lambda]}{\Pr[Q_\ell]} = \log_2 \tfrac{1}{\pi(-\infty : x+1)} \\ I_Q(r) - I_Q(\lambda) &= \log_2 \tfrac{1}{\pi(x+1 : \infty)} \end{aligned}$$

Remember that our rule for a good separator was $\Delta I_Q \leq \rho \cdot \Delta H_N$. On the left side, $I_Q(\ell) - I_Q(\lambda) = H_N(\lambda) - H_N(\ell)$, so the rule is trivially satisfied. On the right, the rule asks that: $\log_2 \frac{1}{\pi(x+1:\infty)} \leq \rho \cdot \log_2 \frac{1}{\pi(x-1:\infty)}$. Thus, x is a good separator iff $\pi(x+1 : \infty) \geq \big[\pi(x-1 : \infty)\big]^\rho$.

Finale. As defined above, a good separator satisfies the bound on the information progress, and guarantees the desired space bound of $O(n^\rho)$. We now ask what happens when no good separator exists.

We may assume by translation that the median of π is 0, so $\pi([1 : \infty]) \leq \frac{1}{2}$. If $x = 1\frac{1}{2}$ is not a good separator, it means that $\pi(3 : \infty) < \big[\pi(1 : \infty)\big]^\rho \leq 2^{-\rho}$. If $x = 3\frac{1}{2}$ is not a good separator, then $\pi(5 : \infty) < \big[\pi(3 : \infty)\big]^\rho \leq 2^{-\rho^2}$. By induction, the lack of a good separator implies that $\pi(2j+1 : \infty) < 2^{-\rho^j}$. The reasoning works simmetrically to negative values, so $\pi(-\infty : -2j-1) < 2^{-\rho^j}$.

Thus, if no good separator exists on coordinate i, the distribution of the values on that coordinate is very concentrated around the median. In particular, only a fraction of $\frac{1}{2d}$ of the database points can have $|x_i| \geq R = 1 + 2\log_\rho \log_2 \frac{n}{4d}$. Since there is no good separator on any coordinate, it follows that less than $d \cdot \frac{n}{2d} = \frac{n}{2}$ points have *some* coordinate exceeding R. Let $D^\star$ be the set of such database points.

To handle the case when no good separator exists, we can introduce a different type of node in the decision tree. This node tests whether the query lies in an ℓ_∞ ball of radius

$R+1$ (which is equivalent to d coordinate comparisons). If it does, the decision tree simply outputs any point in $D \setminus D^\star$. Such a point must be within distance $2R+1$ of the query, so it is an $O(\log_\rho \log d)$ approximation.

If the query is outside the ball of radius $R+1$, a near neighbor must be outside the ball of radius R, i.e. must be in $D^\star$. We continue with the recursive construction of a decision tree for point set $D^\star$. Since $|D^\star| \leq |D|/2$, we get a one-bit reduction in the entropy of the answer for free. (Formally, our μ just assigns probability one to the query being outside the ball of radius $R+1$, because in the "inside" case the query algorithm terminates immediately.)

Intuition for a lower bound. After obtaining this information-theoretic understanding of Indyk's algorithm, the path to a lower bound should be intuitive. We will consider a distribution on coordinates decaying like $2^{-\rho^j}$ (we are free to consider only the right half, making all coordinates positive). Database points will be generated i.i.d., with each coordinate drawn independently from this distribution.

In the communication view, Alice's message sends a certain amount of information restricting the query space to some Q. The entropy of the answer is given by the measure of $Q + [-1,1]^d$ (each of the n points lands in $Q + [-1,1]^d$ independently with the same probability). The question that must be answered is how much bigger $Q + [-1,1]^d$ is, compared to Q. We show an isoperimetric inequality proving that the least expanding sets are exactly the ones generated by Indyk's algorithm: intersections of coordinate cuts $\{p_i \geq x\}$.

Then, if Alice's message has $o(\rho \lg n)$ bits of information, the entropy of the near neighbor decreases by $o(\lg n)$ bits. In other words, $n^{1-o(1)}$ of the points are still candidate near neighbors, and we can use this to lower bound the message that Bob must send.

3 The Communication Lower Bound

We denote the communication problem c-NNS by the partial function F. We complete the function F by setting $\bar{F}(q,D) = F(q,D)$ whenever $F(q,D)$ is defined (i.e., when we are either in a yes or no instance), and $\bar{F}(q,D) = \star$ otherwise. Note that the domain of $\bar{F}$ is $X \times Y$, where $X = \{0,1,\dots m\}^d$ and $Y = \left(\{0,1,\dots m\}^d\right)^n$.

An $[a,b]$-protocol is a protocol by which Alice sends a total of a bits and Bob sends a total of b bits. To prove Theorem 2, assume that there exists some $[a,b]$-protocol Π computing the function $\bar{F} : X \times Y \to \{0,1,\star\}$.

As explained already, our lower bound only applies to deterministic (zero error) protocols. However, at many stages it requires conceptual use of distributions on the input domains X and Y. These distributions are defined below.

We start with the following variant of the richness lemma of [MNSW98, Lemma 6] for *randomized* protocols.

Lemma 5. *Consider a problem $f : X \times Y \to \{0,1\}$, and some probability distributions η_X, η_Y over sets X, Y respectively. Suppose $\Pr_{x \in X, y \in Y}[f(x,y) = 0] \geq \Omega(1)$.*

If f has a randomized two-sided error $[a,b]$-protocol, then there is a rectangle $\mathcal{X} \times \mathcal{Y}$ of f of sizes at least $\eta_X(\mathcal{X}) \geq 2^{-O(a)}$ and $\eta_Y(\mathcal{Y}) \geq 2^{-O(a+b)}$ in which the density (i.e., conditional measure) of ones is at most ϵ. Also, the protocol outputs value 0 on $\mathcal{X} \times \mathcal{Y}$.

First define the following measure (probability distribution) π over the set $\{0,1,\dots m\}$: for $i = 1,2,\dots c$, let $\pi(\{i\}) = 2^{-(2\rho)^i}$ and $\pi(\{0\}) = 1 - \sum_{i \geq 1} \pi(\{i\}) \geq 1/2$. For simplicity, we denote $\pi_i = \pi(\{i\})$. Similarly, define the measure μ_d over $\{0,1,\dots m\}^d$ as $\mu_d(\{(x_1, x_2 \dots x_d)\}) = \pi(\{x_1\}) \cdot \pi(\{x_2\}) \cdots \pi(\{x_d\})$.

In our hard distribution, we generate q at random from $\{0,1,\dots m\}^d$ according to the distribution μ_d. Also, we take the set D by choosing n points i.i.d. from μ_d.

Claim 6. *If we choose q and D as above, then $\Pr[\bar{F}(q,D) \neq 0] \leq e^{-\log^{1+\epsilon/3} n}$.*

Proof. Consider q and some $p \in D$: they differ in the j^{th} coordinate by at least c with probability at least $2\pi_0 \pi_c \geq \pi_c$ (when one is 0 and the other is c). Thus, $\Pr[\|q-p\|_\infty < c] \leq (1-\pi_c)^d \leq e^{-\pi_c d} \leq e^{-\log^{1+\epsilon/2} n}$. By a union bound over all $p \in D$, we get that $\|q-p\|_\infty \geq c$ for all $p \in D$ with probability at least $1 - e^{-\log^{1+\epsilon/3} n}$. $\square$

Claim 7. *There exists a combinatorial rectangle $\mathcal{Q} \times \mathcal{D} \subset \{0,1,\dots m\}^d \times (\{0,1,\dots m\}^d)^n$ on which the presumed protocol outputs 0, and such that $\mu_d(\mathcal{Q}) \geq 2^{-O(a)}$ and $\mu_{d \cdot n}(\mathcal{D}) \geq 2^{-O(a+b)}$.*

The claim follows immediately from the richness lemma 5, applied to the function F' that is the function the presumed protocol Π actually computes. In particular, note that since the protocol is deterministic, $F'(q,D) = \bar{F}(q,D)$ whenever $\bar{F}(q,D) \in \{0,1\}$, and $F'(q,D)$ is either 0 or 1 when $\bar{F}(q,D) = \star$.

Since the protocol computes all of $\mathcal{Q} \times \mathcal{D}$ correctly, it must be that $\bar{F}(q,D) \in \{0,\star\}$ for all $q \in \mathcal{Q}$ and $D \in \mathcal{D}$. It remains to prove the following claim.

Claim 8. *Consider any set $\mathcal{Q} \subseteq \{0,1,\dots m\}^d$ and $\mathcal{D} \subseteq (\{0,1,\dots m\}^d)^n$ of size $\mu_d(\mathcal{Q}) \geq 2^{-\delta\rho \log n}$ and $\mu_{d \cdot n}(\mathcal{D}) \geq 2^{-O(n^{1-\delta})}$. Then, there exists some $q \in \mathcal{Q}$ and $D \in \mathcal{D}$ such that $\bar{F}(q,D) = 1$ (i.e., there exists a point $p \in D$ such that $\|q-p\|_\infty \leq 1$).*

The claim is based on the following lemma that we prove in Section 4. This lemma is a somewhat involved isoperimetric inequality on space with our distributions, and it is the core component of our lower bound.

Lemma 9. *Consider any set $S \subseteq \{0, 1, \ldots m\}^d$. Let $N(S)$ be the set of points at distance at most 1 from S under ℓ_∞: $N(S) = \{p \mid \exists s \in S : \|p - s\|_\infty \le 1\}$. Then $\mu_d(N(S)) \ge (\mu_d(S))^{1/\rho}$.*

Proof of Claim 8. Let $N = N(\mathcal{Q})$ be the set of points at distance at most 1 from $\mathcal{Q}$. By the above lemma, $\mu_d(N) \ge (\mu_d(\mathcal{Q}))^{1/\rho} \ge 1/n^\delta$. We need to prove that there exists a set $D \in \mathcal{D}$ that intersects with N. For $D \in (\{0, 1, \ldots m\}^d)^n$, let $\sigma(D) = |D \cap N|$.

Suppose D would be chosen at random from $(\{0, 1, \ldots m\}^d)^n$ (instead of $\mathcal{D}$). Then $\mathbb{E}_D[\sigma(D)] \ge n \cdot n^{-\delta} = n^{1-\delta}$. By Chernoff bound, $\sigma(D) < 1$ happens only with probability at most $e^{-\Omega(n^{1-\delta})}$.

Thus, if we restrict to $D \in \mathcal{D}$, we obtain $\Pr_D[\sigma(D) < 1 \mid D \in \mathcal{D}] \le \frac{\Pr_D[\sigma(D)<1]}{\Pr[D\in\mathcal{D}]} = e^{-\Omega(n^{1-\delta})} \cdot 2^{O(n^{1-\delta})} < e^{-\Omega(n^{1-\delta})}$.

Concluding, there exists some $D \in \mathcal{D}$ such that $|N(\mathcal{Q}) \cap D| \ge 1$, and thus there exists some $q \in \mathcal{Q}$ and some $p \in D$ such that $\|q - p\|_\infty \le 1$. □

Finally, Claims 7 and 8 imply that either $a = \Omega(\delta\rho \log n)$ or $b = \Omega(n^{1-\delta})$. This concludes the proof of Theorem 2.

4 An Isoperimetric Inequality (Lemma 9)

We prove the following lemma, where we use the notation from the previous section.

Lemma 9. *Consider any set $S \subseteq \{0, 1, \ldots m\}^d$. Let $N(S)$ be the set of points at distance at most 1 from S under ℓ_∞: $N(S) = \{p \mid \exists s \in S : \|p - s\|_\infty \le 1\}$. Then $\mu_d(N(S)) \ge (\mu_d(S))^{1/\rho}$.*

The core of the lemma is the following one-dimensional isoperimetic inequality. The rest of the Lemma 9 results by an induction on the dimension.

Theorem 10. *Let ρ be a large positive integer, and for $i = 1 \ldots m$, $\pi_i = 2^{-(2\rho)^i}$, $\pi_0 = 1 - (\pi_1 + \cdots + \pi_m)$. Then for any non-negative real numbers $\beta_0, \ldots, \beta_m$ satisfying*

$$\pi_0\beta_0^\rho + \pi_1\beta_1^\rho + \cdots + \pi_m\beta_m^\rho = 1$$

the following inequality holds (where we set $\beta_{-1} = \beta_{m+1} = 0$)

$$\sum_{i=0}^{m} \pi_i \max\{\beta_{i-1},\ \beta_i,\ \beta_{i+1}\} \ge \ 1 \tag{2}$$

Before proving Theorem 10, we complete the proof of Lemma 9 assuming this one-dimensional theorem. Let's prove first the case of $d = 1$. We have a set $S \subset \{0, 1, \ldots m\}$, and let $\beta_i = 1$ iff $i \in S$. Then $\mu_1(S) = \pi(S) = \sum \pi_i\beta_i = \sum \pi_i\beta_i^\rho$. The set $N(S) = \Big\{i \in \{0, 1, \ldots m\} \mid \max\{\beta_{i-1}, \beta_i, \beta_{i+1}\} = 1\Big\}$ has measure, by Theorem 10,

$$\mu_1(N(S)) = \sum_{i=0}^{m} \pi_i \max\{\beta_{i-1}, \beta_i, \beta_{i+1}\} \ge (\mu_1(S))^{1/\rho}.$$

Now let's prove the induction step. Consider $S \subset \{0, 1, \ldots m\}^d$, and, for $i \in \{0, 1, \ldots m\}$, let $S_i = \{(s_2, s_3, \ldots s_d) \mid (i, s_2, \ldots s_m) \in S\}$ be the set of points in S that have the first coordinate equal to i. Then, letting $\beta_i^\rho = \mu_{d-1}(S_i)$, we have that

$$\sum_{i=0}^{m} \pi_i\beta_i^p = \sum_i \pi_i\mu_{d-1}(S_i) = \mu_d(S).$$

We can lower bound the measure of $N(S)$ as

$$\mu_d(N(S)) \ge \sum_{i=0}^{m} \pi_i \cdot \max \begin{cases} \mu_{d-1}(N(S_{i-1})) \\ \mu_{d-1}(N(S_i)) \\ \mu_{d-1}(N(S_{i-1})) \end{cases}$$

where we assume, by convention, that $S_{-1} = S_{m+1} = 0$.

By inductive hypothesis, $\mu_{d-1}(N(S_i)) \ge (\mu_{d-1}(S_i))^{1/\rho} = \beta_i$ for all i. Thus, applying Theorem 10 once again, we conclude,

$$\mu_d(N(S)) \ge \sum_i \pi_i \max\{\beta_{i-1}, \beta_i, \beta_{i+1}\} \ge (\mu_d(S))^{1/\rho}.$$

This finishes the proof of Lemma 9.

4.1 The 1D Case (Theorem 10)

Let $\Gamma = \big\{(\beta_0, \ldots, \beta_m) \in \mathbb{R}^{m+1} \mid \pi_0\beta_0^\rho + \pi_1\beta_1^\rho + \ldots + \pi_m\beta_m^\rho = 1\big\}$, and denote by $f(\beta_0, \ldots, \beta_m)$ the left hand side of (2). Then f is a continuous function on the compact set $\Gamma \subset \mathbb{R}^{m+1}$, so it achieves its minimum. Call an $(m+1)$-tuple $(\beta_0, \ldots, \beta_m) \in \Gamma$ *optimal* if $f(\beta_0, \ldots, \beta_m)$ is minimal. Our proof strategy will be to show that if $(\beta_0, \ldots, \beta_m)$ is optimal, then $\beta_i = 1$.

We consider several possible configurations for sizes of β_i's in an optimal β in three separate lemmas, and prove they are not possible. We then conclude the theorem by showing these configurations are all the configurations that we need to consider.

Lemma 11. *If there exists an index $i \in \{1, \ldots, m-1\}$ such that $\beta_{i-1} > \beta_i < \beta_{i+1}$, then $\bar{\beta} = (\beta_0, \ldots, \beta_m)$ is not optimal.*

Proof. Define a new vector $\bar{\beta}' = (\beta_0, \ldots, \beta_{i-2}, \beta_{i-1} - \epsilon, \beta_i + \delta, \beta_{i+1} - \epsilon, \beta_{i+2}, \ldots, \beta_m)$, where $\epsilon,\ \delta > 0$ are chosen suitably so that $\bar{\beta}' \in \Gamma$, and $\beta_{i-1} - \epsilon > \beta_i + \delta < \beta_{i+1} - \epsilon$. It's easy to see that $f(\bar{\beta}) > f(\bar{\beta}')$, which contradicts the optimality of $\bar{\beta}$. □

Lemma 12. *If there exists an index $i \in \{1, \dots, m\}$ such that $\beta_{i-1} > \beta_i \geq \beta_{i+1}$, then $\bar{\beta} = (\beta_0, \dots, \beta_m)$ is not optimal.*

Proof. Let $\beta = \left(\frac{\pi_{i-1}\beta_{i-1}^\rho + \pi_i \beta_i^\rho}{\pi_{i-1}+\pi_i}\right)^{1/\rho}$ and define $\bar{\beta}' = (\beta_0, \dots, \beta_{i-2}, \beta, \beta, \beta_{i+1}, \dots \beta_m)$. Then $\bar{\beta}' \in \Gamma$, and $\beta_{i-1} > \beta > \beta_i$.

We claim that $f(\bar{\beta}) > f(\bar{\beta}')$. Comparing the expressions for $f\left(\bar{\beta}\right)$ and $f\left(\bar{\beta}'\right)$ term by term, we see that it's enough to check that

$$\begin{aligned} \pi_i \max\{\beta_{i-1}, \beta_i, \beta_{i+1}\} + \pi_{i+1} \max\{\beta_i, \beta_{i+1}, \beta_{i+2}\} \quad > \\ > \pi_i \max\{\beta, \beta_{i+1}\} + \pi_{i+1} \max\{\beta, \beta_{i+1}, \beta_{i+2}\} \end{aligned}$$

where the terms involving π_{i+1} appear unless $i = m$. For $i = m$, the inequality becomes $\beta_{i-1} > \beta$ which holds by assumption. For $i = 1, \dots, m-1$, the above inequality is equivalent to

$$\pi_i(\beta_{i-1} - \beta) > \pi_{i+1} \cdot (\max\{\beta, \beta_{i+2}\} - \max\{\beta_i, \beta_{i+2}\})$$

which, in its strongest form (when $\beta_i \geq \beta_{i+2}$), is equivalent to $\pi_i(\beta_{i-1} - \beta) > \pi_{i+1}(\beta - \beta_i)$. The last inequality is equivalent to

$$\left(\frac{\pi_i \beta_{i-1} + \pi_{i+1}\beta_i}{\pi_i + \pi_{i+1}}\right)^\rho > \frac{\pi_{i-1}\beta_{i-1}^\rho + \pi_i \beta_i^\rho}{\pi_{i-1} + \pi_i}$$

which we can rewrite as

$$\left(\frac{c_i + t}{c_i + 1}\right)^\rho - \frac{c_{i-1} + t^\rho}{c_{i-1} + 1} > 0, \tag{3}$$

where $c_i = \pi_i / \pi_{i+1} \geq 2^{(2\rho)^{i+1} - (2\rho)^i}$ (for $i > 0$ we have equality and for $i = 0$ we have inequality because p is large), and $t = \beta_i / \beta_{i-1} \in [0, 1)$. Let $F(t)$ denote the left hand side of inequality (3) (which we are left to prove). Note that $F(0) > 0$, because:

$$\begin{aligned} \left(\frac{c_i}{c_i + 1}\right)^\rho &= \left(1 - \frac{1}{c_i + 1}\right)^\rho \geq 1 - \frac{\rho}{c_i + 1} \\ &> 1 - \frac{1}{c_{i-1} + 1} = \frac{c_{i-1}}{c_{i-1} + 1} \end{aligned}$$

where we have used Bernoulli's inequality $(1-x)^n \geq 1 - nx$ for $0 < x < 1/n$ and $c_i + 1 > 2^{(2\rho)^{i+1} - (2\rho)^i} > \rho \cdot (2^{(2\rho)^i} + 1) = \rho(\frac{1}{\pi_{i-1}} c_{i-1} + 1) > \rho(c_{i-1} + 1)$. Now we let $t \in (0, 1)$ and write $F(t) = F(0) + t^\rho G(t)$, where

$$\begin{aligned} G(t) &= \frac{1}{(c_i + 1)^\rho}\left(\binom{\rho}{1} c_i^{\rho-1} \frac{1}{t} + \binom{\rho}{2} c_i^{\rho-2} \frac{1}{t^2} \right. \\ &\quad \left. + \dots + \binom{\rho}{\rho - 1} c_i \frac{1}{t^{\rho-1}}\right) + \\ &\quad + \left(\frac{1}{(c_i+1)^\rho} - \frac{1}{c_{i-1} + 1}\right). \end{aligned}$$

If $G(t) \geq 0$, then clearly $F(t) \geq F(0) > 0$, so we are done. Otherwise, $G(t) < 0$, and in this case it easily follows that $G(1) < G(t) < 0$, hence $F(t) = F(0) + t^\rho G(t) > F(0) + G(1) = F(1) = 0$, as desired. This concludes the proof of the lemma. □

Lemma 13. *If there is an index $i \in \{0, 1 \dots, m-1\}$ such that $\beta_{i-1} \leq \beta_i < \beta_{i+1}$, then $\beta = (\beta_0, \beta_1, \dots, \beta_m)$ is not optimal.*

Proof. We proceed as in the previous lemma. Let $\beta = \left(\frac{\pi_i \beta_i^\rho + \pi_{i+1}\beta_{i+1}^\rho}{\pi_i + \pi_{i+1}}\right)^{1/\rho}$, and define $\bar{\beta}' = (\beta_0, \dots, \beta_{i-1}, \beta, \beta, \beta_{i+2}, \dots, \beta_m)$. As before, $\bar{\beta}' \in \Gamma$ and $\beta_i < \beta < \beta_{i+1}$. We claim that $f(\bar{\beta}) > f(\bar{\beta}')$. Comparing the expressions for $f\left(\bar{\beta}\right)$ and $f\left(\bar{\beta}'\right)$ term by term, we see that it's enough to check that

$$\begin{aligned} \pi_{i-1} \max\{\beta_{i-2}, \beta_{i-1}, \beta_i\} + \pi_i \max\{\beta_{i-1}, \beta_i, \beta_{i+1}\} \quad > \\ > \pi_{i-1} \max\{\beta_{i-2}, \beta_{i-1}, \beta\} + \pi_i \max\{\beta_{i-1}, \beta, \beta\} \end{aligned}$$

where the terms involving π_{i-1} appear unless $i = 0$. If $i = 0$, the above inequality becomes $\beta_{i+1} > \beta$ and we are done. For $i = 1, \dots m-1$, the inequality is equivalent to

$$\pi_i(\beta_{i+1} - \beta) > \pi_{i-1} \cdot (\max\{\beta, \beta_{i-2}\} - \max\{\beta_i, \beta_{i-2}\})$$

which, in its strongest form (when $\beta_i \geq \beta_{i-2}$) is equivalent to $\pi_i(\beta_{i+1} - \beta) > \pi_{i-1}(\beta - \beta_i)$. The latter inequality is equivalent to

$$\left(\frac{\pi_i \beta_{i+1} + \pi_{i-1}\beta_i}{\pi_i + \pi_{i-1}}\right)^\rho > \frac{\pi_{i+1}\beta_{i+1}^\rho + \pi_i \beta_i^\rho}{\pi_{i+1} + \pi_i}$$

which we can rewrite as

$$\left(\frac{c_{i-1} t + 1}{c_{i-1} + 1}\right)^\rho - \frac{c_i t^\rho + 1}{c_i + 1} > 0, \tag{4}$$

where $c_i = \pi_i / \pi_{i+1}$ as before, and $t = \beta_i / \beta_{i+1} \in [0, 1)$. Let $F(t)$ denote the left hand side of (4) (which we are left to prove). Note that $F(0) > 0$, because

$$\begin{aligned} \left(\frac{1}{c_{i-1} + 1}\right)^\rho &> \frac{1}{(2c_{i-1})^\rho} = \frac{1}{\pi_{i-1}^\rho} \cdot 2^{-\rho \cdot (2\rho)^i - \rho} \\ &> 2^{-\rho \cdot (2\rho)^i - \rho} \geq 2^{(2\rho)^i - (2\rho)^{i+1}} = \frac{1}{c_i} > \frac{1}{c_i + 1} \end{aligned}$$

Now we let $t \in (0, 1)$ and write $F(t) = F(0) + t^\rho G(t)$, where

$$\begin{aligned} G(t) &= \frac{1}{(c_{i-1} + 1)^\rho}\left(\binom{\rho}{1} c_{i-1} \frac{1}{t} + \binom{\rho}{2} c_{i-1}^2 \frac{1}{t^2} \right. \\ &\quad \left. + \dots + \binom{\rho}{\rho - 1} c_{i-1}^{\rho-1} \frac{1}{t^{\rho-1}}\right) + \\ &\quad + \left(\left(\frac{c_{i-1}}{c_{i-1} + 1}\right)^\rho - \frac{c_i}{c_{i-1} + 1}\right). \end{aligned}$$

If $G(t) \geq 0$, then clearly $F(t) \geq F(0) > 0$, so we are done. Otherwise, $G(t) < 0$, in which case it easily follows that $G(1) < G(t) < 0$, hence $F(t) = F(0) + t^\rho G(t) > F(0) + G(1) = F(1) = 0$, as desired. This concludes the proof of the lemma. □

To prove Theorem 10, assume $\bar{\beta} = (\beta_0, \dots, \beta_m) \in \Gamma$ is optimal. By Lemmas 11 and 12, it follows that $\beta_0 \leq \beta_1 \leq \dots \leq \beta_m$. Now Lemma 13 implies that $\beta_0 = \beta_1 = \dots = \beta_m$, so since $\bar{\beta} \in \Gamma$, we have $\beta_i = 1$, and hence the minimal value of f over $\Gamma$is $f(1, 1, \dots, 1) = 1$.

This concludes the proof of the Theorem 10.

5 Lower Bounds for High Approximation

In this section, we present an argument why it is difficult to prove any non-trivial lower bounds for randomized NNS problems for high approximation. Namely, we show that the current techniques are not able to prove communication complexity lower bounds for randomized NNS problems for an approximation bigger than 3. The approximation factor of 3 seems to be fundamental here. For approximation less than 3, we actually know lower bounds for NNS under ℓ_∞, by reduction to the partial match problem.

Our arguments apply to NNS over any metric. Let us consider a metric $\mathcal{M}$ with distance function $d_\mathcal{M}$ and the following problem.

Definition 14 (NNS under $\mathcal{M}$)**.** *Fix $R > 0, \alpha > 0$. Suppose Alice is given a point $q \in \mathcal{M}$, and Bob is given the dataset $D \subset \mathcal{M}$ of size n. Then, in the R-*Near Neighbor Search *problem, Alice and Bob compute the following function $\mathfrak{N}(q, D)$:*

- *$\mathfrak{N}(q, D) = 1$ if there exists some $p \in D$ such that $d_\mathcal{M}(x, y) \leq R$;*
- *$\mathfrak{N}(q, D) = 0$ if for all $p \in D$, we have that $d_\mathcal{M}(x, t) \geq \alpha R$.*

As before, when neither is the case, we set $\mathfrak{N}(x, y) = \star$.

In a randomized $[a, b]$-protocol Π, Alice sends at most a bits, Bob sends at most b bits, and they produce the correct answer with probability at least 0.9.

An almost ubiquitous technique to prove a lower bound for the communication complexity is by applying Yao's minimax principle. The principle says that if there exists a randomized $[a, b]$-protocol, then for any distribution μ on $\mathcal{M} \times \mathcal{M}^n$, there exists some deterministic protocol Π_μ succeeding on 0.9 mass of the distribution μ: $\mathbb{E}_{(q,D)\in\mu} [\Pi_\mu(x, y) = \mathfrak{N}(x, y)] \geq 0.9$ Thus one just need to exhibit a "hard" distribution where no deterministic protocol succeeds. Most candidates for the "hard" distribution μ are *product distributions*, namely $\mu = \mu_q \times \mu_D$, where μ_q and μ_D are independent distributions on $q \in \mathcal{M}$ and $D \in \mathcal{M}^n$ respectively.

Indeed, to the best of our knowledge, all known asymmetric communication complexity lower bounds are proven via this approach with product distributions. It seems quite challenging to prove asymmetric communication complexity lower bounds for distributions that are non-product.

We prove that it is not possible to prove lower bound for the NNS problem with product distributions when the approximation is bigger than 3. In fact, the argument applies even to one-way protocol lower bounds, where one-way protocols are $[a, 0]$-protocols in which just Alice sends a message of length a.

Lemma 15. *Consider the problem $\mathfrak{N}$ for approximation α. Consider any product distribution $\mu = \mu_q \times \mu_D$ on $\mathcal{M} \times \mathcal{M}^n$, and suppose for any $[a, 0]$-protocol Π, we have $\mathbb{E}_\mu [\Pi(x, y) \neq \mathfrak{N}(x, y)] < 0.9$. Then either $\alpha \leq 3$ or $a = O(\log n)$ or there exists (q, D) in the support of μ such that $\mathfrak{N}(q, D) = \star$.*

Proof. Assume that $\alpha > 3$ and that $a \geq C \log n$ for some big constant C. Let $\mathcal{Q}$ be the support of μ_q and $\mathcal{D}$ be the support of μ_D. We will prove that there exists some $(\tilde{q}, \tilde{D}) \in \mathcal{Q} \times \mathcal{D}$ such that $\mathfrak{N}(\tilde{q}, \tilde{D}) = \star$.

We will use a characterization of [KNR99] for one-way protocols for product distributions to construct $\tilde{q}, \tilde{D}$.

First we need to give a definition. Consider the matrix M of size $|\mathcal{Q}| \times |\mathcal{D}|$ where $M_{ij} = \mathfrak{N}(q_i, D_j)$, where q_i is the i^{th} element of $\mathcal{Q}$ in, say, lexicographic order, and same with D_j. The *VC-dimension* of M is the maximum $v \in \mathbb{N}$ such that there exists $D_{j_1}, \dots D_{j_v} \in \mathcal{D}$ such that for any boolean vector $z \in \{0, 1\}^v$, there exist $q_z \in \mathcal{Q}$ with $\mathfrak{N}(q_z, D_{j_k}) = z_k$ for all $k \in [v]$.

Since $a \geq C \log n$, the result of [KNR99] implies that the VC-dimension of the matrix M is at least $v \geq \log_2 n + 2$ (choosing C accordingly). Then, take a set of z's that is $Z \subset \{1\} \times \{0, 1\}^{v-1}$ and has size $|Z| \geq n + 1$. Suppose $Z = \{z^{(1)}, z^{(2)}, \dots, z^{(n+1)}\}$ and let $q_{z_1} \dots q_{z_{n+1}}$ be the queries such that, for all $i = 1 \dots n + 1$, we have that $\mathfrak{N}(q_{z^{(i)}}, D_{j_k}) = z_k^{(i)}$ for all $k \in [v]$. In particular, for $D = D_{j_1}$, we have that $\mathfrak{N}(q_z, D) = 1$, i.e., there exists $p_z \in D$, for each $z \in Z$, such that $d_\mathcal{M}(q_z, p_z) \leq R$. By pigeonhole principle, there exists some $p \in D$ and distinct $z', z'' \in Z$ such that $d_\mathcal{M}(q_{z'}, p) \leq R$ and $d_\mathcal{M}(q_{z''}, p) \leq R$. Thus, by triangle inequality, $d_\mathcal{M}(q_{z'}, q_{z''}) \leq 2R$. However, since $z' \neq z''$, there is some $j \in \{2, \dots v\}$ such that $z'_j \neq z''_j$. In other words, wlog, $d_\mathcal{M}(q_{z'}, D_j) \leq R$ and $d_\mathcal{M}(q_{z''}, D_j) \geq \alpha R > 3R$. But this is not possible since, by triangle inequality, $d_\mathcal{M}(q_{z''}, D_j) \leq d_\mathcal{M}(q_{z''}, q_{z'}) + d_\mathcal{M}(q_{z'}, D_j) \leq 2R + R = 3R$ — a contradiction. □

Aknowledgements

We thank T.S. Jayram, Piotr Indyk, Robi Krauthgamer, and James Lee for useful comments and discussions.

References

[AHL01] Helmut Alt and Laura Heinrich-Litan. Exact L_∞ nearest neighbor search in high dimensions. In *Proc. ACM Symposium on Computational Geometry*, pages 157–163, 2001.

[AIK08] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. Overcoming the ℓ_1 non-embeddability barrier: Algorithms for product metrics. *Manuscript*, 2008.

[Fag96] Ronald Fagin. Combining fuzzy information from multiple systems. In *Proc. ACM Symposium on Principles of Database Systems*, pages 216–227, 1996.

[Fag98] Ronald Fagin. Fuzzy queries in multimedia database systems (invited paper). In *Proc. ACM Symposium on Principles of Database Systems*, 1998.

[FCI99] Martin Farach-Colton and Piotr Indyk. Approximate nearest neighbor algorithms for hausdorff metrics via embeddings. In *Proc. IEEE Symposium on Foundations of Computer Science*, 1999.

[Ind01a] Piotr Indyk. Approximate algorithms for high-dimensional geometric problems. Invited talk at DIMACS Workshop on Computational Geometry'02. `http://people.csail.mit.edu/indyk/high.ps`, 2001.

[Ind01b] Piotr Indyk. On approximate nearest neighbors in l_∞ norm. *Journal of Computer and System Sciences*, 63(4), 2001. See also FOCS'98.

[Ind02] Piotr Indyk. Approximate nearest neighbor algorithms for Frechet metric via product metrics. In *Proc. ACM Symposium on Computational Geometry*, 2002.

[JKKR04] T. S. Jayram, Subhash Khot, Ravi Kumar, and Yuval Rabani. Cell-probe lower bounds for the partial match problem. *Journal of Computer and Systems Sciences*, 69(3):435–447, 2004. See also STOC'03.

[KLMN05] R. Krauthgamer, J. R. Lee, M. Mendel, and A. Naor. Measured descent: A new embedding method for finite metrics. *Geom. Funct. Anal.*, 15(4):839–858, 2005.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[KNR99] I. Kremer, N. Nisan, and D. Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.

[LLR94] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 577–591, 1994.

[Mat96] J. Matoušek. On the distortion required for embedding finite metric spaces into normed spaces. *Israel Journal of Mathematics*, 93:333–344, 1996.

[MNSW98] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. Data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 1998.

[Păt08] Mihai Pătraşcu. (Data) STRUCTURES. In *Proc. IEEE Symposium on Foundations of Computer Science*, 2008.

[Rao99] Satish Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In *Proc. ACM Symposium on Computational Geometry*, pages 300–306, New York, 1999. ACM.

[Sam06] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Elsevier, 2006.

[SDI06] G. Shakhnarovich, T. Darrell, and P. Indyk, editors. *Nearest Neighbor Methods in Learning and Vision*. Neural Processing Information Series, MIT Press, 2006.

A Decision Trees Lower Bound

We formally define what we mean by a decision tree for a data structure problem (see also [KN97]). Consider a partial problem $F : \mathcal{I} \to \{0,1\}$ with $\mathcal{I} \subset X \times Y$, where X is the set of "queries" and Y is the set of "datasets".

For $y \in Y$, a *decision tree* T_y is a complete binary tree in which:

- each internal node v is labeled with a predicate function $f_v : X \to \{0,1\}$. We assume f_v comes from some set $\mathcal{F}$ of *allowed predicates*.
- each edge is labeled with 0 or 1, indicating the answer to the parent's predicate.
- each leaf is labeled with 0 or 1, indicating the outcome of the computation.

Evaluating T_y on x is done by computing the root's predicate on x, following the corresponding edge, computing the next node's predicate, and so on until a leaf is reached. The label of the leaf is the output, denoted $T_y(x)$.

We let the *size* s of the tree to be the total number of the nodes. The *depth* d of the tree is the longest path from the root to a leaf. The *predicate size* is $w = \lceil \log_2 \mathcal{F} \rceil$.

We say that problem F can be solved by a decision tree of size s, depth d, and predicate size w iff, for any y, there exists some tree T_y of size at most s, depth at most d, and node size at most w, such that $T_y(x) = F(x,y)$ whenever $(x,y) \in \mathcal{I}$.

Our result on the decision tree lower bound follows from the following folklore lemma, which converts an efficient decision tree solving a problem F into an efficient communication protocol.

Lemma 16. *Consider any (promise) problem $F : \mathcal{I} \to \{0,1\}$, where $\mathcal{I} \subset X \times Y$. Suppose there exists a decision tree of size s, depth d, and node size w.*

If Alice receives $x \in X$ and Bob receives $y \in Y$, there exists a communication protocol solving the problem F, in which Alice sends a total of $a = O(\log s)$ bits and Bob sends $b = O(dw \log s)$ bits.

Proof. Before the protocol, Bob constructs his decision tree T_y. Suppose, for a moment, that the decision tree is balanced, that is $d = O(\log s)$. Then, Alice and Bob can run the following "ideal" protocol. In round one, Bob sends the predicate f_r of the root r of the decision tree. Alice computes $f_r(x)$ (a bit) and sends it back. Then Bob follows the corresponding edge in the tree, and sends the predicate of the corresponding child, etc. We obtain communication $a \leq d$ and $b \leq w \cdot d$.

In general, however, the decision tree T_D is not balanced. In this case, Alice and Bob can simulate a standard binary search on a tree. Specifically, Bob finds a separator edge that splits the tree in two components, each of size at least $s/3$. Let this separating edge be (u, v). In round one, Alice and Bob want to detect whether, in the ideal protocol, Alice would eventually follow the edge (u, v). To determine this, Bob sends the predicates for all nodes on the path from the root r to u. Alice evaluates these predicates on x and sends back a 1 if she would follow the edge (u, v), and 0 otherwise. Then, the players recurse on the remaining part of the tree; they are done after $O(\log s)$ such rounds.

In the end, Alice sends only $a = O(\log s)$ bits, i.e. one bit per round. Bob sends $O(d \cdot w)$ bits per round, and thus $b = O(dw \log s)$. $\square$