An Optimal Randomised Cell Probe Lower Bound for Approximate Nearest Neighbour Searching

Amit Chakrabarti
Dartmouth College*

Oded Regev
Tel Aviv University†

October 4, 2009

Abstract

We consider the approximate nearest neighbour search problem on the Hamming cube $\{0,1\}^d$. We show that a randomised cell probe algorithm that uses polynomial storage and word size $d^{O(1)}$ requires a worst case query time of $\Omega(\log\log d/\log\log\log d)$. The approximation factor may be as loose as $2^{\log^{1-\eta} d}$ for any fixed $\eta > 0$. Our result fills a major gap in the study of this problem since all earlier lower bounds either did not allow randomisation [7, 21] or did not allow approximation [6, 3, 19]. We also give a cell probe algorithm that proves that our lower bound is optimal.

Our proof uses a lower bound on the round complexity of the related communication problem. We show, additionally, that considerations of bit complexity alone cannot prove any nontrivial cell probe lower bound for the problem. This shows that the "richness technique" [23] used in a lot of recent research around this problem would not have helped here.

Our proof is based on information theoretic techniques for communication complexity, a theme that has been prominent in recent research [9, 2, 28, 18].

1 Introduction

Nearest neighbour searching is one of those basic and fascinating theoretical problems in computer science that has a host of applications in problems from very diverse fields. To give a sense of this diversity we note that the literature includes applications in molecular biology [31, 26], information retrieval [12, 27], and pattern recognition [10, 13], and that this is far from an exhaustive list of fields. Typically, in these applications, the objects of interest are represented as points in Euclidean space by abstracting their features. The problem of nearest neighbour searching is that of finding, in a *database* of points, the closest one to a given *query* point.

When the ambient space that the database and query points come from is the Euclidean plane, the nearest neighbour search problem has well known efficient solutions using classical computational geometry techniques of space decomposition such as Voronoi diagrams (see, e.g., [11]). However, in most applications

*Computer Science Department, Dartmouth College, Hanover, NH 03755, USA. Work partly done while the author was at the Institute for Advanced Study, Princeton, NJ 08540. Work supported in part by NSF grant CCR-9987845 and NSF grant CCF-0448277.

†Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. Work done partly while the author was at the University of California, Berkeley. Supported by the Binational Science Foundation, by the Israel Science Foundation, by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848, by the Army Research Office grant DAAD19-03-1-0082, and by a European Research Council (ERC) Starting Grant.

the dimension of the ambient space is high: anywhere from tens to thousands. The classical techniques still work in such spaces, but the resulting algorithms require storage and/or running time *exponential* or worse in the dimension: a phenomenon that is sometimes called the "curse of dimensionality."

It is by now well established that the way to avoid this curse is to not insist on the absolute nearest neighbour but to allow some *approximation*; this is certainly acceptable in the aforementioned applications, since the abstraction of objects into points in Euclidean space already involves heuristics and approximations. Efficient algorithms for *approximate nearest neighbour searching* (henceforth, ANN) that scale well with dimension were obtained independently by Indyk and Motwani [17] and Kushilevitz, Ostrovsky, and Rabani [20]; some improvements were then made by Har-Peled [15].

Soon after the discovery of these algorithms, lower bounds on nearest neighbour searching were obtained by Chakrabarti, Chazelle, Gum, and Lvov [7] and simultaneously by Borodin, Ostrovsky, and Rabani [6]. These results had a serious shortcoming: the former applied only to deterministic algorithms and the latter applied only to the exact nearest neighbour (henceforth, ENN) problem. Since all the aforementioned algorithms were randomised and approximate, there was no direct comparison possible between the upper and lower bounds obtained in recent research. Subsequent work by Liu [21] and Barkol and Rabani [3] improved the respective lower bounds quantitatively but did not address this shortcoming.

In this work, for the first time, we obtain a randomised lower bound for ANN, thus addressing this shortcoming. Moreover, we design a nontrivial algorithm showing that our lower bound is optimal.

1.1 Our Results

Data Structure Query Problems: The approximate nearest neighbour search problem is an instance of what may be called *data structure query problems*, i.e., problems in which we are required to build a data structure out of some given data and then efficiently query this data structure. Formally, a data structure query problem involves three spaces: a space of *queries* $\mathscr{A}$, a space of *databases* $\mathscr{B}$, and a space of *answers* $\mathscr{C}$. The problem itself is a relation $\rho \subseteq \mathscr{A} \times \mathscr{B} \times \mathscr{C}$ to be interpreted as follows. We will be a given a $y \in \mathscr{B}$ to preprocess and will then be given a query $x \in \mathscr{A}$ and must produce any $z \in \mathscr{C}$ such that $(x, y, z) \in \rho$. We shall assume, without loss of generality, that at least one such z exists.

The Cell Probe Model: The standard framework for analysing the complexity of such problems is the *cell probe model* first defined by Yao [30]. This is a rather strong model designed to capture all conceivable algorithms for data structure problems, which makes lower bounds proven in this model quite powerful. The model assumes that the preprocessing phase deterministically constructs from y a data structure represented as a table consisting of s cells each of which holds w bits. The query phase gets x as input and then accesses t cells of the table; the choice of cells may, in general, be *randomised* as well as *adaptive*. Based on the information gathered from these cells, the algorithm must then compute an answer z which is required to be correct — i.e., to satisfy $(x, y, z) \in \rho$ — with probability at least $1 - \varepsilon$, for some small non-negative ε. Such an algorithm is called an ε-error t*-probe* algorithm with *table size* s and *word size* w. When ε is not specified we assume that it is $\frac{1}{4}$. Note that the model is unconcerned with computation time.

Like all earlier lower bounds for ANN, our bound is shown with the ambient space being the d-dimensional Hamming cube equipped with the Hamming (i.e., ℓ_1) metric. Note that this immediately implies the same lower bound for $\mathbb{R}^d$ equipped with the ℓ_1 metric and a similar lower bound for Euclidean space (i.e., $\mathbb{R}^d$ with the ℓ_2 metric) up to a square in the approximation ratio. We now precisely define our problem and state our main result.

Definition 1.1 (Approximate Nearest Neighbour). For integers $d, n \geq 1$ and a real number $\beta \geq 1$, we

define the Approximate Nearest Neighbour search problem $\mathsf{ANN}^{\beta}_{d,n}$ as the data structure query problem given by

$$\mathscr{A} = \{0,1\}^d, \quad \mathscr{B} = \binom{\{0,1\}^d}{n}, \quad \mathscr{C} = \{0,1\}^d$$
$$\rho = \left\{(x,y,z) \in \mathscr{A} \times \mathscr{B} \times \mathscr{C} : z \in y \ \wedge \ (\forall z' \in y \ (\mathrm{dist}(x,z) \leq \beta \cdot \mathrm{dist}(x,z')))\right\},$$

where "dist" denotes Hamming distance in $\{0,1\}^d$.

We shall allow a rather loose approximation ratio β. We set

$$\beta := 2^{\log^{1-\eta} d}, \tag{1}$$

where η is an arbitrarily small positive constant which we fix for the remainder of the paper.[1] Notice that obtaining approximation factor d is a trivial problem, since we could return *any* database point as an answer to any query.[2] Our main result is the following lower bound on the randomised cell probe complexity of the problem.

Theorem 1.2 (Main Lower Bound). *For all $c_1, c_2 > 0$ there exists a $c_3 > 0$ such that the following holds. Let $n, d \geq 1$ be large enough integers, and suppose that $2^{\log^{1.01} d} \leq n \leq 2^{d^{0.99}}$ (alternatively, suppose that $\log^{1.01} n \leq d \leq 2^{\log^{0.99} n}$). If $\mathsf{ANN}^{\beta}_{d,n}$ has a randomised t-probe algorithm with table size $s \leq n^{c_1}$ and word size $w \leq d^{c_2}$, then $t \geq c_3 \log\log d / \log\log\log d$.*

Chakrabarti, Chazelle, Gum, and Lvov [7] had obtained the same bound on t as above but only for deterministic algorithms. Liu [21] greatly improved their bound to $d^{1-o(1)}$, still for deterministic algorithms. On the flip side, Borodin, Ostrovsky, and Rabani [6] gave a lower bound of $\Omega(\log d)$ for randomised algorithms that did not allow approximation. This was subsequently strengthened to $\Omega(d/\log n)$ by Barkol and Rabani [3]. Thus, our result is the first lower bound that allows both randomisation and approximation.

The range of values for n in our lower bound cannot be extended significantly, because of a trivial query time upper bound of $t = 1$ at either extreme. At $n = 2^{O(\log d)} = d^{O(1)}$, we can pack the entire database into one cell of the table. At $n = 2^{\Omega(d)}$, a table of size $n^{O(1)}$ is large enough to store a correct answer to each of the 2^d possible queries.

Since the conference presentation of our results [8], Pătraşcu and Thorup [25] have considered cell probe lower bounds for the case of data structures that use near-linear space. Because of the near-linear space assumption, the lower bounds they obtain are rather strong. However, they do not allow both randomisation and approximation. For instance, they prove bounds of the form $t = \Omega(d/\lg d)$ for table size $s = n \log^{O(1)} n$, for deterministic algorithms for ANN as well as randomised algorithms for ENN.

We also prove two upper bounds on ANN: the first is a cell probe upper bound that matches our lower bound, showing that it is tight.[3]

Theorem 1.3 (Main Upper Bound). *Let $\alpha > 1$ be any constant. Then, for $n \geq d$, $\mathsf{ANN}^{\alpha}_{d,n}$ has a cell probe algorithm with $O(\log\log d / \log\log\log d)$ probes, table size $n^{O(1)}$ and word size $O(d)$.*

[1] Throughout the paper, the default base of logarithms is 2.

[2] There is a small catch. If the database contains the query point, the only valid answer is the query itself. But this degenerate case can be handled by, say, perfect hashing [14] which has $O(1)$ cell probe complexity.

[3] A similar upper bound was independently discovered and communicated to us by Beame and Guruswami [5].

The best previous cell probe upper bound on ANN under the same conditions was $O(\log \log d)$. This bound is implicit in the work of Kushilevitz, Ostrovsky, and Rabani [20], Indyk and Motwani [17], and Har-Peled [15]. Our algorithm beats these earlier ones in the cell probe model but, unlike in earlier work, we do not dwell on issues of real running time.

It is worth mentioning that the key technical innovations in the previous upper bounds were focussed on getting an $O(1)$-probe algorithm for a *decision version* of ANN: we are given a threshold λ and must distinguish between the case when the distance from the query to a nearest neighbour is at most λ versus the case when it is above $\alpha\lambda$, for some approximation factor $\alpha > 1$. This is sometimes called the approximate "near neighbour" problem. Of the ANN lower bounds mentioned above, all except for [7] in fact apply to this decision version. The $O(1)$ upper bound for the decision version forces us to work with the search version in order to prove a nontrivial lower bound. While on this issue of decision-vs.-search, we note that the recent work of Andoni, Indyk and Pătraşcu [1] (postdating our conference paper [8]) provides a tight space lower bound of $n^{O(1/\varepsilon^2)}$ for $(1+\varepsilon)$-approximate near neighbour with $O(1)$ queries, even under randomisation. The tightness follows from the upper bounds [20, 17, 15] above.

Our second upper bound is more technical but it proves that a certain simple technique — the so-called "richness technique" — that yields a number of interesting cell probe lower bounds must fail in the case of ANN. We state this result in Section 1.4.

1.2 Our Techniques

The Main Lower Bound: The proof of Theorem 1.2 distills ideas from a host of recent papers on a variety of research problems. We now give an intuitive overview of our techniques, which should make the technical exposition easier to follow and should also highlight the new ideas that we developed in the course of this work. We begin with a pair of reductions. The first is a reduction to ANN from an auxiliary problem that we define here and that we call *longest prefix match* (henceforth, LPM). In LPM we must preprocess a database of m-letter words over a large alphabet so as to quickly find, given a query m-letter word, a word in the database which has the longest prefix that matches a prefix of the query. This reduction is inspired by the work of Chakrabarti et al. [7] who used similar ideas but did not explicitly define the LPM problem. The second is a (natural) reduction to cell probe algorithms from communication protocols and was first made explicit by Miltersen [22]. As a result of these two reductions it suffices to consider LPM as a communication problem — with Alice holding the query and Bob the database — and prove a lower bound on the number of rounds required to solve the problem.

To this end, we use the *round elimination technique*, pioneered by Miltersen, Nisan, Safra, and Wigderson [23], and refined by Sen [28]. In fact, we shall need a further strengthening of Sen's round elimination lemma in order to handle LPM. Loosely speaking, a round elimination lemma lets us "remove" the first round of communication in a protocol for a certain problem, leaving us with a protocol one round shorter that solves a somewhat smaller instance of the same problem. To perform such a round elimination, recall that Alice's input is a length-m string and Bob's input is a set of length-m strings. Imagine chopping up all of these strings into k equal-length substrings, called "pieces." Suppose that Alice's first message to Bob is a bits long. Then it conveys at most a bits of information about Alice's input, and so, there must be an i such that it conveys at most a/k bits of information about the ith piece. Suppose we choose $k \gg a$. If we could somehow "embed" a smaller instance of our problem in the ith pieces of Alice's and Bob's strings, ensuring that the remaining pieces become irrelevant, we could then say that Alice's first message to Bob conveys almost no information about her input. This would let us eliminate the message with only a small change in the protocol's behaviour; the technical tool for doing so is what we call the Uninformative Message Lemma, due to Sen [28].

The above outline of round elimination was recently used by Sen [28] in his work on another data structure query problem called the predecessor problem. For our problem, ANN, the "embedding" step above does not work directly. Fortunately, it does work for LPM and this is why we focus on LPM instead. However, LPM is problematic in another way: to be of use in our study of ANN, the LPM problem we must consider has to use very short strings over a very large alphabet. Our round elimination argument above required chopping up the strings into k pieces for a large k, but now our strings end up so short that we cannot do this. New ideas are needed.

Our first new idea is what we call the Message Switching Lemma. It says that the order of the first two messages in a protocol can be *switched* at the cost of blowing up the sizes of the messages. In the communication problems arising from the cell probe model, Bob's message bits are considerably "cheaper" than Alice's, so it is not too bad if Bob's message size blows up. Our exploitation of this asymmetry is crucial and was missing in previous work.

The Message Switching Lemma does serve to eliminate one round in a communication protocol, but the blowup in the message sizes prevents us from using it repeatedly. This is where our second new idea, called the Message Compression Lemma, comes into play. It says that if the first message in a protocol is long in terms of number of bits but conveys a much smaller amount of information about the input, then we may *compress* this message so that its length is linear in the amount of information conveyed. We note that although this lemma does not follow from earlier work, it is strongly inspired by Jain, Radhakrishnan, and Sen [18] who also compressed messages, but in a different way, in their work on direct sum theorems. Subsequent to the conference presentation of our results [8], Harsha, Jain, McAllester and Radhakrishnan [16] have proven an elegant information theoretic lemma (given as Lemma 3.6 in this work) that can be used in plug-in form to prove our message compression lemma.

Finally, the communication lower bound for LPM follows thus: we first embed a smaller instance of LPM in a larger one, as indicated above. We then apply our Message Compression Lemma to a given protocol, and then the Message Switching Lemma, eliminating Alice's first message. Because we took care to apply compression first, the blowup in message sizes is under control. We can then apply Sen's Uninformative Message Lemma to eliminate Bob's first message. We have now arrived at a protocol that is two rounds shorter and solves somewhat smaller instances of LPM. If we start out with too short a protocol, repeated application of this Compression-Switching-Uninformative loop would eliminate all the rounds while still leaving us with a nontrivial communication problem. This contradiction would prove that such a short protocol cannot exist.

Our exposition uses a version of the *information cost* paradigm introduced by Chakrabarti, Shi, Wirth, and Yao [9] and refined by Bar-Yossef, Jayram, Kumar, and Sivakumar [2].

The Main Upper Bound: We now give an intuitive overview of the proof of Theorem 1.3. We first show that Miltersen's observation of a natural reduction from communication protocols to cell probe algorithms has a partial converse: if we can design a short *memoryless* communication protocol for a problem, then we also have an efficient cell probe algorithm for the problem. We call a protocol memoryless if any particular message of Bob depends only on the last message received from Alice, i.e., Bob is not allowed to "remember" the entire communication history.

We then describe a memoryless communication protocol for $\mathsf{ANN}^{\alpha}_{d,n}$. Let $\gamma = \sqrt{\alpha}$. For any given λ, using "dimension reduction" techniques we can use two rounds of communication to distinguish between the case where there exists a database point within distance λ of the query point and the case where all database points are at least $\gamma\,\lambda$ away from the query. Since our metric space is the Hamming cube, it suffices to try out $\log_\gamma d$ values for λ: $1, \gamma, \gamma^2, \ldots$, to obtain an γ^2-approximation. Using binary search on these special

values of λ gives a memoryless protocol with $O(\log\log d)$ rounds. This is the essence of all previously known ANN algorithms.

In order to improve the round complexity to $O(\log\log d/\log\log\log d)$, we replace the binary search with a t-ary search, i.e., in each phase we partition the current search range into t equal-sized sub-ranges where $t = \Theta(\log\log d/\log\log\log d)$. The main technical difficulty is to zoom in to one of the t sub-ranges spending only a constant number of rounds of communication. A naïve approach would have Alice asking Bob t "questions," one for each sub-range, and thus using t rounds. Our new idea is to perform "coarse" queries: such queries are shorter and hence t of them can fit in one message. On the other hand, they are not perfect: database points that are far can be labelled as close and vice versa. Hence, our t-ary search might sometimes fail to shrink the current search range by a factor of t. We show that in such cases we can still modify the current range in such a way that the *number of database points* whose distance to the target point is within the current range decreases by a factor of $n^{-1/t}$.

To summarise, each phase takes only a constant number of rounds and either shrinks the search range by a factor of t or decreases the size of the database by a factor of $n^{-1/t}$; hence, there are at most $O(t)$ rounds to the protocol.

1.3 The Connection With Communication Complexity

We shall prove a cell probe lower bound for LPM via a lower bound on the corresponding communication problem. We shall consider only two-party communication problems in this paper. A communication problem is defined by a relation $\rho \subseteq \mathscr{A} \times \mathscr{B} \times \mathscr{C}$, just like a data structure query problem. The input is split between two players called *Alice* and *Bob*: Alice is given $x \in \mathscr{A}$ and Bob is given $y \in \mathscr{B}$. The players then take turns exchanging messages according to a possibly randomised *protocol*, at the end of which Alice outputs a z such that $(x, y, z) \in \rho$ with probability at least $1 - \varepsilon$. Each message transfer is called a *round* of the protocol.

Definition 1.4 (Notation for Protocols). An $\langle a_1, a_2, \ldots, a_t\rangle^A$-protocol is one with exactly t rounds, with the message in the ith round being exactly $\lfloor a_i \rfloor$ bits long.[4] The superscript "A" indicates that Alice sends the first message in the protocol; we use a "B" superscript if Bob starts. An $[a, b, t]^A$-protocol is one with t rounds and Alice starting, in which each of Alice's messages is $\lfloor a \rfloor$ bits long and each of Bob's messages is $\lfloor b \rfloor$ bits long. An $[a, b, t]^B$-protocol is the same thing except that Bob starts. An $[a, b, t;\ a_0]^A$-protocol is a t-round protocol where Alice starts, each of Bob's messages is $\lfloor b \rfloor$ bits long, and as for Alice, her *first* message is $\lfloor a_0 \rfloor$ bits long and all subsequent messages are $\lfloor a \rfloor$ bits long. Similarly, an $[a, b, t;\ b_0]^B$-protocol is a t-round protocol where Bob starts, each of Alice's messages is $\lfloor a \rfloor$ bits long, Bob's *first* message is $\lfloor b_0 \rfloor$ bits long and all his subsequent messages are $\lfloor b \rfloor$ bits long.

The following simple observation links the cell probe and communication models.

Fact 1.5 (Miltersen [22]). *If a data structure query problem has a t-probe algorithm with table size s and word size w, then it has a private coin randomised $[\lceil \log s \rceil, w, 2t]^A$-protocol.*

1.4 Is There a Simpler Proof?

We make an additional interesting observation about ANN. Thus far there have been two main techniques used in almost all of the research on cell probe lower bounds: the so-called "richness technique" and the aforementioned round elimination technique. Of these, the richness technique, which establishes a lower

[4] We allow the a's to be non-integral for notational convenience.

bound on the bit complexity of a communication problem, is considerably simpler. Lower bounds given by the richness technique have the following form: "either Alice sends a bits or Bob sends b bits;" notice that round complexity is not a consideration.

Jayram, Khot, Kumar, and Rabani [19] recently used the richness technique to obtain a randomised cell probe lower bound for the so-called *exact partial match* problem, which reduces to the *exact* nearest neighbour (ENN) problem. Even more recently, Liu [21] gave a strong cell probe lower bound for *deterministic* ANN using the richness technique; his proof is considerably simpler than that of Chakrabarti et al. [7] who implicitly used round elimination ideas. In view of this history it is natural to ask whether there is a much simpler proof of the Main Lower Bound using the richness technique.

Suppose that $\mathsf{ANN}^{\alpha}_{d,n}$ has a randomised t-probe algorithm with table size $s = n^{O(1)}$ and word size $w = d^{O(1)}$. Fact 1.5 tells us that it has a randomised $[O(\log n), d^{O(1)}, 2t]^A$-protocol. In this protocol Alice sends $O(t \log n)$ bits and Bob sends $td^{O(1)}$ bits. For the richness technique to yield an interesting result we would have to show that this is impossible for small t, i.e., that there is no protocol in which Alice sends only $O(t \log n)$ bits and Bob sends $td^{O(1)}$.

However, the following theorem shows that such a protocol is possible even with t a constant! Thus the richness technique, which can handle randomised ENN and deterministic ANN, is provably too weak to handle randomised ANN.

Theorem 1.6 (Failure of the Richness Technique). *For any $n \in 2^{\Omega(\log^2 d)}$ and any $\alpha > 1$ there is a private coin randomised communication protocol for $\mathsf{ANN}^{\alpha}_{d,n}$ in which Alice sends $O(\log n)$ bits and Bob sends $d^{O(1)}$ bits.*

1.5 Organisation of the Paper

The rest of the paper is organised as follows. Section 2 formally defines LPM and gives the reduction from LPM to ANN. Section 3 prepares a toolkit of three lemmas for manipulating communication protocols. Section 4 is the heart of our lower bound proof and contains our improved round elimination lemma; it uses the toolkit developed in Section 3. The brief Section 5 puts everything together to prove the Main Theorem. Section 6 contains the proofs of our two upper bounds, one in each subsection.

2 The Longest Prefix Match Problem and a Reduction to ANN

Let Σ be a finite alphabet. For strings $x_1, x_2 \in \Sigma^m$, we let $\mathrm{match}(x_1, x_2)$ denote the length of the longest prefix of x_1 which is also a prefix of x_2; thus $0 \le \mathrm{match}(x_1, x_2) \le m$. We will prove that the following auxiliary problem can be reduced to ANN:

Definition 2.1 (Longest Prefix Match). For integers $m, n \ge 1$ and a finite set Σ we define the Longest Prefix Match problem $\mathsf{LPM}^{\Sigma}_{m,n}$ as the data structure query problem given by

$$\mathscr{A} = \Sigma^m, \quad \mathscr{B} = \binom{\Sigma^m}{n}, \quad \mathscr{C} = \Sigma^m$$

$$\rho = \left\{(x, y, z) \in \mathscr{A} \times \mathscr{B} \times \mathscr{C} : z \in y \ \wedge \ (\forall z' \in y \ (\mathrm{match}(x, z) \ge \mathrm{match}(x, z')))\right\}.$$

For $a \in \{0,1\}^d$ and $r \ge 0$, we shall refer to a subset $\{x \in \{0,1\}^d : \mathrm{dist}(x, a) \le r\}$ of the Hamming cube as the *Hamming ball* of radius r centred at a.

Definition 2.2. For $\alpha > 0$, a family of balls is said to be *α-separated* if the distance between any two points belonging to distinct balls in the family is more than α times the diameter (i.e., twice the radius) of any ball in the family.

Lemma 2.3. *Let $d \geq 1$ be a large enough integer, and let $\beta = 2^{\log^{1-\eta} d}$, as defined in Equation* (1)*. There exists a rooted tree $\mathcal{T}$ whose vertices are Hamming balls in $\{0,1\}^d$ and which satisfies the following properties:*
(i) *If v is a child of u in $\mathcal{T}$, then $v \subseteq u$.*
(ii) *Each non-leaf vertex of $\mathcal{T}$ has exactly $\lceil 2^{d^{0.99}} \rceil$ children.*
(iii) *Each depth-i vertex (the root being a depth-0 vertex) has radius $d/(8\beta)^i$.*
(iv) *The depth-i vertices form a β-separated family.*
(v) *The leaves of $\mathcal{T}$ are at depth $\lfloor \log^{\eta/2} d \rfloor$, where η is the constant from* (1)*.*

Proof. The proof uses a construction by Chakrabarti et al. [7, Lemmas 3.2–3.4]. We first note that to construct a suitable tree $\mathcal{T}$, we need only construct balls of radius at least $d/(8\beta)^{\lfloor \log^{\eta/2} d \rfloor} \geq d^{0.995}$, by (1). We claim that inside a Hamming ball of radius r (where $d^{0.995} \leq r \leq d$) in $\{0,1\}^d$ there exists a β-separated family of $\lceil 2^{d^{0.99}} \rceil$ balls, each of radius $r/(8\beta)$. The lemma then follows from this claim by a natural recursive construction.

To prove the claim, we use a volume argument. Let B be a Hamming ball of radius r within which we wish to find our β-separated family and let B' be a Hamming ball concentric with B and of radius $r/2$. It suffices to find a set S of at least $2^{d^{0.99}}$ points inside B' such that any two distinct points in S are at distance more than $r/3$. The family of radius-$r/(8\beta)$ balls with centers at points in S will then be β-separated, because

$$\frac{r}{3} - \frac{2r}{8\beta} \;\geq\; \frac{r}{4} \;=\; \beta \cdot \frac{2r}{8\beta}\,.$$

A suitable set S can be constructed greedily. Start with all points in B' unmarked and $S = \emptyset$. Repeatedly pick an unmarked point in B', add it to S, and mark all points within distance $r/3$ of it. This can be done at least $V_d(r/2)/V_d(r/3)$ times, where $V_d(\rho)$ denotes the volume of a radius-ρ Hamming ball in $\{0,1\}^d$. Clearly, the distance between two distinct points in S is more than $r/3$, and

$$|S| \;\geq\; \frac{V_d(r/2)}{V_d(r/3)} \;\geq\; \frac{\binom{d}{\lfloor r/2 \rfloor}}{r\binom{d}{\lfloor r/3 \rfloor}} \;=\; \frac{1}{r} \cdot \prod_{i=0}^{\lfloor r/2 \rfloor - \lfloor r/3 \rfloor - 1} \frac{d - \lfloor r/3 \rfloor - i}{\lfloor r/2 \rfloor - i} \;\geq\; \frac{1}{r}\left(\frac{4}{3}\right)^{\lfloor r/2 \rfloor - \lfloor r/3 \rfloor} \;\geq\; 2^{d^{0.99}},$$

where the final inequality holds because $r \geq d^{0.995}$. □

Lemma 2.4 (Reduction from LPM to ANN). *Let $d \geq 1$ be a large enough integer, let β be as defined in* (1)*, and set $m := \lfloor \log^{\eta/2} d \rfloor$. Let Σ be an alphabet of size $\lceil 2^{d^{0.99}} \rceil$. If $\mathsf{ANN}^{\beta}_{d,n}$ has a t-probe algorithm using table size s and word size b, then so does $\mathsf{LPM}^{\Sigma}_{m,n}$.*

Proof. Fix a tree $\mathcal{T}$ whose existence is guaranteed by Lemma 2.3 and a numbering of its vertices so that we can refer to "the ith child" of a vertex. Let $L \subseteq \{0,1\}^d$ be the set of centres of the leaves of $\mathcal{T}$.

Identify the letters in Σ with the integers in $\{1, 2, \ldots, \lceil 2^{d^{0.99}} \rceil\}$ in some arbitrary manner. We can now define a mapping $\varphi : \Sigma^m \to L$ as follows. Given a string $\sigma = a_1 a_2 \ldots a_m \in \Sigma^m$ we consider the root-to-leaf path in $\mathcal{T}$ obtained by starting from the root, going to its a_1th child, then going to the a_2th child of that vertex, and so on (notice that the leaves are at depth m); we define $\varphi(\sigma)$ to be the centre of the leaf reached by this path. By the properties of $\mathcal{T}$ enumerated in Lemma 2.3, φ is clearly a bijection.

Now, based on a cell probe algorithm $\mathcal{A}$ for $\mathsf{ANN}^{\beta}_{d,n}$, we get a cell probe algorithm for $\mathsf{LPM}^{\Sigma}_{m,n}$ as follows. Given a database $y \subseteq \Sigma^m$, we preprocess the set $\varphi(y) := \{\varphi(w) : w \in y\} \subseteq \{0,1\}^d$ as $\mathcal{A}$ would. Then, given a query $x \in \Sigma^m$, we use the query scheme of $\mathcal{A}$ to find a point $\widetilde{z}$ that is a β-approximate nearest neighbour of $\varphi(x)$ in the set $\varphi(y)$. We return $z := \varphi^{-1}(\widetilde{z})$ as the answer to the LPM query. Clearly this algorithm uses the same number of probes, table size, and word size as $\mathcal{A}$.

Let $k := \text{match}(x, z)$ and let z' be an arbitrary string in y. To prove that this algorithm is correct, it suffices to show that $\text{match}(x, z') \leq k$. Suppose $k < m$, for otherwise there is nothing to prove. Then the $(k+1)$th symbols of x and z are different, whence $\varphi(x)$ and $\varphi(z)$ lie in distinct balls each of which is a depth-$(k+1)$ vertex of $\mathcal{T}$. Now

$$\text{dist}(\varphi(x), \varphi(z')) \quad \geq \quad \frac{\text{dist}(\varphi(x), \varphi(z))}{\beta} \quad > \quad \frac{2d}{(8\beta)^{k+1}},$$

where the first inequality holds because $\varphi(z)$ is a β-approximate nearest neighbour and the second follows from Parts (iii) and (iv) of Lemma 2.3. Thus, $\varphi(x)$ and $\varphi(z')$ cannot both lie inside the same depth-$(k+1)$ vertex of $\mathcal{T}$, whence $\text{match}(x, z') \leq k$. □

3 Protocol Manipulations

As mentioned in Section 1.2, our strategy is to prove a certain round elimination lemma for LPM and thus obtain a communication lower bound. For this we first develop a toolkit of three lemmas that allow us to manipulate protocols in certain ways.

The first of these, which we call the Message Switching Lemma, says that the order of the first two messages in a protocol can be switched at the cost of blowing up the sizes of the messages. In the communication problems arising from the cell probe model, Bob's message bits are considerably "cheaper" than Alice's, so it is not too bad if Bob's message size blows up. Our exploitation of this asymmetry is crucial; it was not required in Sen's work on the predecessor problem [28].

The second lemma in the toolkit, which we call the Uninformative Message Lemma, and is due to Sen [28], is concerned with protocols where the first message conveys only a fraction of a bit of information about the input and is thus essentially *uninformative*; the lemma says that we may modify the protocol so that this first message is never sent. Finally, the third of our toolkit lemmas, called the Message Compression Lemma, says that if the first message in a protocol is long in terms of number of bits but conveys a much smaller amount of information about the input, then we may *compress* this message so that its length is linear in the amount of information conveyed. The last two lemmas increase the error of the protocol but only by a small *additive* amount, a fact that will be crucial in our applications.

Lemma 3.1 (Message Switching Lemma). *Let P be a deterministic $[a, b, t;\ a_0]^A$-protocol with $t \geq 2$. Then there exists a deterministic $[a + a_0, b, t-1;\ 2^{a_0}b]^B$-protocol that computes the exact same function as P.*

Proof. There are at most 2^{a_0} different messages that Alice may send as her first message. We design a new protocol Q in which Bob starts by sending his responses, as in P, to every one of these. If $t = 2$, we stop here. Otherwise, we let Alice's first message in Q be the concatenation of her first two messages in P; we can do this since Bob's first message gives Alice all the information she needs. At this point Alice and Bob both have all the information that they would have had after three rounds of P. So from now on they just follow P and clearly this results in their computing the exact same function as P. It is also clear that Q is an $[a + a_0, b, t-1;\ 2^{a_0}b]^B$-protocol. In fact only the first of Alice's messages in Q needs the extra a_0 bits but we will be happy with the weaker conclusion. □

For the other two lemmas in the toolkit, we need some more notation and a definition. Let P be a communication protocol and $\mathcal{D}$ a distribution on the possible inputs to P. We remark that $\mathcal{D}$ has two "parts" — one for Alice, one for Bob — but need not be a product distribution; for such distributions $\mathcal{D}$, we denote Alice's part (i.e., its marginal distribution) by $\mathcal{D}_A$ and Bob's part by $\mathcal{D}_B$. We let $\mathrm{err}(P, \mathcal{D})$ denote the distributional error probability of P under this input distribution. When P is a protocol with Alice starting, we let $\mathrm{msg}(P, x)$ denote Alice's first message in P when her input is x. Note that this may be a random variable if P is a randomised protocol. Slightly abusing notation, we use $\mathrm{msg}(Q, y)$ to denote Bob's first message when his input is y for protocols Q in which Bob starts.

Definition 3.2 (Information Cost). The information cost of a private coin protocol P with respect to input distribution $\mathcal{D}$, denoted $\mathrm{icost}(P, \mathcal{D})$, is defined to be the mutual information $\mathrm{I}(X : \mathrm{msg}(P, X))$, where X is a random input distributed according to $\mathcal{D}_A$ (if Alice starts P) or $\mathcal{D}_B$ (if Bob starts). We stress that our notion of information cost deals only with the *first* message of a protocol.

Lemma 3.3 (Uninformative Message Lemma [28]). *Let P be a private coin $\langle a_1, a_2, \ldots, a_t\rangle^A$-protocol for a communication problem ρ. Then, for any input distribution $\mathcal{D}$, there is a deterministic $\langle a_2, \ldots, a_t\rangle^B$-protocol P' for ρ such that* $\mathrm{err}(P', \mathcal{D}) \leq \mathrm{err}(P, \mathcal{D}) + \sqrt{\mathrm{icost}(P, \mathcal{D})}$. □

Remark. Notice that this lemma says something nontrivial only when $\mathrm{icost}(P, \mathcal{D})$ is a small fraction. Sen's proof of this lemma gave a tighter bound of $\sqrt{(\frac{1}{2}\ln 2)\mathrm{icost}(P, \mathcal{D})}$ on the additional error of P'. We drop the constant in order to simplify our calculations later.

Lemma 3.4 (Message Compression Lemma). *Let P be a private coin $\langle a_1, a_2, \ldots, a_t\rangle^A$-protocol for a communication problem ρ. Then, for any input distribution $\mathcal{D}$ and any $a > 0$, there is a deterministic $\langle a, a_2, \ldots, a_t\rangle^A$-protocol P' for ρ such that* $\mathrm{err}(P', \mathcal{D}) \leq \mathrm{err}(P, \mathcal{D}) + (2 \cdot \mathrm{icost}(P, \mathcal{D}) + C)/a$, *where $C > 0$ is a universal constant.*

We give a new proof of this lemma, different from the one in our original work [8], using a recent result of Harsha, Jain, McAllester and Radhakrishnan [16] on the one-way communication complexity of generating a correlated pair of random variables. We comment on our original proof at the end of this section. The result of Harsha et al. concerns a two-player game that we now formalise.

Definition 3.5 (Correlator). Let (X, Y) be a pair of discrete random variables with joint distribution Π. A *correlator* for (X, Y) is a one-way communication protocol with the following behaviour. Alice and Bob both know Π and share a public coin. Alice gets, as input, a random value X distributed according to the first marginal of Π. She must send a message to Bob who must then output a random value Y' with the property that (X, Y') is distributed according to Π.

The *cost* of a correlator is the expected length of Alice's message, with the expectation taken over the public random string as well as the randomness in Alice's input. The *correlation complexity* $T(X : Y)$ is defined to be the minimum cost of a correlator for (X, Y).

It is easy to see that a correlator exists for any pair of discrete random variables: Alice can simply send her input to Bob. The result we need about correlators is the following upper bound.

Lemma 3.6 (Harsha et al. [16]). *There is a universal constant $C > 0$ such that, for all pairs (X, Y) of discrete random variables, we have* $T(X : Y) \leq \mathrm{I}(X : Y) + 2\log(\mathrm{I}(X : Y) + 1) + C$. □

Proof of Lemma 3.4. Assume that protocol P is parametrized by three *independent* uniform random strings: R_{A1}, the string that Alice uses to provide the randomness in her first message, R_{A2}, the string she uses for

her subsequent messages, and R_B, the string Bob uses to randomise his messages. This assumption does not lose generality, because P can always be cast in this form. (To see this, note that one can simulate a general private coin protocol, in which Alice uses a single random string R_A for all her messages, as follows: after sending her first message, m, based on her input, x, and R_A, Alice uses a fresh random string R_{A2} to generate a random string R'_A distributed identically to $(R_A \mid x, m)$ and then uses R'_A in place of R_A for the rest of the protocol.)

Let ε^P be the following error indicator function for P: $\varepsilon^P(x, y, m, r_{A2}, r_B)$ is either 0 or 1 according as P produces a correct or an incorrect answer on input x, y when $R_{A2} = r_{A2}$, $R_B = r_B$, and Alice sends m as her first message. Let $\mu^P(x, r_{A1})$ be the function that Alice computes to produce her first message in P. Define the function $f(x, m) = \mathrm{E}_{Y,R_{A2},R_B}[\varepsilon^P(x, Y, m, R_{A2}, R_B)]$ where Y is distributed according to $(\mathcal{D}_B|X = x)$. Then

$$\begin{aligned}\mathrm{err}(P, \mathcal{D}) &= \mathrm{E}_{X,Y,R_{A1},R_{A2},R_B}\left[\varepsilon^P(X, Y, \mu^P(X, R_{A1}), R_{A2}, R_B)\right] && \text{(with } (X, Y) \sim \mathcal{D}\text{)}\\ &= \mathrm{E}_{X,R_{A1}}\left[f(X, \mu^P(X, R_{A1}))\right] && \text{(with } X \sim \mathcal{D}_A\text{)}\end{aligned}$$

where (R_{A1}, R_{A2}, R_B) is distributed uniformly.

Consider the protocol Q that is identical to P except for the first round, which is modified as follows. Alice and Bob use a correlator for $(X, \mu^P(X, R_{A1}))$ using a new *public* random string R_P: Alice sends Bob her correlator message and Bob generates his correlator output M with the property that (X, M) has the same distribution as $(X, \mu^P(X, R_{A1}))$. Alice mimics Bob and also generates the same value M. From this point on, they pretend that Alice had sent M as her first message to Bob in protocol P and follow the rest of P. By definition of f, we have

$$\mathrm{err}(Q, \mathcal{D}) = \mathrm{E}_{X,M}[f(X, M)] = \mathrm{E}_{X,R_{A1}}\left[f(X, \mu^P(X, R_{A1}))\right] = \mathrm{err}(P, \mathcal{D}).$$

Notice that $\mathrm{I}(X : \mu^P(X, R_{A1})) = \mathrm{icost}(P, \mathcal{D})$. By Lemma 3.6, the expected length of Alice's first message in Q can be made at most

$$T(X : \mu^P(X, R_{A1})) \leq \mathrm{icost}(P, \mathcal{D}) + 2\log(\mathrm{icost}(P, \mathcal{D}) + 1) + C \leq 2 \cdot \mathrm{icost}(P, \mathcal{D}) + C.$$

Let I denote this upper bound. Let Q' be a protocol that is identical to Q except for the first round, in which Alice truncates her first message in Q to at most a bits if necessary. By Markov's inequality, the probability that this truncation happens is at most I/a, whence $\mathrm{err}(Q', \mathcal{D}) \leq \mathrm{err}(Q, \mathcal{D}) + I/a = \mathrm{err}(P, \mathcal{D}) + I/a$. Fixing the random coins in Q' gives us the desired deterministic $\langle a, a_2, \ldots, a_t\rangle^A$-protocol P'. □

Remark. The original version of this work [8] contained a different proof of Lemma 3.4, which was based on the Substate Theorem of Jain, Radhakrishnan, and Sen [18], together with a rejection sampling argument inspired by the same work. Roughly speaking, what we proved there was a weaker version of Lemma 3.6 in which the correlator is imperfect, and introduces some error in the simulation of the joint distribution (X, Y). The recent Lemma 3.6 given by Harsha et al. [16] is based on more sophisticated arguments, and thereby achieves perfect simulation. This results in a cleaner form of Lemma 3.4, with improved parameters. We note, however, that the improvements in the lemma do not propagate to the applications here, namely to communication and cell probe lower bounds for LPM and ANN.

4 Round Elimination for LPM

We now come to the central part of our proof where we show how to eliminate messages one by one from a protocol for LPM. Fix an alphabet Σ over which to define instances of LPM. We now define a couple of parametrized predicates.

Definition 4.1. Let $\mathcal{A}(m, n, a, b, t, \varepsilon)$ denote the statement "$\mathsf{LPM}^{\Sigma}_{m,n}$ has a public coin randomised ε-error $[a, b, t]^A$-protocol." Let $\mathcal{B}(m, n, a, b, t, \varepsilon;\ b_0)$ denote the statement "$\mathsf{LPM}^{\Sigma}_{m,n}$ has a public coin randomised ε-error $[a, b, t;\ b_0]^B$-protocol."

Lemma 4.2 (Round Elimination Lemmas for LPM). *Let m, n, t, k, and ℓ be positive integers with k dividing m and ℓ dividing n. Let $a, b, \varepsilon, \delta$ be positive reals, and C be the universal constant from Lemma 3.4.*
(i) *If $t \geq 2$ and $2a/k \geq C$, then $\mathcal{A}(m, n, a, b, t, \varepsilon) \Rightarrow \mathcal{B}\big(m/k, n, a\big(1 + \frac{2}{\delta k}\big), b, t - 1, \varepsilon + 2\delta;\ 2^{2a/(\delta k)} b\big)$.*
(ii) *If $\ell \leq |\Sigma|$, then $\mathcal{B}(m, n, a, b, t, \varepsilon;\ b_0) \Rightarrow \mathcal{A}\big(m - 1, n/\ell, a, b, t - 1, \varepsilon + \sqrt{b_0/\ell}\big)$.*

Proof of Part (i). Assume $\mathcal{A}(m, n, a, b, t, \varepsilon)$. We shall demonstrate the existence of a public coin randomised $[a(1 + \frac{2}{\delta k}), b, t - 1;\ 2^{2a/(\delta k)} b]^B$-protocol for $\mathsf{LPM}^{\Sigma}_{m/k,\,n}$ with error at most $\varepsilon + 2\delta$. Let $S := \Sigma^{m/k}$. By Yao's minimax principle [29], it suffices to give, for any input distribution $\mathcal{D}$ on $S \times S^n$, a deterministic protocol for $\mathsf{LPM}^{\Sigma}_{m/k,\,n}$ with the same message lengths and distributional error at most $\varepsilon + 2\delta$. So fix some distribution $\mathcal{D}$, and let us now define several distributions based on $\mathcal{D}$. First, let $\mathcal{I}$ denote the distribution over $[k] \times S^*$ obtained as follows: choose $i \in [k]$ uniformly at random and then choose $\sigma \in S^{i-1}$ from distribution $\mathcal{D}_A^{i-1}$. Recall that $\mathcal{D}_A$ is our notation for the marginal distribution of "Alice's portion" of $\mathcal{D}$. Next, let s be some fixed element of S. For each pair (i, σ) in the support of $\mathcal{I}$ we define a distribution $\mathcal{D}_{i,\sigma}$ on $S^k \times S^{kn}$ as follows: draw a sample (x, y) from $\mathcal{D}$ and output $(\sigma\, x X_{i+1} \ldots X_k,\ \sigma\, y s^{k-i})$ where $X_{i+1}, \ldots, X_k$ are random strings drawn independently from $\mathcal{D}_A$.[5] Finally, let $\widetilde{\mathcal{D}}$ be the distribution on $S^k \times S^{kn}$ obtained by choosing (i, σ) from $\mathcal{I}$ and outputting a sample from $\mathcal{D}_{i,\sigma}$. By (the easy half of) Yao's minimax principle, there is a deterministic $[a, b, t]^A$-protocol P for $\mathsf{LPM}^{\Sigma}_{m,n}$ with distributional error at most ε under input distribution $\widetilde{\mathcal{D}}$. By definition, we have

$$\mathrm{E}_{i,\sigma}[\mathrm{err}(P, \mathcal{D}_{i,\sigma})] \;=\; \mathrm{err}(P, \widetilde{\mathcal{D}}) \;\leq\; \varepsilon\,, \tag{2}$$

where the expectation is over (i, σ) distributed according to $\mathcal{I}$.

Now consider the information cost of P with respect to $\widetilde{\mathcal{D}}$. On the one hand, $\mathrm{icost}(P, \widetilde{\mathcal{D}}) \leq a$, since Alice's first message is of length a. On the other hand, if $X = X_1 X_2 \ldots X_k$ is distributed according to $\widetilde{\mathcal{D}}_A = \mathcal{D}_A^k$, then

$$\begin{aligned}
\mathrm{icost}(P, \widetilde{\mathcal{D}}) &= \mathrm{I}(X : \mathrm{msg}(P, X)) \\
&= \sum_{i \in [k]} \mathrm{I}(X_i : \mathrm{msg}(P, X) \mid X_1 \ldots X_{i-1}) \\
&= \sum_{i \in [k]} \mathrm{E}_{\sigma}[\mathrm{I}(X_i : \mathrm{msg}(P, X) \mid X_1 \ldots X_{i-1} = \sigma)] \\
&= k \cdot \mathrm{E}_{i,\sigma}[\mathrm{I}(X_i : \mathrm{msg}(P, X) \mid X_1 \ldots X_{i-1} = \sigma)]
\end{aligned}$$

where the second equality is by the chain rule for mutual information. Hence we see that

$$\mathrm{E}_{i,\sigma}[\mathrm{I}(X_i : \mathrm{msg}(P, X) \mid X_1 \ldots X_{i-1} = \sigma)] \;\leq\; \frac{a}{k}\,. \tag{3}$$

Combining (2) and (3), and using an averaging argument, we see that there exists an integer $i \in [k]$ and a string $\sigma \in S^{i-1}$ such that

$$\mathrm{err}(P, \mathcal{D}_{i,\sigma}) + \frac{2 \cdot \mathrm{I}(X_i : \mathrm{msg}(P, X) \mid X_1 \ldots X_{i-1} = \sigma)}{2a/(\delta k)} \;\leq\; \varepsilon + \delta\,. \tag{4}$$

[5] If σ is a string and y a set of strings, $\sigma\, y$ denotes the set $\{\sigma\, \tau :\ \tau \in y\}$ of strings.

Fix a pair (i, σ) as above. We shall now define a private coin protocol Q'' for $\mathsf{LPM}^{\Sigma}_{m/k,\,n}$ that uses P as a black box. It works as follows: on input $(x, y) \in S \times S^n$, Alice constructs the string $\widetilde{x} := \sigma x X_{i+1} \ldots X_k$ where the X_j's are random strings she draws independently from $\mathcal{D}_A$ using her private coins, and Bob constructs the set $\widetilde{y} := \sigma y s^{k-i}$ of strings; they then run protocol P on input $(\widetilde{x}, \widetilde{y})$ and output the ith block of whatever string P outputs. Note that if (x, y) is chosen from $\mathcal{D}$, then $(\widetilde{x}, \widetilde{y})$ is distributed according to $\mathcal{D}_{i,\sigma}$. Moreover, from the description of the LPM problem it is clear that Q'' works whenever its call to P works. Therefore, we have

$$\mathrm{err}(Q'', \mathcal{D}) \;\leq\; \mathrm{err}(P, \mathcal{D}_{i,\sigma})\,.$$

Moreover,

$$\mathrm{icost}(Q'', \mathcal{D}) \;=\; \mathrm{I}(X_i : \mathrm{msg}(P, X) \mid X_1 \ldots X_{i-1} = \sigma)\,.$$

Applying the Message Compression Lemma 3.4 to Q'', gives us a deterministic $[a, b, t;\ 2a/(\delta k)]^A$-protocol Q' for $\mathsf{LPM}^{\Sigma}_{m/k,\,n}$ whose distributional error can be bounded as follows, using (4) and the fact that $2a/k \geq C$.

$$\mathrm{err}(Q', \mathcal{D}) \;\leq\; \varepsilon + \delta + \frac{C}{2a/(\delta k)} \;\leq\; \varepsilon + 2\delta\,.$$

Applying the Message Switching Lemma 3.1 with $a_0 = 2a/(\delta k)$ to Q' gives us a deterministic $[a(1 + 2/(\delta k)), b, t - 1;\ 2^{2a/(\delta k)} b]^B$-protocol Q with the same error probability as Q'. The protocol Q has all the properties we sought and we are done. □

Proof of Part (ii). Assume $\mathcal{B}(m, n, a, b, t, \varepsilon;\ b_0)$. Let $S = \Sigma^{m-1}$. As before, for an arbitrary input distribution $\mathcal{D}$ on $S \times S^{n/\ell}$, we demonstrate the existence of a deterministic $[a, b, t - 1]^A$-protocol for $\mathsf{LPM}^{\Sigma}_{m-1,n/\ell}$ with low distributional error. Fix ℓ distinct strings $s_1, \ldots, s_\ell \in \Sigma$; we can do this because $|\Sigma| \geq \ell$. For each $i \in [\ell]$, let $\mathcal{D}_i$ be the distribution on $\Sigma S \times (\Sigma S)^n$ obtained as follows: draw ℓ independent samples $(x_1, y_1), \ldots, (x_\ell, y_\ell)$ from $\mathcal{D}$, and output $(s_i x_i, s_1 y_1 \cup \cdots \cup s_\ell y_\ell)$. We also construct a distribution $\widetilde{\mathcal{D}}$ on $\Sigma S \times (\Sigma S)^n$ as follows: choose $i \in [\ell]$ uniformly at random, and output a sample from $\mathcal{D}_i$. By the easy half of Yao's minimax principle there is a deterministic $[a, b, t;\ b_0]^B$-protocol P for $\mathsf{LPM}^{\Sigma}_{m,n}$ with distributional error at most ε under input distribution $\widetilde{\mathcal{D}}$. By definition, we have

$$\mathrm{E}_i\,[\mathrm{err}(P, \mathcal{D}_i)] \;=\; \mathrm{err}(P, \widetilde{\mathcal{D}}) \;\leq\; \varepsilon\,. \tag{5}$$

Now consider $\mathrm{icost}(P, \widetilde{\mathcal{D}})$. On the one hand, it is at most b_0 since Bob's first message is of length b_0. On the other hand, if $Y = Y_1 Y_2 \ldots Y_\ell$ is distributed according to $\widetilde{\mathcal{D}}_B$, then

$$\mathrm{icost}(P, \widetilde{\mathcal{D}}) \;=\; \mathrm{I}(Y : \mathrm{msg}(P, Y)) \;\geq\; \sum_{i \in [\ell]} \mathrm{I}(Y_i : \mathrm{msg}(P, Y))\,,$$

where the inequality holds because the Y_i's are independent. Hence, we obtain that

$$\mathrm{E}_i\left[\mathrm{I}(Y_i : \mathrm{msg}(P, Y))\right] \;\leq\; \frac{b_0}{\ell}\,. \tag{6}$$

Combining (5) and (6), and using the concavity of the square root function and an averaging argument, we conclude that there exists an $i \in [\ell]$ such that

$$\mathrm{err}(P, \mathcal{D}_i) + \sqrt{\mathrm{I}(Y_i : \mathrm{msg}(P, Y))} \;\leq\; \varepsilon + \sqrt{b_0/\ell}\,.$$

Fix such an $i \in [\ell]$, and consider the following private coin protocols Q' for $\mathsf{LPM}^{\Sigma}_{m-1,n/\ell}$ which uses P as a black box: on input $(x, y) \in S \times S^{n/\ell}$, Alice constructs the string $\widetilde{x} := s_i x$ and Bob constructs the

set $\widetilde{y} := s_1 y_1 \cup \cdots \cup s_{i-1} y_{i-1} \cup s_i y \cup s_{i+1} y_{i+1} \cup \cdots \cup s_\ell y_\ell$ of strings, where the y_j's are random sets of strings drawn independently from $\mathcal{D}_B$; they then run protocol P on input $(\widetilde{x}, \widetilde{y})$ and output whatever string P outputs with the first symbol deleted. Note that if (x, y) is chosen from $\mathcal{D}$, then $(\widetilde{x}, \widetilde{y})$ is distributed according to $\mathcal{D}_i$. Moreover, from the description of the LPM problem it is clear that Q' works whenever its call to P works. Thus, we have

$$\mathrm{err}(Q', \mathcal{D}) \;\le\; \mathrm{err}(P, \mathcal{D}_i)\,.$$

Moreover,

$$\mathrm{icost}(Q', \mathcal{D}) \;=\; \mathrm{I}(Y_i : \mathrm{msg}(P, Y))\,.$$

Applying the Uninformative Message Lemma 3.3 to Q' we see that there exists a deterministic $[a, b, t-1]^A$-protocol Q with distributional error at most $\varepsilon + \sqrt{b_0/\ell}$ under $\mathcal{D}$ which has all the properties we sought. This completes the proof. □

Combining the two parts of the above round elimination lemma, and weakening the resulting statement (using $m/k - 1 \ge m/(2k)$), gives us the following corollary.

Corollary 4.3. *With $m, n, a, b, t, k, \ell, \varepsilon, \delta$ and C as above, $\ell \le |\Sigma|$, $2a/k \ge C$, and $t \ge 2$,*

$$\mathcal{A}(m, n, a, b, t, \varepsilon) \implies \mathcal{A}\left(\frac{m}{2k},\ \frac{n}{\ell},\ a\left(1 + \frac{2}{\delta k}\right),\ b,\ t-2,\ \varepsilon + 2\delta + \sqrt{\frac{2^{2a/(\delta k)} b}{\ell}}\right).$$

We can now prove the following result about the communication complexity of $\mathsf{LPM}^{\Sigma}_{m,n}$.

Theorem 4.4. *For all $c_1, c_2 > 0$ there exists a $c_3 > 0$ such that the following holds. Suppose $d, n \ge 1$ are sufficiently large integers satisfying $2^{\log^{1.01} d} \le n \le 2^{d^{0.99}}$. Let $m = \lfloor \log^{\eta/2} d \rfloor$, Σ be a set of cardinality $\lceil 2^{d^{0.99}} \rceil$, $a = c_1 \log n$ and $b = d^{c_2}$. If $\mathcal{A}(m, n, a, b, t, \frac{1}{4})$, then $t \ge c_3 \log\log d / \log\log\log d$.*

Proof. Assume, without loss of generality, that $c_1 \ge 1$. Let us define

$$\xi \;:=\; \frac{\eta \log\log d}{2 \log\log\log d}\,, \tag{7}$$

so that $\xi^\xi \le \lfloor \log^{\eta/2} d \rfloor = m$. We shall start by assuming $\mathcal{A}(m, n, a, b, \frac{\xi}{3c_1}, \frac{1}{4})$ and derive a contradiction, which will prove that $t > \frac{\xi}{3c_1}$, as desired. We ignore divisibility issues to avoid notational clutter. Set $\delta = \xi^{-1}$, $k = \xi^2$ (so that $\delta k = \xi$), and $\ell = n^{5c_1/\xi}$. We claim that for any non-negative integer $i \le \frac{\xi}{6c_1}$, it is the case that

$$\mathcal{A}\left(\frac{m}{(2k)^i},\ \frac{n}{\ell^i},\ a\left(1 + \frac{2}{\xi}\right)^i,\ b,\ \frac{\xi}{3c_1} - 2i,\ \frac{1}{4} + 3i\delta\right). \tag{8}$$

We prove our claim by induction on i. The base case $i = 0$ holds by our initial assumption. Suppose (8) holds for some particular $i \le \frac{\xi}{6c_1} - 1$. We would like to apply Corollary 4.3, so we need to ensure that $\ell \le |\Sigma|$ and that $2a(1 + 2/\xi)^i/k \ge C$. Using the upper bound on n, we see that $\ell \le n \le |\Sigma|$, and using the lower bound on n, we see that $2a(1 + 2/\xi)^i/k \ge a/k = \omega(1)$. Thus, the corollary applies and we conclude that

$$\mathcal{A}\left(\frac{m}{(2k)^{i+1}},\ \frac{n}{\ell^{i+1}},\ a\left(1 + \frac{2}{\xi}\right)^{i+1},\ b,\ \frac{\xi}{3c_1} - 2(i+1),\ \frac{1}{4} + 3i\delta + 2\delta + \delta'\right), \tag{9}$$

where the term δ' can be bounded as follows:

$$a\left(1+\frac{2}{\xi}\right)^{i} \;\le\; a\left(1+\frac{2}{\xi}\right)^{\xi/6} \;\le\; 2a\,,$$

$$\delta' \;\le\; \sqrt{\frac{2^{2(2a)/(\delta k)}b}{\ell}} \;=\; \sqrt{\frac{n^{4c_1/\xi}d^{c_2}}{n^{5c_1/\xi}}} \;\le\; \delta\,,$$

where the final inequality follows from the lower bound on n. Plugging this bound for δ' into (9), we conclude that (8) holds for $i+1$ as well. This proves the claim.

Now set $i = \frac{\xi}{6c_1}$ in (8). Some simple algebra shows that

$$(2k)^i \;=\; (2\xi^2)^{\xi/(6c_1)} \;\le\; (\xi^3)^{\xi/(6c_1)} \;\le\; m^{3/(6c_1)} \;\le m^{1/2}\,,$$

$$\ell^i \;=\; n^{(5c_1/\xi)(\xi/(6c_1))} \;=\; n^{5/6}\,,$$

$$\frac{1}{4}+3i\delta \;=\; \frac{1}{4}+3\cdot\frac{\xi}{6c_1}\cdot\frac{1}{\xi} \;=\; \frac{1}{4}+\frac{1}{2c_1} \;\le\; \frac{3}{4}\,,$$

whence we obtain $\mathcal{A}(m^{1/2}, n^{1/6}, 2a, b, 0, \frac{3}{4})$. But this is a contradiction, as we are solving a nontrivial communication problem with non-negligible success probability but with zero communication. □

Remarks: We note that our techniques in the proof of Lemma 4.2 in fact yield a new general round elimination lemma in the style of Sen [28] and not just one for LPM. A precise statement of such a lemma would introduce too much extra notation, so we refer the interested reader to Sen's work instead.

5 The Cell Probe Lower Bound for ANN

It is now simple to put together the results from the previous sections to obtain our main theorem.

Theorem 5.1 (Main Theorem restated). *For all $c_1, c_2 > 0$ there exists a $c_3 > 0$ such that the following holds. Let $n, d \ge 1$ be large enough integers, and suppose that $2^{\log^{1.01} d} \le n \le 2^{d^{0.99}}$. If $\mathsf{ANN}^{\beta}_{d,n}$ has a randomised t-probe algorithm with table size $s \le n^{c_1}$ and word size $w \le d^{c_2}$, then $t \ge c_3 \log\log d / \log\log\log d$.*

Proof. Let $m := \lfloor \log^{\eta/2} d \rfloor$ and let Σ be an arbitrary set of cardinality $\lceil 2^{d^{0.99}} \rceil$. By Lemma 2.4, $\mathsf{LPM}^{\Sigma}_{m,n}$ has a t-probe algorithm with table size s and word size w. Next, let $a := \lceil \log s \rceil$ and $b := w$. By Fact 1.5, $\mathsf{LPM}^{\Sigma}_{m,n}$ has a private coin randomised $[a, b, 2t]^A$-protocol, i.e., the statement $\mathcal{A}(m, n, a, b, 2t, \frac{1}{4})$ holds. Theorem 4.4 gives us the desired lower bound on t. □

6 Upper Bounds

Kushilevitz et al. [20] implicitly obtained an $O(\log\log d)$ cell probe upper bound for $\mathsf{ANN}^{\alpha}_{d,n}$, for any constant $\alpha > 1$, via a "dimension reduction" technique for the Hamming cube. In this section we prove a couple of upper bounds which use this technique, but in more complex ways than [20]. For our first result, which shows that the lower bound in the Main Theorem is tight, we need to bring in ideas used by Beame and Fich [4] in their work on upper bounds for the predecessor problem. Incidentally, [4] actually gives a cell probe algorithm for LPM; here we show that the harder ANN problem can also be similarly solved.

We need two lemmas which closely follow lemmas from [20]; we include the proofs for completeness. In this section we shall often treat points in Hamming cubes as column vectors over the field $GF(2)$, so that we can use linear algebraic notation. We will let n and d have their usual roles.

Definition 6.1. Let k be a positive integer and r a real number with $r \geq 1$. We define $\mathcal{V}_r$ to be the distribution of a random d-coordinate row vector in which each coordinate is independently chosen to be 1 with probability $1/(4r)$ and 0 otherwise. We define $\mathcal{M}_r^k$ to be the distribution of a random $k \times d$ matrix where each row is independently chosen from distribution $\mathcal{V}_r$.

Lemma 6.2. *Let $r \geq 1$ and $\gamma > 1$. Then, there exist two numbers $\delta_1(r, \gamma) < \delta_2(r, \gamma)$, both in $[0, 1]$, such that $\delta_2(r, \gamma) - \delta_1(r, \gamma)$ is at least some constant that depends only on γ and such that for all $d \geq 1$ and for all $x, z \in \{0, 1\}^d$,*

$$\begin{aligned} \mathrm{dist}(x, z) \leq r &\Rightarrow \Pr[Yx \neq Yz] \leq \delta_1(r, \gamma), \text{ and} \\ \mathrm{dist}(x, z) > \gamma r &\Rightarrow \Pr[Yx \neq Yz] > \delta_2(r, \gamma), \end{aligned}$$

where Y is a random row vector drawn from distribution $\mathcal{V}_r$.

Proof. Consider the following equivalent way of choosing Y: first choose a set $C \subseteq [d]$ where each integer in $[d]$ is put in C independently with probability $1/(2r)$. Then, for each $i \in C$, let the ith coordinate of Y be chosen uniformly from $\{0, 1\}$. For $i \notin C$, set the ith coordinate of Y to zero. Let $z \in \{0, 1\}^d$ be arbitrary and let $h = \mathrm{dist}(x, z)$. If C does not contain any of the coordinates on which x and z differ, then clearly $Yx = Yz$. This happens with probability $(1 - 1/(2r))^h$. Otherwise, if C contains at least one of the coordinates on which x and z differ, the probability that $Yx \neq Yz$ is precisely $1/2$. Hence,

$$\Pr[Yx \neq Yz] = \tfrac{1}{2}\left(1 - \left(1 - \tfrac{1}{2r}\right)^h\right).$$

It can be seen that this is a monotonically increasing function of h and that by plugging in r and γr for h one obtains two numbers whose difference is as claimed. □

Lemma 6.3. *Let $r \geq 1$ and $\gamma > 1$. Define $\delta(r, \gamma) = (\delta_1(r, \gamma) + \delta_2(r, \gamma))/2$, where δ_1, δ_2 are as in Lemma 6.2. Then, for all $d \geq 1$, all $u, v \in \{0, 1\}^d$, and all $k \geq 1$,*

$$\begin{aligned} \mathrm{dist}(u, v) \leq r &\Rightarrow \Pr[\,\mathrm{dist}(Mu, Mv) > \delta(r, \gamma) \cdot k\,] = e^{-\Omega(k)}, \text{ and} \\ \mathrm{dist}(u, v) > \gamma r &\Rightarrow \Pr[\,\mathrm{dist}(Mu, Mv) \leq \delta(r, \gamma) \cdot k\,] = e^{-\Omega(k)}, \end{aligned}$$

where M is a random matrix drawn from distribution $\mathcal{M}_r^k$ and the constant in the $\Omega(k)$ depends only on γ.

Proof. The lemma follows by combining Lemma 6.2 with the following Chernoff bound: For a sequence of m independent random variables on $\{0, 1\}$ such that for all i, $\Pr[X_i = 1] = p$ for some p, $\Pr[\sum X_i > (p + \tau)m] \leq e^{-2m\tau^2}$ and similarly $\Pr[\sum X_i < (p - \tau)m] \leq e^{-2m\tau^2}$. □

6.1 A Tight Cell Probe Upper Bound for ANN

In this section we show that the lower bound given by the main theorem is tight. We assume throughout that $n \geq d$ (say), for otherwise upper bounds for the problem are arguably not too interesting. We start by showing that it is enough to give a special kind of communication protocol for ANN.

Definition 6.4 (Memoryless Protocols). A communication protocol is said to be memoryless if each of Bob's messages depends only on the following: Bob's input, a random string in case the protocol is randomised, and the *most recent* message received from Alice (in a general protocol a message from Bob would depend on the entire communication history). Note that there is no restriction on Alice.

Lemma 6.5. *If $\mathsf{ANN}^{\alpha}_{d,n}$ has a memoryless public coin $[\lambda \log n,\ w, 2t]^A$-protocol, then it has a cell probe algorithm with t probes, table size at most $O(dn^{\lambda+1})$ and word size w.*

Proof. Note that the total input to the $\mathsf{ANN}^{\alpha}_{d,n}$ problem is $d + dn$ bits long. Therefore, the private versus public coin theorem of Newman [24] implies that it is enough to choose the public random string uniformly from a set of at most $s = O(d + dn)$ special strings. Thus we can modify the protocol so that only Alice is randomised and Bob is deterministic: Alice starts by choosing at most $\lceil \log s \rceil$ random bits to index into the list of special strings. She includes these $\lceil \log s \rceil$ bits in each of her messages to Bob so that Bob has access to the random coins of the original protocol and can behave deterministically. Notice that by including the coins in each message (and not just the first one), we ensure that the modified protocol — call it P — is also memoryless.

We now obtain the desired cell probe algorithm as follows. Number Alice's messages in P, which are each at most $(\lambda \log n + \log s)$ bits long, from 1 to $n^{\lambda} s = O(dn^{\lambda+1})$. The preprocessing phase produces a table whose ith entry contains Bob's response, in P, to message i from Alice; this is well-defined since Bob behaves deterministically and memorylessly in P. Also, the word size needed to fit Bob's messages remains w. The query phase simply simulates Alice's behaviour in P, using table lookups instead of messages from Bob. □

Theorem 6.6 (Cell Probe Algorithm for ANN). *Let $\alpha > 1$ be any constant. Then, for $n \geq d$, $\mathsf{ANN}^{\alpha}_{d,n}$ has a cell probe algorithm with $O(\log\log d / \log\log\log d)$ probes, table size $n^{O(1)}$ and word size $O(d)$.*

Remark. A similar upper bound has been independently discovered by Beame and Guruswami [5].

Proof. Without loss of generality, assume that $\alpha < 4$ and let $\gamma = \sqrt{\alpha}$. Let $x \in \{0,1\}^d$ denote the query point and $B \subseteq \{0,1\}^d$ denote the database. For $i \in \{0, 1, \ldots, \log_\gamma d\}$, let B_i be the set of all database points within distance γ^i of x. We start by checking for the degenerate case in which $x \in B$. This can be done with a constant number of cell probes using the technique of perfect hashing [14]. If indeed $x \in B$, the algorithm outputs x and ends. Similarly, in order to avoid certain boundary cases later, let us check if there exists a point in B within distance 1 of x. This can also be done with $O(1)$ cell probes by perfect hashing of all the points within distance 1 of B (there are at most dn such points). Again, if such a point is found, the algorithm outputs it and ends. Hence, since $\gamma < 2$, we can assume from now on that both B_0 and B_1 are empty.

Set $t = c_0 \log\log d / \log\log\log d$, with c_0 chosen so that

$$(t/2)^t \quad \geq \quad \log_\gamma d\,. \tag{10}$$

By Lemma 6.5, a memoryless public coin $[O(\log n), O(d), O(t)]^A$-protocol for $\mathsf{ANN}^{\alpha}_{d,n}$ will suffice.

Our protocol will find an i such that B_i is empty but B_{i+2} is not and will output a point in B_{i+2}; such a point is clearly an $\alpha = \gamma^2$-approximate nearest neighbour of x.

We start the protocol by choosing independent random matrices M_i from distribution $\mathcal{M}^{c_1 \log n}_{\gamma^i}$ and independent random matrices N_i from distribution $\mathcal{M}^{(c_2 \log n)/t}_{\gamma^i}$ (see Definition 6.1), for each $i \in \{0, \ldots, \log_\gamma d\}$. The constants c_1 and c_2 will be specified later. Since we are in the public coin model, these matrices are known to both Alice and Bob. For $0 \leq j \leq i \leq \log_\gamma d$ we define the sets

$$\begin{aligned} C_i &= \{z \in B : \operatorname{dist}(M_i x, M_i z) \leq \delta(\gamma^i, \gamma) \cdot c_1 \log n\}, \\ D_{i,j} &= \{z \in C_i : \operatorname{dist}(N_j x, N_j z) \leq \delta(\gamma^j, \gamma) \cdot (c_2 \log n)/t\}, \end{aligned}$$

where δ is as in Lemma 6.3. Lemma 6.3 says that C_i is an approximation to B_i in the following sense: a point in B_i may be left out of C_i (and a point not in B_{i+1} may get into C_i) with probability at most n^{-2}, provided we choose c_1 large enough. Similarly, $D_{i,j}$ is an approximation to the set of points in C_i that are within distance γ^j of x. Our protocol will *assume* that $B_i \subseteq C_i \subseteq B_{i+1}$ for all i. Under this assumption, by Lemma 6.3, a point in B_j is left out of $D_{i,j}$ (and a point in $C_i \setminus B_{j+1}$ may get into $D_{i,j}$) with probability at most $n^{-2/t}$ provided we choose c_2 large enough. Our protocol will additionally *assume* that at most a fraction $n^{-1/t}$ of B_j is not in $D_{i,j}$ and that at most a fraction $n^{-1/t}$ of $C_i \setminus B_{j+1}$ is in $D_{i,j}$.

Taking the union bound over all i and all n database points, we see that the first assumption is false with probability at most $(\log_\gamma d)\cdot n\cdot n^{-2} \le \frac{1}{8}$. For the second assumption, an application of Markov's inequality followed by a union bound over all i, j and the two parts of the assumption shows that it is false with probability at most $n^{-1/t}\cdot(\log_\gamma d)^2\cdot 2 \le \frac{1}{8}$. Thus, an assumption is false with probability at most $\frac{1}{4}$; this will bound the error probability of the protocol.

The protocol proceeds as follows. Alice maintains two integers r and s, initialised to 0 and $\log_\gamma d$ respectively. The protocol is composed of at most $3t$ *shrinking phases*, each of which consists of at most 4 rounds, followed by a *completion phase*, which consists of at most $6t$ rounds. The protocol maintains the invariant that at the start of each shrinking phase $r < s$, C_r is empty and C_s is nonempty. Note that this holds at the very beginning (C_0 is empty since it is contained in B_1). Moreover, each shrinking phase updates r and/or s in such a way that either $s' - r' \le (s-r)/t + 3$ or $|C_{s'}| \le 2n^{-1/t}|C_s|$, where r' and s' denote the updated values of r and s, respectively. When $s - r$ drops below (say) $3t$, the protocol stops the shrinking phases and moves on to the completion phase. As long as $s - r \ge 3t$, $(s-r)/t + 3 \le 2(s-r)/t$. Hence, in view of (10), there can be at most t shrinking phases in which $(s-r)$ shrinks by a factor of $2/t$. On the other hand, since C_s stays nonempty, there are at most $2t$ shrinking phases in which $|C_s|$ drops by a factor of $2n^{-1/t} \le n^{-1/(2t)}$. Thus, overall there are at most $3t$ shrinking phases, as claimed.

We now describe the completion phase. For i from $r+1$ to s, Alice sends Bob the vector $M_i x$ which is $O(\log n)$ bits long and gives Bob complete information about C_i. If C_i is empty, Bob replies "empty," otherwise he replies with an arbitrary point in C_i which is d bits long. Notice that Bob can do this memorylessly. Alice stops as soon as she receives a point, which, by the invariant, she eventually must. Since we enter the completion phase only when $s - r < 3t$, there are at most $6t$ rounds in this phase. Suppose Alice ends up with a point in C_{k+1}, so that C_k is empty. By our first assumption, $B_k \subseteq C_k$ is empty and $B_{k+2} \supseteq C_{k+1}$ contains this point. As observed earlier, this solves $\mathsf{ANN}^{\alpha}_{d,n}$.

Finally, we describe a shrinking phase. For $j \in [t-1]$ define $\rho_j = \lfloor r + \frac{j}{t}(s-r)\rfloor$. In the first round of the phase, Alice sends Bob the vectors $M_s x, N_{\rho_1}x, N_{\rho_2}x, \ldots, N_{\rho_{t-1}}x$. Examining the shapes of the matrices N_i, we see that this message of Alice is only $O(\log n)$ bits long. Alice's message gives Bob complete information about C_s as well as D_{s,ρ_j} for all $j \in [t-1]$. Bob replies (again, memorylessly) with the smallest $j \in [t-1]$ such that $|D_{s,\rho_j}| > n^{-1/t}|C_s|$, or with $j = t$ if no such j exists. If Bob's response is $j = 1$ (CASE 1), we skip the third and fourth rounds of this phase and Alice updates s to $\rho_1 + 1$, leaving r unchanged. Otherwise, in the third round Alice sends Bob the vector $M_{\rho_{j-1}-1}x$ and Bob replies with a bit indicating whether or not $C_{\rho_{j-1}-1}$ is empty. If it *is* empty (CASE 2), Alice updates r to $\rho_{j-1} - 1$ and if $j < t$, updates s to $\rho_j + 1$. If it is nonempty (CASE 3), Alice updates s to $\rho_{j-1} - 1$, leaving r unchanged.

Let us now verify that all the invariants hold after the phase ends. Clearly, in all three cases, $r < s$. Moreover, in CASE 1 and CASE 3, C_r is empty since r was not changed and in CASE 2, C_r is empty according to Bob's message. In CASE 3, C_s is nonempty according to Bob's message and in CASE 2 with $j = t$, C_s is nonempty because s was not changed. In order to show that C_s is nonempty in the remaining cases, recall that by our assumption, D_{s,ρ_j} contains at most $n^{-1/t}|C_s|$ points from outside B_{ρ_j+1}. Therefore, since $|D_{s,\rho_j}| > n^{-1/t}|C_s|$, it must contain at least one point from B_{ρ_j+1}. In particular, B_{ρ_j+1} and hence C_{ρ_j+1}

are nonempty.

In order to complete the proof, notice that in CASE 1 and CASE 2 the difference between the updated values of r and s is at most

$$(\lfloor r + \tfrac{j}{t}(s-r)\rfloor + 1) - (\lfloor r + \tfrac{j-1}{t}(s-r)\rfloor - 1) \quad \leq \quad \tfrac{s-r}{t} + 3\,,$$

and that in CASE 3 the size of the new C_s is

$$|C_{\rho_{j-1}-1}| \quad \leq \quad |B_{\rho_{j-1}}| \quad \leq \quad |D_{s,\rho_{j-1}}|/(1-n^{-1/t}) \quad \leq \quad 2|D_{s,\rho_{j-1}}| \quad \leq \quad 2n^{-1/t}|C_s|\,,$$

where we used our assumptions from above. Hence, in all three cases the phase shrinks either $s-r$ or $|C_s|$ as promised. □

6.2 A Protocol for ANN with Low Bit Complexity

Finally, we prove our other upper bound on ANN which shows that the richness technique would have failed to prove a nontrivial lower bound.

Theorem 6.7 (Bit Complexity Upper Bound for ANN). *For $n \in 2^{\Omega(\log^2 d)}$ and $\alpha > 1$ there is a private coin randomised communication protocol for $\mathsf{ANN}^{\alpha}_{d,n}$ in which Alice sends $O(\log n)$ bits and Bob sends $d^{O(1)}$ bits.*

Proof. We will present a public coin protocol; a private coin protocol follows from the theorem of Newman [24]. Assume without loss of generality that $\alpha < 4$ and let $\gamma = \sqrt{\alpha}$. Let $x \in \{0,1\}^d$ denote the query point given to Alice and $B \subseteq \{0,1\}^d$ denote the database given to Bob. For $i \in \{0, \ldots, \log_\gamma d\}$ define B_i as the set of points in B within distance γ^i of x. Moreover, as in the previous upper bound, we can assume without loss of generality that both B_0 and B_1 are empty. This is done using perfect hashing, and requires Alice to send only $O(\log n)$ bits and Bob to send only $O(d)$ bits.

Let $r_0 \geq 2$ be the minimum number such that B_{r_0} is non-empty and fix y to be an arbitrary point in B_{r_0}. Our protocol outputs a point either in B_{r_0} or in B_{r_0+1}; this clearly implies a solution to $\mathsf{ANN}^{\alpha}_{d,n}$. Bob maintains a set of points $B' \subseteq B$ which is initially set to B. Alice keeps a value r which is initially set to $\log_\gamma d$. In the following description of the protocol, we describe certain *bad events* and we proceed assuming that they never happen. Later, we show that with high probability none of these events happens.

Our protocol consists of phases where each phase consists of two rounds. In the first round, Bob sends d^2 randomly chosen points from B'. If Alice finds a point in B_{r-1} then she decreases r by one and sends a message to Bob indicating that the phase is complete. Otherwise, we say that a bad event of the first type happened if $|B' \cap B_{r-1}| \geq |B'|/d$. Next, Alice uses the shared randomness to choose a matrix M from the distribution $\mathcal{M}^{c_1 \log d}_{\gamma^{r-2}}$ where c_1 is some constant to be specified later. She sends r and Mx to Bob; this takes $O(\log d)$ bits. Since Bob knows M, he can compute the set

$$\{z \in B' : \operatorname{dist}(Mx, Mz) \leq \delta(\gamma^{r-2}, \gamma) \cdot c_1 \log d\},$$

where δ is as in Lemma 6.3, and he sets the new B' to be this set. This ends the phase. We say that a bad event of the second type happened if the cardinality of the new B' is greater than $2/d$ times that of the old B'. We say that a bad event of the third type happened if $r \geq r_0 + 2$ and y is no longer in B'.

The protocol ends when the set B' becomes empty and then Alice outputs the point in B_r which she received when she decreased r to its current value.

Assuming none of the bad events happens, each phase either decreases r by one or shrinks the size of B' by $2/d$. Hence, the number of phases performed by the protocol is at most $\log_\gamma d + \log_{d/2} n$ which is

$O(\log n/\log d)$ by our assumption. Since Alice sends $O(\log d)$ bits in each phase, she sends $O(\log n)$ bits overall. Moreover, given that bad events of the third type do not happen, we know that the protocol stops when $r \leq r_0 + 1$. Also, since Alice decrements r only after seeing an element in B_{r-1}, we know that she never decrements r below r_0. Thus, the final r is either r_0 or r_0+1 and so Alice outputs a point in $B_{r_0} \cup B_{r_0+1}$ as promised. It remains to bound the probability of the bad events.

Let us consider one phase of the protocol. The probability that a bad event of the first type happens in this phase is at most

$$\left(1 - \tfrac{1}{d}\right)^{d^2} \quad \leq \quad e^{-d}\,.$$

So assume from now on that $|B' \cap B_{r-1}| < |B'|/d$ and that Alice does not find any point in B_{r-1}. According to Lemma 6.3, each element of $B' \setminus B_{r-1}$ is in the new B' with probability at most $1/d^2$ for large enough c_1. By Markov's inequality, the probability that more than a $1/d$ fraction of the points in $B' \setminus B_{r-1}$ remain is at most $1/d$. Hence, with probability at least $1 - 1/d$, the number of points in the new B' is at most

$$\tfrac{1}{d} \cdot |B'| + \tfrac{1}{d} \cdot |B' \setminus B_{r-1}| \quad \leq \quad \tfrac{1}{d} \cdot |B'| + \tfrac{1}{d} \cdot |B'| \quad = \quad \tfrac{2}{d} \cdot |B'|\,,$$

and therefore a bad event of the second type happens with probability at most $1/d$. According to Lemma 6.3, for $r \geq r_0 + 2$, the probability of y being thrown out of B' (the third bad event) is at most $1/d$ for large enough c_1. Summing up over the three bad events and using the union bound over all the phases, the probability that a bad event happens during the protocol is at most

$$O\left(\frac{\log n}{\log d} \cdot \left(e^{-d} + \frac{1}{d} + \frac{1}{d}\right)\right) \quad \leq \quad \frac{1}{4}\,,$$

which bounds the error probability of the protocol. □

Acknowledgments

We would like to thank Paul Beame, T. S. Jayram, Hartmut Klauck, Jaikumar Radhakrishnan, Pranab Sen, and Xiaodong Sun for many helpful discussions about various aspects of this paper. We also thank the anonymous referees for helpful comments.

References

[1] A. Andoni, P. Indyk, and M. Pătraşcu. On the optimality of the dimensionality reduction method. In *Proc. 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006. 449–458.

[2] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 209–218, 2002.

[3] O. Barkol and Y. Rabani. Tighter lower bounds for nearest neighbor search and related problems in the cell probe model. In *Proc. 32nd Annual ACM Symposium on the Theory of Computing*, pages 388–396, 2000.

[4] P. Beame and F. E. Fich. Optimal bounds for the predecessor problem. In *Proc. 31st Annual ACM Symposium on the Theory of Computing*, pages 295–304, 1999.

[5] P. Beame and V. Guruswami. Private communication. 2003.

[6] A. Borodin, R. Ostrovsky, and Y. Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. In *Proc. 31st Annual ACM Symposium on the Theory of Computing*, pages 312–321, 1999.

[7] A. Chakrabarti, B. Chazelle, B. Gum, and A. Lvov. A lower bound on the complexity of approximate nearest-neighbor searching on the hamming cube. *Disc. Comput. Geom.: The Goodman-Pollack Festschrift*, pages 313–328, 2003. Preliminary version in *Proc. 31st Annu. ACM Symp. Theory Comput.*, pages 305–311, 1999.

[8] A. Chakrabarti and O. Regev. An optimal randomised cell probe lower bound for approximate nearest neighbour searching. In *Proc. 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 473–482, 2004.

[9] A. Chakrabarti, Y. Shi, A. Wirth, and A. C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 270–278, 2001.

[10] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13(1):21–27, 1967.

[11] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Heidelberg, 2000.

[12] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *J. Amer. Soc. Inf. Sci.*, 41(6):391–407, 1990.

[13] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis*. John Wiley, New York, NY, 1973.

[14] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *J. ACM*, 31(3):538–544, 1984. Preliminary version in *Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 165–169, 1982.

[15] S. Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103, 2001.

[16] P. Harsha, R. Jain, D. McAllester, and J. Radhakrishnan. The communication complexity of correlation. In *Proc. 22nd Annual IEEE Conference on Computational Complexity*, pages 10–23, 2007.

[17] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. 30th Annual ACM Symposium on the Theory of Computing*, pages 604–613, 1998.

[18] R. Jain, J. Radhakrishnan, and P. Sen. A direct sum theorem in communication complexity via message compression. In *Proc. 30th International Colloquium on Automata, Languages and Programming*, pages 300–315, 2003.

[19] T. S. Jayram, S. Khot, R. Kumar, and Y. Rabani. Cell-probe lower bounds for the partial match problem. *J. Comput. Syst. Sci.*, 69(3):435–447, 2004. Preliminary version in *Proc. 35th Annual ACM Symposium on the Theory of Computing*, pages 667–672, 2003.

[20] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high-dimensional spaces. *SIAM J. Comput.*, 30(2):457–474, 2000. Preliminary version in *Proc. 30th Annual ACM Symposium on the Theory of Computing*, pages 614–623, 1998.

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

[22] P. B. Miltersen. Lower bounds for union-split-find related problems on random access machines. In *Proc. 26th Annual ACM Symposium on the Theory of Computing*, pages 625–634, 1994.

[23] 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. Preliminary version in *Proc. 27th Annual ACM Symposium on the Theory of Computing*, pages 103–111, 1995.

[24] I. Newman. Private vs. common random bits in communication complexity. *Inform. Process. Lett.*, 39(2):67–71, 1991.

[25] M. Pătraşcu and M. Thorup. Higher lower bounds for near-neighbor and further rich problems. *SIAM J. Comput.*, 39(2):730–741, 2009. Preliminary version in *Proc. 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 325–332, 2006.

[26] A. A. Salamov and V. V. Solovyev. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *J. Mol. Biol.*, 247(1):11–15, 1995.

[27] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, 1983.

[28] P. Sen. Lower bounds for predecessor searching in the cell probe model. In *Proc. 18th Annual IEEE Conference on Computational Complexity*, pages 73–83, 2003.

[29] A. C. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proc. 18th Annual IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1977.

[30] A. C. Yao. Should tables be sorted? *J. ACM*, 28(3):615–628, 1981.

[31] T. M. Yi and E. S. Lander. Protein secondary structure prediction using nearest-neighbor methods. *J. Mol. Biol.*, 232(4):1117–1129, 1993.