Electronic Colloquium on Computational Complexity, Report No. 70 (2003)

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

Amit Chakrabarti
Dartmouth College
ac@cs.dartmouth.edu *

Oded Regev
UC Berkeley
odedr@cs.berkeley.edu †

August 18, 2003

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 $\eta > 0$. This generalises an earlier result [6] on the deterministic complexity of the same problem and, more importantly, fills a major gap in the study of this problem since all earlier lower bounds either did not allow randomisation [6, 19] or did not allow approximation [5, 2, 16]. We also give a cell probe algorithm which 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 [20] 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 [7, 1, 24, 15]. In particular, we make heavy use of the round elimination and message compression ideas in the recent work of Sen [24] and Jain, Radhakrishnan, and Sen [15], and also introduce a new technique which we call message switching.

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 [27, 22], information retrieval [10, 23], and pattern recognition [8, 11], 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., [9]). However,

*Computer Science Department, Dartmouth College, Hanover, NH 03755. Most of this work was done while the author was at the Institute for Advanced Study, Princeton, NJ 08540. Work supported in part by NSF grant CCR-9987845.

†EECS Department, UC Berkeley, Berkeley, CA 94720. Work supported by the Army Research Office grant DAAD19-03-1-0082.

ISSN 1433-8092

in most applications 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 which 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 [14] and Kushilevitz, Ostrovsky, and Rabani [18]; some improvements were then made by Har-Peled [13].

Soon after the discovery of these algorithms, lower bounds on nearest neighbour searching were obtained by Chakrabarti, Chazelle, Gum, and Lvov [6] and simultaneously by Borodin, Ostrovsky, and Rabani [5]. 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 [19] and Barkol and Rabani [2] improved the respective lower bounds quantitatively but did not address this shortcoming.

In this paper, for the first time, we obtain a randomised lower bound for ANN, thus addressing this shortcoming. Moreover, we show that our lower bound is optimal.

1.1 Our Results

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, w.l.o.g., that at least one such z exists.

The standard framework for analysing the complexity of such problems is the *cell probe model* first defined by Yao [26]. The model assumes that the preprocessing phase deterministically constructs from y a data structure which is 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 $\frac{1}{4}$.

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) *Consider the data structure query problem given*

by

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

where "dist*" denotes Hamming distance in* $\{0,1\}^d$*. We call this the Approximate Nearest Neighbour search problem and abbreviate it as* $\mathsf{ANN}^{\beta}_{d,n}$*.*

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

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

where η is an arbitrarily small positive constant. Notice that taking $\eta = 0$ would make $\mathsf{ANN}^{\beta}_{d,n}$ a completely trivial problem, since we could return *any* database point as an answer to any query.[1] Our main result is the following lower bound on the randomised cell probe complexity of the problem.

Theorem 1.2 (Main Theorem) *Suppose* $n \in 2^{\Omega(\log^2 d)} \cap 2^{O(\sqrt{d})}$*. If* $\mathsf{ANN}^{\beta}_{d,n}$ *has a randomised* t*-probe algorithm with table size* $s = n^{O(1)}$ *and word size* $w = d^{O(1)}$*, then* $t = \Omega(\log\log d / \log\log\log d)$*.*

Let us examine our result in the light of past work on the problem. Chakrabarti, Chazelle, Gum, and Lvov [6] had obtained the same bound as above but only for deterministic algorithms. Liu [19] greatly improved their bound to $d^{1-o(1)}$, still for deterministic algorithms. On the flip side, Borodin, Ostrovsky, and Rabani [5] 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 [2]. Thus, our result is the first lower bound that allows both randomisation and approximation.

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. A similar upper bound has been independently discovered by Beame and Guruswami [4].

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

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 [18], Indyk and Motwani [14], and Har-Peled [13]. 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.

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 1.3.

1.2 Overview of the Proof

We give a brief overview of the proof of the Main Theorem. We start by showing a reduction from another data structure query problem problem, which we call *longest prefix match* (henceforth, LPM), to ANN. 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. The point of introducing this auxiliary problem will be explained soon.

[1] There is a small catch. If the database contains the query point, the only valid answer is the query. But this degenerate case can be handled by, say, perfect hashing [12] which has $O(1)$ cell probe complexity.

Following Miltersen, Nisan, Safra, and Wigderson [20], 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 the same thing as a data structure query problem, i.e., a relation $\rho \subseteq \mathscr{A} \times \mathscr{B} \times \mathscr{C}$. 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.

Remark : Throughout this paper all randomised communication protocols will be assumed to be public coin unless explicitly qualified otherwise.

Definition 1.4 (Notation for Protocols) *An $\langle a_1, a_2, \ldots, a_t \rangle^A$-protocol is one with exactly t messages exchanged, the i^{th} being a_i bits long. 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 a bits long and each of Bob's messages is b 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 b bits long, and as for Alice, her first message is a_0 bits long and all subsequent messages are a 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 a bits long, Bob's first message is b_0 bits long and all his subsequent messages are b bits long.*

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

Fact 1.5 *If a data structure query problem has a t-probe algorithm with table size s and word size w, then it has a $[\log s, w, 2t]^A$-protocol.*

We prove a communication lower bound for LPM based on the *round elimination* techniques pioneered by Miltersen et al. [20] and recently refined by Sen [24] in his work on another data structure query problem called the predecessor problem. In fact, we shall need a further strengthening of Sen's round elimination lemma for the problem we consider here. We shall establish this stronger lemma by using a version of the *information cost* paradigm introduced by Chakrabarti, Shi, Wirth, and Yao [7] and refined by Bar-Yossef, Jayram, Kumar, and Sivakumar [1], together with a new technique which we refer to as *message switching*.

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. If we start with too short a protocol, repeated application of the lemma 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.

The proof of a round elimination lemma involves "embedding" a smaller instance of the problem under consideration into a larger instance. Such an embedding turns out to be natural if the problem has to do with strings and substrings: a property that LPM has but ANN does not. This is why we work with LPM. We remark that our use of LPM is inspired by the work of Chakrabarti et al. [6] who used similar ideas but did not explicitly define the LPM problem.

1.3 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. Both these techniques originated in the work of

Miltersen et al [20]. Of these, the richness technique, which establishes a lower 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 [16] 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 [19] 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. [6] who implicitly used round elimination ideas. In view of this history it is natural to ask whether there is a much easier proof of the Main Theorem using the richness technique.

Suppose $\mathsf{ANN}^{\alpha}_{d,n}$ 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 $n \in 2^{\Omega(\log^2 d)}$ 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.*

1.4 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 protocols; included here is our new message switching lemma, as well as our adaptation of the message compression ideas of Jain et al [15]. 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

In this section we fix d to be a sufficiently large integer and we also fix a finite alphabet Σ with $|\Sigma| = 2^{\sqrt{d}}$. 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 $0 \leq \mathrm{match}(x_1, x_2) \leq m$.

We will prove that the following auxiliary problem can be reduced to ANN:

Definition 2.1 (Longest Prefix Match) *Consider the data structure query problem given by*

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

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

We call this the Longest Prefix Match problem and abbreviate it as $\mathsf{LPM}^{d}_{m,n}$.

For $a \in \{0,1\}^d$ and r an integer, we shall refer to a subset $\{x \in \{0,1\}^d : \mathrm{dist}(x,a) \leq r\}$ of the Hamming cube as the *Hamming ball* of radius r centred at a. Notice that every Hamming ball, except for the entire cube $\{0,1\}^d$ itself, has a unique centre.

Definition 2.2 *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 distance between any two points belonging to any one ball in the family. Here α is any positive real quantity.*

Lemma 2.3 *Let β be 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 $2^{\sqrt{d}}$ children.*
(iii) *Each depth-i vertex (the root being a depth-0 vertex) has radius $d/(16\beta)^i$.*
(iv) *The depth-i vertices form a β-separated family.*
(v) *The leaves of $\mathcal{T}$ are at depth $\log^{\eta/2} d$, where η is the constant from* (1)*.*

Proof : This lemma follows directly from a construction by Chakrabarti et al. [6, Lemmas 3.2–3.4] and involves routine volume arguments. We omit the details. ∎

Lemma 2.4 (Reduction from LPM to ANN) *Let β be as defined in* (1) *and set $m := \log^{\eta/2} d$. If $\mathsf{ANN}^{\beta}_{d,n}$ has a t-probe algorithm using table size s and word size b, then so does $\mathsf{LPM}^{d}_{m,n}$.*

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 i^{th} child" of a vertex. Let $L \subseteq \{0,1\}^d$ be the set of centres of the leaves of $\mathcal{T}$.

Recalling that $|\Sigma| = 2^{\sqrt{d}}$, identify the letters in Σ with the integers in $[2^{\sqrt{d}}]$ 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_1^{th} child, then going to the a_2^{th} 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 $\mathsf{LPM}^{d}_{m,n}$ 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 $k := \mathrm{match}(x,z)$ and let z' be an arbitrary string in y. To prove that this algorithm is correct, it suffices to show that $\mathrm{match}(x,z') \leq k$. Suppose $k < m$, for otherwise there is nothing to prove. Then the $(k+1)^{\text{th}}$ 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

$$\mathrm{dist}(\varphi(x),\varphi(z')) \quad \geq \quad \frac{\mathrm{dist}(\varphi(x),\varphi(z))}{\beta} \quad > \quad \frac{2d}{(16\beta)^{k+1}},$$

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

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 and was missing in Sen's work on the predecessor problem [24].

The second lemma in the toolkit, which we call the Uninformative Message Lemma, 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. This lemma was proven by Sen [24] in just the form we need. 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. This lemma is new and was not required in earlier work, but it is strongly inspired by the work of Jain, Radhakrishnan, and Sen [15] on direct sum theorems.

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 exactly 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 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.*

Now let us define a universal constant:

$$\gamma \;:=\; \sqrt{\frac{\ln 2}{2}}\,. \tag{2}$$

Lemma 3.3 (Uninformative Message 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 $\mathcal{D}$, there is a deterministic $\langle a_2, \ldots, a_t\rangle^B$-protocol P' for ρ such that $\mathrm{err}(P', \mathcal{D}) \le \mathrm{err}(P, \mathcal{D}) + \gamma\sqrt{\mathrm{icost}(P, \mathcal{D})}$.*

Remark : Notice that this lemma says something nontrivial only when $\mathrm{icost}(P, \mathcal{D})$ is a small fraction.

Proof : A proof can be found in [24]. We remark that the constant γ comes from the so-called Average Encoding Theorem [17] which is used in the proof of this lemma. ∎

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. Suppose, for an input distribution $\mathcal{D}$, we have $\mathrm{icost}(P, \mathcal{D}) = a$. Then, for any $\delta > 0$, there is a deterministic $\langle (1+a)/\delta^2 + 1/\delta, a_2, \ldots, a_t\rangle^A$-protocol P' for ρ such that $\mathrm{err}(P', \mathcal{D}) \le \mathrm{err}(P, \mathcal{D}) + 4\delta$.*

To prove this lemma, we need the following information theoretic lemma (inspired by the work of Jain, Radhakrishnan, and Sen [15] on direct sum theorems) whose proof we defer to Appendix A in order to avoid a digression into technicalities.

Lemma 3.5 *Let X and M be correlated random variables with ranges $\mathscr{X}$ and $\mathscr{M}$ respectively, and let $a = \mathrm{I}(X : M)$. Then, for any function $f : \mathscr{X} \times \mathscr{M} \to [0, 1]$ and any sufficiently small $\delta > 0$, there is a subset $\mathscr{M}'$ of $\mathscr{M}$ and a function $g : \mathscr{X} \to \mathscr{M}'$ such that*
(i) $\log|\mathscr{M}'| \le (1+a)/\delta^2 + 1/\delta$, *and*
(ii) $\mathrm{E}_X[f(X, g(X))] \le \mathrm{E}_{X,M}[f(X, M)] + 4\delta$.

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. It is not hard to see that P can always be cast in this form, so this assumption does not lose generality. 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. Then

$$\mathrm{err}(P, \mathcal{D}) \;=\; \mathrm{E}_{X,Y,R_{A1},R_{A2},R_B}\left[\varepsilon^P(X, Y, \mu^P(X, R_{A1}), R_{A2}, R_B)\right]\,, \tag{3}$$

where (X, Y) is distributed according to $\mathcal{D}$ and the R's are distributed uniformly. The key observation is that if a protocol Q is identical to P except for Alice's first message which is set to $\mu^Q(x)$ for some (deterministic) function μ^Q, then $\mathrm{err}(Q, \mathcal{D}) = \mathrm{E}_{X,Y,R_{A2},R_B}[\varepsilon^P(X, Y, \mu^Q(X), R_{A2}, R_B)]$; note that we are still using the same error indicator function ε^P.

Let $\mathscr{A}$ and $\mathscr{M}$ be the domains of Alice's input and her first message respectively. Let $M = \mu^P(X, R_{A1})$ and define $f : \mathscr{A} \times \mathscr{M} \to [0, 1]$ by $f(x, m) = \mathrm{E}_{Y,R_{A2},R_B}[\varepsilon^P(x, Y, m, R_{A2}, R_B)]$ where Y is distributed according to $(\mathcal{D}_B | X = x)$. Noting that $\mathrm{I}(X : \mu^P(X, R_{A1})) = \mathrm{icost}(P, \mathcal{D}) = a$ and applying Lemma 3.5, we see that there exists a subset $\mathscr{M}'$ of messages with $\log|\mathscr{M}'| \le (1+a)/\delta^2 + 1/\delta$ and a function $g : \mathscr{A} \to \mathscr{M}'$ such that

$$\mathrm{E}_X[f(X, g(X))] \;\le\; \mathrm{E}_{X,M}[f(X, M)] + 4\delta \;=\; \mathrm{err}(P, \mathcal{D}) + 4\delta\,,$$

where the final equality follows from (3). Let $\widetilde{P}$ be a protocol that follows P precisely except that Alice sends $g(x)$ as her first message on input x. By our key observation above, $\mathrm{E}_X[f(X, g(X))]$ is precisely $\mathrm{err}(\widetilde{P}, \mathcal{D})$. Now, fixing the random coins in $\widetilde{P}$ gives us a deterministic protocol P' whose distributional error under $\mathcal{D}$ is at most $\mathrm{err}(\widetilde{P}, \mathcal{D}) \leq \mathrm{err}(P, \mathcal{D}) + 4\delta$. The upper bound on $\log|\mathscr{M}'|$ implies P' is a $\langle (1+a)/\delta^2 + 1/\delta, a_2, \ldots, a_t\rangle^A$-protocol which completes the proof. ∎

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 a sufficiently large d and an alphabet Σ with $|\Sigma| = 2^{\sqrt{d}}$ over which to define instances of LPM. We define a couple of parametrized predicates (recall that protocols are by default public coin randomised).

Definition 4.1 *$\mathcal{A}(m, n, a, b, t, \varepsilon)$ denotes the statement "$\mathsf{LPM}^d_{m,n}$ has an ε-error $[a, b, t]^A$-protocol." Similarly $\mathcal{B}(m, n, a, b, t, \varepsilon;\ b_0)$ denotes the statement "$\mathsf{LPM}^d_{m,n}$ has an ε-error $[a, b, t;\ b_0]^B$-protocol."*

Lemma 4.2 (Round Elimination Lemmas for LPM) *Let $m, n, a, b, t, k,$ and ℓ be positive integers with k dividing m and ℓ dividing n, and let ε, δ be sufficiently small positive reals. Let γ be the constant defined in* (2)*.*
(i) *If $t \geq 2$ and $k \leq a$, then $\mathcal{A}(m, n, a, b, t, \varepsilon) \Rightarrow \mathcal{B}\big(m/k, n, a\big(1 + \frac{2}{\delta^3 k}\big), b, t-1, \varepsilon + 5\delta;\ 2^{2a/(\delta^3 k)} b\big)$.*
(ii) *If $\ell \leq 2^{\sqrt{d}}$, then $\mathcal{B}(m, n, a, b, t, \varepsilon;\ b_0) \Rightarrow \mathcal{A}\big(m-1, n/\ell, a, b, t-1, \varepsilon + \gamma\sqrt{b_0/\ell}\big)$.*

Remark : Actually, our technique does yield a new general round elimination lemma in the style of Sen [24] and not just one for LPM. However, we choose to avoid extra notation by not stating it.

Proof of Part (i): Assume $\mathcal{A}(m, n, a, b, t, \varepsilon)$. We shall demonstrate the existence of a randomised $[a(1 + \frac{2}{\delta^3 k}), b, t-1;\ 2^{2a/(\delta^3 k)} b]^B$-protocol for $\mathsf{LPM}^d_{m/k,\,n}$ with low error. Let $S := \Sigma^{m/k}$. By Yao's minimax principle [25], it suffices to give, for any input distribution $\mathcal{D}$ on $S \times S^n$, a deterministic protocol for $\mathsf{LPM}^d_{m/k,\,n}$ with the same message lengths and distributional error $\varepsilon + 5\delta$. For this we first construct a distribution $\widetilde{\mathcal{D}}$ on $S^k \times S^{kn}$ as follows: draw k independent samples $(x_1, y_1), \ldots, (x_k, y_k)$ from $\mathcal{D}$, choose $i \in [k]$ uniformly at random, and output $(x_1 x_2 \ldots x_k,\ x_1 \ldots x_{i-1} y_i s^{k-i})$ where $s \in S$ is some arbitrary fixed string.[2] By (the easy half of) Yao's minimax principle, there is a deterministic $[a, b, t]^A$-protocol P for $\mathsf{LPM}^d_{m,n}$ with distributional error ε under input distribution $\widetilde{\mathcal{D}}$.

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 $\mathcal{D}$.

We shall now construct a family $\{Q_{i,\sigma}\}$, indexed by (i, σ) in the support of $\mathcal{I}$, of private coin protocols for $\mathsf{LPM}^d_{m/k,\,n}$, each of which uses P as a black box. Protocol $Q_{i,\sigma}$ 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 drawn independently from $\mathcal{D}_A$, 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 i^{th} block of whatever string P outputs. From the description of the LPM problem it is clear that $Q_{i,\sigma}$ works whenever its call to P works. Therefore we have

$$\mathrm{E}_{i,\sigma}[\mathrm{err}(Q_{i,\sigma}, \mathcal{D})] \quad \leq \quad \mathrm{err}(P, \widetilde{\mathcal{D}}) \quad \leq \quad \varepsilon\,, \tag{4}$$

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

where the expectation is over (i,σ) distributed according to $\mathcal{I}$. Furthermore, if $X = X_1X_2\ldots X_k$ denotes a random string drawn from $\widetilde{\mathcal{D}}_A$, with each $X_j \in S$, then $\mathrm{msg}(Q_{i,\sigma}, X_i)$ has the same distribution as $\mathrm{msg}(P,X)$ conditioned on the event that $X_1\ldots X_{i-1} = \sigma$. This gives us

$$\begin{aligned}
\mathrm{E}_{i,\sigma}[\mathrm{icost}(Q_{i,\sigma},\mathcal{D})] &= \tfrac{1}{k}\textstyle\sum_{i\in[k]} \mathrm{E}_\sigma[\mathrm{I}(X_i : \mathrm{msg}(Q_{i,\sigma},X_i))] \\
&= \tfrac{1}{k}\textstyle\sum_{i\in[k]} \mathrm{E}_\sigma[\mathrm{I}(X_i : \mathrm{msg}(P,X) \mid X_1\ldots X_{i-1}=\sigma)] \\
&= \tfrac{1}{k}\textstyle\sum_{i\in[k]} \mathrm{I}(X_i : \mathrm{msg}(P,X) \mid X_1\ldots X_{i-1}) \\
&= \tfrac{1}{k}\cdot \mathrm{I}(X : \mathrm{msg}(P,X)) \qquad (5)\\
&\le a/k\,, \qquad (6)
\end{aligned}$$

where (5) is a standard result in information theory and (6) holds because $\mathrm{msg}(P,X)$ is a distribution over a-bit strings. Combining (4) and (6) gives

$$\mathrm{E}_{i,\sigma}\left[\mathrm{err}(Q_{i,\sigma},\mathcal{D}) + \frac{\delta k\cdot \mathrm{icost}(Q_{i,\sigma},\mathcal{D})}{a}\right] \;\le\; \varepsilon+\delta\,. \qquad (7)$$

Therefore there exists an i and a $\sigma \in S^{i-1}$ such that $\mathrm{err}(Q_{i,\sigma},\mathcal{D}) + (\delta k/a)\mathrm{icost}(Q_{i,\sigma},\mathcal{D}) \le \varepsilon+\delta$. Fix this pair (i,σ). By the above inequality, the corresponding protocol $Q_{i,\sigma}$ has distributional error at most $\varepsilon+\delta$ and, assuming $\varepsilon+\delta<1$, has information cost at most $a/(\delta k)$ under input distribution $\mathcal{D}$. Applying the Message Compression Lemma 3.4 to $Q_{i,\sigma}$, we see that there exists a deterministic $[a,b,t;\ a_0]^A$-protocol Q' with distributional error at most $\varepsilon+5\delta$ under $\mathcal{D}$, for

$$a_0 \;=\; (1+a/(\delta k))/\delta^2 + 1/\delta \;\le\; 2a/(\delta^3 k)\,,$$

where the last inequality follows because $k \le a$. Applying the Message Switching Lemma 3.1 to Q' gives us a deterministic $[a\big(1+\frac{2}{\delta^3 k}\big), b, t-1;\ 2^{2a/(\delta^3 k)}b]^B$-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 $\mathsf{LPM}^d_{m-1,n/\ell}$ with low distributional error. Fix ℓ distinct strings $s_1,\ldots,s_\ell \in \Sigma$; we can do this because $|\Sigma| = 2^{\sqrt{d}} \ge \ell$. We construct a distribution $\widetilde{\mathcal{D}}$ on $\Sigma S \times (\Sigma S)^n$ as follows: draw ℓ independent samples $(x_1,y_1),\ldots,(x_\ell,y_\ell)$ from $\mathcal{D}$, choose $i\in[\ell]$ uniformly at random, and output $(s_ix_i, s_1y_1\cup\cdots\cup s_\ell y_\ell)$. By the easy half of Yao's minimax principle there is a deterministic $[a,b,t;\ b_0]^B$-protocol P for $\mathsf{LPM}^d_{m,n}$ with distributional error ε under input distribution $\widetilde{\mathcal{D}}$.

We construct a family $\{Q_i\}_{i\in[\ell]}$ of private coin protocols for $\mathsf{LPM}^d_{m-1,n/\ell}$, each of which uses P as a black box. In Q_i, on input $(x,y)\in S\times S^{n/\ell}$, Alice constructs the string $\widetilde{x} := s_ix$ and Bob constructs the set $\widetilde{y} := s_1Y_1\cup\cdots\cup s_{i-1}Y_{i-1}\cup s_iy\cup s_{i+1}Y_{i+1}\cup\cdots\cup s_\ell Y_\ell$ 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. From the description of the LPM problem it is clear that Q_i works whenever its call to P works. Thus, arguing as in the proof of Part (i), $\mathrm{E}_i[\mathrm{err}(Q_i,\mathcal{D})] \le \varepsilon$. Meanwhile, if $Y = Y_1Y_2\ldots Y_\ell$ denotes a random string drawn from $\widetilde{\mathcal{D}}_B$, with each $Y_j \in S^{n/\ell}$,

$$\begin{aligned}
\mathrm{E}_i[\mathrm{icost}(Q_i,\mathcal{D})] &= \tfrac{1}{\ell}\textstyle\sum_{i\in[\ell]} \mathrm{I}(Y_i : \mathrm{msg}(Q_i,Y_i)) \\
&= \tfrac{1}{\ell}\textstyle\sum_{i\in[\ell]} \mathrm{I}(Y_i : \mathrm{msg}(P,Y)) \\
&\le \tfrac{1}{\ell}\cdot \mathrm{I}(Y : \mathrm{msg}(P,Y)) \qquad (8)\\
&\le b_0/\ell\,, \qquad (9)
\end{aligned}$$

where (8) holds because the Y_i's are independent and (9) holds because $\mathrm{msg}(P, Y)$ is a distribution over b_0-bit binary strings. Set $\varepsilon' := \varepsilon + \gamma\sqrt{b_0/\ell}$. By the concavity of the square root function,

$$\mathrm{E}_i\left[\mathrm{err}(Q_i, \mathcal{D}) + \gamma\sqrt{\mathrm{icost}(Q_i, \mathcal{D})}\right] \;\leq\; \varepsilon' \,,$$

whence there exists an i such that $\mathrm{err}(Q_i, \mathcal{D}) + \gamma\sqrt{\mathrm{icost}(Q_i, \mathcal{D})} \leq \varepsilon'$. Applying the Uninformative Message Lemma 3.3 to this Q_i we see that there exists a deterministic $[a, b, t-1]^A$-protocol Q with distributional error ε' 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 $m/k - 1 \geq m/(2k)$), gives us the following corollary.

Corollary 4.3 *With m, n, a, b, t, k, ℓ, and ε as above, $\ell \leq 2^{\sqrt{d}}$, $k \leq a$, and $t \geq 2$,*

$$\mathcal{A}(m, n, a, b, t, \varepsilon) \implies \mathcal{A}\left(\frac{m}{2k},\, \frac{n}{\ell},\, a\left(1 + \frac{2}{\delta^3 k}\right),\, b,\, t-2,\, \varepsilon + 5\delta + \gamma\sqrt{\frac{2^{2a/(\delta^3 k)} b}{\ell}}\right) .$$

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

Theorem 4.4 *Suppose $m = \log^{\eta/2} d$ and $n \in 2^{\Omega(\log^2 d)} \cap 2^{O(\sqrt{d})}$. If $\mathcal{A}(m, n, a, b, t, \frac{1}{4})$ with $a = \lambda \log n$ and $b = d^\mu$ for some constants λ, μ, then $t = \Omega(\log\log d / \log\log\log d)$.*

Remark : We have not tried to optimise the range of n in this theorem.

Proof : Assume, w.l.o.g., that $\lambda \geq 2$. Define

$$\xi \;:=\; \frac{\eta \log\log d}{2\log\log\log d} \,, \tag{10}$$

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

$$\mathcal{A}\left(\tfrac{m}{(2k)^i},\, \tfrac{n}{\ell^i},\, a\left(1 + \tfrac{2}{\xi}\right)^i,\, b,\, \tfrac{\xi}{3\lambda} - 2i,\, \tfrac{1}{4} + 6i\delta\right) . \tag{11}$$

We prove by our claim by induction on i. The base case $i = 0$ holds by our initial assumption. Suppose (11) holds for some particular $i \leq \frac{\xi}{6\lambda} - 1$. Note that our setting of parameters gives us

$$\begin{aligned}
\delta^3 k &= \xi \,, \\
a\left(1 + \tfrac{2}{\xi}\right)^i \left(1 + \tfrac{2}{\delta^3 k}\right) &= a\left(1 + \tfrac{2}{\xi}\right)^{i+1} \,, \\
a\left(1 + \tfrac{2}{\xi}\right)^i &\leq a\left(1 + \tfrac{2}{\xi}\right)^{\xi/8} \;\leq\; 2a \,, \\
\gamma\sqrt{\frac{2^{2(2a)/(\delta^3 k)} b}{\ell}} &= \gamma\sqrt{\frac{n^{4\lambda/\xi} d^\mu}{n^{5\lambda/\xi}}} \;\leq\; \delta \,,
\end{aligned} \tag{12}$$

where the final inequality follows from the lower bound on n. The upper bound on n ensures that $\ell \leq 2^{\sqrt{d}}$, so Corollary 4.3 applies and we conclude that (11) holds for $i+1$ as well. This completes the proof of the claim.

Now set $i = \frac{\xi}{6\lambda}$ in (11). Some simple algebra shows that

$$\begin{aligned}(2k)^i \;&=\; (2\xi^4)^{\xi/(6\lambda)} \;\leq\; (\xi^5)^{\xi/(6\lambda)} \;\leq\; m^{5/(6\lambda)} \;\leq m^{5/6}\,,\\ \ell^i \;&=\; n^{(5\lambda/\xi)(\xi/(6\lambda))} \;=\; n^{5/6}\,,\\ \tfrac{1}{4}+6i\delta \;&=\; \tfrac{1}{4}+6\cdot\tfrac{\xi}{6\lambda}\cdot\tfrac{1}{\xi} \;=\; \tfrac{1}{4}+\tfrac{1}{\lambda} \;\leq\; \tfrac{3}{4}\,,\end{aligned}$$

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

5 The Cell Probe Lower Bound for ANN

Finally, putting everything together gives us our main theorem.

Theorem 5.1 (Main Theorem restated) *Suppose $n \in 2^{\Omega(\log^2 d)} \cap 2^{O(\sqrt{d})}$. If $\mathsf{ANN}^{\beta}_{d,n}$ has a t-probe algorithm with table size $s = n^{O(1)}$ and word size $w = d^{O(1)}$, then $t = \Omega(\log\log d/\log\log\log d)$.*

Proof : Set $m := \log^{\eta/2} d$, $a := \log s = O(\log n)$, and $b := w = d^{O(1)}$. By Lemma 2.4, $\mathsf{LPM}^{d}_{m,n}$ has a t-probe algorithm with table size s and word size w. By Fact 1.5, $\mathsf{LPM}^{d}_{m,n}$ has an $[a,b,2t]^A$-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. [18] 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 [18]. 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 [3] in their work on upper bounds for the predecessor problem. Incidentally, [3] 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 [18]; 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 $\mathcal{M}^k_r$ 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 $x \in \{0,1\}^d$, $r \geq 1$, and $\alpha > 1$. Then, there exist two numbers $\delta_1(r), \delta_2(r) \in [0,1]$, $\delta_1(r) < \delta_2(r)$ such that $\delta_2(r) - \delta_1(r)$ is at least some constant that depends only on α and such that for all $z \in \{0,1\}^d$,*

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

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 i^{th} coordinate of Y be chosen uniformly from $\{0,1\}$. For $i \notin C$, set the i^{th} coordinate of y to zero. Let $z \in \{0,1\}^d$ be arbitrary and let $h = \text{dist}(x,z)$. 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 αr for h one obtains two constants whose difference is as claimed. ∎

Lemma 6.3 *Let $u, v \in \{0,1\}^d$, $r \geq 1$, $\alpha > 1$, and let k be a positive integer. Let $\delta(r) = (\delta_1(r) + \delta_2(r))/2$, where $\delta_1(r), \delta_2(r)$ are as in Lemma 6.2. Then,*

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

where M is a random matrix drawn from distribution $\mathcal{M}_r^k$.

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 $n > d^2$ (say), for otherwise an $O(1)$-probe algorithm is trivial. 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 $\mathsf{ANN}_{d,n}^{\alpha}$ has a memoryless public coin $[\lambda \log n,\, d^{\mu}, 2t]^A$-protocol, then it has a cell probe algorithm with t probes, table size at most $n^{\lambda+2}$ and word size d^{μ}.*

Proof : Note that the total input to the $\mathsf{ANN}_{d,n}^{\alpha}$ problem is $d + dn$ bits long. Therefore, the private versus public coin theorem of Newman [21] implies that it is enough to choose the public random string uniformly from a set of at most $O(d + dn) \leq n^2$ special strings. Thus we can modify the protocol so that only Alice is randomised and Bob is deterministic: Alice starts by choosing at most $2\log n$ random bits to index into the list of special strings. She includes these $2\log n$ 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 $(\lambda + 2)\log n$ bits long, from 1 to $n^{\lambda+2}$. The preprocessing phase produces a table

whose i^{th} 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 is d^{μ}. 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 $\mathsf{ANN}^{\alpha}_{d,n}$ has a cell probe algorithm with $O(\log\log d/\log\log\log d)$ probes, table size $n^{O(1)}$ and word size $d^{O(1)}$.*

Remark : A similar upper bound has been independently discovered by Beame and Guruswami [4].

Proof : Without loss of generality, assume that $\alpha < 2$. Let $x \in \{0,1\}^d$ denote the query point and $B \subseteq \{0,1\}^d$ denote the database. For $i \in \{0,1,\ldots,\log_\alpha d\}$, let B_i be the set of all database points within distance α^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 [12]. If indeed $x \in B$, the protocol 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 protocol outputs it and ends. Hence, 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

$$(t/2)^t \;\geq\; \log_\alpha d\,. \tag{13}$$

By Lemma 6.5, a memoryless public coin $[O(\log n), d^{O(1)}, O(t)]^A$-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 α^2-approximate nearest neighbour of x. This is just as good as an α-approximation; simply readjust α as necessary.

We start the protocol by choosing independent random matrices M_i from distribution $\mathcal{M}^{c_1\log n}_{\alpha^i}$ and independent random matrices N_i from distribution $\mathcal{M}^{(c_2\log n)/t}_{\alpha^i}$ (see Definition 6.1), for each $i \in \{0,\ldots\log_\alpha d\}$. 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 $0 \leq j \leq i \leq \log_\alpha d$ we define the sets

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

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 α^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 $(\log_\alpha d)\cdot n\cdot n^{-2} \leq \frac{1}{8}$. 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 $n^{-1/t}\cdot(\log_\alpha d)^2\cdot 2 \leq \frac{1}{8}$. 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 $\log_\alpha d$ 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 $s' - r' \leq (s-r)/t + 3$ or $|C_{s'}| \leq 2n^{-1/t}|C_s|$, 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 $s - r \geq 3t$, $(s-r)/t + 3 \leq 2(s-r)/t$. Hence, in view of (13), 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 $2n^{-1/t} \leq n^{-1/(2t)}$. 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 $\mathsf{ANN}^{\alpha^2}_{d,n}$.

Finally, we describe a shrinking phase. For $j \in [t]$ define $\rho_j = \lfloor r + \frac{j}{t}(s-r) \rfloor$. In the first round of the phase, Alice sends Bob the vectors $M_s x, N_{\rho_1} x, N_{\rho_2} x, \ldots, N_{\rho_t} x$. 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 $j \in [t]$. Bob replies (again, memorylessly) with the smallest $j \in [t]$ such that $|D_{s,\rho_j}| > n^{-1/t}|C_s|$. If this j is 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 s to $\rho_j + 1$ and r to $\rho_{j-1} - 1$. 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. Finally, in CASE 3, C_s is nonempty according to Bob's message. In order to show that C_s is nonempty in the two other 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

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

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 [21]. 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 $i \in \{0, \ldots, \log_\alpha d\}$ define B_i as the set of points in B within distance α^i of x. Assume without loss of generality that $\alpha < 2$. 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 $\mathsf{ANN}^{\alpha^2}_{d,n}$ and by readjusting α appropriately, this completes the proof. 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 $\log_\alpha d$. 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 randomly chooses a matrix M from the distribution $\mathcal{M}^{c_1 \log d}_{\alpha^{r-2}}$ where c_1 is some constant to be specified later. She sends r and Mx to Bob; this takes $O(\log d)$ bits. Since we are in the public coin model, Bob knows M and can compute the set

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

and he sets the new B' to be this set. This ends the phase. We expect that this will shrink $|B'|$ to at most $2/d$ times its previous value; if not, we say that a bad event of the second type happened. 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 $\log_\alpha d + \log_{d/2} n$ which is $O(\log n / \log d)$. 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

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

Assuming that a bad event of the first type did not happen, $|B' \cap B_{r-1}| < |B'|/d$. 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, Hartmut Klauck, Pranab Sen, and Xiaodong Sun for many helpful discussions about various aspects of this paper.

References

[1] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proc. 43rd Annu. IEEE Symp. Found. Comp. Sci.*, 2002. To appear.

[2] O. Barkol and Y. Rabani. Tighter lower bounds for nearest neighbor search and related problems in the cell probe model. In *Proc. 32nd Annu. ACM Symp. Theory Comput.*, pages 388–396, 2000.

[3] P. Beame and F. Fich. Optimal bounds for the predecessor problem. In *Proc. 31st Annu. ACM Symp. Theory Comput.*, pages 295–304, 1999.

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

[5] A. Borodin, R. Ostrovsky, and Y. Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. In *Proc. 31st Annu. ACM Symp. Theory Comput.*, pages 312–321, 1999.

[6] A. Chakrabarti, B. Chazelle, B. Gum, and A. Lvov. A lower bound on the complexity of approximate nearest-neighbor searching on the hamming cube. In *Proc. 31st Annu. ACM Symp. Theory Comput.*, pages 305–311, 1999.

[7] A. Chakrabarti, Y. Shi, A. Wirth, and A. C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proc. 42nd Annu. IEEE Symp. Found. Comp. Sci.*, pages 270–278, 2001.

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

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

[10] 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.

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

[12] 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 Annu. IEEE Symp. Found. Comp. Sci.*, pages 165–169.

[13] S. Har-Peled. A replacement for Voronoi diagrams of near linear size. In *Proc. 42nd Annu. IEEE Symp. Found. Comput. Sci.*, pages 94–103, 2001.

[14] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. 30th ACM Symp. Theory Comput.*, pages 604–613, 1998.

[15] R. Jain, J. Radhakrishnan, and P. Sen. A direct sum theorem in communication complexity via message compression. In *Proc. 30th ICALP*, 2003.

[16] T. S. Jayram, S. Khot, R. Kumar, and Y. Rabani. Cell-probe lower bounds for the partial match problem. In *Proc. 35th Annu. ACM Symp. Theory Comput.*, pages 667–672, 2003.

[17] H. Klauck, A. Nayak, A. Ta-Shma, and D. Zuckerman. Interaction in quantum communication and the complexity of set disjointness. In *Proc. 33rd Annu. ACM Symp. Theory Comput.*, pages 124–133, 2001.

[18] 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 Annu. ACM Symp. Theory Comput.*, pages 614–623, 1998.

[19] D. Liu. A strong lower bound for approximate nearest neighbor searching in the cell probe model. manuscript, 2003.

[20] 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 Annu. ACM Symp. Theory Comput.*, pages 103–111, 1995.

[21] I. Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67–71, 1991.

[22] 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.

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

[24] P. Sen. Lower bounds for predecessor searching in the cell probe model. In *Proc. 18th Annu. IEEE Conf. Comput. Complexity*, 2003. to appear.

[25] A. C. Yao. Probabilistic computations: Towards a unified measure of complexity. In *Proc. 18th Annu. IEEE Symp. Found. Comp. Sci.*, pages 222–227, 1977.

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

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

A Proof of Lemma 3.5

We now provide a proof of the information theoretic lemma we used in Section 3. In the proof we use the following Substate Theorem of Jain, Radhakrishnan, and Sen [15].

Fact A.1 (Substate Theorem [15]) *Suppose P and Q are probability distributions on some set $\mathscr{A}$ such that $\mathrm{D}_{KL}(P \,\|\, Q) = a$. Let $r \geq 1$. Then, there exists a probability distribution on $\mathscr{A}$ such that $\|P - P'\|_1 \leq \frac{2}{r}$ and for all $i \in \mathscr{A}$, $\rho P'(i) \leq Q(i)$ where $\rho := (1 - \frac{1}{r})2^{-r(a+1)}$.*

Lemma A.2 (Lemma 3.5 restated) *Let X and M be correlated random variables with ranges $\mathscr{X}$ and $\mathscr{M}$ respectively, and let $a = \mathrm{I}(X : M)$. Then, for any function $f : \mathscr{X} \times \mathscr{M} \to [0, 1]$ and any sufficiently small $\delta > 0$, there is a subset $\mathscr{M}'$ of $\mathscr{M}$ and a function $g : \mathscr{X} \to \mathscr{M}'$ such that*
(i) $\log |\mathscr{M}'| \leq (1 + a)/\delta^2 + 1/\delta$, *and*
(ii) $\mathrm{E}_X[f(X, g(X))] \leq \mathrm{E}_{X,M}[f(X, M)] + 4\delta$.

Proof : Let Π denote the distribution of M and let Π_x denote the distribution of M conditioned on $X = x$. By Fact A.1, for any $x \in \mathscr{X}$ there exists a probability distribution Π'_x such that

$$\|\Pi_x - \Pi'_x\|_1 \quad \leq \quad 2\delta \tag{14}$$

and

$$\forall i \in \mathscr{M} : \quad \rho_x \Pi'_x(i) \;\leq\; \Pi(i)$$

where $\rho_x := (1-\delta)2^{-(1+\mathrm{D}_{KL}(\Pi_x \,\|\, \Pi))/\delta}$.

Consider the following (randomised) procedure h, which takes an input $x \in \mathscr{X}$ and a suffi ciently long string r which is supposed to be set to a uniform random value, and produces an output in $\mathscr{M}$.

```
Given x and r, repeat the following:
   Using r, choose an element i ∈ M according to Π
   With probability ρ_x Π'_x(i)/Π(i) output this i and stop
   Otherwise, continue
```

Let $h(r,x) \in \mathscr{M}$ denote the output of the procedure. Notice that $\rho_x \Pi'_x(i)/\Pi(i) \leq 1$, so that the procedure is well defi ned. At any particular iteration the procedure stops with probability

$$\sum_i \Pi(i) \cdot \rho_x \Pi'_x(i)/\Pi(i) \quad = \quad \rho_x \,.$$

Therefore, the probability of outputting $i \in \mathscr{M}$ is

$$\textstyle\sum_{k=1}^{\infty}(1-\rho_x)^{k-1} \cdot \Pi(i) \cdot \rho_x \Pi'_x(i)/\Pi(i) \quad = \quad \rho_x \Pi'_x(i) \cdot \sum_{k=1}^{\infty}(1-\rho_x)^{k-1} \quad = \quad \Pi'_x(i) \,.$$

In other words, if R denotes a uniform random variable uncorrelated with X, then the distribution of $h(R,x)$ is precisely Π'_x. Let $n(r,x)$ the number of iterations performed by h on input (r,x). For any $x \in \mathscr{X}$, $n(R,x)$ is a geometric random variable with expectation $\mathrm{E}_R[n(R,x)] = 1/\rho_x$. By the concavity of the log function,

$$\begin{aligned} \mathrm{E}_R[\log n(R,x)] \;\leq\; -\log \rho_x \quad &= \quad \log \frac{1}{1-\delta} + \big(1 + \mathrm{D}_{KL}(\Pi_x \,\|\, \Pi)\big)/\delta \\ &\leq \quad 1 + \big(1 + \mathrm{D}_{KL}(\Pi_x \,\|\, \Pi)\big)/\delta \,. \end{aligned}$$

Taking the expectation over X and noting that $\mathrm{E}_X[\mathrm{D}_{KL}(\Pi_X \,\|\, \Pi)] = \mathrm{I}(X:M) = a$,

$$\mathrm{E}_{X,R}[\log n(R,X)] \quad \leq \quad 1 + (1+a)/\delta \,,$$

whence, by Markov's inequality,

$$\Pr_{(X,R)}\big[\log n(R,X) \geq (1+a)/\delta^2 + 1/\delta\big] \quad \leq \quad \delta \,. \tag{15}$$

Now let h' be a modifi ed version of the procedure h that, after $2^{(1+a)/\delta^2 + 1/\delta}$ iterations, stops and outputs some arbitrary fi xed element of $\mathscr{M}$. Then, by (15), the probability (over X, R) that the output of h' differs from the output of h is at most δ. From this, it is not hard to show that if Π''_x denotes the distribution of $h'(R,x)$,

$$\sum_{x \in \mathscr{X}} \Pr[X = x] \cdot \|\Pi''_x - \Pi'_x\|_1 \quad \leq \quad 2\delta \,.$$

Combining the above with (14) we get $\sum_{x\in\mathscr{X}} \Pr[X=x]\cdot\|\Pi''_x-\Pi_x\|_1 \le 4\delta$, whence

$$\begin{aligned}
\mathrm{E}_R[\mathrm{E}_X[f(X,h'(R,X))]] &= \mathrm{E}_X[\mathrm{E}_R[f(X,h'(R,X))]] \\
&= \sum_{x\in\mathscr{X}} \Pr[X=x]\cdot \mathrm{E}_R[f(x,h'(R,x))] \\
&= \sum_{x\in\mathscr{X}} \Pr[X=x]\cdot \mathrm{E}_{i\sim\Pi''_x}\big[f(x,i)\big] \\
&\le 4\delta + \sum_{x\in\mathscr{X}} \Pr[X=x]\cdot \mathrm{E}_{i\sim\Pi_x}\big[f(x,i)\big] \qquad (16)\\
&= 4\delta + \mathrm{E}_{X,M}[f(X,M)]\,,
\end{aligned}$$

where (16) holds because f takes values in $[0,1]$. Therefore, there exists some fixed r_0 such that

$$\mathrm{E}_X[f(X,h'(r_0,X))] \le \mathrm{E}_{X,M}[f(X,M)] + 4\delta\,.$$

Let $g:\mathscr{X}\to\mathscr{M}$ be defined by $g(x)=h(r_0,x)$ for all $x\in\mathscr{X}$, and let $\mathscr{M}'$ be the range of g. Since the procedure h', by design, stops after $2^{(1+a)/\delta^2+1/\delta}$ iterations, $\log|\mathscr{M}'| \le (1+a)/\delta^2 + 1/\delta$. This completes the proof. ∎

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'