# Higher Lower Bounds for Near-Neighbor and Further Rich Problems

Mihai Pătraşcu
mip@mit.edu

Mikkel Thorup
mthorup@research.att.com

December 4, 2008

**Abstract**

We convert cell-probe lower bounds for polynomial space into stronger lower bounds for near-linear space. Our technique applies to any lower bound proved through the richness method. For example, it applies to partial match, and to near-neighbor problems, either for randomized exact search, or for deterministic approximate search (which are thought to exhibit the curse of dimensionality). These problems are motivated by search in large databases, so near-linear space is the most relevant regime.

Typically, richness has been used to imply $\Omega(d/\lg n)$ lower bounds for polynomial-space data structures, where $d$ is the number of bits of a query. This is the highest lower bound provable through the classic reduction to communication complexity. However, for space $n \lg^{O(1)} n$, we now obtain bounds of $\Omega(d/\lg d)$. This is a significant improvement for natural values of $d$, such as $\lg^{O(1)} n$. In the most important case of $d = \Theta(\lg n)$, we have the first superconstant lower bound. From a complexity theoretic perspective, our lower bounds are the highest known for *any* static data structure problem, significantly improving on previous records.

## 1 Introduction

Despite assiduous research, the best known bounds for many geometric problems explode exponentially in the dimension, a phenomenon known as the "curse of dimensionality". Establishing in which cases this curse is real is the holy grail of high-dimensional computational geometry.

Among data structure problems which seem to exhibit the curse, nearest neighbor search is perhaps the best known example, in no little part due to its central importance to many fields of computer science. To develop lower bounds, we consider a simplified decision version $\text{NN}_n^d$ defined as follows. We have $n$ points in the $d$-dimensional Hamming cube $\{0,1\}^d$, and a distance threshold $\lambda \leq d$. A *near* neighbor query asks whether a given point has any neighbor at distance at most $\lambda$. If we knew the nearest neighbor we could trivially answer the near neighbor query for any $\lambda$, so a lower bound for the near-neighbor problem implies a lower bound for the more common nearest-neighbor problem. Similarly, lower bounds for the Hamming space imply lower bounds for the $\ell_1$ or $\ell_2$ distances in $\mathbb{R}^d$.

The approximate near-neighbor problem $\text{ANN}_n^{\gamma,d}$ is defined similarly, except that the query must answer "yes" when there exists a neighbor within distance $\lambda$, "no" when the nearest neighbor is at least $\gamma\lambda$ away, and can answer anything otherwise. $\text{ANN}_n^{\gamma,d}$ is related to, and not harder than searching for a $\gamma$-approximate nearest neighbor.

If we allow both approximation and randomization, it is possible to avoid the curse of dimensionality, at least for constant approximation. If $\gamma = 1 + \varepsilon$, [8] and [10] provide data structures of size $n^{O(1/\varepsilon^2)}$ which can solve the problem with $O(1)$ cell probes. However, prohibiting either randomization or approximation seems to make the problem much harder. Despite extensive research, all known solutions have either space or query time growing exponentially in $d$ (see references in [4]).

Another famous problem which seems to exhibit a curse of dimensionality is the partial match problem. Given $n$ elements in $\{0,1\}^d$, and a query in $\{0,1,\star\}^d$, the goal is to find a matching element in the database,

where $\star$ matches anything. For a small number of $\star$ values, the problem can be solved efficiently [7], but the bounds grow exponentially with the number of wild cards.

**Models for lower bounds.** Upper bounds are typically designed in the Word RAM, a computational model meant to describe formally what is possible in a general-purpose programming language like `C`. The model allows random access to a memory organized in cells of $w$ bits, where $w = \Omega(\lg n)$ to permit constant-time manipulation of pointers and indices. The model also allows common constant-time operations such as addition, multiplication, and bitwise operations.

For lower bounds, the cell-probe model is preferred. This is a nonuniform model of computation, in which a querier (CPU) tries to answer a query by reading memory words (cells). The internal computation of the querier is not bounded, and the cost is only the number of memory accesses. Since any operation inside the CPU is allowed, the only assumption of the model is the CPU/memory distinction. For this reason, it is often claimed that lower bounds in the cell-probe model apply to all models of general-purpose computers deployed today.

For static data structures, most known lower bounds have been shown in an even stronger model: asymmetric communication complexity. In this model, Alice holds a query, and Bob holds a database. The two players communicate to answer the query on the database. To obtain a lower bound for data structures, one converts a cell-probe data structure into a communication protocol. Each round simulates a cell probe: the querier sends $\lg S$ bits, where $S$ is the space (the number of cells used by the data structure), and the database responds with $w$ bits, where $w$ is the cell size. This model is stronger than the cell-probe model because Bob can be adaptive (i.e., can remember Alice's previous messages), whereas a precomputed table sitting in memory is fixed *a priori*. The *asymmetry* in the model refers to the asymmetric message sizes: usually, $\lg S \ll w$.

**Previous work.** Motivated by the conjectured curse of dimensionality, there has been much recent work on cell-probe lower bounds for the problems discussed above. As mentioned already, these results actually prove lower bounds for asymmetric communication complexity.

Lower bounds in asymmetric communication complexity have been shown through variants of only two techniques: richness and round elimination. In general, richness is useful for "harder" problems, and can show significantly higher bounds than round elimination. Thus, it is more interesting when thinking about the curse of dimensionality. The richness method shows lower bounds of the following form: either Alice must send $a$ bits, or Bob must send $b$ bits. If the data structure makes $T$ cell probes to answer the query, in the communication protocol, Alice sends $T \lg S$ bits, and Bob sends $Tw$ bits. Comparing with the lower bounds, one concludes that the cell-probe complexity must be at least $T \geq \min\{\frac{a}{\lg S}, \frac{b}{w}\}$. In general, $b$ is prohibitively large, so the first bound dominates for reasonable word size.

Note that polynomial changes in the space only affect constant factors in the lower bound on $T$, as the bound only depends on $\lg S$. In the rest of the discussion, we assume $S = n^{O(1)}$ for simplicity.

Asymmetric communication complexity was formally introduced by Miltersen et al. [12] in STOC'95, who also described the richness and round elimination techniques. Using round elimination, they proved an $\Omega(\sqrt{\lg d})$ randomized cell-probe lower bound for partial match.

All subsequent lower bounds for the problems we are considering used the richness approach. In STOC'99, Borodin, Ostrovsky, and Rabani [4] showed that either the querier sends $a = \Omega(\lg d \cdot \lg n)$ bits, or the database sends $b = \Omega(n^{1-\varepsilon})$ bits, for any $\varepsilon > 0$. Furthermore, they observed a very simple reduction from partial match to near-neighbor search: simply map $\star$ to the value $\frac{1}{2}$ (which can be simulated in the Hamming cube by doubling $d$). The minimum distance is proportional to the number of stars in the query for a matched query, and strictly larger otherwise. Thus, both partial match and exact near-neighbor search have cell-probe complexity $\Omega(\lg d)$, for any $w = O(n^{1-\varepsilon}/\lg d)$.

In STOC'00, Barkol and Rabani [2] revisited randomized near-neighbor search and showed a lower bound with $a = \Omega(d)$ and $b = \Omega(n^{1/8-\varepsilon})$. Note that this is optimal with regard to the querier's communication. Furthermore, $b$ is still large enough that it is irrelevant for reasonable word size, giving a cell-probe complexity of $\Omega(d/\lg n)$.

This lower bound did not apply to partial match, but in STOC'03, Jayram et al. [9] analyzed partial match directly and proved an almost maximal communication bound with $a = \Omega(d/\lg n)$ and $b = \Omega(n^{1-\varepsilon})$. Very recently [13], this bound was improved to the optimal $a = \Omega(d)$ and $b = \Omega(n^{1-\varepsilon})$.

Finally, Liu [11] showed a tight communication lower bound for *deterministic* $O(1)$-approximate near neighbor search, giving $a = \Omega(d)$ and $b = \Omega(nd)$, hence a cell-probe lower bound of $\Omega(d/\lg n)$.

We note that a parallel thread of research has been investigating the randomized approximate *nearest* neighbor problem. Remember that when both approximation and randomization are allowed, the *near* neighbor problem can be solved with $O(1)$ cell probes. Approximate nearest neighbor can be reduced to approximate near neighbor with an $O(\lg\lg d)$ slow-down, by binary searching for an $i$ such that the nearest neighbor is at a distance between $(1+\varepsilon)^i$ and $(1+\varepsilon)^{i+1}$.

Building on work of Chakrabarti et al. [5] from STOC'99, Chakrabarti and Regev [6] in FOCS'04 showed a cell-probe lower bound bound of $\Omega(\lg\lg d/\lg\lg\lg d)$, using a variant of round elimination. Furthermore, they slightly improved the binary search, obtaining a matching upper bound of $O(\lg\lg d/\lg\lg\lg d)$.

**Our results.** As mentioned above, the lower bounds in the communication model are now optimal. However, the implications for data structures are far from satisfactory. For example, the entire strategy is, by design, insensitive to polynomial changes in the space (up to constants in the lower bound). However, the near-neighbor problems are motivated by searching in large databases. In this context, the difference between space $n\lg^{O(1)} n$ and (say) space $O(n^3)$ is plainly the difference between an interesting solution and an uninteresting one.

To put this in a different light, note that a communication complexity of $O(d)$ bits from the querier equates data structures of size $2^{O(d)}$ which solve the problem in constant time, and data structures of size $O(n)$ which solve the problem in a mere $O(d/\lg n)$ time. Needless to say, this equivalence appears unlikely. Thus, we need new approaches which can understand the time/space trade-offs in the cell-probe model at a finer granularity than direct reduction to communication. Our contribution makes progress in this direction, in the case when the space is $n^{1+o(1)}$.

Interestingly, we do not need to throw out the old work in the communication model. We can take any lower bound shown by the richness method, for problems with a certain compositional structure, and obtain a better lower bound for small-space data structures by *black-box* use of the old result. Thus, we can boost old bounds for polynomial space, in the case of near-linear space.

Let $S$ be the space in cells used by the data structure. If one uses richness to show a lower bound of $\Omega(d)$ bits for the communication of the querier, the standard approach would imply a cell-probe lower bound of $\Omega(d/\lg S)$. In contrast, we can show a lower bound of $\Omega(d/\lg \frac{Sd}{n})$, which is an improvement for $S = n^{1+o(1)}$. In the most interesting case of near-linear space $S = n(d\lg n)^{O(1)}$, the bound becomes $\Omega(d/\lg d)$. Compared to $\Omega(d/\lg n)$, this is a significant improvement for natural values of $d$, such as $d = \lg^{O(1)} n$. In particular, for $d = O(\lg n)$, previous lower bounds could not exceed a constant, whereas we obtain $\Omega(\lg n/\lg\lg n)$. Note that for $d = O(\lg n)$ we have constant upper bounds via tabulation, given large enough *polynomial* space.

Specifically, our paradigm gives the following results, based on previous richness analyses:

1. $\Omega(d/\lg \frac{Sd}{n})$ for randomized solutions to partial match, using [13].
2. $\Omega(d/\lg \frac{Sd}{n})$ for randomized, exact near-neighbor search $\mathrm{NN}_n^d$, by reduction from item 1., or by using [2].
3. $\Omega(\frac{d}{\gamma^3}/\lg \frac{Sd}{n})$ for deterministic, approximate near-neighbor search $\mathrm{ANN}_n^{\gamma,d}$, using [11].

Like previous lower bounds, our results hold in the Hamming cube, and extend by reductions to Euclidean spaces. It should be noted that all known lower bounds, including ours, are still very far from the "holy grail" of showing that the curse of dimensionality is inherent to these problems (for instance, showing that a query time of $O(n^{1-\varepsilon})$ requires space that is exponential in $d$).

The case $d = O(\lg n)$, where we give the first superconstant lower bound, is particularly interesting for the exact near neighbor. This is because approximate solutions essentially work by dimensionality reduction into $O(\frac{1}{\varepsilon^2}\lg n)$ dimensions, and applying an exact solution which is exponential in this dimension.

Thus, understanding exact search in $O(\lg n)$ dimensions is also conceptually interesting for the approximate problem.

**Technical contributions.** We first describe the intuition for why a lower bound of $\Omega(d/\lg n)$ for space $S = n^{O(1)}$, should also imply a lower bound of $\Omega(d/\lg d)$, when the space is $S = n \cdot (d \lg n)^{O(1)}$. For very small databases, namely $n = d^{O(1)}$, the lower bound for polynomial space can be rewritten as $\Omega(d/\lg d)$. If $n$ is larger, one can hope to partition the problem into $k = n/d^{O(1)}$ independent subproblems, each with database of size $N = d^{O(1)}$. Intuitively, each subproblem "gets" space $S/k = (d \cdot \lg n)^{O(1)} = N^{O(1)}$, and hence it requires $\Omega(d/\lg d)$ cell probes.

Transforming this intuition into an actual lower bound is surprisingly simple. Instead of simulating one query as part of a communication protocol, we will simulate $k$ queries in parallel. In each step, the queriers need to send the *subset* of $k$ cells which are probed, among the $S$ cells in memory. Sending this information requires $O(\lg \binom{S}{k}) = O(k \lg \frac{S}{k})$ bits. This is $O(\lg \frac{S}{k})$ bits "on average" per query, whereas the normal reduction sends $O(\lg S)$ bits for one query. We will typically use $k = n/\lg^{O(1)} n$.

The trouble with this approach is that one now needs to show communication lower bounds for a direct-sum problem, involving $k$ independent copies of the input. Rather than doing this on a per problem basis, we (essentially) show that the richness measure obeys a direct-sum law: considering $k$ independent copies increases the communication lower bound by a factor of $\Omega(k)$. Thus, any problem which could be analyzed by the standard reduction to communication and the richness method can also be analyzed in our improved framework.

Richness comes in two flavors: deterministic and randomized. In the deterministic case, we give a proper direct-sum result[1] for communication complexity. This is essentially at the level of an exercise. Section 2 contains a simple exposition of this result and our general framework.

For the randomized case (Section 3), things become slightly more technical. The heart of our result is a lemma about combinatorial rectangles, of a clear direct-sum flavor. However, we choose not to state the result in terms of communication games, because our setup is rather different from the common direct-sum setup in communication complexity.

**Relation to previous techniques.** Since it is known that direct-sum properties are not true for arbitrary functions, recent research has concentrated on proving such properties for specific lower bound measures. To our knowledge, none of these measures is relevant to asymmetric communication. In the symmetric case, corruption is the closest analog to our richness measure. A direct *product* result for corruption was recently shown by [3]. At a superficial level, the approach of our Lemma 9 resembles the approach of [3].

As mentioned already, the known techniques for showing lower bounds on asymmetric communication are richness and round elimination (with variants). In STOC'06, we [14] gave the only previous cell-probe lower bound which could exceed communication complexity. For linear space, our bound was roughly $\Omega(\lg d)$, which beats the $\Omega(d/\lg n)$ from communication complexity for small $d$. The approach of that paper can be seen as showing direct-sum results for round elimination, whereas here we show direct-sum results for richness.

Round elimination and richness are generally used for rather different sets of problems. Round elimination leads to a very precise understanding of a restricted class of problems related to predecessor search. (Note that the bounds of [14] are optimal for predecessor search, and richness does not yield any bound there.) On the other hand, richness applies to a much larger class of "very hard" problems, for which it shows maximal lower bounds in the communication model. In the cell-probe model, our $\Omega(d/\lg d)$ bounds for these "hard problems" are exponentially higher than the optimal bound for predecessor search.

[1] We have learned that Paul Beame and Matthew Cary independently observed this property.

# 2 Deterministic Lower Bounds

## 2.1 Data Structures and Communication

Consider a decision problem $f : X \times Y \to \{0,1\}$. When interpreting $f$ as a data structure problem, an input $y \in Y$ is given at preprocessing time, and the data structure must store a representation of it in space $S$. A query $x \in X$ is given at query time, and $f(x,y)$ must be computed through cell probes. For now, we restrict the preprocessing and query algorithms to be deterministic.

Consider a vector of problems $\mathbf{f} = (f_1, \ldots, f_k)$, where $f_i : X \times Y \to \{0,1\}$. We define another data structure problem $\bigoplus^k \mathbf{f} : ([k] \times X) \times Y^k \to \{0,1\}$ as follows. The data structure receives a vector of inputs $(y_1, \ldots, y_k) \in Y^k$. The representation depends arbitrarily on all of these inputs. The query is the index of a subproblem $i \in [k]$, and an element $x \in X$. The output of $\bigoplus^k \mathbf{f}$ is $f_i(x, y_i)$.

When interpreting a decision problem $f$ as a communication problem, Alice receives an input $x \in X$, Bob receives some $y \in Y$, and they must determine $f(x,y)$. For a vector of problems $\mathbf{f} = (f_1, \ldots, f_k)$, we also consider the communication problem $\bigwedge^k \mathbf{f} : X^k \times Y^k \to \{0,1\}$ defined by $\bigwedge^k \mathbf{f}(x,y) = \prod_i f_i(x_i, y_i)$. In other words, the output is the logical `and` of the $k$ component outputs.

The following describes our direct-sum reduction to communication complexity:

**Lemma 1.** *Assume $\bigoplus^k \mathbf{f}$ can be solved in the cell-probe model with $w$-bit cells by a data structure using space $S$, making $T$ cell probes. Then $\bigwedge^k \mathbf{f}$ has a communication protocol in which Alice sends $O(Tk \lg \frac{S}{k})$ bits and Bob sends $Tkw$ bits.*

*Proof.* Given input $(x_1, \ldots, x_k)$, Alice simulates the $k$ queries $(i, x_i)$ in parallel. In each of $T$ rounds, she sends the set of cells which are probed next by the $k$ queries. Sending the set requires $O(\lg \binom{S}{k}) = O(k \lg \frac{S}{k})$ bits. Bob replies with the contents of the cells, taking $kw$ bits. At the end of $T$ rounds, Alice has simulated all queries and knows their answers, so she can send their logical `and` using one more bit. □

## 2.2 Direct Sum for Richness

Consider the truth table of $f$, where rows are indexed by elements of $X$, and columns by elements of $Y$. The problem $f$ is called $[u,v]$-*rich* if at least $v$ columns of the truth table contain at least $u$ one entries.

Central to the analysis of communication protocols is the notion of (combinatorial) rectangles. A rectangle of a function $f$ is a submatrix of the truth table of $f$. When all entries of the submatrix are 1, this is called a 1-rectangle. It can be observed that the set of inputs leading to the same bits being communicated between the players is a rectangle; we call such rectangles *canonical rectangles*.

**Lemma 2** (the richness lemma [12])**.** *Let $f$ be a $[u,v]$-rich problem. If $f$ has a deterministic protocol in which Alice sends $a$ bits and Bob sends $b$ bits, then $f$ contains a canonical 1-rectangle of size at least $u/2^a \times v/2^{a+b}$.*

Thus, to prove a lower bound for a problem $f$, one shows that $f$ is $[u,v]$-rich, and that it does not contain any large 1-rectangle. For a vector of problems where each $f_i$ has these properties, we can show a lower bound for $\bigwedge^k \mathbf{f}$ via the following new direct-sum result:

**Theorem 3.** *Let $f_1, \ldots, f_k : X \times Y \to \{0,1\}$ be $[\rho|X|, v]$-rich, and assume $\bigwedge^k \mathbf{f}$ has a communication protocol in which Alice sends $k \cdot a$ bits and Bob sends $k \cdot b$ bits. Then some $f_i$ has a 1-rectangle of size $\rho^{O(1)}|X|/2^{O(a)} \times v/2^{O(a+b)}$.*

Note that this theorem is only interesting when the problem is very rich with respect to rows, i.e. $u$ is close to $|X|$. This is because the gap between $|X|$ and $u = \rho|X|$ is amplified polynomially. However, this is not an issue in real applications, since $\rho$ turns out to be large naturally. For example, if the function is balanced (say, it outputs 1 in a constant fraction of the cases), it is easy to show $\rho = \Omega(1)$.

The rest of this section is dedicated to proving the theorem. Without loss of generality, we can assume $|Y| = v$. Indeed, we can restrict the problem to only the columns which contain $\rho|X|$ ones. This maintains richness, and any deterministic protocol which works for the original domain also works for a subdomain.

**Claim 4.** *$\bigwedge^k \mathbf{f}$ is $[(\rho|X|)^k, v^k]$-rich.*

*Proof.* Since $\bigwedge^k \mathbf{f}$ only has $v^k$ columns, we want to show that *all* columns contain enough ones. Let $y \in Y^k$ be arbitrary. We have $\{x \in X^k \mid \bigwedge^k \mathbf{f}(x,y) = 1\} = \prod_i \{x' \in X \mid f_i(x', y_i) = 1\}$. But each set in the product has at least $\rho|X|$ elements by richness of $f_i$. □

Now we apply Lemma 2 to find a 1-rectangle of $\bigwedge^k \mathbf{f}$ of size $(\rho|X|)^k/2^{ak} \times v^k/2^{(a+b)k}$, which can be rewritten as $(\frac{\rho}{2^a}|X|)^k \times (\frac{1}{2^{a+b}}|Y|)^k$. Then, we complete the proof of the theorem by applying the following claim:

**Claim 5.** *If $\bigwedge^k \mathbf{f}$ contains a 1-rectangle of dimensions $(\alpha|X|)^k \times (\beta|Y|)^k$, then there exists $i \in [k]$ such that $f_i$ contains a 1-rectangle of dimensions $\alpha^3|X| \times \beta^3|Y|$.*

*Proof.* Let $\mathcal{X} \times \mathcal{Y}$ be the 1-rectangle of $\bigwedge^k \mathbf{f}$. Also let $\mathcal{X}_i$ and $\mathcal{Y}_i$ be the projections of $\mathcal{X}$ and $\mathcal{Y}$ on the $i$-th coordinate. Note that $(\forall)i$, $\mathcal{X}_i \times \mathcal{Y}_i$ is a 1-rectangle for $f_i$. Indeed, for any $(x', y') \in \mathcal{X}_i \times \mathcal{Y}_i$, there exists $(x, y) \in \mathcal{X} \times \mathcal{Y}$ with $x_i = x', y_i = y'$, hence $\bigwedge^k \mathbf{f}(x,y) = \prod_j f_j(x_j, y_j) = 1$, so $f_i(x', y') = 1$.

Now note that there must be at least $\frac{2}{3}k$ dimensions with $|\mathcal{X}_i| \geq \alpha^3|X|$. Otherwise, we would have $|\mathcal{X}| \leq \prod_i |\mathcal{X}_i| < (\alpha^3|X|)^{k/3} \cdot |X|^{2k/3} = (\alpha|X|)^k = |\mathcal{X}|$. Similarly, there must be at least $\frac{2}{3}k$ dimensions with $|\mathcal{Y}_i| \geq \beta^3|Y|$. Consequently, there must be at an overlap of dimensions, satisfying the statement of the lemma. □

This completes the proof of Theorem 3.

## 2.3 Application

Recall that $\mathrm{ANN}_n^{\gamma,d}$ is the $\gamma$-approximate near neighbor problem on $n$ points in $\{0,1\}^d$. We can view $\mathrm{ANN}_n^{\gamma,d}$ as a partial function, or alternatively as a family of (complete) functions that give the correct output whenever the partial function is defined. In this section, we slightly abuse notation and write $\mathrm{ANN}_n^{\gamma,d}$ for an arbitrary member of the family. Also, $\bigoplus^k \mathrm{ANN}_n^{\gamma,d}$ means $\bigoplus^k \mathbf{f}$ for an arbitrary vector $\mathbf{f} = (f_1, \ldots, f_k)$ such that $(\forall)i$, $f_i \in \mathrm{ANN}_n^{\gamma,d}$.

**Theorem 6.** *Consider a deterministic data structure solving $\mathrm{ANN}_n^{\gamma,d}$ in the cell-probe model with cells of $d^{O(1)}$ bits, which uses a space of $S$ cells. Assuming $d \geq (1+5\gamma)\lg n$, a query requires $\Omega(\frac{d}{\gamma^3} / \lg \frac{Sd}{n})$ cell probes in the worst case.*

*Proof.* Assume a solution for $\mathrm{ANN}_n^{\gamma,d}$ using $T$ cell probes. Let $D = d/(1+5\gamma) \geq \lg n$ and $k = n/N$ for $N < n$ to be chosen later. We now construct a solution for $\bigoplus^k \mathrm{ANN}_N^{\gamma,D}$ with the same complexity as the solution for $\mathrm{ANN}_n^{\gamma,d}$. To do that, consider a code on $5\gamma D$ bits, with minimum distance $\gamma D$. By the Gilbert-Varshamov bound, there exists such a code with $2^{(1-H(1/5)-0.01)5\gamma D} > 2^D \geq n$ codewords, for sufficiently large $D$. Since we work in a nonuniform model, a good code can be hardcoded in the algorithm.

Now we identify each of the $k \leq n$ subproblems by a unique codeword, and concatenate the $5\gamma D$ bits of the codeword to the $D$ bits of each point in the database corresponding to that subproblem. At query time, the codeword of the subproblem index is also concatenated to the query, and we search for a near neighbor to this extended query in the entire set of points. Our transformation guarantees that the near neighbor can only be in the subproblem where the query is intended to run, hence the structure of $\bigoplus^k \mathrm{ANN}_N^{\gamma,D}$ is respected. Indeed, the nearest neighbor of any query point is at most $D-1$ away from the query. Hence, a $\gamma$-approximate answer must be at distance strictly less than $\gamma D$. But any point in a different subproblem will be at distance at least $\gamma D$ due to the distance among codewords.

Now that we are working with $\bigoplus^k \mathrm{ANN}_N^{\gamma,D}$, we can apply our direct-sum framework. Remember that $\mathrm{ANN}_N^{\gamma,D}$ is the family of functions giving the correct output in the "yes" and "no" cases and any output when the promise fails. The richness lower bound of Liu [11] applies to *any* function in this family, and shows that for any $f \in \mathrm{ANN}_N^{\gamma,D}$:

- by [11, Claim 10], $f$ is $[2^{D-1}, 2^{ND}]$-rich.
- by [11, Claim 11], $f$ does not have a 1-rectangle of size $2^{D-D/(169\gamma^2)} \times 2^{ND-ND/(32\gamma^2)}$.

Remember that the query domain is $|X| = \{0,1\}^D$, so the problem is $[|X|/2, 2^{ND}]$-rich. Applying Theorem 3, we find that if $\bigwedge^k \mathbf{f}$ is solved by a protocol in which Alice sends $ka$ bits and Bob sends $kb$ bits, either $a = \Omega(D/\gamma^2)$, or $a + b = \Omega(ND/\gamma^2)$.

Now, by Lemma 1 we have a protocol in which Alice sends $k \cdot O(T \lg \frac{S}{k})$ bits, and Bob sends $k \cdot Tw$ bits. Thus, $T = \Omega(\min\{\frac{D}{\gamma^2}/\lg\frac{S}{k}, \frac{ND}{\gamma^2}/w\})$. Fixing $N = w = d^{O(1)}$, the first term is smallest. This means $T = \Omega(\frac{d}{\gamma^3}/\lg\frac{SN}{n}) = \Omega(\frac{d}{\gamma^3}/\lg\frac{Sd}{n})$. □

# 3 Randomized Lower Bounds

## 3.1 Obtaining Big Rectangles

Let us first describe how randomized richness is normally applied to communication games. We say problem $f$ is $\alpha$-*dense* if $\mathbf{E}_{x\in X, y\in Y}[f(x,y)] \geq \alpha$, i.e. at least an $\alpha$ fraction of the truth table of $f$ contains ones. Then, one applies the following lemma:

**Lemma 7** ([12])**.** *Let $\alpha, \varepsilon > 0$ be arbitrary constants. If $f$ is $\alpha$-dense and has a randomized protocol with error rate $\leq \frac{1}{3}$ in which Alice sends $a$ bits and Bob sends $b$ bits, there is a rectangle of $f$ of dimensions $|X|/2^{O(a)} \times |Y|/2^{O(a+b)}$ in which the density of zeros is at most $\varepsilon$.*

In this lemma, and below, the $O$-notation hides constants depending on $\alpha$ and $\varepsilon$.

Thus, to prove a communication lower bound, one shows $f$ is $\alpha$-dense, and every large rectangle contains $\Omega(1)$ zeros. Unfortunately, we cannot use this lemma directly because we do not know how to convert $k$ outputs, some of which may contain errors, into a single meaningful boolean output. Instead, we need a new lemma, which reuses ideas of the old Lemma 7, but in a more subtle way. A technical difference is that our new lemma will talk directly about data structures, instead of going through communication complexity.

Define $\rho_i : X \times Y \times \{0,1\} \to \{0,1\}$ by $\rho_i(x,y,z) = 1$ if $f_i(x,y) \neq z$, and 0 otherwise. Also let $\rho : X^k \times Y^k \times \{0,1\}^k \to [0,1]$ be $\rho(x,y,z) = \frac{1}{k}\sum_i \rho_i(x_i, y_i, z_i)$. In other words, $\rho$ measures the fraction of the outputs from $z$ which are wrong.

**Lemma 8.** *Let $\varepsilon > \frac{99}{k}$ be arbitrary, and $f_1, \ldots, f_k$ be $\varepsilon$-dense. Assume $\bigoplus^k \mathbf{f}$ can be solved in the cell-probe model with $w$-bit cells, using space $S$, cell-probe complexity $T$, and error rate $\leq \varepsilon$. Then there exists a canonical rectangle $\mathcal{X} \times \mathcal{Y} \subset X^k \times Y^k$ for some output $z \in \{0,1\}^k$ satisfying:*

$$|\mathcal{X}| \geq |X|^k/2^{O(Tk\lg\frac{S}{k})}, \qquad |\mathcal{Y}| \geq |Y|^k/2^{O(Tkw)}$$
$$\sum_i z_i \geq \frac{\varepsilon}{3}k, \qquad \mathbf{E}_{x\in\mathcal{X}, y\in\mathcal{Y}}[\rho(x,y,z)] \leq \varepsilon^2.$$

*Proof.* First we decrease the error probability of the data structure to $\frac{\varepsilon^2}{9}$. This requires $O(1)$ repetitions, so it only changes constant factors in $S$ and $T$. Now we use the easy direction of Yao's minimax principle to fix the coins of the data structure (nonuniformly) and maintain the same error over the uniform distribution on the inputs.

We now convert the data structure to a communication protocol as in Lemma 2. We simulate one query to each of the $k$ subproblems in parallel. In each round, Alice sends the subset of $k$ cells probed, and Bob replies with the contents of the cells. As in Lemma 1, Alice sends a total of $O(Tk\lg\frac{S}{k})$ bits, and Bob a total of $O(Tkw)$ bits. At the end, the protocol outputs the vector of $k$ answers.

Let $P_i(x_i, y)$ be the output of the data structure when running query $(i, x_i)$ on input $y$. Note that this may depend arbitrarily on the entire input $y$, but depends only on one query (since the query algorithm cannot consider parallel queries). When the communication protocol receives $x$ and $y$ as inputs, it

will output $P(x,y) = (P_1(x_1,y),\ldots,P_k(x_k,y))$. Note that some values $P_i(x_i,y)$ may be wrong (different from $f_i(x_i,y_i)$), hence some coordinates of $P(x,y)$ will contain erroneous answers. To quantify that, note $\mathbf{E}_{x,y}[\rho(x,y,P(x,y))] = \mathbf{E}_{i,x_i,y}[\rho_i(x_i,y_i,P_i(x_i,y))] \le \frac{\varepsilon^2}{9}$, i.e. the average fraction of wrong answers is precisely the error probability of the data structure.

We now wish to show that the set $W = \{(x,y) \mid \sum_i P_i(x_i,y) \ge \frac{\varepsilon}{3}k\}$ has density $\Omega(1)$ in $X^k \times Y^k$. First consider the set $W_1 = \{(x,y) \mid \sum_i f_i(x_i,y_i) \ge \frac{2\varepsilon}{3}k\}$. As $(x,y)$ is chosen uniformly from $X^k \times Y^k$, $f_i(x_i,y_i)$ are independent random variables with expectation $\ge \varepsilon$. Then, by the Chernoff bound, $\Pr_{x,y}[(x,y) \in W_1] \ge 1 - e^{k\varepsilon/18} \ge 1 - e^{-99/18} \ge \frac{2}{3}$. Now consider $W_2 = \{(x,y) \mid \rho(x,y,P(x,y)) \ge \frac{\varepsilon}{3}\}$. Since $\mathbf{E}_{x,y}[\rho(x,y,P(x,y))] = \frac{\varepsilon^2}{9}$, the Markov bound shows that the density of $W_2$ is at most $\frac{\varepsilon}{3}$. Finally, observe that $W_1 \setminus W_2 \subseteq W$, so $W$ has density $\ge \frac{1}{3}$.

The communication protocol breaks $X^k \times Y^k$ into disjoint canonical rectangles, over which $P(x,y)$ is constant. Consider all rectangles for which $P(x,y)$ has at least $\frac{\varepsilon}{3}k$ one entries. The union of these rectangles is $W$. Now eliminate all rectangles $R$ with $\mathbf{E}_{(x,y)\in R}[\rho(x,y,P(x,y))] \ge \varepsilon^2$, and let $W'$ be the union of the remaining ones. Since the average of $\rho(x,y,P(x,y))$ over $X^k \times Y^k$ is $\frac{\varepsilon^2}{9}$, a Markov bound shows the total density of the eliminated rectangles is at most $\frac{1}{9}$. Then, $|W'| \ge \frac{2}{3}|W|$.

Now observe that membership in $W'$ is $[\Omega(|X|^k), \Omega(|Y|^k)]$-rich. Indeed, since $|W'| = \Omega(|X|^k|Y|^k)$, a constant fraction of the rows must contain $\Omega(|Y|^k)$ elements from $W'$. Now note that the communication protocol can be used to decide membership in $W'$, so we apply Lemma 2. This shows that one of the rectangles reached at the end of the protocol must contain only elements of $W'$, and have size $\Omega(|X|^k)/2^{O(Tk\lg(S/k))} \times \Omega(|Y|^k)/2^{O(Tkw)}$. In fact, because Lemma 2 finds a large canonical rectangle, this must be one of the rectangles composing $W'$, so we know the answer corresponding to this rectangle has at least $\frac{\varepsilon}{3}k$ ones, and the average $\rho(x,y,P(x,y))$ over the rectangle is at most $\varepsilon^2$. □

The direct-sum result that we want will rely on the following key combinatorial lemma, whose proof is deferred to Section 3.2:

**Lemma 9.** *For $i \in [d]$, consider a family of functions $\phi_i : X \times Y \to \{0,1\}$, and define $\phi : X^d \times Y^d \to [0,1]$ by $\phi(x,y) = \frac{1}{d}\sum_i \phi_i(x_i,y_i)$. Let $\mathcal{X} \subset X^d, \mathcal{Y} \subset Y^d$ with $|\mathcal{X}| \ge (|X|/\alpha)^d, |\mathcal{Y}| \ge (|Y|/\beta)^d$, where $\alpha,\beta \ge 2$. Then there exists $i \in [d]$ and a rectangle $A \times B \subset X \times Y$ with $|A| \ge |X|/\alpha^{O(1)}, |B| \ge |Y|/\beta^{O(1)}$, such that $\mathbf{E}_{a\in A, b\in B}[\phi_i(a,b)] = O(\mathbf{E}_{x\in\mathcal{X}, y\in\mathcal{Y}}[\phi(x,y)])$.*

Using this technical result, we can show our main direct-sum property:

**Theorem 10.** *Let $\varepsilon > \frac{99}{k}$ be arbitrary, and $f_1,\ldots,f_k$ be $\varepsilon$-dense. Assume $\bigoplus^k \mathbf{f}$ can be solved in the cell-probe model with $w$-bit cells, using space $S$, cell-probe complexity $T$, and error $\varepsilon$. Then some $f_i$ has a rectangle of dimensions $|X|/2^{O(T\lg(S/k))} \times |Y|/2^{O(Tw)}$ in which the density of zeros is at most $\varepsilon$.*

*Proof.* First we apply Lemma 8, yielding a rectangle $\mathcal{X} \times \mathcal{Y}$. By reordering coordinates, assume the first $d = \frac{\varepsilon}{3}k$ elements of $z$ are ones. We now wish to fix $x_{d+1},\ldots,x_k$ and $y_{d+1},\ldots,y_k$ such that the remaining $d$-dimensional rectangle is still large, and the average of $\rho(x,y,z)$ over it is small. There are at most $|X|^{k-d}$ choices for fixing the $x$ elements. We can eliminate all choices which would reduce the rectangle by a factor of at least $3|X|^{k-d}$. In doing so, we have lost a $\frac{1}{3}$ fraction of the density. Similarly, we eliminate all choices for the $y$ elements which would reduce the rectangle by a factor of $3|Y|^{k-d}$.

We still have a third of the mass remaining, so the average of $\rho(x,y,z)$ can only have increased by a factor of 3. That means $\mathbf{E}_{i\in[k]}[\rho_i(x_i,y_i,z_i)] \le 3\varepsilon^2$, which implies $\mathbf{E}_{i\in[d]}[\rho_i(x_i,y_i,z_i)] \le 3\varepsilon^2 \cdot \frac{k}{d} = 9\varepsilon$. We now fix $x_{d+1},\ldots,x_k$ and $y_{d+1},\ldots,y_k$ among the remaining choices, such that this expected error is preserved. Thus, we have found a rectangle $\mathcal{X}' \times \mathcal{Y}' \subset X^d \times Y^d$ with $|\mathcal{X}'| \ge |X|^d/2^{O(Tk\lg(S/k))}$ and $|\mathcal{Y}'| \ge |Y|^d/2^{O(Tkw)}$. Since $d = \Theta(k)$, we can freely substitute $d$ for $k$ in these exponents. Besides largeness, the rectangle satisfies $\mathbf{E}_{i\in[d], x\in\mathcal{X}', y\in\mathcal{Y}'}[\rho_i(x_i,y_i,1)] \le 9\varepsilon$.

We now apply Lemma 9 on the rectangle $\mathcal{X}' \times \mathcal{Y}'$, with $\alpha = 2^{O(T\lg(S/k))}$, $\beta = 2^{O(Tw)}$ and $\phi_i(x,y) = \rho_i(x,y,1)$. We obtain a rectangle $A \times B \subset X \times Y$ of dimensions $|A| \ge |X|/2^{O(T\lg(S/k))}, |B| \ge |Y|/2^{O(Tw)}$, which has the property $\mathbf{E}_{a\in A, b\in B}[\rho_i(a,b,1)] = O(\varepsilon)$, i.e. $\Pr_{a\in A, b\in B}[f_i(a,b) = 0] = O(\varepsilon)$. □

## 3.2 Proof of Lemma 9

Define $\mathcal{X}_i$ to be the weighted projection of $\mathcal{X}$ on dimension $i$ (i.e. a distribution giving the frequency of every value on coordinate $i$). Thus, $\mathcal{X}_i$ is a distribution on $X$ with density function $\wp_{\mathcal{X}_i}(z) = \frac{|\{x \in \mathcal{X} | x_i = z\}|}{|\mathcal{X}|}$.

We identify sets like $\mathcal{X}$ and $\mathcal{Y}$ with the *uniform distributions* on the sets. Treating $\phi$ and $\phi_i$ as random variables (measuring some error to be minimized), let $\varepsilon = \mathbf{E}_{\mathcal{X}\times\mathcal{Y}}[\phi] = \frac{1}{d}\sum_i \mathbf{E}_{\mathcal{X}_i\times\mathcal{Y}_i}[\phi_i]$.

We now interpret the lower bound on the size of $\mathcal{X}$ as bounding the entropy, and use submodularity of the Shannon entropy $H$ to write:

$$\sum_i H(\mathcal{X}_i) \;\geq\; H(\mathcal{X}) \;\geq\; d \cdot (\lg|X| - \lg\alpha) \quad\Rightarrow\quad \frac{1}{d}\cdot\sum_i \big(\lg|X| - H(\mathcal{X}_i)\big) \;\leq\; \lg\alpha$$

Observe that each term in the sum is positive, since $H(\mathcal{X}_i) \leq \lg|X|$. We can conclude that:

$$(\exists) i: \qquad \lg|X| - H(\mathcal{X}_i) \;\leq\; 3\lg\alpha; \qquad \lg|Y| - H(\mathcal{Y}_i) \;\leq\; 3\lg\beta; \qquad \mathbf{E}_{\mathcal{X}_i\times\mathcal{Y}_i}[\phi_i] \leq 3\varepsilon,$$

because there are strictly less than $\frac{d}{3}$ coordinates that violate each of these three constraints. For the remainder of the proof, fix some $i$ satisfying these constraints.

Let $A'$ be the set of elements $z \in X$ with $\wp_{\mathcal{X}_i}(z) \leq \alpha^8/|X|$, where $\wp_{\mathcal{X}_i}$ is the density function of the distribution $\mathcal{X}_i$. In the probability space on which distribution $\mathcal{X}_i$ is observed, $A'$ is an event. We have:

$$\begin{aligned} H(\mathcal{X}_i) &= H(A') \;+\; \Pr[A']\cdot H(\mathcal{X}_i \mid A') \;+\; (1-\Pr[A'])\cdot H(\mathcal{X}_i \mid \neg A') \\ &\leq 1 \;+\; \Pr[A']\cdot\lg|X| \;+\; \big(1-\Pr[A']\big)\cdot\lg\frac{|X|}{\alpha^8} \;=\; \lg|X| + 1 - \big(1-\Pr[A']\big)\cdot 8\lg\alpha \end{aligned}$$

We claim that $\Pr[A'] \geq \frac{1}{2}$. Otherwise, we would have $H(\mathcal{X}_i) \leq \lg|X| + 1 - 4\lg\alpha$, contradicting the lower bound $H(\mathcal{X}_i) \geq \lg|X| - 3\lg\alpha$, given $\alpha \geq 2$.

Now let $\mathcal{X}'$ be the distribution $\mathcal{X}_i$ conditioned on $A'$ (equivalently, the distribution restricted to the support $A'$). Performing an analogous analysis on $\mathcal{Y}_i$, we define a support $B'$ and restricted distribution $\mathcal{Y}'$. Observe that:

$$\mathbf{E}_{\mathcal{X}'\times\mathcal{Y}'}[\phi_i] \;=\; \mathbf{E}_{\mathcal{X}_i\times\mathcal{Y}_i}[\phi_i \mid A'\wedge B'] \;\leq\; \frac{\mathbf{E}_{\mathcal{X}_i\times\mathcal{Y}_i}[\phi_i]}{\Pr[A'\wedge B']} \;\leq\; 4\cdot\mathbf{E}_{\mathcal{X}_i\times\mathcal{Y}_i}[\phi_i] \;\leq\; 12\varepsilon$$

We now want to conclude that $\mathbf{E}_{A'\times B'}[\phi_i]$ is small. This is not necessarily true, because changing from some distribution $\mathcal{X}'$ on support $A'$ to the uniform distribution on $A'$ may increase the average error. To fix this, we consider a subset $A \subseteq A'$, discarding from $A'$ every value $x$ with $\mathbf{E}_{\{x\}\times\mathcal{Y}'}[\phi_i] > 24\varepsilon$. Since the expectation over $x$ is $12\varepsilon$, a Markov bound implies that $\Pr[A] \geq \frac{1}{2}\Pr[A'] \geq \frac{1}{4}$. We now have a bound for every $x \in A$, and thus $\mathbf{E}_{A\times\mathcal{Y}'}[\phi_i] \leq 24\varepsilon$. Now perform a similar pruning of $B$, concluding that $\mathbf{E}_{A\times B}[\phi_i] \leq 48\varepsilon$.

Finally, we must show that $|A| \geq |X|/\alpha^{O(1)}$. This follows because $\Pr_{\mathcal{X}_i}[A] \geq \frac{1}{4}$, and for any $x \in A$ we had $\wp_{\mathcal{X}_i}(x) \leq \alpha^8/|X|$. The same analysis holds for $|B|$.

## 3.3 Applications

In this section, we discuss our applications to exact near neighbor and partial match. Recall that $\mathrm{NN}_n^d$ is the exact near neighbor problem on a database of $n$ points in $\{0,1\}^d$. Similarly, let $\mathrm{PM}_n^d$ be the partial match problem with a query in $\{0,1,\star\}^d$ and a database of $n$ strings in $\{0,1\}^d$.

**Theorem 11.** *Consider a bounded error (Monte Carlo) data structure solving* $\mathrm{PM}_n^d$ *in the cell-probe model with cells of* $d^{O(1)}$ *bits, using space* $S$*. Assuming* $d \geq 2\lg n$*, the cell-probe complexity of a query must be* $\Omega(d/\lg\frac{Sd}{n})$.

*Proof.* It is easy to convert a solution to $\mathrm{PM}_n^d$ into a solution to $\bigoplus^k \mathrm{PM}_N^D$, where $N = n/k$ and $D = d - \lg k \geq d/2$. One simply prefixes query and database strings with the subproblem number, taking $\lg k$ bits.

In [13], it is shown how a lower bound for the communication complexity of partial match can be obtained by a very simple reduction from a lower bound for lopsided set disjointness. A lower bound for set disjointness was described earlier [1], and this lower bound is by richness. Interpreting this richness lower bound in the context of partial match, we see that on a certain domain $X \times Y$ for $\mathrm{PM}_N^D$, we have:

- By [1, Lemma 4], $\mathrm{PM}_N^D$ is $\frac{1}{2}$-dense.
- By [1, Lemma 5] (more specifically, the variant claimed in the full version of that paper), for any $\delta > 0$, in any rectangle of size $|X|/2^{O(\delta D)} \times |Y|/2^{O(N^{1-\delta}/D^2)}$, the density of zeros is $\Omega(1)$.

For concreteness, set $\delta = \frac{1}{2}$ in the above result. Applying Theorem 10 to $\bigoplus^k \mathrm{PM}_N^D$, we obtain that either $T \lg \frac{S}{k} = \Omega(D)$, or $T'w = \Omega(\sqrt{N}/D^2)$. Setting $N = w^2 \cdot D^4 \cdot d = d^{O(1)}$, the second inequality becomes $T = \Omega(d)$, while the first becomes $T = \Omega(\lg \frac{Sd}{n})$. We thus conclude that $T \geq \min\{\Omega(d),\ \Omega(d/\lg \frac{Sd}{n})\} = \Omega(d/\lg \frac{Sd}{n})$. □

By the well-known reduction from partial match to exact near neighbor (outlined in the introduction), we immediately conclude that:

**Corollary 12.** *There exists a constant $C$ such that the following holds. Consider a bounded error (Monte Carlo) data structure solving $\mathrm{NN}_n^d$ in the cell-probe model with cells of $d^{O(1)}$ bits, using space $S$. Assuming $d \geq C \lg n$, the cell-probe complexity of a query must be $\Omega(d/\lg \frac{Sd}{n})$.*

This result can also be obtained without going through the partial match problem: use the earlier lower bound for $\mathrm{NN}_n^d$ by Barkol and Rabani [2], in the same manner we used a partial match lower bound above.

# 4 Conclusions

While randomized $(1+\varepsilon)$-approximate near neighbor does not suffer from the curse of dimensionality, all known solutions require space $n^{O(1/\varepsilon^2)}$, which significantly limits applications. Showing that small-space solutions do not exist for constant approximation is a very interesting problem. Recent independent work by [1] presents a richness lower bound for this problem, showing that the querier must send $\Omega(\frac{1}{\varepsilon^2} \lg n)$ bits. But this suffers from the inherent limitations of communication bounds, and cannot disprove that a linear-space data structure can solve the problem in constant time with constant approximation.

Though the bound of [1] is through richness, we obtain nothing interesting by combining it with our framework. This is not a coincidence, but reflects a deep fact about the structure of randomized ANN: dimensionality reduction can bring the problem down to logarithmic dimension, regardless of the original dimension. Thus, if we break the problem into subproblems, we gain nothing, because the dimension can be reduced corresponding to the size of the subproblem.

**Acknowledgements.** We are grateful for a suggestion by an anonymous referee, which simplified the proof of Lemma 9 considerably.

# References

[1] Alexandr Andoni, Piotr Indyk, and Mihai Pătraşcu. On the optimality of the dimensionality reduction method. In *Proc. 47th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 449–458, 2006.

[2] Omer Barkol and Yuval Rabani. Tighter lower bounds for nearest neighbor search and related problems in the cell probe model. *Journal of Computer and System Sciences*, 64(4):873–896, 2002. Announced at STOC'00.

[3] Paul Beame, Toniann Pitassi, Nathan Segerlind, and Avi Wigderson. A strong direct product theorem for corruption and the multiparty communication complexity of set disjointness. *Computational Complexity*, 15(4):391–432, 2006. Announced at CCC'05.

[4] Allan Borodin, Rafail Ostrovsky, and Yuval Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. In *Proc. 31st ACM Symposium on Theory of Computing (STOC)*, pages 312–321, 1999.

[5] Amit Chakrabarti, Bernard Chazelle, Benjamin Gum, and Alexey Lvov. A lower bound on the complexity of approximate nearest-neighbor searching on the hamming cube. In *Proc. 31st ACM Symposium on Theory of Computing (STOC)*, pages 305–311, 1999.

[6] Amit Chakrabarti and Oded Regev. An optimal randomised cell probe lower bound for approximate nearest neighbour searching. In *Proc. 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 473–482, 2004.

[7] Richard Cole, Lee-Ad Gottlieb, and Moshe Lewenstein. Dictionary matching and indexing with errors and don't cares. In *Proc. 36th ACM Symposium on Theory of Computing (STOC)*, pages 91–100, 2004.

[8] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. 30th ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

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

[10] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*, 30(2):457–474, 2000. Announced at STOC'98.

[11] Ding Liu. A strong lower bound for approximate nearest neighbor searching. *Information Processing Letters*, 92(1):23–29, 2004.

[12] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998. Announced at STOC'95.

[13] Mihai Pătraşcu. (Data) STRUCTURES. In *Proc. 49th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2008.

[14] Mihai Pătraşcu and Mikkel Thorup. Time-space trade-offs for predecessor search. In *Proc. 38th ACM Symposium on Theory of Computing (STOC)*, pages 232–240, 2006.