COMS E6998-8: Advanced Data Structures (Spring'19) March 7, 2019

Lecture 7: Cell-probe lower bounds for Nearest Neighbor Search

Instructor: *Omri Weinstein* Scribe: *Victor Lecomte*

1 Introduction

Last time, we considered the (c, r)-*approximate nearest neighbor* (ANN) problem in $\{0,1\}^d$:[1] the task of finding a point at distance $\leq cr$ from query point x, if we are guaranteed that the closest point is at distance $\leq r$. To solve it, we used *locality-sensitive hashing* (LSH) techniques: a hash with the property that close points will map to the same output with high probability. The particular LSH we used was to randomly sample a $\frac{k}{d} = \Theta(\frac{\log n}{cr})$ fraction of the coordinates, and put points into buckets according their coordinates in this reduced $\{0,1\}^k$ space. This gave us

$$s = O(n^{1+1/c}) \qquad t = O(n^{1/c} \cdot d), \tag{1}$$

while in homework 3 we will show that we can shift the space-time tradeoff to

$$s = n^{O(1/\epsilon^2)} \qquad t = O(1)$$

for $c = (1+\epsilon)$-ANN, a regime which would have required nearly linear query time with the previous result. The space we pay for this is very large, but still far off from the naive "precompute everything" space of 2^d.

Today, we will show that if we insist on constant query time, then the given space is tight. More formally:

Theorem 1. *Any data structure that solves $(1+\epsilon)$-ANN with $t = O(1)$ must have $s \geq n^{\Omega(1/\epsilon^2)}$ in the cell-probe model.*

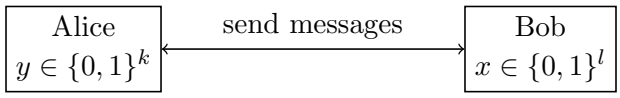
Note. The tightness of (1) is an open problem. For example, if we have $s = O(n^{1+1/c})$ it is not known whether we can get $n^{o(1)}$ query time. There are some conditional lower bounds, but nothing is known in the cell-probe model.

2 Asymmetric communication complexity

The tool we will use is *asymmetric communication complexity*. In this model, Alice is given an input $y \in \{0,1\}^k$, while Bob is given input $x \in \{0,1\}^l$, and they need to communicate with each

[1] Note that in $\{0,1\}^d$, it doesn't matter which norm we choose: since $0^p = 0$ and $1^p = 1$, all ℓ_p-norms are equivalent to ℓ_1.

other in order to compute the result of a function $f(x, y)$. What's asymmetric here is the amount of information each person gets: $k \ll l$, so Alice gets much less information than Bob.



Alice and Bob will interact by following some protocol Π, and try to compute f while minimizing the *number of bits exchanged*. That is, they are allowed to do arbitrarily hard computations (even undecidable if they wanted to) based on the information that they have, but they will be limited in terms of communication.

Definition 2. We write $\mathrm{ACC}(f) \leq (a, b)$ iff there exists some deterministic protocol Π computing f in which Alice sends $\leq a$ bits and Bob sends $\leq b$ bits. We call Π an (a, b)-*protocol*.

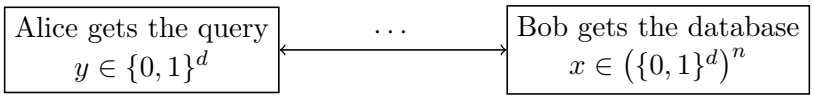
Note. In this lecture, for simplicity we will assume that everything Alice and Bob do is deterministic. But the results carry over to the probabilistic case.

This definition is a more fine-grained definition than the symmetric communication complexity definition we saw before. This will allow us to take into account the differing sizes between Alice's and Bob's inputs.

Definition 3. We write $\mathrm{ACC}(f) \geq (a, b)$ iff any deterministic protocol Π computing f requires either Alice to send $\geq a$ bits, or Bob to send $\geq b$ bits.

3 ACC formulation: first attempt

For any data structure problem P working on a database $x = (x_1, \ldots, x_n)$ and a query y, all in $\{0,1\}^d$, we can consider the following communication complexity problem:



This model intuitively encapsulates a situation in which the database is stored in some distant location at Bob's place, and Alice will only be allowed to query a limited number of locations in it in order to answer her query.

In our case, the function f to be computed will be $g_{(1+\epsilon)\text{-ANN}}$, the decision version of $(1+\epsilon)$-ANN. That is,

$$f = g_{(1+\epsilon)\text{-ANN}} \coloneqq \begin{cases} 1 \text{ if } \exists\, x_i \text{ such that } \|y - x_i\|_1 \leq r \\ 0 \text{ if } \forall\, x_i,\ \|y - x_i\|_1 \geq (1+\epsilon)r \\ \text{otherwise, any answer is valid} \end{cases}$$

The key observation is that any cell-probe data structure can be efficiently simulated in this setup.

Claim 4. *If there is a space s, time t, word-size w cell-probe data structure for $(1+\epsilon)$-ANN, then* $\mathrm{ACC}(g_{(1+\epsilon)\text{-}ANN}) \leq (t \lg s, tw)$.

Proof. Alice and Bob will play the game this way.

- First, Bob, builds the data structure independently. Once precomputation is done, the data structure has size s.
- Then Bob answers the t cell accesses that Alice needs for running the cell-probe query algorithm:
  - Alice sends some address in the data structure, using $\lg(s)$ bits;
  - Bob sends the contents of the cell, using w bits.

□

Corollary 5. *If we can show that* $\mathrm{ACC}(g_{(1+\epsilon)\text{-}ANN}) \geq (a, b)$*, then we will have proved that either*

$$t \lg s \geq a \Leftrightarrow s \geq 2^{a/t}$$
$$\text{or} \qquad tw \geq b \Leftrightarrow t \geq b/w.$$

Note that in this game, the best lower bound that we can every hope to prove would be (d, nd), i.e. either Alice or Bob has to send its full input. But the first option would only imply $s \geq 2^{d/t}$, so at best $s \geq 2^d = 2^{O(\lg n)} = n^{O(1)}$ for the $d = \Theta(\log n)$ regime. This means we have no chance of proving our desired lower bound of $s \geq n^{\Omega(1/\epsilon^2)}$. There are generally two options to make the problem harder:

1. Increase the number of dimensions d to $O\left(\frac{\lg n}{\epsilon^2}\right)$. This is what we'll do, and we will be able to use this d to obtain a reduction from a well-known hard ACC problem.
2. Instead of giving Alice only one query, we could give her k queries $(y_1, \ldots, y_k) \in (\{0,1\}^d)^k$. To see how this can help, let's imagine we're trying to get space $s = O(n)$, and looking at lower bounds for t. At first, giving k queries instead of 1 seems to multiply both Alice's input size and required communication a by a factor k, yielding no improvement: this would again lead to
$$kt \lg s \geq kd \Leftrightarrow t \lg s \geq d,$$
which gives $t \geq \Omega(\lg n / \lg s) = \Omega(1)$ (i.e. no lower bound) for the $d = \Theta(\log n)$ regime.

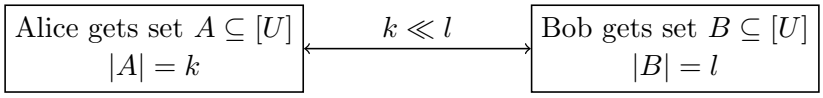
   But if Alice is given k queries, she is able to more efficiently encode her questions to Bob by executing them all in parallel. Indeed, assuming queries require $r = O(1)$ cell probes in the worst case, then she can perform them in r rounds. In the i^{th} round she will request the i^{th} memory cell required in the execution of each of the k queries. There are only $\binom{s}{k}$ sets of k addresses, so she could get away with sending only $O\left(\lg \binom{s}{k}\right) = O(k \lg(s/k))$ bits, a subtle but important improvement. If the optimal lower bound of $a = kd$ were obtained, we would then get
$$kt \lg(s/k) \geq kd \Leftrightarrow t \geq d / \lg(s/k),$$

which for $k = \Theta(n)$ (the biggest possible value of k that doesn't necessarily lead Bob to transmit his whole input) would yield

$$t \geq \Omega(\log n \,/\, \log(s/n)) = \Omega(\log n).$$

4 The Lopsided Set Disjoint problem

In the Lopsided Set Disjoint problem LSD(k, l), both Alice and Bob are given a subset of a large space $[U]$, and they have to decide whether their intersection is empty or not.



Goal: if $A \cap B = \emptyset$, output 1, otherwise output 0.

This problem has the following hardness result:

Theorem 6. *For all n, all $k = O(1)$ and all $\delta > 0$, we have* $\text{ACC}(\text{LSD}(k, n)) \geq \left(\delta k \lg n,\, n^{1-2\delta}\right)$.

Note. The deterministic version of this theorem is very easy to prove (and we will briefly show how). On the other hand, the randomized version is highly nontrivial, and took 10 years to prove!

Note that this result is very close to the maximal theoretically possible lower bound of $(k \lg n, n)$. Also, note that since k is small, we can interpret this problem as giving Alice k queries of the form "is this element in Bob's set" to answer. So we are essentially using option 2 from the previous section to make this problem harder.

Roadmap Assume we have an (s, t, w)-data structure for $(1+\epsilon)$-ANN. Then we will set $\delta = 0.1$ in the above theorem to obtain

$$\left(0.1k \log n,\, n^{0.8}\right) \overset{\text{Thm 6}}{\leq} \text{ACC}(\text{LSD}(k, n)) \overset{?}{\leq} \text{ACC}(g_{(1+\epsilon)\text{-ANN}}) \overset{\text{Claim 4}}{\leq} (t \lg s, tw).$$

If we manage to prove that middle reduction with $k = 1/\epsilon^2$, then we're done: we would have

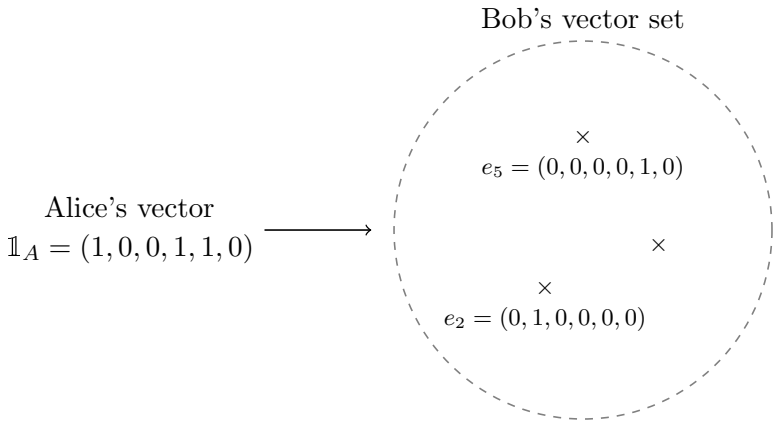
$$t \lg s \geq \Omega\left(\frac{\lg n}{\epsilon^2}\right) \Leftrightarrow s \geq n^{\Omega(1/\epsilon^2 t)} = n^{\Omega(1/\epsilon^2)}$$

since $t = O(1)$. The alternative is $tw \geq n^{0.8}$, which is of course impossible.

5 Reducing LSD to $(1+\epsilon)$-ANN

Let's attempt a first reduction from LSD $\left(k = \frac{1}{\epsilon}, n\right)$ to $g_{(1+\epsilon)\text{-ANN}}$ as a warmup. Note that this would only prove an $s \geq n^{\Omega(1/\epsilon)}$ lower bound.

In this version, we set $U = 2n$. So Alice gets set $A \subseteq [2n]$, $|A| = \frac{1}{\epsilon}$, and Bob gets set $B \subseteq [2n]$, $|B| = n$. Bob will interpret each element $i_j \in B$ of its set as a unit vector $e_{i_j} \in \{0,1\}^{2n}$, while Alice will interpret her $(1/\epsilon)$-size set as indicator vector $\mathbb{1}_A \in \{0,1\}^{2n}$ of A, which has a 1 for each position in A, and a 0 everywhere else.



Alice will try to determine the vector e_{i_j} in Bob's set that is closest to her own vector $\mathbb{1}_A$. There are two possible cases:

- if $A \cap B = \emptyset$, then for every j, $\|\mathbb{1}_A - e_{i_j}\| = \frac{1}{\epsilon} + 1$;
- otherwise, there is some j such that $\|\mathbb{1}_A - e_{i_j}\| = \frac{1}{\epsilon} - 1$.

So this LSD problem reduces to the decision version of (c, r)-ANN by setting $r = \frac{1}{\epsilon} - 1$ and $c = \frac{1/\epsilon+1}{1/\epsilon-1} = 1 + \Theta(\epsilon)$, which is what we were hoping to show (up to constants).

But there is a catch: here we used dimension $d = 2n$, which is too big to be realistic or useful. In general, we assume that $w \geq d$ in order to be able to store a point in a single memory word. So here even Bob's large input would hold in a single memory word, which makes no sense. We will fix the issue by using *metric embeddings*. We first define them informally:

Definition 7. A *distortion-D metric embedding* f for $D = 1 + \epsilon$ is a mapping between two metric spaces (X, d_1) and (Y, d_2) such that for all $a, b \in X$, we have $d_1(a, b) \in (1 \pm \epsilon) d_2(f(a), f(b))$.

Theorem 8 (Johnson-Lindenstrauss)**.** *For any polynomial $p(n)$, there is a randomized embedding*

$$f : (\{0,1\}^d, \ell_1) \to \left(\{0,1\}^{O\left(\frac{\log n}{\epsilon^2}\right)}, \ell_1\right)$$

which achieves $(1+\epsilon)$-distortion on a set of n points with probability at least $1 - \frac{1}{p(n)}$.

This means that if Alice and Bob have access to shared randomness for the setup of f, all they need to do is apply f to their $2n$-dimensional vectors to get them down to $d = O\left(\frac{\log n}{\epsilon^2}\right)$ dimensions, with only a small $\Theta(\epsilon)$ distortion. They are then able to perform the rest of the protocol unchanged, which completes the reduction from LSD $\left(\frac{1}{\epsilon}, n\right)$ to $g_{(1+\epsilon)\text{-ANN}}$.

6 Strengthening the reduction

But the previous reduction only gives us a lower bound of $s \geq n^{\Omega(1/\epsilon)}$, which falls somewhat short of our goal of $n^{\Omega(1/\epsilon^2)}$. To attain a stronger reduction, we will need to start with a more powerful metric space: $(\mathbb{R}^{2n}, \ell_2)$. This means we will take a more sinuous path: instead of using a single metric embedding to go directly to our goal metric space as before

$$A, B \subseteq U \to (\{0,1\}^{2n}, \ell_1) \to \left(\{0,1\}^{O\left(\frac{\log n}{\epsilon^2}\right)}, \ell_1\right),$$

we will use three metric embedding to perform the following successive steps

$$A, B \subseteq U \to (\mathbb{R}^{2n}, \ell_2) \to (\mathbb{R}^{D}, \ell_1) \to (\{0,1\}^{D'}, \ell_1) \to \left(\{0,1\}^{O\left(\frac{\log n}{\epsilon^2}\right)}, \ell_1\right)$$

for potentially huge dimensions D and D'.

But let's first have a look at how we can embed A and B into $\mathbb{R}^{2n}$ while preserving a decent gap. This time, Alice has to handle a bigger set $|A| = 1/\epsilon^2$, while Bob remains at $|B| = n$. We will encode A as $\epsilon \cdot \mathbb{1}_A \in \mathbb{R}^{2n}$ instead of $\mathbb{1}_A$, and B will be represented as before as the set of unit vectors e_{i_j} for all $i_j \in B$. Again, there are two cases:

- if $A \cap B = \emptyset$, then for every j, $\|\mathbb{1}_A - e_{i_j}\| = \sqrt{\frac{1}{\epsilon^2} \cdot \epsilon^2 + 1} = \sqrt{2}$;
- otherwise, there is some j such that $\|\mathbb{1}_A - e_{i_j}\| = \sqrt{\left(\frac{1}{\epsilon^2} - 1\right) \cdot \epsilon^2 + (1-\epsilon)^2} \leq (1 - \epsilon/2)\sqrt{2}$.

So we still managed to get a $1 + \Theta(\epsilon)$ ratio even though the problem has become harder.

All that remains to show is that we can obtain the desired embeddings.

Theorem 9 (Dvoretzky's theorem, informally)**.** *Any normed space in $\mathbb{R}^{N(k,\epsilon)}$ has a subspace of dimension k on which the Euclidean distances and the ℓ_p distances are equivalent up to a $1 \pm \epsilon$ factor, for $N(k, \epsilon) \geq \exp(k/\epsilon^2)$.*

Note. The original proof by Dvoretzky in 1961 was very complicated and did not have the above dependence on k and ϵ. Later, Milman and Alon gave a simpler proof based on random projections and Chernoff bounds that gave the improved dependence on k, ϵ. Their argument essentially shows that if a large space is projected on a small subspace, then everything looks Euclidean, in the same way that the geometry of $\{0,1\}^D$ can be well-approximated by a ball for large D by the law of large numbers.

This theorem directly implies that for any d, we can find an embedding with $1 \pm \epsilon$ distortion from $(\mathbb{R}^{2n}, \ell_2)$ to $(\mathbb{R}^D, \ell_1)$, with $D = N(n, \epsilon)$ potentially depending exponentially on n and ϵ. The last remaining step, going from $(\mathbb{R}^D, \ell_1)$ to $(\{0,1\}^{D'}, \ell_1)$, was not covered during the lecture.

7 Proof sketch for LSD lower bound

Recall we want to prove that $\mathrm{ACC}(\mathrm{LSD}(k,n)) \geq (\delta k \lg n, n^{1-2\delta})$ for all $\delta > 0$ (theorem 6). We will demonstrate the general approach one can use to show this.

Let's consider the communication matrix M_f for this problem, that is, the matrix indexed by all possible inputs for Alice and Bob, and whose elements are the corresponding outputs of f in $\{0,1\}$. In other words, we define each cell of M_f as

$$(M_f)_{xy} \coloneqq f(x,y).$$

We say that a communication problem is (u,v)-rich if M_f has at least v columns that have at least u 1's in them. For example, the problem corresponding to the matrix below is $(2,3)$-rich because three of the columns have at least two 1's.

$$\begin{array}{c} \\ x \end{array} \begin{array}{c} \begin{array}{cccc} \multicolumn{4}{c}{y} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

Lemma 10 (Richness lemma)**.** *If problem f is (u,v)-rich and $\mathrm{ACC}(f) \leq (a,b)$, then there is a rectangle $R = A \times B \subseteq M_f$ (A, B need not be contiguous) such that all elements in $A \times B$ are 1, and*

$$|A| \geq \frac{u}{2^a}, \qquad |B| \geq \frac{v}{2^{a+b}}.$$

Proof. We proceed by induction on the length of protocol Π. The general idea is that after Alice and Bob have exchanged a few bits, the problem is entirely determined by the submatrix of M_f whose rows and columns are still consistent with the bits that were sent by Alice and Bob. By "adversarially" choosing this submatrix after each exchanged bit, we will show that the (u,v)-richness is at least partially preserved by the end of the protocol. More precisely:

- If Bob sends a bit first, that partitions the columns of M_f into two groups C_0, C_1 where C_0 is the set of inputs on which Bob would send 0, and C_1 is the set of inputs on which Bob would send 1. Then the rest of the protocol depends only on the submatrix of M_f restricted to either C_0 or C_1. But if M_f has v columns with at least u 1's, then one of those halves must contain at least $v/2$ of those columns, so one of the two resulting submatrices must be $(u, v/2)$-rich.

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{array}\right)$$

- Similarly, if Alice sends a bit first, that splits the rows of M_f into two groups R_0, R_1 depending on the bit that Alice sends for each input. Let's say one of those halves R_0, R_1 "wins" a column if it contains at least half of the 1's that that column contains. If M_f has v columns with at least u 1's, then clearly, at least one of R_0, R_1 must win at least half of the v columns,

so at least one of the resulting submatrices must have at least $v/2$ columns with at least $u/2$ 1's. Thus one of them must be $(u/2, v/2)$-rich.

$$\left(\begin{array}{cccc} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 1 & 1 \end{array}\right)$$

Since Alice sends at most a bits and Bob sends at most b bits, it is clear that by the end of the protocol, one of the remaining submatrices will be $(\frac{u}{2^a}, \frac{v}{2^{a+b}})$-rich. But since the protocol has ended, to ensure correctness, this submatrix must contain either only 0's or only 1's. Assuming that $u \geq 2^a$, $v \geq 2^{a+b}$ (otherwise the lemma holds vacuously), we deduce that it contain only 1's, and has dimensions at least $\frac{u}{2^a} \times \frac{v}{2^{a+b}}$. □

The last step in proving theorem 6, which consists in showing that the communication matrix M_f cannot have large 1-monochromatic rectangles for the LSD problem, is left as an exercise to the reader. :)