On the Optimality of the Dimensionality Reduction Method*

Alexandr Andoni
MIT
andoni@mit.edu

Piotr Indyk
MIT
indyk@mit.edu

Mihai Pătraşcu
MIT
mip@mit.edu

Abstract

We investigate the optimality of $(1+\epsilon)$*-approximation algorithms obtained via the dimensionality reduction method. We show that:*

- *Any data structure for the* $(1+\epsilon)$*-approximate* nearest neighbor *problem in Hamming space, which uses constant number of probes to answer each query, must use* $n^{\Omega(1/\epsilon^2)}$ *space.*
- *Any algorithm for the* $(1+\epsilon)$*-approximate* closest substring *problem must run in time exponential in* $1/\epsilon^{2-\gamma}$ *for any* $\gamma > 0$ *(unless 3SAT can be solved in sub-exponential time)*

Both lower bounds are (essentially) tight.

1. Introduction

Dimensionality reduction is a powerful method for designing efficient approximation algorithms for problems of a geometric nature. Its main idea is simple: to solve a problem defined over a high-dimensional geometric space $\Re^d$, map that space onto $\Re^k$ where k is "low", and solve the problem in the latter space. A prototypical tool used for such purpose is the theorem by Johnson and Lindenstrauss [24], which states that there exists a randomized mapping $A : \Re^d \to \Re^k$, $k = O(\log(1/P)/\epsilon^2)$, such for any $x \in \Re^d$ we have $\Pr_A[\|Ax\|_2 = (1 \pm \epsilon)\|x\|_2] \geq 1 - P$. This theorem is often instantiated with $P = 1/n^{O(1)}$ where n is the size of the input data set in $\Re^d$. In that case we have $k = O(\log n/\epsilon^2)$, which can be much smaller than the original dimension.

The above theorem (or its variants[1]) has led to numerous algorithmic results for several algorithmic domains:

- DATA STREAM COMPUTATION: A low-space algorithm for maintaining the second frequency moment of the data stream was shown in [4]. The algorithm provides a $(1+\epsilon)$-approximation to the estimated quantity while using only $O(1/\epsilon^2)$ memory words. The problem has numerous applications, see [31].
- DATA STRUCTURES: Several data structures for the $(1+\epsilon)$-approximate nearest neighbor problem in $\Re^d$ were given in [21, 26, 17, 11, 2]. In particular, the data structure of [26] uses $n^{O(1/\epsilon^2)}$ space and guarantees $(d + \log n + 1/\epsilon)^{O(1)}$ query time.
- APPROXIMATION ALGORITHMS: Several $n^{1/\epsilon^{O(1)}}$-time $(1+\epsilon)$-approximate algorithms for various clustering and pattern analysis problems were given, e.g., in [32, 27, 23].

These algorithmic developments raise the natural question: how close to optimal are the space/time bounds derived using the dimensionality reduction method ? In addition to the theoretical importance of this question, resolving it is of significant practical interest. The difference between (say) $1/\epsilon^{0.5}$ and $1/\epsilon^2$ could easily mean the difference between a practical algorithm and an impractical one. This is especially the case[2] if the expression appears in the exponent of the time/space bound.

What is known about this issue ? It is known [3] that some n-point data sets cannot be mapped into a space of dimension $k = o\left(\frac{\log n/\epsilon^2}{\log(1/\epsilon)}\right)$ without distorting the distances by a factor larger than $(1+\epsilon)$; for the l_1 norm, a polynomial lower bound for the dimension k is known [7] for any constant ϵ. However, this does not shed much light on the issue of optimality of bounds for *concrete applications* of that lemma. On the latter front, we are aware of only two results:

- The aforementioned second frequency moment problem does require $\Omega(1/\epsilon^2)$ space [22, 35]. Thus, the

*This work was supported in part by NSF CAREER award CCR-0133849, David and Lucille Packard Fellowship and Alfred P. Sloan Fellowship.

[1] This is arguably a broad statement, since the dimensionality reduction techniques used in those papers are quite diverse. Nevertheless, all of them can be replaced by Johnson-Lindenstrauss lemma (e.g., see [18] and the Appendix), which in our opinion justifies this point of view.

[2] Even if the bound is polynomial in $1/\epsilon$, the degree of the polynomial is of key practical importance. E.g., see [13] for the discussion of this issue in the context of streaming algorithms.

dimensionality reduction approach yields the optimal bound for this problem.

- It has been observed in [19] that the techniques of [32] yield a $(1+\epsilon)$-approximation algorithm for the k-*center problem* in $\Re^d$ under the Euclidean metric, with the running time exponential in k/ϵ^2. However, it was later showed that the same problem can be solved in time $dnk^{O(k/\epsilon)}$ [8]. Thus, here the dimensionality reduction gives a suboptimal bound.

In this paper we consider two problems which belong to the two algorithmic domains (data structures and approximation algorithms) where the optimality of the dimensionality reduction method is not yet well-understood. Specifically, we focus on the Approximate Near Neighbor (NN) problem, and the Approximate Closest Substring (CSS) problem. For both problems, we show lower bounds indicating that the dimensionality reduction approach yields algorithms whose time/space bounds have essentially optimal (i.e., nearly quadratic) dependence on $1/\epsilon$.

Approximate Near Neighbor. We consider a decision version [3] of the approximate nearest neighbor problem over the Hamming space. Given a set $P \subset \{0,1\}^d$ of n points and a distance λ, build a data structure which given $q \in \{0,1\}^d$ does the following, with probability at least, say, $2/3$:

- If there is $p \in P$ such that $\|q-p\| \leq \lambda$, answer YES
- If there is no $p \in P$ such that $\|q-p\| \leq (1+\epsilon)\lambda$, answer NO

Here we use $\|\cdot\|$ for the Hamming norm. It is standard to assume cells have $\Theta(d)$ bits, i.e. a point can be stored in one cell. The lower bound holds for the Euclidean space as well.

This problem is closely related to the approximate *nearest* neighbor problem. In fact, the aforementioned paper [26] provides an algorithm for the $(1+\epsilon)$-NN problem, using $n^{O(1/\epsilon^2)}$ space. Moreover, the algorithm has constant query time (measured by the number of probes to the data structure).

In this paper, we complement that result by showing that any data structure for $(1+\epsilon)$-NN which achieves constant query time must use $n^{\Omega(1/\epsilon^2)}$ space[4]. To prove the lower bound, we consider the *asymmetric communication complexity* of the problem for dimension $d = (\frac{1}{\epsilon}\lg n)^{O(1)}$. That is, we consider the setting where two parties, Alice and Bob, are communicating in order to answer a query q. We assume that Alice holds q, while Bob holds P. We show that to solve the problem, either Alice sends $\Omega(\frac{1}{\epsilon^2}\lg n)$ bits, or Bob sends $\Omega(n^{1-\delta})$ bits, for any constant $\delta > 0$. By the standard relation to cell-probe complexity [29], this implies that the lower bound on space. Therefore, the aforementioned algorithms are space-optimal. Our result is obtained by showing a close relationship between the complexity of the $(1+\epsilon)$-NN problem and the complexity of *set disjointness*. Lower bounds for the latter problem appeared in [29] for the case of randomized protocols with one-sided error. We give an analogous lower bound for the two-sided error case, solving an open problem posed in that paper.

There has been a considerable number of results on lower bounds for the near and nearest neighbor problems (e.g. see [6, 10, 5, 11, 34] or [20] for a survey). Most apply to more restrictive (i.e, harder) versions of the problem, where either randomization or approximation are disallowed. For randomized approximation algorithms for the *nearest* neighbor problem, a tight query time bound of $\Theta(\lg\lg d/\lg\lg\lg d)$ is known [11], for any constant ϵ and polynomial space.

In contrast to that work, our result holds for the approximate *near* neighbor problem, and establishes a *quantitative* dependence between the approximation factor and the exponent in the space bound (for the constant query time case). Given that the exponent must be quadratic in $1/\epsilon$, our results indicate a fundamental difficulty in designing practical data structures which are very accurate *and* very fast.

Our space lower bound also holds for a closely related $(1+\epsilon)$-*far* neighbor problem (defined formally in section 2.3).

Approximate Closest Substring. The Closest Substring problem is a fundamental pattern analysis problem in computational biology. Assume we are given a set of n strings $s_1 \ldots s_n$ of length L over some alphabet Σ (in this paper we focus on the case $\Sigma = \{0,1\}$). Let $D(\cdot,\cdot)$ be the Hamming metric. The goal is to find $s \in \Sigma^d$ which is "close" to some substring of each input string s_i. More formally, the goal is to minimize

$$C(s) = \max_{i=1\ldots n} \min_{s'=s_i[j\ldots j+d-1]} D(s,s')$$

The closest substring problem is a combinatorial formalization of the task of finding motifs in DNA sequences, which is of major interest in molecular biology (see [33, 25] for background and references). The problem is NP-hard, but its $(1+\epsilon)$-approximate version can be solved in polynomial time. Specifically [27] provided an algorithm for this problem with running time (roughly) $(nL)^{O(1/\epsilon^4)}$. In the

[3] The definition of the approximate near neighbor problem employed here is somewhat weaker than the usual one. Specifically, it does not require the algorithm to provide a "near" point in the YES case. However, this definition is more suitable for the reductions used in this paper. Clearly, the lower bound for this version holds for stronger versions as well.

[4] In this extended abstract we present a proof of a slightly weaker lower bound of $n^{\Omega((1/\epsilon^2)/\log(1/\epsilon))}$. The proof of the optimal bound is deferred to the full version of this paper.

appendix we show that, by combining known techniques, one can reduce the exponent to $1/\epsilon^2 \cdot \log(1/\epsilon)$.

In this paper we show a result indicating that any $(1+\epsilon)$-approximate algorithm for the closest substring problem must have running time that is exponential[5] in $1/\epsilon^{2-\gamma}$, for any $\gamma > 0$. The result is based on a strong assumption about hardness of the 3-SAT problem. That is, we assume that 3-SAT for formulas with n variables and $O(n)$ constraints cannot be solved in time $2^{O(n^b)}$ for any fixed constant $b < 1$. Our hardness result is then obtained by showing the following: if, for *any*[6] $\epsilon = \epsilon(n) > 0$ there is a $(1+\epsilon)$-approximate algorithm for CSS with running time polynomial in d, n and $2^{1/\epsilon^{2a}}$ for a fixed constant $a < 1$, then there exists an algorithm for 3-SAT with running time $2^{O(n^b)}$ for a fixed constant $b < 1$.

We note that earlier work [28, 12] investigated the issue of optimality of approximation schemes for substring problems developed in [27] and their followups. However, they approached this problem from the fixed-parameter tractability point of view. That is, they asked if one can obtain a running time of the form $f(\epsilon)n^{O(1)}$ for some function $f(\cdot)$. In comparison, the premise of our paper is that, in the exponent of running time bound, $1/\epsilon$ can create as much trouble as $\log n$; therefore, we focus on the dependence on $1/\epsilon$.

Our lower bound for the closest substring problem is shown using the following approach. Firstly, we use the short PCP construction of Dinur [15] to transform a given 3SAT formula ϕ into another formula ϕ' of comparable size, such that ϕ' is either satisfiable, or no assignment can satisfy more than $1-\alpha < 1$ fraction of ϕ's clauses, for a fixed constant $\alpha > 0$.

In the next step, we transform the formula ϕ' into an instance of the hitting set (HS) problem, with similar gap properties. Finally, we reduce the hitting set problem to the CSS problem. Although somewhat involved, the reduction is intuitively natural, since the goal of the CSS problem is to find a "center" string that "hits" (i.e., is "close" to a substring of) each input string. The main part of the reduction is establishing a relation between cardinality (of the hitting set) and proximity (i.e., the distance between the "center" string and the "close" substrings of the input strings). This task is accomplished by a variant of a theorem known in the list-decoding literature as the Johnson bound (see [16]). That theorem establishes an upper bound on the number of codewords (of a "good" error-correcting code) in a ball of certain radius. However, that upper bound is a constant factor away from the lower bound that we can establish for the (randomly constructed) error-correcting code that we use; for our reduction, we need that factor to be arbitrarily close to 1. We resolve this issue by establishing a tighter Johnson-type bound for random error-correcting codes.

Other implications. The techniques introduced in this paper have applications beyond the ones mentioned so far. Specifically, consider the following *Minimum Enclosing Ball (MEB)* problem (with respect to the l_p norm): given $Q \subset R^d$, minimize $R = R_p(Q)$ so that there exists $x \in \Re^d$ such that $\|x - q\|_p \leq R$ for all $q \in Q$. A *weak ϵ-coreset* [9] for an MEB instance is a subset $S \subset Q$ such that $R_p(S) \geq (1-\epsilon)R_p(Q)$. Weak coresets have numerous applications for clustering of high-dimensional data (see [1] for a survey); often, the coreset size appears in the exponent of the running time bound.

It it known [8] that, in the l_2 norm, any set contains a weak ϵ-coreset of size $1/\epsilon$. In this paper we show that, in the l_1 norm, a weak ϵ-coreset must have size $\Omega(1/\epsilon^2)$.

2. Lower bounds for the approximate near neighbor

For proving the lower bound, we analyze the asymmetric communication complexity of $(1+\epsilon)$-NN via a reduction from the set disjointness problem. In the set disjointness problem, Alice receives a set S from a universe $[U] = \{1 \dots U\}$, $|S| = m$, and Bob receives a set $T \subset [U]$ of size n. They need to decide whether $T \cap S = \varnothing$. We prove the following asymmetric communication complexity lower bound for the latter problem.

Theorem 1. *Assume Alice receives a set $S, |S| = m$ and Bob receives a set $T, |T| = n$, both sets coming from a universe of size $2mn$, for $m < n^\gamma$, where $\gamma < 1$ is a constant. In any randomized, two-sided error communication protocol deciding disjointness of S and T, either Alice sends $\Omega(m \lg n)$ bits or Bob sends $\Omega(n^{1-\delta})$ bits, for any $\delta > 0$.*

The proof of this theorem is deferred to the full version. In Section 2.2, we present the proof of a slightly weaker version of this theorem, which implies a dependence of $\frac{1/\epsilon^2}{\log(1/\epsilon)}$ instead of $1/\epsilon^2$.

From the reduction in Section 2.1 and the above theorem for $m = \frac{1}{9\epsilon^2}$, we derive the following theorem on asymmetric communication complexity of the $(1+\epsilon)$-NN problem:

Theorem 2. *Consider the communication complexity version of $(1+\epsilon)$-NN in $\{0,1\}^d$, $d = O(\frac{\log^2 n}{\epsilon^5})$, where Alice receives the query $q \in \{0,1\}^d$ and Bob receives the set $P \subset \{0,1\}^d$. Then, for any $\epsilon = \Omega(n^{-\gamma})$, $\gamma < 1/2$, in any randomized protocol deciding the $(1+\epsilon)$-NN problem, either Alice sends $\Omega(\frac{\log n}{\epsilon^2})$ bits or Bob sends $\Omega(n^{1-\delta})$ bits, for any $\delta > 0$.*

[5] Actually, we can show that the exponent is at least $1/\epsilon^2/\log^{O(1)}(1/\epsilon)$, assuming even stronger hardness of 3-SAT. We'll ignore it for now.

[6] Our reduction uses values of ϵ that are subconstant in n (that is, $\epsilon \rightarrow 0$ as $n \rightarrow \infty$).

From the above theorem, we can obtain the $n^{O(1/\epsilon^2)}$ lower bound on space for any data structure implementing $(1+\epsilon)$-NN problem with a constant time query. Specifically, we apply Lemma 1 from [29], which states:

Lemma 1 ([29], Lemma 1)**.** *If there is a solution to the data structure problem with space s, query time t, and cell size b, then there exists a protocol where Alice sends $2t\lceil\log s\rceil$ bits and Bob sends $2tb$ bits.*

For $t = O(1)$, and cell size $b < O(n^{1-\delta})$, for some $\delta > 0$, Bob sends an insufficient number of bits. Thus, Alice needs to send $2t\lceil\log s\rceil > \Omega(m \log n)$ bits. Solving for s, we obtain that space is $s = n^{\Omega(1/\epsilon^2)}$. Note that the cell size b is usually much smaller than $n^{1-\delta}$, typically $b = d\log^{O(1)} n$.

2.1. Reduction from asymmetric set disjointness to $(1+\epsilon)$-near neighbor

We prove that we can reduce asymmetric set disjointness problem to the approximate near neighbor. A randomized $[a, b]$-protocol for a communication problem is a protocol in which Alice sends a bits and Bob sends b bits, and the error probability of the protocol is bounded away from $1/2$.

Lemma 2. *Suppose there exists a randomized $[a, b]$-protocol for the $(1+\epsilon)$-NN problem with $d = O\left(\frac{\log^2 n}{\epsilon^5}\right)$, where Alice receives the query $q \in \{0,1\}^d$ and Bob receives the dataset $P \subset \{0,1\}^d$ of size n. Then there exists a randomized $[a, b]$-protocol for asymmetric set disjointness in an arbitrary universe $[U]$, where Alice receives a set $S \subset [U]$ of size $m = \frac{1}{9\epsilon^2}$, and Bob receives a set $T \subset U$ of size n.*

Proof. We show how to map an instance of asymmetric set disjointness, given by T and S, into an instance of $(1+\epsilon)$-NN, given by respectively the dataset $P \subset \{0,1\}^d$ and the query $q \in \{0,1\}^d$. For this purpose, first, Alice and Bob map their sets S and T into query $\tilde{q} \in \Re^U$ and dataset $\tilde{P} \subset \Re^U$, i.e., an $(1+\epsilon)$-NN instance in Euclidean U-dimensional space, l_2^U. Then, Alice and Bob map their points from the l_2^U metric to Hamming cube $\{0,1\}^{O(\log^2 n/\epsilon^5)}$, essentially preserving the distances among all the points $\tilde{q}$ and $\tilde{P}$. This method for reducing a communication complexity problem into an approximate problem involving Hamming distance appeared earlier in [22], albeit in the context of different problems.

For the set $T \subset [U]$, we define $\tilde{P} \triangleq \{e_u \mid u \in T\}$, where e_u is a standard $\Re^d$ basis vector, with 1 in the u^{th} coordinate, and 0 everywhere else. For the set S, we set the query $\tilde{q} \triangleq 3\epsilon \cdot \sum_{u\in S} e_u$; note that $\|\tilde{q}\|_2^2 = m \cdot (3\epsilon)^2 = 1$.

We show that if $S \cap T = \varnothing$, then $\|\tilde{q} - \tilde{p}\|_2 = \sqrt{2}$ for all $\tilde{p} \in \tilde{P}$, and, if $S \cap T \neq \varnothing$, then there exists a point $\tilde{p} \in \tilde{P}$ such that $\|\tilde{q} - \tilde{p}\|_2 \leq (1 - \frac{4\epsilon}{3})\sqrt{2}$. Indeed, we have that

- if $S \cap T = \varnothing$, then for any $\tilde{p} \in \tilde{P}$, we have that $\|\tilde{q} - \tilde{p}\|_2^2 = \|\tilde{q}\|_2^2 + \|\tilde{p}\|_2^2 - 2\tilde{q} \cdot \tilde{p} = 2$;
- if $S \cap P \neq \varnothing$, then for $u^* \in S \cap P$ and for $\tilde{p} = e_{u^*} \in P$, we have $\|\tilde{q} - \tilde{p}\|_2^2 = \|\tilde{q}\|_2^2 + \|\tilde{p}\|_2^2 - 2\tilde{q} \cdot \tilde{p} = 2 - 2(3\epsilon e_{u^*}) \cdot e_{u^*} = 2(1 - 3\epsilon)$.

To construct $P \subset \{0,1\}^d$ and $q \in \{0,1\}^d$, Alice and Bob perform a randomized mapping of l_2^U into $\{0,1\}^d$ for $d = O(\log^2 n/\epsilon^5)$, such that the distances are only insignificantly distorted, with high probability. Alice and Bob use a source of public random coins to construct the same randomized mapping. First, they construct a randomized embedding f_1 mapping l_2^U into $l_1^{O(\log n/\epsilon^2)}$ with distortion less than $(1 + \epsilon/16)$ (cf. [19]). Then, they construct the standard embedding f_2 mapping $l_1^{O(\log n/\epsilon^2)}$ into $\{0,1\}^{O(\log^2 n/\epsilon^5)}$. The embedding f_2 first scales up all coordinates by $D = O(\frac{\log n}{\epsilon^3})$, then rounds the coordinates, and finally transforms each coordinate into its unary representation. We set the constants such that the resulting approximation of f_2 is an additive term $O(\frac{\log n}{\epsilon^2}) < \frac{D\epsilon\sqrt{2}}{16}$.

Next, Alice and Bob construct $q = f_2(f_1(\tilde{q})) \in \{0,1\}^d$ and $P = \{f_2(f_1(\tilde{p})) \mid \tilde{p} \in \tilde{P}\} \subset \{0,1\}^d$. Notice that for any $p = f_2(f_1(\tilde{p})) \in P$, if $\|\tilde{q} - \tilde{p}\|_2 \geq \sqrt{2}$, then $\|q - p\|_H \geq D\sqrt{2}(1 - \epsilon/16) - \frac{D\epsilon\sqrt{2}}{16} = D\sqrt{2}(1 - \frac{\epsilon}{8})$, and if $\|\tilde{q} - \tilde{p}\|_2 \leq \sqrt{2}(1 - \frac{4\epsilon}{3})$, then $\|q - p\|_H \leq D\sqrt{2}(1 - \frac{4\epsilon}{3})(1 + \epsilon/16) + \frac{D\epsilon\sqrt{2}}{16} \leq D\sqrt{2}(1 - \epsilon - \frac{5\epsilon}{24})$.

Finally, Alice and Bob can run the $(1+\epsilon)$-NN communication protocol with $\lambda = D\sqrt{2}(1 - \epsilon - \frac{5\epsilon}{24})$ to decide whether $S \cap T = \varnothing$. Note that the error probability of the resulting set disjointness protocol is bounded away from $1/2$ since $(1+\epsilon)$-NN communication protocol has error probability bounded away from $1/2$, and the embedding $f_2 \circ f_1$ fails with probability at most $n^{-\Omega(1)}$. □

2.2. Lower bound for asymmetric set disjointness

In this section, we prove a slightly weaker version of Theorem 1:

Theorem 3. *Assume Alice receives a set $S, |S| = m$ and Bob receives a set $T, |T| = n$, both sets coming from a universe of size $2mn$, for $m < n^\gamma$, where $\gamma < 1/3$ is a constant. In any randomized, two-sided error communication protocol deciding disjointness of S and T, either Alice sends $\Omega(\frac{m}{\log m} \lg n)$ bits or Bob sends $\Omega(n^{1-\delta}/m^2)$ bits, for any $\delta > 0$.*

First we define the hard instance. The elements of our sets come from the universe $[2m] \times [n]$. Alice receives $S = \{(i, s_i) \mid i \in [m]\}$, for $s_1, \ldots, s_m$ chosen independently at random from $[n]$. Bob receives $T = \{(t_j, j) \mid j \in [n]\}$, for $t_1, \ldots, t_n$ chosen independently from $[2m]$. The output should be 1 iff the sets are disjoint. Note that the number of choices is n^m for S and $(2m)^n$ for T, and that S and T are chosen independently.

The lower bound follows from the following variant of the richness lemma, based on [29, Lemma 6]. The only change is that we make the dependence on ϵ explicit, because we will use $\epsilon = o(1)$.

Lemma 3. *Consider a problem $f : X \times Y \rightarrow \{0,1\}$, such that the density of $\{(x,y) \mid f(x,y) = 1\}$ in $X \times Y$ is $\Omega(1)$. If f has a randomized two-sided error $[a,b]$-protocol, then there is a rectangle of f of dimensions at least $|X|/2^{O(a \lg(1/\epsilon))} \times |Y|/2^{O((a+b)\lg(1/\epsilon))}$ in which the density of zeros is at most ϵ.*

To apply the lemma, we first show the disjointness function is 1 with constant probability.

Lemma 4. *As S and T are chosen randomly as described above,* $\Pr[S \cap T = \varnothing] = \Omega(1)$.

Proof. Note that $S \cap T \subset [n] \times [m]$. We have $\Pr[(i,j) \in S \cap T] = \frac{1}{n(2m)}$ when $i \in [n], j \in [m]$. Then by linearity of expectation $\mathbf{E}[|S \cap T|] = \frac{1}{2}$. Since $|S \cap T| \in \{0,1,2,\dots\}$, we must have $\Pr[|S \cap T| = 0] \geq \frac{1}{2}$. □

Thus, it remains to show that no big enough rectangle has a small density of zeros. Specifically, we show the following:

Lemma 5. *Let $\delta > 0$ be arbitrary. If we choose $S \in \mathcal{S}, T \in \mathcal{T}$ uniformly and independently at random, where $|\mathcal{S}| > 2n^{(1-\delta)m}$ and $\mathcal{T} \geq (2m)^n \cdot 2/e^{n^{1-\delta}/(8m^2)}$, then the probability $S \cap T \neq \varnothing$ is at least $\frac{1}{16m^2}$.*

We use the richness lemma with $\epsilon = \frac{1}{32m^2}$. If there exists an $[a,b]$ protocol for our problem, we can find a rectangle of size $\left(n^m/2^{O(a \lg m)}\right) \times \left((2m)^n/2^{O((a+b)\lg m)}\right)$, in which the fraction of zeros is at most ϵ. To avoid contradicting Lemma 5, we must either have $2^{O(a \lg m)} > n^{\delta m}/2$, or $2^{O((a+b)\lg m)} > e^{n^{1-\delta}/(8m^2)}/2$. This means either $a = \Omega(\frac{m}{\lg m} \lg n)$ or $a + b = \Omega(n^{1-\delta}/(m^2 \lg m))$. If $m < n^\gamma$, for constant $\gamma < \frac{1}{3}$, this implies that $a = \Omega(\frac{m}{\lg m} \lg n)$ or $b = \Omega(n^{1-\delta}/m^2)$, for any $\delta > 0$.

Proof. (of Lemma 5) Choosing S at random from $\mathcal{S}$ induces a marginal distribution on $[n]$. Now consider the heaviest $n^{1-\delta}$ elements in this distribution. If the total probability mass of these elements is at most $1 - \frac{1}{2m}$, we call i a *well-spread coordinate*.

Lemma 6. *If $|\mathcal{S}| > 2n^{(1-\delta)m}$, there exists a well-spread coordinate.*

Proof. Assume for contradiction that no coordinate is well-spread. Consider the set $\mathcal{S}'$ formed by $S \in \mathcal{S}$ such that no s_i is outside the heaviest $n^{1-\delta}$ elements in S_i. By a union bound, the probability over $S \in \mathcal{S}$ that some s_i is not among the heavy elements is at most $m\frac{1}{2m} = \frac{1}{2}$. Then, $|\mathcal{S}'| \geq |\mathcal{S}|/2$. On the other hand $|\mathcal{S}'| \leq (n^{1-\delta})^m$, since for each coordinate we have at most $n^{1-\delta}$ choices. This contradicts the lower bound on $|\mathcal{S}|$. □

Let i be a well-spread coordinate. We now lower bound the probability of $S \cap T \neq \varnothing$ by the probability of $S \cap T$ containing an element on coordinate i. Furthermore, we ignore the $n^{1-\delta}$ heaviest elements of S_i. Let the remaining elements be W, and $p(j) = \Pr[s_i = j]$ when $j \in W$. Note that $p(j) \leq 1/n^{1-\delta}$, and $\sum_{j \in W} p(j) \geq \frac{1}{2m}$.

Define $\sigma(T) = \sum_{j \in W : t_j = i} p(j)$. For some choice of T, $\sigma(T)$ gives exactly the probability of an interesting intersection, over the choice of $S \in \mathcal{S}$. Thus, we want to lower bound $\mathbf{E}_T[\sigma(T) \mid T \in \mathcal{T}]$.

Assume for now that T is uniformly distributed in the original space (not in the subspace $\mathcal{T}$). Note that $\sigma(T) = \sum_{j \in W} X_j$, where X_j is a variable equal to $p(j)$ when $t_j = i$ and 0 otherwise. By linearity of expectation, $\mathbf{E}_T[\sigma(T)] = \sum_{j \in W} \frac{p(j)}{2m} \geq 1/(2m)^2$. Since X_j's are independent (t_j's are independent when T is not restricted), we can use a Chernoff bound to deduce $\sigma(T)$ is close to this expectation with very high probability over the choice of T. Indeed, $\Pr[\sigma(T) < \frac{1}{2} \cdot \frac{1}{(2m)^2}] < e^{-n^{1-\delta}/(8m^2)}$.

Now we can restrict ourselves to $T \in \mathcal{T}$. The probability $\sigma(T) < \frac{1}{8m^2}$ is so small, that it remains small even in this restricted subspace. Specifically, this probability is at most $\Pr[\sigma(T) < \frac{1}{8m^2}]/\Pr[T \in \mathcal{T}] \leq \exp(-n^{1-\delta}/(8m^2))/(2\exp(-n^{1-\delta}/(8m^2))) = \frac{1}{2}$. Since $\sigma(T) \geq 0, (\forall)T$, we conclude that $\mathbf{E}_T[\sigma(T) \mid T \in \mathcal{T}] \geq \frac{1}{2} \cdot \frac{1}{8m^2} = \frac{1}{16m^2}$. □

2.3. Approximate far neighbor problem

The above lower bound for the $(1+\epsilon)$-NN problem can also be transferred to the $(1+\epsilon)$-*far* neighbor problem, yielding exactly the same space lower bound. Formally, we define the $(1+\epsilon)$-far neighbor as follows. Given a set $P \subset \{0,1\}^d$ of n points and a distance λ, build a data structure which given $q \in \{0,1\}^d$ does the following, with probability at least, say, 2/3:

- If there is $p \in P$ such that $\|q - p\| \geq \lambda$, answer YES
- If there is no $p \in P$ such that $\|q - p\| \geq \lambda/(1+\epsilon)$, answer NO

The lower bound results from the following lemma, an equivalent of lemma 2.

Lemma 7. *Suppose there exists a randomized $[a,b]$-protocol for the $(1+\epsilon)$-far neighbor problem with $d =$*

$O\left(\frac{\log^2 n}{\epsilon^5}\right)$, where Alice receives the query $q \in \{0,1\}^d$ and Bob receives the dataset $P \subset \{0,1\}^d$ of size n. Then there exists a randomized $[a,b]$-protocol for asymmetric set disjointness in an arbitrary universe $[U]$, where Alice receives a set $S \subset [U]$ of size $m = \frac{1}{9\epsilon^2}$, and Bob receives a set $T \subset U$ of size n.

As before, together with theorem 1, this lemma implies that any data structure for $(1+\epsilon)$-far neighbor problem achieving constant number of cell probes, has space $n^{\Omega(1/\epsilon^2)}$.

Proof (of lemma 7). Same as the proof of lemma 2, except set the query $\tilde{q} = -3\epsilon \sum_{u \in S} e_u$. □

3. Lower bounds: Approximate closest substring problem

In this section we focus on the *Approximate Closest Substring* problem.

Our goal is to show a lower bound of $2^{1/\epsilon^{2a}}$, for any $a \in (0,1)$, for a running time of an algorithm solving $(1+\epsilon)$-approximate CSS. We do it by using the following assumption. Recall that in a *Hitting Set (HS)* problem, we are given sets $A_1 \dots A_n \subset [m]$, and the goal is to find $H \subset [m]$ which intersects each A_i and minimizes $|H|$. Our assumptions are stated as the following two conjectures.

Conjecture 1. *For any constant $a \in (0,1)$, there exists a constant $C_a > 1$ such that no C_a-approximate algorithm for the hitting set problem has running time $2^{O(m^a)}$.*

In the following, we show that this conjecture is implied by another (more palatable) one.

Conjecture 2. *For any $a \in (0,1)$, there is no algorithm solving 3-SAT with $O(n)$ constraints over n variables, with running time $2^{O(n^a)}$.*

Theorem 4. *Conjecture 2 implies Conjecture 1. That is, if there exists $a < 1$ such that for every constant $C_a > 1$ there exists a C_a-approximation algorithm for HS with running time $2^{O(m^a)}$, then there exists an algorithm for 3SAT over n variables with $O(n)$ constraints, with running time $2^{O(n^b)}$ for some $b < 1$.*

The proof will follow from the following two reductions. The first one follows from the PCP construction by [15]. Specifically, for any 3SAT formula ϕ, let $SAT(\phi)$ be the maximum fraction of clauses satisfiable by any assignment. Dinur [15] proved the following:

Fact 1. *There is a polynomial-time algorithm which, given a 3SAT formula ϕ with m variables and $O(m)$ constraints, outputs a 3SAT formula ϕ' with $m' = m \log^{O(1)} m$ variables such that each variable occurs in exactly the same (and constant) number of constraints, and:*

- *If ϕ satisfiable then ϕ' satisfiable.*
- *If ϕ is not satisfiable, then $SAT(\phi') \le 1 - \alpha$.*

where $\alpha > 0$ is an absolute constant.

The relation between the hitting set problem and the 3SAT problem is captured in the following lemma.

Lemma 8. *There is a polynomial-time algorithm which, given a 3 SAT formula ϕ with n variables, where each variable occurs in exactly the same number B'' constraints, produces an instance of HS with $m = 2n$ such that:*

1. *If ϕ is satisfiable, then there is a hitting set of size n.*
2. *If there is a hitting set of size $(1+\gamma)n$, for $\gamma > 0$, then $SAT(\phi) \ge (1 - 3\gamma)$.*

Proof. The reduction is as follows. The universe of HS consists of all literals $x_i, \overline{x_i}$, where x_i is a variable. The family of sets contains all pairs $\{x_i, \overline{x_i}\}$ (called *literal* sets), and all constraints of ϕ interpreted as sets (called *constraint* sets).

The first statement of the lemma is immediate. Consider now a set H of size $(1+\gamma)n$ which hits all sets. There are $(1-\gamma)n$ literal sets which are hit once - this defines the assignment of the corresponding variable. For the γn literal sets which are hit twice, define the assignment in an arbitrary way. For each such variable we "unsatisfy" at most B'' constraints, thus a total of $\gamma B'' n$ constraints are unsatisfied. Since ϕ contains exactly $B''/3 \cdot n$ constraints, the lemma follows. □

The main part of the reduction is encapsulated in the following theorem.

Theorem 5. *If, for some $a \in (0,1)$, there is an algorithm for the $(1+\epsilon)$-approximate CSS problem, with running time $2^{O(1/\epsilon^{2a})} \cdot (dn)^{O(1)}$, then for any (constant) $C_b > 1$, there is a C_b-approximation algorithm for the hitting set problem with the running time $2^{O(m^a \log^{O(1)} m)}$.*

Proof. We exploit the following nice combinatorial structure. Consider a code $C \subset \{0,1\}^d$ with codewords $c_1 \dots c_m$, and $C' \subset \{0,1\}^d$ with codewords $c'_1 \dots c'_{nm}$ (alternatively referred to as $c'_{1,1} \dots c'_{n,m}$) with the following properties parametrized by constants $b, b' > 0$, $b'' > 1$, as well as a parameter $t > 0$:

1. For any $T \subset [m]$, $|T| = t$, $R(\{c_i : i \in T\}) \le r$.
2. Let $t' = b''t$. Consider any sequence P of t' pairs $(c_{i_1}, c'_{j_1}) \dots (c_{i_{t'}}, c'_{j_{t'}})$, such that all of the indexes $i_1 \dots i_{t'}$ and $j_1 \dots j_{t'}$ are pairwise distinct. Then, for each pair (c_{i_k}, c'_{j_k}), take q_k to be any d-length substring of $c_{i_k} \circ c'_{j_k}$ or $c'_{j_k} \circ c_{i_k}$. We want to have a property that $R(\{q_1 \dots q_{t'}\}) > r'$ for $r' = r(1+\epsilon)$, where $\epsilon = b'/\sqrt{t}$ for some (tiny) $b' > 0$.

For completeness we mention that $r = d/2(1 - b/\sqrt{t})$, where $b \approx \sqrt{2/\pi}$. Observe that the Property 2 essentially states that the code obtained by taking substrings of a product of C and C' has good list-decodable properties. Arguably, the definition would be more intuitive if we simply insisted that C has good list-decodable properties. Indeed, that suffices if our goal is to show hardness for just the *Group Closest String* problem, where the goal is to find a substring that is close to at least one string from each of n *groups* of strings. The latter problem strictly generalizes CSS, since we can define the ith group to contain all substrings of the ith input string. The more complicated definition is a consequence of proving hardness for the more restrictive CSS problem.

Lemma 9. *For any fixed constant $b'' > 1$, and variables $m > 1$, $t > 1$, $\epsilon > 0$, the "nice combinatorial structure" defined above can be constructed probabilistically with success probability at least $2/3$, with $d = t^{O(1)} \log m$, and positive b' strictly bounded away from 0.*

We defer the proof till later. For now, we assume C and C' as above.

The reduction from HS to CSS is as follows. For each set $A_i = A = \{a_1 \dots a_l\}$ we generate a string

$$s_i = c_{a_1} \circ c'_{i,1} \dots c_{a_l} \circ c'_{i,l}$$

The intuition is that the codewords of C represent the input, while the codeword of C' are placeholders, to make sure that for each string s_i, the substring of s_i that is "close" to the solution string does not overlap with more than one codeword from C.

We now show that:

- If there is a hitting set H of size t, then there is a solution to CSS with cost at most r.
- If there is a solution to CSS with cost $r' = r(1 + \epsilon)$, then there is a hitting set of size at most $b''t$.

The first part is easy. If there is a hitting set $H = \{a_1 \dots a_t\}$ of size t, then (by Property 1) $R(\{c_{a_1} \dots c_{a_t}\}) \leq r$. The corresponding string provides a solution to CSS with cost at most r.

The second part is as follows. Suppose that we are given an x and indexes $l_1 \dots l_n$ such that for each $i_1 \dots i_n$ we have $D(x, s_i[l_i \dots l_i + d - 1]) \leq r(1 + \epsilon)$. Denote $p_i = s_i[l_i \dots l_i + d - 1]$. Note that each p_i is a d-length substring of $c_{j_i} \circ c'_{j'_i}$ or $c'_{j'_i} \circ c_{j_i}$, for some $c_{j_i} \in C, c'_{j'_i} \in C'$. Also, note that all $c'_{j'_i}$ are distinct.

Consider $H = \{j_1 \dots j_n\}$. By construction it is a hitting set. The question is, how many *distinct* elements it contains. Assume it has at least t' elements $a_1 \dots a_{t'}$. But then we know, by Property 2 of the "nice combinatorial structure", that $R(\{p_1 \dots p_n\}) > r'$. Thus, $|H| < t'$. □

Proof of Lemma 9. It suffices to construct the code with the desired properties. We use the probabilistic method, that is, for each $i = 1 \dots m$, $j = 1 \dots d$, we select c_i independently uniformly at random from $\{0,1\}^d$. We do the same for C'.

Observe that the bits in $S_T = \{c_i : i \in T\}$ (as in Property 1), as well as the bits in $\{q_1 \dots q_{t'}\}$ (as in Property 2) are independent Bernoulli variables. Our strategy is therefore to show that the respective properties hold for t or t' strings randomly chosen from $\{0,1\}^d$, with probability $\exp(-\Omega(d/t^{O(1)}))$. This means that, if we set $d = t^{O(1)} \log m$, then the respective properties will hold for all required sets of strings with high probability.

Consider $c_1 \dots c_t$, chosen independently from $\{0,1\}^d$. First we take care of the high probability bound. Define $R_t = R(\{c_1 \dots c_t\})$, where each c_i is chosen independently and uniformly at random from $\{0,1\}^d$.

In the following $\delta > 0$ denotes a (tiny) constant.

Lemma 10. *The random variable $R = R_t$ is sharply concentrated around its mean. That is, for any $\delta > 0$:*

$$\Pr[|R - E[R]| > d/2 \cdot \delta/\sqrt{t}] \leq 2 \exp\left(-\frac{\delta^2 d}{8t^2}\right)$$

Proof. Observe that, for any arguments $c_1 \dots c_t$, changing one coordinate of c_i changes the value of R by at most 1. The bound then follows from Azuma's inequality, since

$$Pr[|R - E[R]| > d/2 \cdot \delta/\sqrt{t}] \leq 2 \exp\left(-\frac{(d/2 \cdot \delta/\sqrt{t})^2}{2dt}\right)$$

$$= 2 \exp\left(-\frac{\delta^2 d}{8t^2}\right)$$

□

Now we proceed with the upper bound on R_t. Consider l Bernoulli variables $u_1 \dots u_l$, and let $u = \sum_i u_i$. Consider now the quantity $E_l = E[u | u \leq l/2]$. For concreteness, we mention that $E_l = l/2(1 - (\sqrt{2/\pi} + o(1))/\sqrt{l})$, which follows from the value of the influence of a variable in a majority function [30]. We express the lower and upper bound for $E[R]$ in terms of E_t and $E_{t'}$, respectively.

Lemma 11.

$$\Pr[R_t > (E_t/t + \frac{\delta}{2\sqrt{t}}) \cdot d] \leq 2d \exp\left(-\frac{\delta^2 d}{8t^2}\right)$$

Proof. We construct a vector x such that, with high probability over the choice of the code, $D(x, c_i) \leq (E_t/t + \frac{\delta}{2\sqrt{t}}) \cdot d$ for each $i = 1 \dots d$. Our approach is to use the majority vote, that is, to define $x_i = Majority((c_1)_i \dots (c_t)_i)$, $i = 1 \dots d$. By symmetry argument, for any $i = 1 \dots d$, we have $\Pr[(c_j)_i \neq x_i] = E_t/t$. Thus, $E[D(c_i, x)] = dE_t/t$. Application of Lemma 10 finishes the proof. □

Now, we need to show a lower bound for $R_{t'}$. Before we do that, we mention that a weaker lower bound can be obtained by using Johnson bound for error correcting codes. Specifically, assume that d is large enough so that the minimum distance of the code $Q = \{q_1 \dots q_{t'}\}$ is $\approx d/2$. Then we can use Johnson bound, as in [16]. It says that:

$$R(Q) \ge d/2(1 - 1/\sqrt{t'})$$

Unfortunately, this lower bound is not very tight – the aforementioned upper bound would guarantee about $d/2(1 - \sqrt{2/\pi}/\sqrt{t'})$. This discrepancy occurs because the Johnson bound works for *any* code with sufficiently large minimum distance. To circumvent this difficulty, we are going to show a better Johnson-type bound for a *random* code

Lemma 12. *$E[R_{t'}] \ge dE_{t'}/t'$. Therefore*

$$\Pr[R_{t'} < (E_{t'}/t' - \frac{\delta}{2\sqrt{t'}}) \cdot d] \le 2d \exp\left(-\frac{\delta^2 d}{8t'^2}\right)$$

Proof. Consider a linear relaxation of the problem of finding $R(Q)$. Specifically, the linear program finds the smallest (with respect to the l_1 metric) ball enclosing Q, centered at *any* point $x \in [0,1]^d$.

$$\begin{array}{lll}
\text{minimize} & r & \\
\text{subject to:} & & \\
x_j + r_{ij} & \ge (q_i)_j & \text{for all } i = 1 \dots t', j = 1 \dots d \\
-x_j + r_{ij} & \ge -(q_i)_j & \\
r - \sum_j r_{ij} & \ge 0 & \text{for all } i = 1 \dots t' \\
r_{ij}, x_i, r & \ge 0 &
\end{array}$$

To show a lower bound for this program, we consider the dual LP, with dual variables y^+_{ij}, y^-_{ij}, y_i corresponding to the respective inequalities in the primal LP.

$$\begin{array}{lll}
\text{maximize} & \multicolumn{2}{l}{\sum_{i,j} (y^+_{ij} - y^-_{ij})(q_i)_j} \\
\text{subject to:} & & \\
y^+_{ij} + y^-_{ij} - y_i & \le 0 & \text{for all } i = 1 \dots t', j = 1 \dots d \\
\sum_i (y^+_{ij} - y^-_{ij}) & \le 0 & \text{for all } j = 1 \dots d \\
\sum_i y_i & \le 1 & \\
y^+_{i,j}, y^-_{ij}, y_i & \ge 0 &
\end{array}$$

We will now demonstrate, for each input q_{ij}, a feasible solution. The expected value of the objective function will be equal to $dE_{t'}/t'$.

We set $y_i = 1/t'$, for each $i = 1 \dots t'$. For each i, j, we set either y^+_{ij} or y^-_{ij} to 0. The other possible value for these variables is $1/t'$.

The specific assignment is as follows. For each $j = 1 \dots d$, let $M_j = \{i : (q_i)_j = 1\}$. Let $m_j = \min(|M_j|, t' - |M_j|)$. For m_j indexes $i \in M_j$, we set $y^+_{ij} = 1/t$. For m_j indexes $i \notin M_j$, we set $y^-_{ij} = 1/t$. The remainder variables are set to 0.

It is easy to see that the resulting solution is feasible. Moreover, the value of the objective function is at least $\sum_{j=1}^d m_j/t'$. It follows that its expected value is equal to $dE_{t'}/t'$. □

We now finalize the proof of Lemma 9. By the last three lemmas, we know that, for large enough $d = t^{O(1)} \log m$, C and C' satisfy the following two conditions with high probability:

1. C satisfies Property 1 for
$$r = d(E_t/t + \frac{\delta}{2\sqrt{t}}) \le d/2\left(1 - \frac{\sqrt{2/\pi} - o(1) - \delta}{\sqrt{t}}\right)$$
2. C and C' satisfy Property 2 for
$$r' = d(E_{t'}/t' - \frac{\delta}{2\sqrt{t'}}) \ge d/2\left(1 - \frac{\sqrt{2/\pi} + o(1) + \delta}{\sqrt{t'}}\right)$$

If $b'' = t'/t > 1$, then $1 + \epsilon = r'/r \ge 1 + b'/\sqrt{t}$ for some $b' > 0$, by taking sufficiently small δ and large enough t. Lemma 9 follows. □

Remark 1. *From the above discussion it follows that for any subset $S \subset C$, $|S| = t$, we have $R_1(S) \le R(S) \le d/2(1 - b/\sqrt{t})$. At the same time, $R_1(C) \ge d/2(1 - 1/\sqrt{m}) \ge d/2(1 - \frac{b}{2}/\sqrt{t})$ for large enough m. Thus, we have that $R_1(S) < R_1(C)(1 - \epsilon)$ as long as $t < C/\epsilon^2$ for some constant $C > 0$. Therefore, any core-set for C under the l_1 norm must have size $\Omega(1/\epsilon^2)$.*

References

[1] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via coresets - survey. *Combinatorial and Computational Geometry (MSRI publication)*, 52, 2005.

[2] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. *Proceedings of the Symposium on Theory of Computing*, 2006.

[3] N. Alon. Problems and results in extremal combinatorics i. *Discrete Mathematics*, 273:31–53, 2003.

[4] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Proceedings of the Symposium on Theory of Computing*, pages 20–29, 1996.

[5] O. Barkol and Y. Rabani. Tighter bounds for nearest neighbor search and related problems in the cell probe model. *Proceedings of the Symposium on Theory of Computing*, 2000.

[6] A. Borodin, R. Ostrovsky, and Y. Rabani. Lower bounds for high dimensional nearest neighbor search and related problems. *Proceedings of the Symposium on Theory of Computing*, 1999.

[7] B. Brinkman and M. Charikar. On the impossibility of dimension reduction in l1. *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003.

[8] M. Bădoiu and K. Clarkson. Smaller core-sets for balls. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[9] M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. *Proceedings of the Symposium on Theory of Computing*, 2002.

[10] A. Chakrabarti, B. Chazelle, B. Gum, and A. Lvov. A lower bound on the complexity of approximate nearest-neighbor searching on the hamming cube. *Proceedings of the Symposium on Theory of Computing*, 1999.

[11] A. Chakrabarti and O. Regev. An optimal randomised cell probe lower bounds for approximate nearest neighbor searching. *Proceedings of the Symposium on Foundations of Computer Science*, 2004.

[12] J. Chen, X. Huang, I. Kanj, and G. Xia. Linear fpt reductions and computational lower bounds. *Proceedings of the Symposium on Theory of Computing*, 2004.

[13] G. Cormode and S. Muthukrishnan. Improved data stream summaries: The count-min sketch and its applications. *FSTTCS*, 2004.

[14] S. Dasgupta. Learning mixtures of gaussians. *Proceedings of the Symposium on Foundations of Computer Science*, pages 634–644, 1999.

[15] I. Dinur. The pcp theorem by gap amplification. *Proceedings of the Symposium on Theory of Computing*, 2006.

[16] V. Guruswami. List decoding of error-correcting codes. *Ph.D thesis, Massachusetts Institute of Technology*, August, 2001.

[17] S. Har-Peled. A replacement for voronoi diagrams of near linear size. *Proceedings of the Symposium on Foundations of Computer Science*, 2001.

[18] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. *Proceedings of the Symposium on Foundations of Computer Science*, 2000.

[19] P. Indyk. Tutorial: Algorithmic applications of low-distortion geometric embeddings. *Proceedings of the Symposium on Foundations of Computer Science*, 2001.

[20] P. Indyk. Nearest neighbor in high dimensional spaces. *CRC Handbook of Discrete and Computational Geometry, 2nd edition*, 2003.

[21] P. Indyk and R. Motwani. Approximate nearest neighbor: towards removing the curse of dimensionality. *Proceedings of the Symposium on Theory of Computing*, 1998.

[22] P. Indyk and D. Woodruff. Tight lower bounds for the distinct elements problem. *Proceedings of the Symposium on Foundations of Computer Science*, pages 283–290, 2003.

[23] Y. Jiao, J. Xu, and M. Li. On the k-closest substring and k-consensus pattern problems. *Proceedings of the Symposium on Combinatorial Pattern Matching*, pages 130–144, 2004.

[24] W. Johnson and J. Lindenstrauss. Extensions of lipshitz mapping into hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[25] N. Jones and P. Pevzner. *An introduction to Bioinformatics Algorithms*. MIT Press, 2004.

[26] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *Proceedings of the Thirtieth ACM Symposium on Theory of Computing*, pages 614–623, 1998.

[27] M. Li, B. Ma, and L. Wang. On the closest string and substring problems. *Journal of the ACM. Early versions appeared in STOC 99 and CPM 00.*, 49(2):157–171, 2002.

[28] D. Marx. The closest substring problem with small distances. *Proceedings of the Symposium on Foundations of Computer Science*, 2005.

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

[30] E. Mossel and R. O'Donnell. On the noise sensitivity of monotone functions. *Random Struct. Algorithms*, 23(3):333–350, 2003.

[31] S. Muthukrishnan. Data streams: Algorithms and applications (invited talk at soda'03). *Available at http://athos.rutgers.edu/∼muthu/stream-1-1.ps*, 2003.

[32] R. Ostrovsky and Y. Rabani. Polynomial time approximation schemes for geometric k-clustering. *Proceedings of the Symposium on Foundations of Computer Science*, 2000.

[33] P. Pevzner and S. Sze. Combinatorial approaches to finding subtle signals in dna sequences. *International Conference on Intelligent Systems for Molecular Biology*, 2000.

[34] M. Pătraşcu and M. Thorup. Higher lower bounds for near-neighbor and further rich problems. *Proceedings of the Symposium on Foundations of Computer Science*, 2006.

[35] D. Woodruff. Optimal space lower bounds for all frequency moments. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2004.

A. Upper bounds for clustering problems

A.1. $(1+\epsilon)$-approximate closest string in $n^{O(\log(1/\epsilon)/\epsilon^2)}$ time

In this section we describe an improved algorithm for the closest string problem.

In [27], an algorithm with running time of $n^{O(1/\epsilon^4)}$ was given. As we show, it is not difficult to reduce the exponent to $O(1/\epsilon^2 \cdot \log(1/\epsilon))$. Firstly, we need to briefly review the algorithm of [27].

Let $R = R(S)$. If $R > C \log n/\epsilon^2$ for some constant C, then the problem can be solved as follows:

- Write an integer program optimizing $R(S)$, with variables $x_i \in \{0,1\}$, $i = 1 \dots d$

- Relax it to a linear program, with variables $x'_i \in [0,1]$; this can be solved in polynomial time.

- Use randomized rounding to convert x'_is into x_i's. This works as long as the expectations of $D(s_i, x)$ are $\Omega(\log n/\epsilon^2)$, which is the case by our assumption.

So, if R is large, we are done. What if $R = O(\log n/\epsilon^2)$? A simple approach would be to take any $s \in S$, and enumerate all x such that $D(s,x) \leq R$. This clearly identifies an optimal solution. The drawback is that this results in quasi-polynomial time of $d^{O(\log n/\epsilon^2)}$.

To avoid this problem, the paper [27] proposed the following "dimensionality reduction" idea. For each set $P \subset S$, we define the set $I(P) = \{i : p_i \neq q_i$, for some $p, q \in P\}$; note that $I(P)$ is efficiently computable given P. Let $I'(P)$ be the complement of $I(P)$. The idea is to show that there exists a small set P such that for an optimum solution x, and $I' = I'(P)$, we have $D(x_{|I'}, s_{|I'}) \leq \epsilon R$ for all $s \in P$. This is good news, since now we can create a solution x', such that $x'_{|I'} = s_{|I'}$ for some $s \in S$, and $x'_{|I} = x_{|I}$. By the above, x' is a $(1+\epsilon)$-approximate solution to the problem. To find it, we only need to find $x_{|I}$. However, $|I| \leq |P| \cdot 2R = O(|P| \log n/\epsilon^2)$. Thus, as long as $|P|$ is small, we can find $x'_{|I}$ by exhaustive enumeration.

How small can $|P|$ be ? The original paper [27] showed a bound polynomial in $1/\epsilon$. However, Lemma 2.2 of [28] (more specifically, the statement in the second line of the proof of that Lemma) gives an upper bound of just $\log(1/\epsilon)$. Therefore, we obtain an algorithm with the running time of $n^{O(\log(1/\epsilon)/\epsilon^2)}$.

A.2. Other clustering problems

The papers [14, 32] discovered a method for clustering in high dimensional spaces using dimensionality reduction. As it turns out, this method can be generalized so that it applies to a wide variety of problems, including the closest substring problem. In fact, the algorithm of [27] can be viewed as an instantiation of that method.

The general method is as follows. Assume that each cluster has a center; we denote the centers by $c_1 \dots c_k$.

1. Construct a $(1+\epsilon)$-approximate mapping A from the original d-dimensional space $\Re^d$, to the host space $\Re^k$. It suffices that this mapping is correct for the input points and $c_1 \dots c_k$, which can be guaranteed by taking $k = O(\log n \cdot 1/\epsilon^2)$.

2. Map (using A) all input points P into $\Re^k$.

3. Enumerate "all" possible images $Ac_1 \dots Ac_k$ (after a proper discretization)

4. Infer the "combinatorial structure" of an optimum clustering from $A(P)$ and $Ac_1 \dots Ac_k$. In the context of clustering, for each $Ap, p \in P$, find the nearest point Ac_i.

5. Using the above information, solve the problem in $\Re^d$. E.g., use the information to partition P into clusters, and find optimum center in $\Re^d$ for each cluster.

This approach nicely applies to the closest substring problem. Let $Group_d(s)$ denote the set of all contiguous d-length substrings of s. We map all d-length substrings $Group_d(s_i)$ of the input strings into $\Re^k$.

In the next step, we enumerate "all" candidates for the optimal center string s. This is implemented as follows. First, we "guess" the value $C = C(s)$ of the objective function at the optimum string s. Then, we guess the index j which minimizes $D(s, s_1[j \dots j+d-1])$. Let $s' = s_1[j \dots j+d-1]$. Since $D(s,s') \leq C$, it follows that $\|s - s'\|_2^2 \leq C$, and therefore $\|As - As'\|_2^2 \leq C(1+\epsilon)$.

We now find a "good enough" approximation to As as follows. First, we impose an $\epsilon\sqrt{C}$-net N on the l_2 norm ball $B(As', \sqrt{C(1+\epsilon)})$. It is possible to construct such a net so that $|N| \leq (1/\epsilon)^{O(k)}$ in time polynomial in $|N|$. Then, we "guess" $p \in N$ that is closest to As. Note that we have $\|As - p\|_2 \leq \epsilon\sqrt{C}$.

Now we choose, for each $i = 1 \dots n$, the index j_i such that the substring $s'_i = s[j_i \dots j_i + d - 1]$ minimizes $\|As'_i - p\|_2$. Observe that $\|As'_i - As\|_2 \leq (1+\epsilon)\sqrt{C}$. Therefore we have

$$\|As'_i - p\|_2 \leq \|As'_i - As\|_2 + \|As - p\|_2 \leq (1+2\epsilon)\sqrt{C}$$

At the same time, consider any other substring s' of $s_1 \dots s_n$ such that $D(s,s') \geq (1+12\epsilon)C$. As before we get

$$\|As' - p\|_2 \geq \|As' - As\|_2 - \epsilon\sqrt{C} \geq (1-\epsilon)\|s' - s\|_2 - \epsilon\sqrt{C}$$

$$\geq \sqrt{C(1+12\epsilon)}(1-\epsilon) - \epsilon\sqrt{C}$$

which for ϵ small enough is at least

$$\sqrt{C}(1+5\epsilon)(1-\epsilon) - \epsilon\sqrt{C} > \sqrt{C}(1+2\epsilon)$$

Therefore, all strings s'_i chosen by the algorithms must satisfy $D(s'_i, s) < (1+12\epsilon)C$.

It follows that solving a $(1+\epsilon)$-approximate closest string problem for $s'_1 \dots s'_n$ yields a $(1+O(\epsilon))$-approximate solution to the closest substring problem for $s_1 \dots s_n$. The total time needed to enumerate all "guesses" is at most $(1/\epsilon)^{O(\log n/\epsilon^2)} = n^{O(\log(1/\epsilon)/\epsilon^2)}$.