Efficient and Private Distance Approximation in the Communication and Streaming Models

by

David P. Woodruff

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2007

© Massachusetts Institute of Technology 2007. All rights reserved.

Author . . . .
Department of Electrical Engineering and Computer Science
August 31, 2007

Certified by . . . .
Piotr Indyk
MIT Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Efficient and Private Distance Approximation in the Communication and Streaming Models

by

David P. Woodruff

Submitted to the Department of Electrical Engineering and Computer Science
on August 31, 2007, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis studies distance approximation in two closely related models - the streaming model and the two-party communication model.

In the streaming model, a massive data stream is presented in an arbitrary order to a randomized algorithm that tries to approximate certain statistics of the data with only a few (usually one) passes over the data. For instance, the data may be a flow of packets on the internet or a set of records in a large database. The size of the data necessitates the use of extremely efficient randomized approximation algorithms. Problems of interest include approximating the number of distinct elements, approximating the surprise index of a stream, or more generally, approximating the norm of a dynamically-changing vector in which coordinates are updated multiple times in an arbitrary order.

In the two-party communication model, there are two parties who wish to efficiently compute a relation of their inputs. We consider the problem of approximating L_p distances for any $p \geq 0$. It turns out that lower bounds on the communication complexity of these relations yield lower bounds on the memory required of streaming algorithms for the problems listed above. Moreover, upper bounds in the streaming model translate to constant-round protocols in the communication model with communication proportional to the memory required of the streaming algorithm. The communication model also has its own applications, such as secure datamining, where in addition to low communication, the goal is not to allow either party to learn more about the other's input other than what follows from the output and his/her private input.

We develop new algorithms and lower bounds that resolve key open questions in both of these models. The highlights of the results are as follows.

1. We give a 1-pass $\tilde{O}(m^{1-2/p})\text{poly}(1/\epsilon)$-space streaming algorithm for $(1 \pm \epsilon)$-approximating the L_p norm of an m-dimensional vector presented as a data stream for any $p \geq 2$. This improves the previous $\tilde{O}(m^{1-1/(p-1)})\text{poly}(1/\epsilon)$ bound, and is optimal up to polylogarithmic factors. As a special case our algorithm can be used to approximate the frequency moments F_p of a data stream with the same optimal amount of space. This resolves the main open question of the 1996 paper by Alon, Matias, and Szegedy.

2. We give an $\Omega(1/\epsilon^2)$ lower bound for approximating the number of distinct elements of a data stream in one pass to within a $(1 \pm \epsilon)$ factor with constant probability, as well as the p-th frequency moment F_p for any $p \geq 0$. This is tight up to very small factors, and greatly improves upon the earlier $\Omega(1/\epsilon)$ lower bound for these problems. It also

gives the same quadratic improvement for the communication complexity of 1-round protocols for approximating the L_p distance for any $p \geq 0$.

3. In the two-party communication model, we give a protocol for *privately* approximating the Euclidean distance (L_2) between two m-dimensional vectors, held by different parties, with only polylog m communication and $O(1)$ rounds. This tremendously improves upon the earlier protocol of Feigenbaum, Ishai, Malkin, Nissim, Strauss, and Wright, which achieved $O(\sqrt{m})$ communication for privately approximating the Hamming distance only.

This thesis also contains several previously unpublished results concerning the second item above, including new lower bounds for the communication complexity of approximating the L_p distances when the vectors are uniformly distributed and the protocol is only correct for most inputs, as well as tight lower bounds for the multiround complexity for a restricted class of protocols that we call linear.

Thesis Supervisor: Piotr Indyk
Title: MIT Professor

Acknowledgments

First, I would like to thank my advisor Piotr Indyk for making all of this possible. Piotr has always helped me find important and interesting problems and has had very insightful comments about all of my work. His constant presence in the lab was extremely useful whenever I had a question. He was always willing to listen to my ideas, even if they were not related to his main research interests.

Next, I would like to thank my Master's thesis advisor Ron Rivest and my coauthor Marten van Dijk. Marten and Ron introduced me to cryptographic research and shaped me in my early days. This influence has helped me ever since, and kept my passion for cryptography alive.

I am also very grateful for a few fantastic internships. I thank Jessica Staddon at PARC for introducing inference control to me and putting up with my endless emails on privacy-preserving data mining. I also thank the tag team Craig Gentry and Zulfikar Ramzan at DoCoMo Labs. With their talent and contagious enthusiasm, we made a lot of progress on broadcast encryption that summer.

I was also extremely fortunate to visit Andy Yao for a year at Tsinghua University in Beijing. The hospitality and support of Andy was perfect for my development. The students there - Hongxu Cai, Lan Liu, Daniel Preda, Xiaoming Sun, Hoeteck Wee, Jing Zhang, and many others - made for an amazing learning experience. On a personal note, I would like to thank Andris, Hoeteck, Lan, and especially Zhang Yi for inspiring me and helping me stay alive in Beijing.

I would like to thank all of my coauthors for their hard work - Arnab Bhattacharyya, Marten van Dijk, Craig Gentry, Robert Granger, Elena Grigorescu, Piotr Indyk, Kyomin Jung, Dan Page, Zully Ramzan, Sofya Raskhodnikova, Karl Rubin, Alice Silverberg, Jessica Staddon, Martijn Stam, Xiaoming Sun, Sergey Yekhanin, and Hanson Zhou.

I also thank many others for enlightening discussions - Alex Andoni, Victor Chen, Khanh DoBa, Nick Harvey, Mohammad Taghi Hajiaghayi, Yuval Ishai, Jon Kelner, Swastik Kopparty, Silvio Micali, Vahab Mirrokni, Payman Mohassel, Jelani Nelson, Mihai Patrascu, Chris Peikert, Seth Pettie, Benny Pinkas, Ronitt Rubinfeld, Tasos Sidiropoulos, Adam Smith, Madhu Sudan, Vinod Vaikuntanathan, Grant Wang, and those whose names I cannot remember at the moment.

Finally, I'd like to thank my parents, David and Marsha, for unflagging encouragement and support.

Financially, I would like to thank Akamai and the Department of Defence for fellowships (Presidential and NDSEG, respectively) that gave me the freedom to pursue my own interests as a graduate student.

Contents

Chapter 1

Introduction

Consider the following scenario: there are two players, Alice and Bob, holding inputs x and y respectively, who wish to compute a function $f(x, y)$. To do this, they need to communicate with each other. They'd like to do this by transmitting as few bits as possible. This is the classical two-party communication model introduced by Yao [69, 53].

Depending on the function f and the resources available, this task may require a lot of communication or very little. Examples of such resources include space and time complexity, as well as the ability to flip random coins. One class of functions of considerable interest is the class of *distance* functions. In this case, x and y are finite strings of length m, with characters drawn from an alphabet Σ, and $f(x, y)$ measures how similar the strings are.

For example, if $\Sigma = \{0, 1\}$, then $f(x, y)$ could be the Hamming distance between x and y, that is, the number of positions which differ. If Σ is the set of real numbers $\mathbb{R}$, a natural distance function f is the L_p distance. Here, $L_p(x, y)$ is defined to be $(\sum_{i=1}^{m} |x_i - y_i|^p)^{1/p}$. If $p = 2$, this is the Euclidean distance. When $p = 0$, it is natural to define $L_0(x, y)$ as the Hamming distance between x and y.

Most of this thesis is concerned with some form of study of the communication complexity of approximating L_p distances. In this case, it is not a function $f(x, y)$ that we are after, but rather a relation $S \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, where $(x, y, z) \in S$ if and only if $(1 - \epsilon)L_p(x, y) \leq z \leq (1 + \epsilon)L_p(x, y)$, where $\epsilon \geq 0$ is an approximation parameter.

A major reason for studying distance approximation in the communication model is the strong connection with the *data-stream model* [2, 58].

The Data-Stream Model: Imagine an internet router with only limited storage and processing power. Everyday, gigabytes of information travel through the router in the form of packets. Despite its limited abilities, the router would like to make sense of this information, but it will settle for a few basic statistics. For example, it might want to know what fraction of network IP addresses have daily traffic, which IP addresses are the most popular, or which, if any, IP addresses have received a sudden spike in traffic. In the data-stream model a huge stream of elements is presented to an algorithm in an arbitrary, possibly adversarial order, and the algorithm is only given a constant number of passes (usually one) over the stream. In most cases the limited resources of the algorithm imply that only probabilistic estimates are possible.

More formally, suppose A is an algorithm which, in an online fashion, receives items labeled by integers $i \in [m]$. For example, i could be the destination IP address of a packet passing through a router running A. Note that A may receive as input the same item i many times. Let f_i be the number of occurrences of item i, and for $p \geq 0$, define the p-th frequency moment $F_p = \sum_{i \in [m]} f_i^p$. This statistic was defined by Alon, Matias, and Szegedy [2], and is just the p-th power of the L_p norm of the vector represented by the stream. When $p = 0$, we let $0^0 = 0$. Thus, F_0 is the number of distinct elements in the stream.

There are several practical motivations for designing space-efficient algorithms for approximating the frequency moments. In the networking example above, F_0 can be used to determine the fraction of network IP addresses that have daily traffic. There are also algorithms (see, e.g., [21]) for finding the most popular IP addresses with error that depends on the F_2-value of the packet stream, so estimating this quantity is useful for bounding the error of such algorithms. F_p for higher p gives an indication of the *skewness* of the data stream. As $p \to \infty$, F_p approaches $\max_{i \in [m]} f_i$. For an application of F_p-estimation algorithms to detecting Denial of Service attacks, see [1].

The frequency moments are also very valuable to the database community. With commercial databases approaching the size of 100 terabytes, it is infeasible to make multiple passes over the data. Query optimizers can use F_0 to find the number of unique values in a database without having to perform an expensive sort. F_2 is a quantity known as the *surprise index* or *Gini's index of homogeneity*. Efficient algorithms for F_2 are useful for determining the output size of self-joins (see, e.g., [38]). Finally, F_p for $p > 2$ can be used to measure the size of a self-join on more than two tables of a database or to approximate the maximum frequency of an entry.

The Connection Between Communication and Data-Stream Models: Suppose, for example, that there is a streaming algorithm A that outputs $\tilde{F}_0$ such that

$$\Pr\left[(1-\epsilon)F_0 \leq \tilde{F}_0 \leq (1+\epsilon)F_0\right] \geq 1-\delta,$$

where the probability is taken only over A's coin tosses. A natural bound of interest is how much memory A needs. This is where communication complexity comes into play.

Suppose Alice has $x \in \{0,1\}^m$, Bob has $y \in \{0,1\}^m$, and they would like to estimate the Hamming distance $\Delta(x,y)$. Alice creates a stream $\mathcal{S}_x$ as follows. For each i such that $x_i = 1$, Alice appends i to $\mathcal{S}_x$. Similarly, Bob creates $\mathcal{S}_y$. Alice then runs A on $\mathcal{S}_x$. When she is finished, she transmits the memory contents of A to Bob, who continues the computation on $\mathcal{S}_y$. At the end, A will have been run on $\mathcal{S} = \mathcal{S}_x \circ \mathcal{S}_y$, the concatenation of the two streams.

For $s \in \{0,1\}^m$, let $wt(s)$ be the number of ones in the string s. Alice also transmits $wt(x)$ to Bob, using $O(\log m)$ bits. A simple calculation shows that

$$F_0(\mathcal{S}) = \frac{wt(x)+wt(y)}{2} + \frac{\Delta(x,y)}{2}.$$

If A were to compute $F_0(\mathcal{S})$, Bob could compute $\Delta(x,y)$. Thus, the memory required of A must be at least the communication required for computing the Hamming distance, minus the $O(\log m)$ bits needed for transmitting $wt(x)$. Similarly, approximating F_0 translates into approximating $\Delta(x,y)$, and so lower bounds on the memory required can be obtained even for randomized approximation algorithms. Similarly, lower bounds on the communication of approximating L_p distances translate into lower bounds on approximating F_p in the streaming model. Thus, important statistics in the data-stream model translate into distance computations in the communication setting.

Not only do lower bounds in the communication model yield lower bounds in the streaming model, but oftentimes upper bounds in the communication model yield upper bounds in the streaming model. Here the connection is informal, and on a case-by-case basis.

A large part of this thesis will be devoted to the streaming complexity of the frequency moments of a data stream. We will give nearly optimal upper bounds for F_p when $p > 2$, and improve the known lower bounds for all $p \geq 0$.

Secure Datamining: Another fairly recent application of communication complexity is secure datamining (see, e.g., [55]). Imagine there are two hospitals, which for medical research purposes, would like to mine their joint data. Due to patient confidentiality, the hospitals would not like to share more of their data than necessary. As private distance approximation is a subroutine in private classification and private clustering algorithms, it is important to understand its complexity in order to understand that of the larger algorithms.

In this thesis we will give very efficient protocols for privately approximating the Hamming and Euclidean distance, so that no party learns more about each other's input other than what is necessary. We will also discuss extensions of private distance approximation to private near neighbor problems. To do this, we will introduce a new notion of privacy suitable for search problems.

Previous Results: The previous results in the non-private communication setting are summarized by the following table. The notation $\tilde{O}, \tilde{\Omega}$, and $\tilde{\Theta}$ will suppress factors that are logarithmic in m, and when talking about the streaming model, that are logarithmic in mn, where n is the length of the stream. All quantities refer to the total communication between the two parties, which are assumed to run in polynomial time.

	Upper Bounds	Lower Bounds
Hamming Distance	$\tilde{O}(1/\epsilon^2)$ [31, 9]	$\Omega(1/\epsilon)$ [5]
L_2	$\tilde{O}(1/\epsilon^2)$ [49]	$\Omega(1/\epsilon)$ [5]
$L_p,\ p > 2$	$\tilde{O}(m^{1-1/(p-1)})\text{poly}(1/\epsilon)$ [25, 33]	$\Omega(m^{1-2/p})\text{poly}(1/\epsilon)$ [7, 6, 65, 20]

Note that the bounds for the Hamming and Euclidean distance depend only polylogarithmically on m, whereas for $L_p, p > 2$, the dependence is polynomial. Thus, for the Hamming and Euclidean distance, the main parameter of interest is the dependence on $1/\epsilon$. Indeed, a $\tilde{O}(1/\epsilon^2)$ upper bound versus a $\tilde{O}(1/\epsilon)$ upper bound can make the difference in practice between setting $\epsilon = .01$, say, and setting $\epsilon = .0001$.

For the L_p distance when $p > 2$, there is a polynomial gap in the upper and lower bounds. For instance, if $p = 3$ the upper bound is $\tilde{O}(m^{1/2})\text{poly}(1/\epsilon)$ while the lower bound is $\Omega(m^{1/3})\text{poly}(1/\epsilon)$. Since m is very large in practice (e.g., in the streaming applications mentioned above), the main parameter of interest here is the dependence on m.

The bounds for the corresponding streaming problems are very similar. For F_0 and F_2, the upper bounds are $\tilde{O}(1/\epsilon^2)$ bits of space [31, 2, 9], while the lower bounds are $\Omega(1/\epsilon)$ [5]. For $F_p, p > 2$, the upper bound is $\tilde{O}(m^{1-1/(p-1)})\text{poly}(1/\epsilon)$ [25, 33][1] while the lower bound is $\Omega(m^{1-2/p})\text{poly}(1/\epsilon)$ [7, 6, 65, 20]. The upper bounds are all realized by algorithms that get only one pass over the input, while the lower bounds allow any constant number of passes.

In the private setting, the previous state of affairs is much worse. It is summarized by the following table.

	Upper Bounds	Lower Bounds
Hamming Distance	$\tilde{O}(\sqrt{m}/\epsilon)$ [29]	$\Omega(1/\epsilon)$ [5]
L_2	$\tilde{O}(m)$ [36, 70]	$\Omega(1/\epsilon)$ [5]
$L_p,\ p > 2$	$\tilde{O}(m)$ [36, 70]	$\Omega(m^{1-2/p})\text{poly}(1/\epsilon)$ [7, 6, 65, 20]

Even for the Hamming and Euclidean distance, the upper bound with privacy depends polynomially on m. For the L_p distance for $p \geq 2$, the dependence is even linear. In all cases, the lower bounds just follow from the communication lower bounds in the setting without privacy.

Our results: Both in the non-private and private settings, we provide resolutions to many of the existing bounds. In the setting without privacy, our results are summarized as follows.

	Upper Bounds	Lower Bounds
Hamming Distance	$\tilde{O}(1/\epsilon^2)$	$\Omega(1/\epsilon^2)$ for 1-round protocols [46, 68]
L_2	$\tilde{O}(1/\epsilon^2)$	$\Omega(1/\epsilon^2)$ for 1-round protocols [68]
$L_p,\ p > 2$	$\tilde{O}(m^{1-2/p})\text{poly}(1/\epsilon)$ [47]	$\Omega(m^{1-2/p})\text{poly}(1/\epsilon)$

Our contribution for the L_p distance for $p > 2$ is a new upper bound of $\tilde{O}(m^{1-2/p})\text{poly}(1/\epsilon)$. Perhaps surprisingly, our new protocol achieving this bound only uses one round of communication. Moreover, it can be implemented in the streaming model by an algorithm that makes only one pass over the data stream and uses only $\tilde{O}(m^{1-2/p})\text{poly}(1/\epsilon)$ bits of space. This gives a streaming algorithm with the same complexity for approximating the frequency moments, and resolves the main question left open in the paper of Alon, Matias, and Szegedy [2].

[1] Independently of our work, Ganguly [33] achieved $\tilde{O}(m^{1-2/(p+1)})\text{poly}(1/\epsilon)$ space.

While our main contribution for the L_p distance for $p > 2$ is a new upper bound, our main contribution for the Hamming and Euclidean distances is a new lower bound. For the Hamming and Euclidean distances, we show a matching $\Omega(1/\epsilon^2)$ lower bound, but our bound holds only for 1-round protocols. For streaming algorithms, this gives an $\Omega(1/\epsilon^2)$ lower bound for algorithms approximating the number of distinct elements, as well as for algorithms approximating F_p for any p, using only one pass over the input. This is not much of a restriction, since the most common setting in the data stream model is when the algorithm only has one pass. We also show that for a certain natural class of protocols, which we call *linear*, the $\Omega(1/\epsilon^2)$ bound holds for multi-round protocols.

Note that for constant p, approximating $L_p(x, y)$ has the same complexity as approximating $L_p^p(x, y)$. Moreover, when $x, y \in \{0, 1\}^m$, $L_p^p(x, y) = \Delta(x, y)$. Thus, for any $x, y \in \{0, 1, \ldots, z\}^m$, for any integer z, we obtain the $\Omega(1/\epsilon^2)$ bound for approximating $L_p(x, y)$ based on our lower bound for approximating $\Delta(x, y)$. We also extend our result to show an $\Omega(1/\epsilon^2)$ lower bound even when x and y are uniformly distributed and the protocol need only be correct for most pairs of inputs.

In the case with privacy, we provide an exponential improvement to the communication complexity of privately approximating both the Hamming and the Euclidean distance. Moreover, our protocol can be implemented with a constant number of rounds. This shows, rather unexpectedly, that privately approximating the Hamming or the Euclidean distance is not that much harder than approximating it without privacy. For the L_p distance, $p > 2$, we leave it as an open question to resolve the polynomial gap in known bounds. This is summarized by the following table. We also give new protocols for exact near neighbor and develop new models and non-trivial upper bounds for approximate near neighbor queries.

	Upper Bounds	Lower Bounds
Hamming Distance	$\tilde{O}(1/\epsilon^2)$ [48]	$\Omega(1/\epsilon)$
L_2	$\tilde{O}(1/\epsilon^2)$ [48]	$\Omega(1/\epsilon)$
$L_p,\ p > 2$	$\tilde{O}(m)$	$\Omega(m^{1-2/p})\text{poly}(1/\epsilon)$

Our Techniques: This thesis unifies techniques in three different worlds - algorithm design (Section 3), communication complexity (Section 4), and cryptography (sections 5 and 6). The common theme is the study of distance approximation.

Our upper bounds for approximating L_p, due to Indyk and the author [47], significantly depart from earlier algorithms, which were obtained by constructing a single estimator, which was shown to equal L_p in expectation and have small variance. We instead group coordinates into buckets based on their values, and try to estimate the size of each bucket. This involves looking at certain randomly chosen substreams of the original stream and invoking a heavy-hitter algorithm on the substream. We can then sum up the contributions of each bucket to obtain an approximation to L_p.

Our lower bounds for streaming algorithms for L_p are derived using the classical framework of communication complexity. Most previous lower bounds in the streaming literature came from studying the disjointness (see, e.g., [50, 64])and indexing functions (see, e.g., [51]). We, however, introduce a new problem - the gap Hamming distance problem, and give a surprising way to lower bound its communication complexity. This problem was first suggested by Indyk and the author [46] and studied explicitly by the author [68]. This problem captures approximating distinct elements of a data stream, as well as L_p norms for $p > 0$, as shown by the author in [68].

Our upper bound for privately approximating L_2 in the communication model, due to Indyk and the author [48], involves a new way of making a certain dimensionality reduction technique private. We use a secure subprotocol to carefully truncate sensitive information from the view of the parties. Our security definitions for private approximate near neighbor problems were first formulated in [48], and were adopted by others (see, e.g., Section 4 of [10]).

Followup Work: After our work, the polylogarithmic factors in our upper bounds for approximating L_p distances and the frequency moments [47] were improved by Bhuvanagiri, Ganguly, Kesh, and Saha [15]. We will present our original proof of the upper bound. Our main idea in [47] of classifying frequencies into buckets and estimating bucket sizes using CountSketch [21] is also used by [15]. Techniques similar to ours [47] as well as those in [15] also appeared in work by Bhuvanagiri and Ganguly [14] on upper bounding the complexity of entropy estimation in the streaming model.

The proofs of our lower bounds for approximating L_p distances and the frequency moments [46, 68] were simplified by Bar-Yossef *et al* [8], and we will present a similarly simplified proof. We will also present a new proof using distributional complexity due to the author which has a number of additional features. The lower bound techniques of Indyk and the author [47] were also used by Andoni, Indyk, and Pătraşcu [4] to show optimality of the dimensionality reduction method. Moreover, the gap Hamming Distance problem and the techniques developed by Indyk and the author [46] and the author [68] appeared in work by Chakrabarti, Cormode, and McGregor [19] for lower bounding the complexity of entropy estimation in the streaming model. Various attempts at bypassing our $\Omega(1/\epsilon^2)$ bounds were considered. For example, Cormode and Muthukrishnan [26] look at streams generated according to Zipfian distributions and can then approximate F_0 and F_2 in $o(1/\epsilon^2)$ space.

Our private approximation algorithm for the Euclidean distance and related techniques [48] were used by Strauss and Zheng [66] for privately approximating the heavy hitters. Our upper bounds for private near neighbor problems [48] were considered by Chmielewski and Hoepman [22], who tried to make them more practical. Finally, our new models for private approximations of search problems [48] were used by Beimel, Hallak, and Nissim [10] in the context of clustering algorithms.

Roadmap: In the next section, we formalize the streaming model.

In Section 3 we obtain a new streaming algorithm for approximating L_p, $p > 0$, to within an arbitrarily small $\epsilon > 0$. Our algorithm achieves optimal space, up to polylogarithmic factors, and consequently gives essentially optimal communication for approximating the L_p distance in the communication model.

In Section 4, we prove lower bounds for the one-way communication complexity of approximating L_p distances for any $p \geq 0$. This yields an $\Omega(1/\epsilon^2)$ lower bound for the space complexity of approximating the frequency moments F_p in the streaming model for any $p \geq 0$. We extend this result in several ways.

In Section 5, we develop a new *private* protocol for approximating the Euclidean distance between two parties to within an arbitrarily small $\epsilon > 0$ using only polylogarithmic communication.

In Section 6, we continue the study of private approximations. We first look at the complexity of several exact near neighbor problems. Then, in an attempt to further improve the complexity, we define a new notion of approximate privacy suitable for near neighbor problems, and we give non-trivial private protocols for these problems.

Chapter 2

Streaming Algorithms

In the data stream model we think of there being an underlying array $A[1], A[2], \ldots, A[m]$. We then see a stream of n elements of the form (j, U_j), where $j \in [m]$ and U_j is a number. We may see the same pair (j, U_j) more than once, or see (j, U'_j) for $U'_j \neq U_j$. There are four common models of data streams: the time series model, the cash register model, the strict turnstile model, and the turnstile model (see, e.g., [58] for an exposition).

In the *time series model*, for each $j \in [m]$, there is only pair in the stream with first coordinate j. Moreover, the pairs are sorted by first coordinate, so the stream appears as $(1, U_1), (2, U_2), \ldots, (m, U_m)$. This model is often unrealistic due to the sortedness of the stream, and we do not consider it in this thesis.

A more general model is the *cash register model*. In this model, the only restriction is that for each pair (j, U_j) in the stream, $U_j \geq 0$. This is perhaps the most popular data stream model ([58]), and is suitable for monitoring IP addresses that access a web server.

An even more general model is the *strict turnstile model*. Here, the U_j may be arbitrary numbers subject to the following constraint. Consider the (ordered) substream of pairs

$$(j, U_{j,1}), (j, U_{j,2}), \ldots, (j, U_{j,f_j}),$$

where f_j is the number of occurrences of j in the data stream. The constraint is that for any i, $1 \leq i \leq f_j$, we have $\sum_{k=1}^{i} U_{j,k} \geq 0$. This models many applications, such as in a database where you can only delete items that have already been entered.

Finally, the most general model is the *turnstile model*. This is like the strict turnstile model, but without any restrctions on the U_js. This models signals that may be both positive and negative at any point in time.

In our lower bounds, we will consider the cash register model, which is the second weakest model (only the time series model is weaker, and actually quite trivial to compute statistics in). In our upper bounds, we will consider the turnstile model, which is the most general model.

We will be interested in computing norms on data streams. Formally, we are given a stream of elements $\mathcal{S} = (1, a_1), (2, a_2), ..., (n, a_n)$, which appear in arbitrary order, and we have an algorithm A which computes a function $f : [m]^n \rightarrow \mathbb{Z}^{\geq 0}$ on the stream.

Definition 1 *An algorithm A is an (ϵ, δ)-approximation algorithm if for all streams $\mathcal{S}$ of n elements of the universe $[m]$,*

$$\Pr[(1-\epsilon)f(\mathcal{S}) \leq A(\mathcal{S}) \leq (1+\epsilon)f(\mathcal{S})] \geq 1-\delta,$$

where the probability is over the coin tosses of A.

Various efficiency measures of A are possible, including its space, update time, and query time.

Definition 2 *The space complexity of an (ϵ, δ)-approximation algorithm A for a function f, denoted $S(A)$ is the maximum amount of space the algorithm uses, over all possible input streams $\mathcal{S}$ and all random coin tosses of the algorithm. Its update time, denoted $UT(A)$ is the maximum, over all input streams $\mathcal{S}$, all integers $i \in [n]$, and all random coin tosses, of the time taken to process a_i given $a_1, \ldots, a_{i-1}$. The query time, denoted $QT(A)$, is the maximum, over all input streams $\mathcal{S}$ and all random coin tosses, of the time taken to report $f(\mathcal{S})$ after given $a_1, \ldots, a_n$.*

Definition 3 *For a function f its (ϵ, δ)-space complexity, denoted $S_{\epsilon,\delta}(f)$ is the minimum over all (ϵ, δ)-approximation algorithms A for f, of $S(A)$. We similarly define $UT_{\epsilon,\delta}(f)$ and $QT_{\epsilon,\delta}(f)$.*

We will mostly be concerned with the space complexity of functions, since low space complexity usually implies a reasonably low time complexity. Also, as we will see in Section 4, there is an intimate connection between the space complexity of a streaming algorithm and the communication complexity of a related problem.

One class of functions f we will focus on is the set of frequency moments of a data stream.

Definition 4 *The pth frequency moment is defined to be* $F_p = \sum_{i \in [m]} g_i^p$, *where* g_i *is the sum of second coordinates of all pairs of the form*

$$(i, U_{i,1}), (i, U_{i,2}), \ldots, (i, U_{i,f_i}),$$

where f_i *is the number of such pairs. If* $p = 0$, *we interpret* 0^0 *as* 0, *and thus* F_0 *is the number of distinct elements in* $\mathcal{S}$.

Remark 5 In most of the previous work on frequency moments, the setting considered was a restricted cash register model with pairs of the form $(j, 1)$ for $j \in [m]$ that appear in any number of times in an arbitrary order. We will, however, handle the more general turnstile model.

Chapter 3

Upper Bounds for L_p Distances

We describe our upper bounds for L_p, $p > 2$, as originally presented in an extended abstract by Indyk and the author [46]. We will describe the upper bound by first giving an upper bound for approximating the k-th frequency moment F_k in the restricted cash register data stream model where we see a stream of pairs of the form $(j, 1)$ for $j \in [m]$. Later, we will extend it to the turnstile model and make the connection from frequency moments to L_p norms.

The earlier algorithms for estimating F_k were obtained by constructing a single estimator, which was shown to equal F_k in expectation, and to have small variance. Our algorithm departs from this approach. Instead, the main idea of our algorithm is as follows. First, we (conceptually) divide the elements into classes S_i, such that the elements in class S_i have frequency $\approx (1+\epsilon)^i$. We observe that, in order for the elements in S_i to contribute significantly to the value of F_k, it must be the case that the size s_i of S_i is comparable to the size of $S_{i+1} \cup \ldots \cup S_{\log_{1+\epsilon} n}$. If this is the case, we have a good chance of finding an element from S_i if we restrict the stream to an $\approx 1/s_i$ fraction of universe elements, and then find the most frequent elements in the substream. The contribution of S_i to F_k can then be approximately estimated by $s_i \cdot (1+\epsilon)^{ik}$. By summing up all estimated contributions, we obtain an estimator for F_k.

Unfortunately, finding the most frequent element in a general stream (even approximately) requires storage that is linear in the stream size. However, if the distribution of the stream elements is not very "heavy-tailed", then a more efficient algorithm is known [21]. This more efficient method is based on the sketching algorithm for F_2 given in [2]. We show that the streams generated by our algorithm (for S_i's that contribute to F_k), satisfy this "tail property", and thus we can use the algorithm of [21] in our algorithm.

A somewhat technical issue that arises in implementing the above algorithm is a need to classify a retrieved element i into one of the classes. For this, we need to know f_i. This information is easy to obtain using a second pass. In the absence of that, we use the estimation of f_i provided by the algorithm of [21], but we define the thresholds defining the classes randomly, to ensure that the error in estimating the frequencies is unlikely to result in misclassification of a frequency.

Our algorithm will be implementable in 1-pass with space $\tilde{O}(m^{1-2/k})\text{poly}(1/\epsilon)$. We do not try to optimize the logarithmic factors or the $\text{poly}(1/\epsilon)$ factors.

3.1 Preliminaries

We are given a stream $\mathcal{S}$ of n elements, each drawn from the universe $[m]$. Our goal is to output an approximation to the kth frequency moment F_k. For simplicity, we assume $k \geq 2$ is a constant (for $k < 2$, space-optimal algorithms already exist [2, 9]), while $m, n, 1/\epsilon$ may be growing. Let $0 < \delta, \epsilon < 1$ be the desired confidence and accuracy of our estimate, respectively. We define the following parameters:

$$c > 0, \quad \epsilon' = c\epsilon, \quad \alpha = 1 + \epsilon', \quad \lambda = \epsilon'/\alpha^k, \quad L = \frac{\lambda}{\log n + 1}.$$

In the analysis we will often assume that c is a sufficiently small constant. W.l.o.g., we may assume that m is a power of two and that n is a power of α. Unless otherwise specified, logs are to the base α. We define the *frequency classes* S_i, for $0 \leq i \leq \log n$, as

$$S_i = \{j \mid \alpha^i \leq f_j < \alpha^{i+1}\}.$$

We use the shorthand notation s_i for $|S_i|$. We say that a class S_i *contributes* if

$$s_i \alpha^{ik} > LF_k.$$

Lemma 6 *If S_i contributes, then $s_i > L \sum_{l>i} s_l$.*

Proof: Since S_i contributes,

$$s_i\alpha^{ik} > LF_k \geq L\sum_l s_l\alpha^{lk} \geq L\sum_{l>i} s_l\alpha^{ik},$$

and the lemma follows by dividing by α^{ik}. ■

We define F_k^C to be the component of F_k due to the contributing frequency classes, namely,

$$F_k^C = \sum_{\text{contributing } S_i} \sum_{j\in S_i} f_j^k.$$

We define F_k^{NC} to be the component due to the non-contributing classes, so $F_k^C + F_k^{NC} = F_k$. The next lemma shows that F_k^{NC} is small.

Lemma 7

$$F_k^{NC} \leq \lambda\alpha^k F_k.$$

Proof: We note that if $j \in S_i$, then $f_j < \alpha^{i+1}$. Therefore,

$$\begin{aligned}
F_k^{NC} &\leq \sum_{\text{non-contr. } S_i} s_i\alpha^{(i+1)k} && \text{(using the definition of } S_i\text{)}\\
&\leq \alpha^k \sum_{\text{non-contr. } S_i} s_i\alpha^{ik} \\
&\leq \alpha^k \sum_{\text{non-contr. } S_i} LF_k && \text{(using the definition of non-contributing } S_i\text{)}\\
&\leq \frac{\alpha^k\lambda F_k}{\log n + 1}(\log n + 1) && \text{(using the definition of } L\text{)}\\
&= \lambda\alpha^k F_k.
\end{aligned}$$

■

We will also make heavy use of the following inequality.

Lemma 8 *Let $0 \leq x < 1$ and $y \geq 1$ be real numbers. Then,*

$$xy - \frac{(xy)^2}{2} \leq 1 - (1-x)^y \leq xy.$$

Proof: For the lower bound,

$$\begin{aligned}
1-(1-x)^y &\geq 1-e^{-xy} && \text{(using that } 1+z \leq e^z \text{ for all reals } z \text{ see, e.g., [57])} \\
&= 1-\left(1-xy+\frac{(xy)^2}{2}-\cdots\right) && \text{(using the Taylor expansion for } e^{-xy}\text{)} \\
&= xy-\frac{(xy)^2}{2}+\cdots \\
&\geq xy-\frac{(xy)^2}{2}.
\end{aligned}$$

For the upper bound, we first show $(1-x)^y \geq 1-xy$. By monotonicity of the $\ln(\cdot)$ function, $(1-x)^y \geq 1-xy$ iff $\ln(1-x)^y \geq \ln(1-xy)$. We use the Taylor expansion for $\ln(1+x)$, that is, for $|x|<1$ we have the expansion $\ln(1+x)=\sum_{n=0}^{\infty}\frac{(-1)^n}{n+1}x^{n+1}$. Then,

$$\ln(1-x)^y = -y\sum_{i=1}^{\infty}\frac{x^i}{i+1}.$$

Also,

$$\ln(1-xy) = -\sum_{i=1}^{\infty}\frac{(xy)^i}{i+1}.$$

We will have $\ln(1-x)^y \geq \ln(1-xy)$ if for all $i \geq 1$,

$$-y\frac{x^i}{i+1} \geq -\frac{(xy)^i}{i+1}.$$

This holds provided $y^{i-1} \geq 1$, which holds for $y \geq 1$, as given by the premise of the lemma. Thus,

$$1-(1-x)^y \leq 1-(1-xy) \leq xy,$$

completing the proof. ■

Corollary 9 *Let $0 \leq x < 1$ and $y \geq 1$ be real numbers. Then,*

$$(1-x)^y \geq (1-xy).$$

Proof: This follows from the proof of the upper bound in the previous lemma. ■

3.2 The Idealized Algorithm

We start by making the unrealistic assumption that we have the following oracle algorithm. Later we remove this assumption by approximating the oracle with the Countsketch algorithm of [21].

Assumption 10 *For some $B = B(m, n)$, there exists a 1-pass B-space algorithm* Max *that outputs the maximum frequency of an item in its input stream.*

We start by describing our algorithm which outputs a $(1 \pm \epsilon)$-approximation to F_k with probability at least 8/9. The main idea is to estimate F_k by estimating each of the set sizes s_i and computing $\sum_i s_i \alpha^{ik}$. Although in general it will not be possible to estimate all of the s_i, we show that we can estimate the sizes of those S_i that contribute. By Lemma 7, this will be enough to estimate F_k. The space complexity will be B up to poly $\left(\frac{1}{\epsilon} \ln n \ln m\right)$ terms.

The algorithm approximates s_i by restricting the input stream to randomly-chosen *substreams.* By this, we mean that it randomly samples subsets of items from $[m]$, and only considers those elements of $\mathcal{S}$ that lie in these subsets. More precisely, the algorithm creates $b = O(\ln \frac{m}{\epsilon' L})$ families of $R = O\left(\frac{1}{\epsilon' L^3} \ln\left(\ln m \log n\right)\right)$ substreams $\mathcal{S}_j^r$, for $j \in [b]$ and $r \in [R]$. We will assume that the constants in the big-Oh notation for both b and R are sufficiently large in several steps in the analysis. For each r, $\mathcal{S}_j^r$ will contain about $m/2^j$ randomly chosen items. If a class contributes, we can show there will be some j for which a good fraction of the maximum frequencies of the $\mathcal{S}_j^r$ come from the class. This fraction is used to estimate the class's size.

We separate the description of the algorithm from its helper algorithm Estimate used in Step 4. In Section 3.6 we will show how to choose the hash functions in Step 1 of the main algorithm F_k-Approximator below.

```
$\underline{F_k\textbf{-Approximator}}$ (stream $\mathcal{S}$):

  1. For $j \in [b]$ and $r \in [R]$, independently sample hash functions $h_j^r\colon [m] \to [2^j]$ using the pseudorandom technique described in Section 3.6.
  2. Let $\mathcal{S}_j^r$ be the restriction of $\mathcal{S}$ to those items $x$ for which $h_j^r(x) = 1$.
  3. For each $j, r$, compute $M_j^r = \mathsf{Max}(\mathcal{S}_j^r)$.
  4. For $i = \log n, \ldots, 0$,
      (a) Find the largest $j$ for which at least $RL(1-\epsilon')\epsilon'/8$ different $r$ satisfy $\alpha^i \leq M_j^r < \alpha^{i+1}$. If no such $j$ exists, set $\tilde{s}_i = 0$.
      (b) Otherwise, set $\mathsf{temp} = \mathsf{Estimate}(i, j, \sum_{l>i} \tilde{s}_l, M_j^1, \ldots, M_j^R)$.
      (c) If $\mathsf{temp} \leq L2^j$, set $\tilde{s}_i = \mathsf{temp}$, otherwise set $\tilde{s}_i = 0$.
  5. Output $\tilde{F}_k = \sum_i \tilde{s}_i \alpha^{ik}$.
```

We now describe Estimate. Define

$$r_{i,j} = (1 - (1 - 2^{-j})^{s_i}).$$

Estimate computes an approximation $\tilde{r}_{i,j}$ to $r_{i,j}$, and uses it to estimate s_i.

```
$\underline{\textbf{Estimate}}$ $(i, j, \sum_{l>i} \tilde{s}_l, M_j^1, \ldots, M_j^R)$:

  1. Set $A_{i,j} = \#r$ for which $\alpha^i \leq M_j^r < \alpha^{i+1}$.
  2. Compute $\tilde{r}_{i,j} = \frac{A_{i,j}}{R(1-2^{-j})^{\sum_{l>i} \tilde{s}_i}}$. If $\tilde{r}_{i,j} < 1$, output $\frac{\ln(1-\tilde{r}_{i,j})}{\ln(1-2^{-j})}$. Otherwise, output 0.
```

Lemma 11 *The output of* Estimate *is non-negative.*

Proof: Note that $0 \leq \tilde{r}_{i,j}$ and $j \geq 1$. If $\tilde{r}_{i,j} \neq 0$ and $\tilde{r}_{i,j} < 1$, then $\ln(1 - \tilde{r}_{i,j})$ and $\ln(1 - 2^{-j})$ are both negative, so the output of Estimate is positive. If $\tilde{r}_{i,j} = 0$, the output is 0, and finally if $\tilde{r}_{i,j} \geq 1$, the output of Estimate is 0 by definition. ■

3.3 Analysis

We first observe that if the $\tilde{s}_i$ are good approximations to the s_i, then $\tilde{F}_k$ is a good approximation to F_k. More precisely, define the event $\mathcal{E}$ as follows:

- for all i, $0 \leq \tilde{s}_i \leq (1+\epsilon)s_i$, and
- for all i, if S_i contributes, then $\tilde{s}_i \geq (1-\epsilon/(k+2))s_i$.

We claim that proving $\tilde{F}_k$ is a $(1 \pm \epsilon)$-approximation reduces to bounding the probability that event $\mathcal{E}$ occurs. More precisely,

Claim 12 *Suppose that with probability at least q, event $\mathcal{E}$ occurs. Then with probability at least q, we have $|\tilde{F}_k - F_k| \leq \epsilon F_k$.*

Proof: Assume $\mathcal{E}$ occurs. Put $\epsilon^* = \epsilon/(k+2)$. Then,

$$\tilde{F}_k = \sum_i \tilde{s}_i \alpha^{ik} \leq \sum_i (1+\epsilon) s_i \alpha^{ik} \leq (1+\epsilon)F_k.$$

For the other direction, write $\tilde{F}_k = \tilde{F}_k^C + \tilde{F}_k^{NC}$, where $\tilde{F}_k^C$ denotes the contribution to $\tilde{F}_k$ due to the contributing S_i. Then, assuming $\epsilon' \leq \epsilon^*$ by setting c to be sufficiently small,

$$\begin{aligned}
\tilde{F}_k^C &= \sum_{\text{contributing } S_i} \tilde{s}_i \alpha^{ik} \\
&\geq \frac{(1-\epsilon^*)}{\alpha^k} \sum_{\text{contributing } S_i} s_i \alpha^{(i+1)k} && \text{(using the definition of event } \mathcal{E} \text{ and } \epsilon^*) \\
&\geq \frac{(1-\epsilon^*)}{\alpha^k} F_k^C \\
&\geq \frac{(1-\epsilon^*)(1-\lambda\alpha^k)}{\alpha^k} F_k && \text{(using Lemma 7)} \\
&= (1-\epsilon^*)\left(\frac{1}{\alpha^k} - \lambda\right) F_k \\
&\geq \frac{(1-\epsilon^*)^2}{\alpha^k} F_k && \text{(using the definition of } \lambda \text{ and } \epsilon' \leq \epsilon^*) \\
&\geq (1-\epsilon^*)^2 (1-\epsilon^*)^k F_k && \text{(definition of } \alpha \text{ and } 1/(1+\epsilon') \geq (1-\epsilon') \geq (1-\epsilon^*)) \\
&\geq (1-\epsilon^*)^2 (1-k\epsilon^*) F_k && \text{(using Corollary 9)} \\
&\geq (1-(k+2)\epsilon^*) F_k && \text{(expanding, and dropping positive terms)} \\
&= (1-\epsilon) F_k. && \text{(using the definition of } \epsilon^*)
\end{aligned}$$

Noting that $\tilde{F}_k^{NC} \geq 0$ by Lemma 11. We conclude that with probability at least q, we have $|\tilde{F}_k - F_k| \leq \epsilon F_k$. ■

Our goal is now to prove the following theorem.

Theorem 13 *For sufficiently large m, n, with probability at least $8/9$, event $\mathcal{E}$ occurs.*

In the analysis we will assume the h_j^r are truly random functions. This assumption will be removed using the techniques of [44], and we wil describe how to do this in Section 3.6. It may also be possible to remove it by a slight modification of the inclusion-exclusion approach used in [9], though we do not attempt that approach here. We start by showing that with probability at least 8/9, a very natural event occurs. We then condition on this event in the remainder of the proof.

Observe that in Estimate,

$$\mathbf{E}[A_{i,j}] = R(1-2^{-j})^{\sum_{l>i} s_l}(1-(1-2^{-j})^{s_i}) = R(1-2^{-j})^{\sum_{l>i} s_l} r_{i,j}.$$

We define $\mathcal{F}$ to be the event that for all $A_{i,j}$,

- If $\mathbf{E}[A_{i,j}] \geq RL(1-\epsilon')\epsilon'/(16e)$, then $|A_{i,j} - \mathbf{E}[A_{i,j}]| \leq L\mathbf{E}[A_{i,j}]$.
- If $\mathbf{E}[A_{i,j}] \leq RL(1-\epsilon')\epsilon'/(16e)$, then $A_{i,j} < RL(1-\epsilon')\epsilon'/8$.

Lemma 14 $\Pr[\mathcal{F}] \geq 8/9$.

Proof: Fix any i, j for which $\mathbf{E}[A_{i,j}] \geq RL(1-\epsilon')\epsilon'/(16e)$. By Chernoff bounds [57],

$$\begin{aligned} \Pr[|A_{i,j} - \mathbf{E}[A_{i,j}]| \geq L\mathbf{E}[A_{i,j}]] &\leq e^{-\Theta(L^2\mathbf{E}[A_{i,j}])} \\ &= e^{-\Theta(L^3R(1-\epsilon')\epsilon')} \\ &= e^{-\Theta(L^3R\epsilon')} \\ &= e^{-\Theta(\ln(\ln m \log m))} \\ &= O\left(\frac{1}{\ln m \log n}\right). \end{aligned}$$

Now suppose $A_{i,j}$ is such that $\mathbf{E}[A_{i,j}] \leq RL(1-\epsilon')\epsilon'/(16e)$. Then, using the fact that $RL(1-\epsilon')\epsilon'/8 \geq 2e\mathbf{E}[A_{i,j}]$, we may apply another Chernoff bound (see, e.g., Exercise 4.1

of [57]) to conclude

$$\begin{aligned}\Pr[A_{i,j} \geq RL(1-\epsilon')\epsilon'/8] &\leq 2^{-RL(1-\epsilon')\epsilon'/8} \\ &\leq 2^{-\Theta(\ln(\ln m \log m))} \\ &= O\left(\frac{1}{\ln m \log n}\right),\end{aligned}$$

where we have used the fact that $\epsilon', L \leq 1$. The lemma follows by a union bound over all i and j, assuming the constant in the big-Oh notation defining R is sufficiently large. ■

In the remainder, we assume that $\mathcal{F}$ occurs.

Definition 15 *We say that* temp *is* **set** *if in step 4 of the main algorithm,* temp *is set to the output of* Estimate. *We say that $\tilde{s}_i$ is* **set** *if in step 4, $\tilde{s}_i$ is set to* temp.

For any stream $\mathcal{S}_j^r$, the probability that $\alpha^i \leq M_j^r < \alpha^{i+1}$ is precisely

$$p_{i,j} = (1-2^{-j})^{\sum_{l>i} s_l}(1-(1-2^{-j})^{s_i}) = (1-2^{-j})^{\sum_{l>i} s_l} r_{i,j}.$$

We would like to approximate s_i by approximating $p_{i,j}$. We start with a few propositions.

Proposition 16 *Suppose $s_i/L \leq 2^j$ and $0 < \gamma < 1$. Let* temp $= \frac{\ln(1-\tilde{r}_{i,j})}{\ln(1-1/2^j)}$.

- *If $\tilde{r}_{i,j} - r_{i,j} \leq \gamma r_{i,j}$, then* temp $- s_i \leq (\gamma + O(L))s_i$.
- *If $r_{i,j} - \tilde{r}_{i,j} \leq \gamma r_{i,j}$, then $s_i -$* temp $\leq (\gamma + O(L))s_i$.

Proof: For $|x| < 1$, we have the Taylor expansion $\ln(1+x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n+1} x^{n+1}$. Thus,

$$\begin{aligned}\ln(1-2^{-j}) &= \sum_{n=0}^{\infty} \frac{(-1)^n}{n+1}(-2^{-j})^{n+1} \\ &= \sum_{n=0}^{\infty} \frac{(-1)^{2n+1}}{n+1}(2^{-j})^{n+1} \\ &= -\sum_{n=0}^{\infty} \frac{(2^{-j})^{n+1}}{n+1} \\ &= -2^{-j} - \sum_{n=1}^{\infty} \frac{(2^{-j})^{n+1}}{n+1}.\end{aligned}$$

So,

$$2^{-j} \leq -\ln(1-2^{-j}) \leq 2^{-j} + \eta_1,$$

where $\eta_1 = O(1/4^j)$. Similarly,

$$\tilde{r}_{i,j} \le -\ln(1 - \tilde{r}_{i,j}) \le \tilde{r}_{i,j} + \eta_2,$$

where $\eta_2 = O(\tilde{r}_{i,j}^2)$. Recall that $r_{i,j} = (1 - (1 - 2^{-j})^{s_i})$. We want to deduce that $s_i 2^{-j} - (s_i 2^{-j})^2/2) \le r_{i,j} \le s_i 2^{-j}$. If $s_i = 0$ this is clear, since all three terms are 0 in this case. Otherwise, since s_i is an integer, $s_i \ge 1$. As $0 \le 1 - 2^{-j} < 1$, we may apply Lemma 8 to deduce that $s_i 2^{-j} - (s_i 2^{-j})^2/2 \le r_{i,j} \le s_i 2^{-j}$. Therefore,

$$\begin{aligned}
\mathsf{temp} &\le 2^j(\tilde{r}_{i,j} + \eta_2) && \text{(using the definition of } \mathsf{temp} \text{ and our bounds above)}\\
&\le 2^j(\tilde{r}_{i,j} + \tilde{r}_{i,j} O(r_{i,j})) && \text{(using the bound on } \tilde{r}_{i,j})\\
&\le 2^j r_{i,j}(1+\gamma)(1 + O(r_{i,j})) && \text{(using the bound on } \tilde{r}_{i,j})\\
&\le s_i(1+\gamma)(1 + O(r_{i,j})) && \text{(using that } r_{i,j} \le s_i 2^{-j})\\
&\le s_i(1 + \gamma + O(L)) && \text{(using that } r_{i,j} \le s_i 2^{-j} \le L).
\end{aligned}$$

Suppose now that $r_{i,j} - \tilde{r}_{i,j} \le \gamma r_{i,j}$. Then,

$$\begin{aligned}
\mathsf{temp} &\ge \frac{\tilde{r}_{i,j}}{2^{-j} + \eta_1} && \text{(using the definition of } \mathsf{temp} \text{ and our bounds above)}\\
&\ge \frac{2^j(1-\gamma) r_{i,j}}{1 + 2^j \eta_1} && \text{(using the new bound on } \tilde{r}_{i,j})\\
&\ge \frac{s_i(1-\gamma)(1 - s_i 2^{-j}/2)}{1 + 2^j \eta_1} && \text{(using that } r_{i,j} \ge s_i 2^{-j} - (s_i 2^{-j})^2/2)\\
&\ge \frac{s_i(1-\gamma)(1 - s_i 2^{-j}/2)}{1 + O(2^{-j})} && \text{(using that } \eta_1 = O(1/4^j))\\
&\ge s_i(1-\gamma)(1 - s_i 2^{-j}/2)(1 - O(2^{-j})) && \text{(using that } 1/(1 + O(2^{-j})) \ge 1 - O(2^{-j}))\\
&\ge s_i(1 - \gamma - O(L)) && \text{(using that } s_i 2^{-j} \le L \text{ and } 2^{-j} \le L \text{ since } s_i \ge 1),
\end{aligned}$$

where in the last step we may assume $s_i \ge 1$ since s_i is a non-negative integer, and if $s_i = 0$ the bound holds because of Lemma 11. ■

Proposition 17 *Suppose for some i and some $0 < \gamma < 1/3$, $\sum_{l>i} \tilde{s}_l \le (1+\gamma) \sum_{l>i} s_l$. If* temp *is set for S_i, then $\tilde{r}_{i,j} \le (1 + \gamma + L) r_{i,j}$.*

Proof: Put $\sigma = \sum_{l>i} s_l$, and $\sigma' = \sum_{l>i} \tilde{s}_l$. In $\mathsf{Estimate}$, $\tilde{r}_{i,j} = \frac{A_{i,j}}{R(1-2^{-j})^{\sigma'}}$. Also, since temp is set, at least $RL(1-\epsilon')\epsilon'/4$ different r satisfy $\alpha^i \le M_j^r < \alpha^{i+1}$. Therefore, since we are conditioning on event $\mathcal{F}$, we must have $\mathbf{E}[A_{i,j}] \geq RL(1-\epsilon')\epsilon'/(16e)$, where $\mathbf{E}[A_{i,j}] = R(1-2^{-j})^{\sigma} r_{i,j}$. Moreover, since we are conditioning on $\mathcal{F}$, this means that

$$|A_{i,j} - \mathbf{E}[A_{i,j}]| \leq L\mathbf{E}[A_{i,j}].$$

Using the definition of $A_{i,j}$ and dividing by $R(1-2^{-j})^{\sigma'}$, we obtain,

$$\tilde{r}_{i,j} \leq (1+L) r_{i,j} (1-2^{-j})^{\sigma-\sigma'}.$$

Using that $\sigma' \leq (1+\gamma)\sigma$,

$$\tilde{r}_{i,j} \leq (1+L)(1-2^{-j})^{-\gamma\sigma} r_{i,j}.$$

Moreover,

$$\begin{aligned}
(1+L)(1-2^{-j})^{-\gamma\sigma} r_{i,j} &\leq (1+L)e^{\gamma\sigma/2^j} r_{i,j} \quad \text{(using that } (1-x)^y \leq e^{-xy} \text{ for all reals } x,y) \\
&\leq (1+L)e^{\gamma/2} r_{i,j} \qquad \text{(using that } \sigma \leq 2^{j-1})
\end{aligned}$$

Now, using the Taylor expansion for $e^{\gamma/2}$ and the fact that $\gamma < 1/3$,

$$e^{\gamma/2} = \sum_{i=0}^{\infty} \frac{\left(\frac{\gamma}{2}\right)^i}{i!} \leq 1 + \sum_{i=1}^{\infty} \left(\frac{\gamma}{2}\right)^i = 1 + \frac{\gamma/2}{1-\gamma/2} \leq 1 + \frac{6\gamma}{10} = 1 + \frac{3\gamma}{5}.$$

Thus, since we may assume $L \leq 1/2$ by setting the parameter c sufficiently small,

$$(1+L)(1-2^{-j})^{-\gamma\sigma} r_{i,j} \leq (1+L)(1+3\gamma/5) r_{i,j} \leq (1+L+3\gamma/5+3\gamma/10) r_{i,j} \leq (1+L+\gamma) r_{i,j},$$

which completes the proof. ■

Our first lemma shows that we do not overestimate a class's size provided our estimates of previous classes were not overestimated.

Lemma 18 *Suppose for some i and some $0 < \gamma < 1/3$, $\sum_{l>i} \tilde{s}_l \leq (1+\gamma)\sum_{l>i} s_l$. Then $0 \leq \tilde{s}_i \leq s_i + (\gamma + O(L))s_i$.*

Proof: If either temp or $\tilde{s}_i$ is not set, then $\tilde{s}_i = 0$ and we're done. Otherwise, $\tilde{s}_i =$ temp and $2^j \geq \tilde{s}_i/L$. If $\tilde{s}_i < s_i$, since the output of Estimate is nonnegative by Lemma 11, we have $0 \leq \tilde{s}_i < s_i$. Otherwise, $2^j \geq \tilde{s}_i/L \geq s_i/L$. The conditions of the lemma together with

Proposition 17 imply that $\tilde{r}_{i,j} \leq (1+\gamma+L)r_{i,j}$. Since $2^j \geq s_i/L$ and $\tilde{s}_i = \mathsf{temp}$, Proposition 16 implies that $\tilde{s}_i \leq s_i + (\gamma + O(L))s_i$. ■

Let $\mu > 0$ be a constant for which, for any γ, $0 < \gamma < 1/3$, and $\sum_{l>i} \tilde{s}_l \leq (1+\gamma) \sum_{l>i} s_l$, $0 \leq \tilde{s}_i \leq s_i + (\gamma + \mu L)s_i$. Such a μ exists by Lemma 18, and we can assume $\mu \geq 1$, since this can only make the inequality $\tilde{s}_i \leq s_i + (\gamma + \mu L)s_i$ weaker.

Define $\beta_i = \mu(\log n + 1 - i)L$ for $i = \log n + 1, \ldots, 0$. We may assume for all i that $\beta_i + \mu L < \epsilon$. Indeed, for all i, $\beta_i \leq \beta_0 = \mu(\log n + 1)L = \mu\lambda \leq \mu\epsilon' = c\mu\epsilon$, using that $L = \lambda/(\log n + 1)$, $\lambda = \epsilon'/\alpha^k \leq \epsilon'$, and $\epsilon' = c\epsilon$. As $L \leq c\epsilon$, for all i, $\beta_i + \mu L \leq 2c\mu\epsilon$, which can be made less than ϵ by setting $c \leq 1/(2\mu)$. We will also assume that $\beta_i < 1/3$ for all i, which we can also achieve by setting c to be sufficiently small.

We now use the previous lemma to show that in step 4 of the algorithm, we do not overestimate the class sizes.

Lemma 19 *For $i = \log n, \ldots, -1$,*

$$0 \leq \sum_{l>i} \tilde{s}_l \leq (1 + \beta_{i+1}) \sum_{l>i} s_l.$$

Proof: For each i, the lower bound $0 \leq \sum_{l>i} \tilde{s}_l$ holds since for each l, $\tilde{s}_l \geq 0$ by Lemma 11. To prove the upper bound, we induct downwards on i. In the base case, $i = \log n$, we have $\sum_{l>i} \tilde{s}_i = \sum_{l>i} s_i = 0$, and so the claim holds. We show the upper bound for some $i < \log n$ assuming it holds for $i+1$.

$$\begin{aligned}
\sum_{l\geq i} \tilde{s}_l &= \tilde{s}_i + \sum_{l>i} \tilde{s}_l \\
&\leq \tilde{s}_i + \sum_{l>i} s_l + \beta_{i+1} \sum_{l>i} s_l && \text{(by the inductive hypothesis)} \\
&\leq s_i + (\beta_{i+1} + \mu L)s_i + \sum_{l>i} s_l + \beta_{i+1} \sum_{l>i} s_l && \text{(by Corollary 18)} \\
&\leq \sum_{l\geq i} s_l + \beta_i \sum_{l\geq i} s_l && \text{(using the definition of } \beta_i\text{),}
\end{aligned}$$

which completes the induction and the proof of the lemma. ■

The following corollary combines the two previous lemmas and shows the first condition of event $\mathcal{E}$ holds.

Corollary 20 *For all i, $0 \leq \tilde{s}_i \leq (1+\epsilon)s_i$.*

Proof: Lemma 19 implies that for all i, $\sum_{l>i} \tilde{s}_l \leq (1+\beta_{i+1}) \sum_{l>i} s_l$. Since $0 < \beta_{i+1} < 1/3$, Corollary 18 implies that for all i, $0 \leq \tilde{s}_i \leq s_i + (\beta_{i+1} + \mu L)s_i$. Since $\beta_{i+1} + \mu L \leq \epsilon$ for all i, the corollary follows. ■

The following lemma shows the second condition of event $\mathcal{E}$ holds.

Lemma 21 *For a sufficiently small choice of the parameter c, which may depend on k, for all i, if S_i contributes, then $\tilde{s}_i \geq (1 - \epsilon/(k+2))s_i$.*

Proof: Define $\sigma = \sum_{l>i} s_l$ and $\sigma' = \sum_{l>i} \tilde{s}_l$. Choose j' for which $s_i/(\epsilon' L) \leq 2^{j'} < 2s_i/(\epsilon' L)$. To see that this is possible, we just need to check that there is a value of $j' \in [b]$ for which $s_i/(\epsilon' L) \leq 2^{j'}$. But $s_i \leq m$ and $b = O(\ln \frac{m}{\epsilon' L})$, so by setting the constant in the big-Oh notation defining b to be sufficiently large, we can find such a j'.

Recall that $\mathbf{E}[A_{i,j'}] = R(1-2^{-j'})^\sigma(1-(1-2^{-j'})^{s_i})$. Since σ is a non-negative integer, $(1-2^{-j'})^\sigma \geq 1 - \sigma/2^{j'}$, since either $\sigma = 0$ and we have equality, or we may apply Corollary 9. Since S_i contributes, Lemma 6 implies that $s_i > L\sigma$. Thus, $(1-2^{-j'})^\sigma > 1 - s_i/(L2^{j'})$. By our choice of j', $s_i/(L2^{j'}) \leq \epsilon'$. Thus, $(1-2^{-j'})^\sigma > 1 - \epsilon'$.

Also, in particular, $s_i > 0$. Thus, applying Lemma 8, $(1-(1-2^{-j'})^{s_i}) \geq s_i/2^{j'} - (s_i/2^{j'})^2/2$. By our choice of j', $s_i/2^{j'} \leq \epsilon' L \leq 1$, so $s_i/2^{j'} - (s_i/2^{j'})^2/2 \geq s_i/2^{j'+1}$. By our choice of j', $s_i/2^{j'+1} > \epsilon' L/4$, and so $(1-(1-2^{-j'})^{s_i}) > \epsilon' L/4$.

Thus, $\mathbf{E}[A_{i,j'}] \geq RL(1-\epsilon')\epsilon'/4$. Since we are conditioning on event $\mathcal{F}$, and since $L \leq 1/2$ for a small enough setting of the parameter c,

$$A_{i,j'} \geq (1-L)\mathbf{E}[A_{i,j'}] \geq RL(1-\epsilon')\epsilon'/8.$$

Since F_k-$\mathsf{Approximator}$ chooses the largest value of j for which $A_{i,j} \geq RL(1-\epsilon')\epsilon'/8$, and we have just shown there is one such value of j (namely, $j = j'$), it follows that temp is set.

Let j be the value in F_k-$\mathsf{Approximator}$ for which temp is set. Then $A_{i,j} \geq RL(1-\epsilon')\epsilon'/8$, and since we are conditioning on $\mathcal{F}$, this means that $\mathbf{E}[A_{i,j}] \geq RL(1-\epsilon')\epsilon'/(16e)$, and further that $|A_{i,j} - \mathbf{E}[A_{i,j}]| \leq L\mathbf{E}[A_{i,j}]$.

Thus,

$$A_{i,j} \geq (1-L)\mathbf{E}[A_{i,j}] = (1-L)R(1-2^{-j})^\sigma r_{i,j},$$

and using the definition of $\tilde{r}_{i,j}$,

$$\tilde{r}_{i,j} \geq (1-L)(1-2^{-j})^{\sigma-\sigma'} r_{i,j}.$$

By Lemma 19, $\sigma' \geq 0$, and thus

$$\tilde{r}_{i,j} \geq (1-L)(1-2^{-j})^{\sigma} r_{i,j} \geq (1-L)(1-\epsilon') r_{i,j} = (1-L-\epsilon') r_{i,j}.$$

Note that $j \geq j'$, and thus $s_i/(\epsilon' L) \leq 2^j$ and so $s_i/L \leq 2^j$. Since also $0 < L + \epsilon' < 1$, we may apply Proposition 16 to infer that $\mathsf{temp} \geq (1 - \epsilon' - O(L))s_i$.

We now show that $\tilde{s}_i$ is set. This happens if $\mathsf{temp} \leq L2^j$. This in turn happens if $\mathsf{temp} \leq s_i/\epsilon'$. We may assume that $\epsilon' \leq 1/2$ by setting the parameter c to be sufficiently small, and therefore, this will happen if $\mathsf{temp} \leq 2s_i$. By Lemma 19, $0 \leq \sigma' \leq (1 + \beta_{i+1})\sigma$. Using the fact that $\beta_{i+1} < 1/3$ and temp is set, Proposition 17 implies that $\tilde{r}_{i,j} \leq (1 + \beta_{i+1} + L)r_{i,j}$. Since $j \geq j'$, $s_i/L \leq 2^j$, and therefore by Proposition 16, $\mathsf{temp} \leq (1 + \beta_{i+1} + O(L))s_i$. For a sufficiently small choice of the parameter c, $1 + \beta_{i+1} + O(L) \leq 2$, and therefore $\mathsf{temp} \leq 2s_i$, so that $\tilde{s}_i$ is set. Thus, $\tilde{s}_i = \mathsf{temp} \geq (1 - \epsilon' - O(L))s_i$.

The last observation is that $\epsilon' + O(L) \leq \epsilon/(k+2)$ for a sufficiently small choice of the parameter c. This completes the proof. ■

Theorem 22 *With probability at least* $8/9$*, we have* $|\tilde{F}_k - F_k| \leq \epsilon F_k$.

Proof: By Claim 12, this will follow if we show that with probability at least $8/9$, event $\mathcal{E}$ occurs. By Lemma 14, event $\mathcal{F}$ occurs with probability at least $8/9$. Conditioned on $\mathcal{F}$, Corollary 20 and Lemma 21 hold, and this shows that $\mathcal{E}$ occurs. Thus, $\Pr[\mathcal{E}] \geq 8/9$. ■

3.4 A 2-pass Algorithm

We instantiate Assumption 10 with the CountSketch algorithm of [21]. We review it in the next section, and then modify it for our application. We then describe how it can be used in place of the Max oracle in F_k-Approximator in the following section.

3.4.1 CountSketch

In [21], the authors solve a problem that they call FindApproxTop$(\mathcal{S}, k, \epsilon)$:

- Given: an input stream $\mathcal{S}$ of length n with elements drawn from $[m]$, an integer k, and a real number $\epsilon > 0$.
- With probability at least $1 - \eta$, output a list of k elements from $\mathcal{S}$ such that every element i in the list has frequency f_i larger than $(1-\epsilon)n_k$, where n_k is the kth largest frequency of an item in $\mathcal{S}$.

The following is Theorem 1 of [21]:

Theorem 23 *The* CountSketch *algorithm solves FindApproxTop*$(\mathcal{S}, k, \epsilon)$ *in space*[1]

$$O\left(\left(\frac{F_2(\mathcal{S})}{\epsilon n_k^2} \ln \frac{n}{\eta} + k\right) \ln m\right).$$

In fact, the authors prove an additional property of their algorithm: with probability at least $1 - \eta$, the list output by CountSketch satisfies the above and the additional property that every element x with $f_x > (1+\epsilon)n_k$ occurs in the list.

Theorem 23 is not quite in the form we need, since the space required may be quite large if n_k is small. For our application, we would like the space to be independent of n_k, which we can do if we relax the problem FindApproxTop. Although our modification of [21] is simple, we describe it here for completeness.

The algorithm is the same as described in [21]. There are $t = O(\ln \frac{n}{\eta})$ hash functions $h_1, h_2, \ldots, h_t$ from $[m]$ to $[B']$, for a parameter B', and t hash functions $s_1, s_2, \ldots, s_t$ from $[m]$ to $\{-1, 1\}$. The h_i and s_i are pairwise independent, and can be represented using only $O(\ln m)$ bits. We think of these hash functions as forming a $t \times B'$ array of counters. For an element $x \in [m]$, we will use the notation $c(h_i(x))$ to refer to the current count of the $h_i(x)$-th counter. Given a new stream element x, for each $i \in [t]$ we add to $c(h_i(x))$ the value $s_i(x)$. For each item $x \in [m]$, we can estimate its frequency f_x as follows:

$$f'_x = \text{median}_i\{c(h_i(x)) \cdot s_i(x)\}.$$

[1] In [21], the authors state that the space is $O(tB + k)$, where $t = \Theta(\ln \frac{n}{\eta})$ and $B = \frac{F_2(\mathcal{S})}{\epsilon n_k^2}$. This gives the same bound that we state, up to the $\ln m$ factor. The reason for the additional $\ln m$ factor is that we are looking at bit complexity, whereas [21] counted the number of machine words.

Indeed, observe that for any given i,

$$\begin{aligned}
\mathbf{E}[c(h_i(x)) \cdot s_i(x)] &= \mathbf{E}\left[\sum_{y \mid h_i(y)=h_i(x)} f_y s_i(y) s_i(x)\right] \\
&= \sum_{y \mid h_i(y)=h_i(x)} f_y \mathbf{E}[s_i(y) s_i(x)] \\
&= f_x s_i^2(x) \\
&= f_x,
\end{aligned}$$

where we have used the pairwise independence of the s_i. Thus $c(h_i(x)) \cdot s_i(x)$ is an unbiased estimator, and the median is taken to reduce the variance of the estimator.

We need the following key lemma from [21], which we state without proof.

Lemma 24 *(Lemma 4 of [21], restated) With probability at least $1 - \eta/2$, for all $\ell \in [n]$, if q is the ℓ-th item appearing in the stream and $n_q(\ell)$ is its frequency after the first ℓ items,*

$$|\mathit{median}\{c(h_i(q))s_i(q)\} - n_q(\ell)| \leq 8\sqrt{\frac{F_2}{B'}},$$

where $c(h_i(q))$ refers to the value of the $h_i(q)$-th counter after processing the first ℓ items.

Given an input parameter B, we would like to use the lemma above to accomplish the following two tasks with probability at least $1 - \eta$ for an input parameter $\eta > 0$:

1. Return all items x for which $f_x \geq \sqrt{\frac{F_2}{B}}$.
2. For all items x that are returned, return an estimate $\tilde{f}_x$ for which $f_x \leq \tilde{f}_x \leq (1+\kappa)f_x$, where $0 < \kappa \leq 1/2$ is an input parameter.

We accomplish these tasks as follows.

Our CountSketchFilter Algorithm

We modify CountSketch as follows. We refer to this new algorithm as CountSketchFilter. Set the parameter $B' = (4096/\kappa^2)B$. We keep a heap of the top B' items as we process the stream. When we encounter an item x in the stream, if x is already in the heap, we update the estimated frequency f'_x of x in the heap. If not, we compute f'_x and insert x into the heap if it is larger than any of the B' items already in the heap, or if there are less than B' items already in the heap.

In parallel, we also run the algorithm of [2] (see the remark after the proof of Theorem 2.2 of [2]) which gives a 2-approximation $\tilde{F}_2(\mathcal{S})$ of $F_2(\mathcal{S})$ using space $O((\ln m + \ln\ln n)\ln 1/\eta)$, where $\eta/2$ is the failure probability. We can assume by scaling that with probability at least $1-\eta/2$,

$$\frac{1}{2}F_2(\mathcal{S}) \le \tilde{F}_2(\mathcal{S}) \le 2F_2(\mathcal{S}).$$

After processing all of the items in $\mathcal{S}$, we remove all elements x from the heap for which

$$f'_x \le \frac{1}{2}\sqrt{\frac{\tilde{F}_2}{B}}.$$

For those x still in the heap, we define the estimate

$$\tilde{f}_x = f'_x + 8\sqrt{\frac{F_2}{B'}}.$$

Analysis

Let $\mathcal{E}$ be the event that the event in Lemma 24 occurs, and also $\frac{1}{2}F_2(\mathcal{S}) \le \tilde{F}_2(\mathcal{S}) \le 2F_2(\mathcal{S})$. By a union bound, $\Pr[\mathcal{E}] \ge 1-\eta$. Let us condition on $\mathcal{E}$ occurring.

First Task: Suppose x is such that $f_x \ge \sqrt{\frac{F_2}{B}}$. Then using that $B' > 4096B$,

$$8\sqrt{\frac{F_2}{B'}} \le \frac{1}{8}\sqrt{\frac{F_2}{B}}.$$

Thus,

$$f'_x \ge f_x - 8\sqrt{\frac{F_2}{B'}} \ge \frac{7}{8}\sqrt{\frac{F_2}{B}} > \frac{7}{8\sqrt{2}}\sqrt{\frac{\tilde{F}_2}{B}} > \frac{1}{2}\sqrt{\frac{\tilde{F}_2}{B}},$$

and so if x occurs in the heap, it will not be removed when we remove the elements y for which $f'_y \le \frac{1}{2}\sqrt{\frac{\tilde{F}_2}{B}}$.

Now by Lemma 24, for any two elements x and y whose frequencies f_x and f_y differ by more than $16\sqrt{\frac{F_2}{B'}}$, their estimates correctly identify the more frequent item. Suppose x is some item for which $f_x \ge \sqrt{\frac{F_2}{B}}$ and x is not in the heap of the top B' items. Note that x is clearly among the top B' items (in fact, the top $B'/4096$) in the stream. Moreover,

$$f_x \ge \sqrt{\frac{F_2}{B}} \ge 64\sqrt{\frac{F_2}{B'}}.$$

Therefore, there must be some y for which $f_y < \sqrt{\frac{F_2}{B'}}$ for which $f'_y \geq f'_x$. But

$$f'_y \leq f_y + 8\sqrt{\frac{F_2}{B'}} \leq 9\sqrt{\frac{F_2}{B'}}.$$

On the other hand,

$$f'_x \geq f_x - 8\sqrt{\frac{F_2}{B'}} \geq 56\sqrt{\frac{F_2}{B'}} > f'_y,$$

a contradiction. Thus, all x for which $f_x \geq \sqrt{\frac{F_2}{B}}$ are returned, and we have accomplished the first task.

Second Task: Using the definition of $\tilde{f}_x$, conditioning on event $\mathcal{E}$ occurring, if x is returned then $\tilde{f}_x \geq f_x$. Moreover, since x was returned we have

$$f'_x \geq \frac{1}{2}\sqrt{\frac{\tilde{F}_2}{B}} \geq \frac{1}{2\sqrt{2}}\sqrt{\frac{F_2}{B}},$$

and thus since $\kappa \leq 1/2$,

$$f_x \geq f'_x - 8\sqrt{\frac{F_2}{B'}} \geq \frac{1}{2\sqrt{2}}\sqrt{\frac{F_2}{B}} - \frac{\kappa}{8}\sqrt{\frac{F_2}{B}} \geq \frac{(2\sqrt{2}-1/2)}{8}\sqrt{\frac{F_2}{B}} > \frac{1}{4}\sqrt{\frac{F_2}{B}}.$$

Using the fact that $B' = 4096B/\kappa^2$, we have

$$16\sqrt{\frac{F_2}{B'}} \leq 16\kappa\sqrt{\frac{F_2}{4096B}} = \frac{\kappa}{4}\sqrt{\frac{F_2}{B}}.$$

Thus, since event $\mathcal{E}$ occurs,

$$\begin{aligned}\tilde{f}_x &= f'_x + 8\sqrt{\frac{F_2}{B'}} \\ &\leq f_x + 16\sqrt{\frac{F_2}{B'}} \\ &\leq f_x + \frac{\kappa}{4}\sqrt{\frac{F_2}{B}}.\end{aligned}$$

As $f_x \geq \frac{1}{4}\sqrt{\frac{F_2}{B}}$, it holds that $\tilde{f}_x \leq (1+\kappa)f_x$, and so we have accomplished the second task as well.

Space Complexity: We need $O(tB' \ln m)$ space to maintain the counters, and $O(B' \ln m)$ space to maintain the heap. We also need $O((\ln m + \ln\ln n)\ln 1/\eta)$ space to compute $\tilde{F}_2(\mathcal{S})$. Recall that $t = O(\ln \frac{n}{\eta})$ and $B' = \Theta(\frac{B}{\kappa^2})$.

The total space complexity is $O(\frac{B}{\kappa^2} \ln m \ln \frac{n}{\eta} + \ln\ln n \ln \frac{1}{\eta})$. We summarize our discoveries by the following theorem.

Theorem 25 *Let $0 < \eta < 1$ and $0 < \kappa < 1/2$. Given a stream $\mathcal{S}$ and a parameter B, there is an algorithm* CountSketchFilter *that with probability at least $1 - \eta$, succeeds in returning all items x for which $f_x^2 \geq F_2/B$, no items x for which $f_x^2 \leq F_2/(2B)$, and for all items x that are returned, there is also an estimate $\tilde{f}_x$ provided which satisfies $f_x \leq \tilde{f}_x \leq (1+\kappa)f_x$. The space complexity is $O(\frac{B}{\kappa^2} \ln m \ln \frac{n}{\eta} + \ln\ln n \ln \frac{1}{\eta})$.*

3.4.2 The new algorithm

We modify F_k-Approximator($\mathcal{S}$) as follows. In step 3 we invoke the algorithm CountSketchFilter of Theorem 25 on $\mathcal{S}_j^r$ with parameters

$$B = O(bRm^{1-2/k}), \quad \eta = O(1/(bR)), \quad \kappa = 1/2.$$

We obtain lists L_j^r of candidate maxima for $\mathcal{S}_j^r$. In parallel, we compute a 2-approximation $\tilde{F}_2(\mathcal{S}_j^r)$ to $F_2(\mathcal{S}_j^r)$ for each j and r. To do this, we run the algorithm (see the remark after the proof of Theorem 2.2 of [2]) which gives a 2-approximation $\tilde{F}_2(\mathcal{S}_j^r)$ using space $O((\ln m + \ln\ln n)\ln 1/\eta)$, where η is the failure probability. We assume $\eta \leq 1/(9bR)$, which can be made by setting the constant in the big-Oh above to be less than 1/9.

Then, before invoking steps 4 and 5, we make a second pass over $\mathcal{S}$ and compute the true frequency of each element of each L_j^r. Since the L_j^r are relatively small, this is efficient. We then prune the lists L_j^r by removing all items with squared frequency less than $2\tilde{F}_2(\mathcal{S}_j^r)/B$. We set M_j^r to be the maximum frequency among the remaining items in L_j^r, if there is at least one remaining item. Otherwise we set $M_j^r = 0$. At the end of the second pass, we proceed with steps 4 and 5 as before.

3.4.3 Conditioning on a few natural events

To analyze the modified algorithm, we start by conditioning on the following event:

$$\mathcal{G}_1 \overset{\text{def}}{=} \forall j, r,\ \mathsf{CountSketchFilter} \text{ succeeds.}$$

Lemma 26 $\Pr[\mathcal{G}_1] \geq 8/9$.

Proof: This follows by our choice of η and a union bound over all $j \in [b]$ and $r \in [R]$. ■

We also condition on the event:

$$\mathcal{G}_2 \overset{\text{def}}{=} \forall j, r,\ F_2(\mathcal{S}_j^r) \leq \frac{9bR \cdot F_2(\mathcal{S})}{2^j}.$$

Lemma 27 $\Pr[\mathcal{G}_2] \geq 8/9$.

Proof: We have $\mathbf{E}[F_2(\mathcal{S}_j^r)] = F_2(\mathcal{S})/2^j$, so

$$\Pr\left[F_2(\mathcal{S}_j^r) \geq \frac{9bRF_2(\mathcal{S})}{2^j}\right] \leq \frac{1}{9bR},$$

by Markov's inequality. By a union bound over all $j \in [b]$ and $r \in [R]$, we have

$$\Pr\left[\exists j, r \mid F_2(\mathcal{S}_j^r) \geq \frac{9bRF_2(\mathcal{S})}{2^j}\right] \leq \frac{1}{9}.$$

■

Finally, define the event:

$$\mathcal{G}_3 \overset{\text{def}}{=} \forall j, r,\ \frac{F_2(\mathcal{S}_j^r)}{2} \leq \tilde{F}_2(\mathcal{S}_j^r) \leq 2F_2(\mathcal{S}_j^r).$$

Lemma 28 $\Pr[\mathcal{G}_3] \geq 8/9$.

Proof: We are running the F_2-approximator of [2] in space $O((\ln m + \ln n)\ln 1/\eta)$, where $\eta = 1/(9bR)$ is the failure probability. A union bound gives the lemma. ■

Combining Lemma 14 with the previous three lemmas and a union bound, we have

Lemma 29 $\Pr[\mathcal{G}_1 \wedge \mathcal{G}_2 \wedge \mathcal{G}_3] \geq 2/3$.

In the remainder, we assume that $\mathcal{G}_1 \wedge \mathcal{G}_2 \wedge \mathcal{G}_3$ occurs. We need a few technical claims.

Claim 30 *For all j, r, either M_j^r is set to the maximum frequency in $\mathcal{S}_j^r$, or to* 0.

Proof: Suppose for some j, r the pruned list L_j^r contains at least one element x of $\mathcal{S}_j^r$, but does not contain the most frequent element y of $\mathcal{S}_j^r$. Then, since $\mathcal{G}_1$ occurs,

$$f_x^2 < f_y^2 < \frac{F_2(\mathcal{S}_j^r)}{B}.$$

Since $\mathcal{G}_3$ occurs,

$$f_x^2 < \frac{F_2(\mathcal{S}_j^r)}{B} \leq \frac{2\tilde{F}_2(\mathcal{S}_j^r)}{B},$$

which contradicts the fact that x wasn't pruned. ■

Corollary 31 *For all i, j, $\mathbf{E}[A_{i,j}] \leq Rp_{i,j}$.*

Proof: Recall that in Section 3.3, we had $\mathbf{E}[A_{i,j}] = R(1 - 2^{-j})^{\sigma} r_{i,j} = Rp_{i,j}$. After instantiating Assumption 10 with CountSketchFilter, Claim 30 implies that M_j^r is what it was before or 0. Thus, for each i, $A_{i,j}$ cannot increase, and so after instantiating with CountSketchFilter, $\mathbf{E}[A_{i,j}] \leq Rp_{i,j}$. ■

At this point we may define the event $\mathcal{F}$ as before, since the definition of $\mathcal{F}$ doesn't depend on what the actual values of the $\mathbf{E}[A_{i,j}]$ are. All that matters is that for any fixed i, and any $j \in [b]$, as we range over the $r \in [R]$, the events $\alpha^i \leq M_j^r < \alpha^{i+1}$ are i.i.d. Bernoulli random variables, and $A_{i,j}$ is the sum of indicators for these events. This allows us to apply Chernoff bounds.

Thus, as before we define $\mathcal{F}$ to be the event that for all $A_{i,j}$,

- If $\mathbf{E}[A_{i,j}] \geq RL(1-\epsilon')\epsilon'/(16e)$, then $|A_{i,j} - \mathbf{E}[A_{i,j}]| \leq L\mathbf{E}[A_{i,j}]$.
- If $\mathbf{E}[A_{i,j}] \leq RL(1-\epsilon')\epsilon'/(16e)$, then $A_{i,j} < RL(1-\epsilon')\epsilon'/8$.

Lemma 14 still holds with the same proof, and we have $\Pr[\mathcal{F}] \geq 8/9$. By a union bound, $\Pr[\mathcal{F} \wedge \mathcal{G}_1 \wedge \mathcal{G}_2 \wedge \mathcal{G}_3] \geq 5/9$. In the remainder, we assume that these four events all occur.

3.4.4 Walking through the previous proofs

Most of the arguments in Section 3.3 go through as before, though there are a few differences. We carefully guide the reader through the modified arguments. First, note that the lemmas and corollary in Section 3.1 continue to hold, since they are independent of the algorithm. Also, the only lemma in Section 3.2, Lemma 11 that states that the output of Estimate is non-negative, continues to hold since this property is independent of the inputs to Estimate, and the algorithm Estimate has not changed. Now we turn to the analysis in Section 3.3. The definitions of $p_{i,j}$ and $r_{i,j}$ remain unchanged.

We define $\mathcal{E}$ as before. Claim 12 that reduces correctness to bounding the probability that event $\mathcal{E}$ occurs, continues to hold. As observed in Section 3.4.3, we may define $\mathcal{F}$ as before and Lemma 14 continues to hold. The definition of **set** for temp and $\tilde{s}_i$, as given by Definition 15, remains the same. Proposition 16, relating the estimate on the $\tilde{r}_{i,j}$ to the value of temp, is the same as before. Indeed, this property does not depend on the fact that the random variables $A_{i,j}$ are now distributed differently.

We now reprove Proposition 17. We will need to make use of Corollary 31.

Proposition 32 *Suppose for some i and some $0 < \gamma < 1/3$, $\sum_{l>i} \tilde{s}_l \leq (1+\gamma)\sum_{l>i} s_l$. If* temp *is set for S_i, then $\tilde{r}_{i,j} \leq (1+\gamma+L)r_{i,j}$.*

Proof: Put $\sigma = \sum_{l>i} s_l$, and $\sigma' = \sum_{l>i} \tilde{s}_l$. In Estimate, $\tilde{r}_{i,j} = \frac{A_{i,j}}{R(1-2^{-j})^{\sigma'}}$. Also, since temp is set, at least $RL(1-\epsilon')\epsilon'/4$ different r satisfy $\alpha^i \leq M_j^r < \alpha^{i+1}$. Therefore, since we are conditioning on event $\mathcal{F}$, we must have $\mathbf{E}[A_{i,j}] \geq RL(1-\epsilon')\epsilon'/(16e)$, where $\mathbf{E}[A_{i,j}] \leq Rp_{i,j} = R(1-2^{-j})^{\sigma} r_{i,j}$ by Corollary 31. Moreover, since we are conditioning on $\mathcal{F}$, this means that

$$|A_{i,j} - \mathbf{E}[A_{i,j}]| \leq L\mathbf{E}[A_{i,j}].$$

Thus, using the definition of $A_{i,j}$ and these bounds, we have

$$\tilde{r}_{i,j} R(1-2^{-j})^{\sigma'} = A_{i,j} \leq (1+L)\mathbf{E}[A_{i,j}] \leq (1+L)R(1-2^{-j})^{\sigma} r_{i,j}.$$

Dividing by $R(1-2^{-j})^{\sigma'}$ we obtain,

$$\tilde{r}_{i,j} \leq (1+L) r_{i,j} (1-2^{-j})^{\sigma-\sigma'}.$$

Using that $\sigma' \leq (1+\gamma)\sigma$,

$$\tilde{r}_{i,j} \leq (1+L)(1-2^{-j})^{-\gamma\sigma} r_{i,j}.$$

Moreover,

$$\begin{aligned}(1+L)(1-2^{-j})^{-\gamma\sigma}r_{i,j} &\leq (1+L)e^{\gamma\sigma/2^j}r_{i,j} \quad \text{(using that } (1-x)^y \leq e^{-xy} \text{ for all reals } x,y\text{)} \\ &\leq (1+L)e^{\gamma/2}r_{i,j} \qquad \text{(using that } \sigma \leq 2^{j-1}\text{)}\end{aligned}$$

Now, using the Taylor expansion for $e^{\gamma/2}$ and the fact that $\gamma < 1/3$,

$$e^{\gamma/2} = \sum_{i=0}^{\infty} \frac{\left(\frac{\gamma}{2}\right)^i}{i!} \leq 1 + \sum_{i=1}^{\infty}\left(\frac{\gamma}{2}\right)^i = 1 + \frac{\gamma/2}{1-\gamma/2} \leq 1 + \frac{6\gamma}{10} = 1 + \frac{3\gamma}{5}.$$

Thus, since we may assume $L \leq 1/2$ by setting the parameter c sufficiently small,

$$(1+L)(1-2^{-j})^{-\gamma\sigma}r_{i,j} \leq (1+L)(1+3\gamma/5)r_{i,j} \leq (1+L+3\gamma/5+3\gamma/10)r_{i,j} \leq (1+L+\gamma)r_{i,j},$$

which completes the proof. ■

Now, Lemma 18 continues to hold, but in the proof we replace Proposition 17 with Proposition 32. We may then define $\mu > 0$ and the β_i as before. Moreover, Lemma 19 concerning the inductive approximation of the $\tilde{s}_i$ continues to hold. Consequently, Corollary 20 continues to hold, that is, the first part of event $\mathcal{E}$ continues to hold.

The main modification of this section is to Lemma 21, which we now reprove.

Lemma 33 *For a sufficiently small choice of the parameter c, which may depend on k, for all i, if S_i contributes, then $\tilde{s}_i \geq (1-\epsilon/(k+2))s_i$.*

Proof: For each $j \in [b]$ and $r \in [R]$, let U_j^r be the indicator random variable for the event that $\alpha^i \leq M_j^r < \alpha^{i+1}$. Note that the U_j^r are independent and identically distributed, and that $A_{i,j} = \sum_{r\in[R]} U_j^r$. We start by showing that for any j' for which $2^{j'} \geq s_i/(\epsilon' L)$, the values $U_{j'}^r$ are distributed just as under Assumption 10, that is, they are i.i.d. Bernoulli$(p_{i,j'})$.

For this, it suffices to show that for an S_i that contributes and j' for which $2^{j'} \geq s_i/(\epsilon' L)$,

$$\forall r \in [R], \text{ if } \alpha^i \leq \mathsf{Max}(\mathcal{S}_{j'}^r) < \alpha^{i+1}, \text{ then } \mathsf{Max}^2(\mathcal{S}_{j'}^r) \geq 4F_2(\mathcal{S}_{j'}^r)/B, \tag{3.1}$$

where $B = O(bRm^{1-2/k})$ is the parameter in $\mathsf{CountSketchFilter}$. Then, by Theorem 25, and the fact that $\mathsf{CountSketchFilter}$ succeeds because $\mathcal{G}_1$ occurs, the item x realizing $\mathsf{Max}(\mathcal{S}_{j'}^r)$ will occur in $L_{j'}^r$. Moreover, this will imply $\mathsf{Max}^2(\mathcal{S}_{j'}^r) \geq 2\tilde{F}_2(\mathcal{S}_{j'}^r)/B$, since $\mathcal{G}_3$ occurs, and therefore x will not be pruned from $L^r_{j'}$. In the second pass, the algorithm will then learn that $\alpha^i \leq f_x < \alpha^{i+1}$, and $U^r_j = 1$. Conversely, if we do not have $\alpha^i \leq \mathsf{Max}(\mathcal{S}^r_{j'}) < \alpha^{i+1}$, then by Claim 30, $U^r_j = 0$.

To show (3.1), it suffices to show that for all $x \in S_i$ and all $r \in [R]$, $f_x^2 \geq 4F_2(\mathcal{S}^r_j)/B$. Indeed, if it happens that $\alpha^i \leq \mathsf{Max}(\mathcal{S}^r_{j'}) < \alpha^{i+1}$, then this will guarantee that $\mathsf{Max}^2(\mathcal{S}^r_{j'}) \geq 4F_2(\mathcal{S}^r_{j'})/B$.

To show this, by Hölder's inequality (a generalization of the Cauchy-Schwartz inequality),

$$F_2 = \sum_{i=1}^m f_i^2 \cdot 1 \leq \left(\sum_{i=1}^m f_i^k\right)^{2/k} \left(\sum_{i=1}^m 1\right)^{1-2/k} = F_k^{2/k} m^{1-2/k}. \tag{3.2}$$

Since S_i contributes, $f_x^k s_i \geq \alpha^{ik} s_i > LF_k$ by definition. Therefore,

$$\begin{aligned}
f_x^2 s_i^{2/k} &\geq F_k^{2/k} L^{2/k} \\
&\geq \frac{F_2 L^{2/k}}{m^{1-2/k}} && \text{(using (3.2))} \\
&\geq \frac{2^{j'} F_2(\mathcal{S}^r_{j'}) L^{2/k}}{9bRm^{1-2/k}} && \text{(since } \mathcal{G}_2 \text{ occurs)} \\
&\geq \frac{2^{j'+1} F_2(\mathcal{S}^r_{j'}) L^{2/k}}{B} && \text{(for a large enough constant in the big-Oh for } B) \\
&\geq \frac{2^{j'+1} F_2(\mathcal{S}^r_{j'}) L}{B}. && \text{(since } L^{2/k} \geq L \text{ for } k \geq 2)
\end{aligned}$$

Since $k \geq 2$, we have $2^{j'+1}/s_i^{2/k} \geq 2^{j'+1}/s_i$. Moreover, our choice of j' is that $2^{j'} \geq s_i/(\epsilon' L) \geq 2s_i/L$, where the second inequality follows for a small enough setting of the parameter c (to ensure $\epsilon' \leq 1/2$). Thus, $2^{j'+1}/s_i^{2/k} \geq 2^{j'+1}/s_i \geq 4/L$. It follows that

$$f_x^2 \geq \frac{2^{j'+1} F_2(\mathcal{S}^r_{j'}) L}{s_i^{2/k} B} \geq \frac{4F_2(\mathcal{S}^r_{j'}) L}{LB} \geq \frac{4F_2(\mathcal{S}^r_{j'})}{B},$$

as desired.

The rest of the proof is now the same as that of Lemma 21. Indeed, using the fact that the $U^r_{j'}$ are distributed just as under Assumption 10, for any j' for which $2^{j'} \geq s_i/(\epsilon' L)$, the $A_{i,j'}$ are also distributed just as under Assumption 10. Thus, an identical analysis shows that temp is set with some value of j for which $2^j \geq s_i/(\epsilon' L)$, and $\mathsf{temp} \geq (1 - \epsilon' - O(L))s_i$.

To see that $\tilde{s}_i$ is set, we use the same analysis except instead of applying Proposition 21 to infer that $\tilde{r}_{i,j} \leq (1+\beta_{i+1}+L)r_{i,j}$, we use Proposition 33. This completes the proof. ■

Up to the pseudorandom technique to be described in Section 3.6, we now have a 2-pass algorithm for F_k.

Theorem 34 *Assuming there is access to an infinite random string, for any $\epsilon, \delta > 0$, there is a 2-pass algorithm which (ϵ, δ)-approximates F_k in space $\tilde{O}(m^{1-2/k})\mathrm{poly}(1/\epsilon)\ln(1/\delta)$.*

Proof: With probability at least 5/9, $\mathcal{F} \wedge \mathcal{G}_1 \wedge \mathcal{G}_2 \wedge \mathcal{G}_3$ occurs. In this case, Corollary 20 continues to hold, that is, the first part of event $\mathcal{E}$ holds. Moreover, Lemma 33 still holds, and so $\mathcal{E}$ occurs. Thus, by Claim 12, $\Pr[|\tilde{F}_k - F_k| \leq \epsilon F_k] \geq 5/9$. Taking the median of $O(\ln 1/\delta)$ independent repetitions makes the output an (ϵ, δ)-approximation to F_k.

The algorithm is a 2-pass algorithm. In our total space calculation, we will suppress a $\mathrm{poly}(\ln\ln m, \ln\ln n, \ln 1/\epsilon)$ factor. Since we have an infinite random string, the random functions chosen in step 1 of F_k-Approximator do not contribute to the space complexity. F_k-Approximator invokes CountSketchFilter $O(bR\ln 1/\delta)$ times, which by Theorem 25 uses space $O(B\ln m \ln n/\eta))$ in each invocation, up to a $\ln\ln n$ factor. Here, $B = O(bRm^{1-2/k})$ and $\eta = O(1/(bR))$. This has total space

$$O\left(\left(bR\ln\frac{1}{\delta}\right)bRm^{1-2/k}\ln m \ln\left(nbR\right)\right) = O\left(m^{1-2/k}b^2R^2\ln m\ln(nbR)\ln\frac{1}{\delta}\right).$$

The algorithm also invokes the 2-approximation algorithm for F_2 of [2] a total of $bR\ln 1/\delta$ times. This has total space

$$O\left(bR(\ln m + \ln\ln n)\ln\frac{1}{\eta}\ln\frac{1}{\delta}\right) = O\left(bR(\ln m + \ln\ln n)\ln(bR)\ln\frac{1}{\delta}\right).$$

The second pass, where we compute the true frequencies of the elements in the L_j^r, can be done with space

$$O\left(B\left(bR\ln\frac{1}{\delta}\right)\ln m\right) = O\left(B\left(bR\ln\frac{1}{\delta}\right)\ln m\right) = O\left(m^{1-2/k}b^2R^2\ln\frac{1}{\delta}\right).$$

Thus, the total space is,

$$O\left(m^{1-2/k}b^2R^2\ln m\ln(nbR)\ln\frac{1}{\delta}\right) + O\left(bR(\ln m + \ln\ln n)\ln(bR)\ln\frac{1}{\delta}\right).$$

Note that since k is constant, α^k is constant, and so we can bound the parameter L as follows. Recall that we have defined $\log n$ to be $\log_{1+\alpha} n$, which is just $O\left(\frac{\ln n}{\epsilon}\right)$.

$$L = \Theta\left(\frac{\lambda}{\log n}\right) = \Theta\left(\frac{\epsilon}{\alpha^k \log n}\right) = \Theta\left(\frac{\epsilon}{\log n}\right) = \Theta\left(\frac{\epsilon}{\log_{1+\alpha} n}\right) = \Theta\left(\frac{\epsilon \ln(1+\alpha)}{\ln n}\right) = \Theta\left(\frac{\epsilon^2}{\ln n}\right).$$

Now, recall that $b = O(\ln \frac{m}{\epsilon' L})$ and $R = O\left(\frac{1}{\epsilon' L^3} \ln\left(\ln m \log n\right)\right)$. Using our bound on L,

$$b = O\left(\ln m + \ln \frac{1}{\epsilon} + \ln \ln n\right),$$

and

$$R = O\left(\frac{\ln^3 n}{\epsilon^7}\left(\ln \ln m + \ln \ln n + \ln \frac{1}{\epsilon}\right)\right).$$

Suppressing a poly$(\ln \ln m, \ln \ln n, \ln 1/\epsilon)$ factor, $b = O(\ln m)$ and $R = O(\frac{\ln^3 n}{\epsilon^7})$. Moreover, $\ln b$ and $\ln R$ are $O(1)$ if we suppress such a factor. Thus, suppressing such a factor, the total space is:

$$O\left(m^{1-2/k} b^2 R^2 \ln m \ln n \ln \frac{1}{\delta}\right) + O\left(bR(\ln m + \ln \ln n) \ln \frac{1}{\delta}\right).$$

The first term dominates, and so up to a poly$(\ln \ln m, \ln \ln n, \ln 1/\epsilon)$ factor, the space is

$$O\left(m^{1-2/k} b^2 R^2 \ln m \ln n \ln \frac{1}{\delta}\right) = O\left(\frac{m^{1-2/k}}{\epsilon^{14}} \ln^3 m \ln^7 n \ln \frac{1}{\delta}\right),$$

and the theorem follows. ■

3.5 The 1-pass Algorithm

In this section we show how to remove the second pass from the algorithm in the previous section, and obtain a 1-pass algorithm. We will again assume the existence of an infinite random string, and remove this in Section 3.6.

Recall that, in the previous section, the algorithm assumed an oracle that we refer to as Partial Max. For each stream $\mathcal{S}_j^r$ and a certain value of a threshold T (namely, for a given j and r we set the threshold $T = 2\tilde{F}_2(\mathcal{S}_j^r)/B$), the oracle reported the element $i^* \in [m]$ with the maximum value of f_{i^*}, but if and only if $f_{i^*} \geq T$. The second pass was needed in order to compute the exact frequencies of the candidate maxima, and check if (a) any of them was greater than T and (b) find the element with the maximum frequency.

We reduce the need of the second pass by transforming the algorithm in such a way that, if we replace each frequency f_i by its estimation $\tilde{f}_i$ provided by CountsketchFilter, the behavior of the transformed algorithm is, with high probability, the same as in the original algorithm.

Let $\kappa = o(1)$ be a function to be determined. Since there are $O(bR \ln 1/\delta)$ invocations of CountSketchFilter, if $\eta = o(1/(bR \ln 1/\delta))$ in the premise of Theorem 25, then with probability $1 - o(1)$, for all invocations of CountSketchFilter and for all items i reported by CountSketchFilter in each invocation,

$$f_i \leq \tilde{f}_i \leq (1 + \kappa) f_i.$$

We assume this event occurs in the rest of this section. The transformations are as follows.

Shifted boundaries: We modify the algorithm so that the thresholds T passed to Partial Max are multiplied by some value $y \in [1, \alpha)$, and the frequency boundaries α^i are multiplied (consistently) by some value $x \in [1/\alpha, 1)$. The algorithm and its analysis of correctness can be easily adapted to this case. This requires natural adjustments, such as replacing each term α^i in step 5 of F_k-Approximator in the estimator by $(x\alpha)^i$. The other modifications are to the calls CountSketchFilter($\mathcal{S}_j^r$) in step 3 of F_k-Approximator.

The reason for this modification is that, if we choose x, y independently at random from a near-uniform distribution, then, for *fixed* i, the outcome of comparisons, say, $f_i \geq yT$ and $\tilde{f}_i \geq yT$, is likely to be the same, as long as f_i and $\tilde{f}_i$ differ by a small multiplicative factor.

Class reporting: We replace the Partial Max oracle by another oracle, called Rounded Partial Max, which does the following: for a permutation $\pi : [m] \rightarrow [m]$, it reports i with the smallest value of $\pi[i]$ such that f_i is in the same frequency class as f_{i^*}, but only if $f_{i^*} \geq yT$. The algorithm and its analysis remain unchanged, since it only performs comparisons of f_i with values $x\alpha^j$ and yT.

In the following we assume π is choosen uniformly at random from the set of all permutations of $[m]$. Later, we show how to reduce the required randomness by choosing π from a family of 2-approximate min-wise independent functions [43].

Approximate frequencies: Now we consider the following key modification to Rounded Partial Max. The modification replaces the use of the exact frequencies f_i by their approximations $\tilde{f}_i$. Specifically, we replace each comparison $f_i \geq v$ by a comparison $\tilde{f}_i \geq v(1+\kappa)$. Note that $\tilde{f}_i \geq v(1+\kappa)$ implies $f_i \geq v$. Call the resulting oracle Approximate Max.

Let i' be such that $\pi(i')$ is the smallest value of $\pi(i)$ over all i for which f_i is in the same frequency class as f_{i^*}, and let $[x\alpha^{k'}, x\alpha^{k'+1})$ be the frequency class containing f_{i^*}. If we invoke Rounded Partial Max with parameters j, r, we denote the values i', i^*, k', T by $i'(j,r)$, $i^*(j,r)$, $k'(j,r)$ and $T(j,r)$. Consider the following event $\mathcal{B}(j,r)$:

1. $f_{i'(j,r)} \geq x\alpha^{k'(j,r)}(1+\kappa)$, and

2. $f_{i^*(j,r)} \geq yT(j,r) \Rightarrow f_{i^*(j,r)} \geq yT(j,r)(1+\kappa)$

This event allows us to use the approximate frequencies provided by CountSketchFilter in lieu of the actual frequencies. The following is easy to verify.

Claim 35 *Fix the random bits of F_k-Approximator. If $\mathcal{B}(j,r)$ holds for all j,r, then the behaviors of all invocations of Rounded Partial Max and Approximate Max, respectively, are exactly the same. Therefore, the output of F_k-Approximator using either oracle is the same.*

Now it suffices to show that each $\mathcal{B}(j,r)$ holds with good probability. We show that this holds even if π is chosen from a family of 2-approximate min-wise permutations i.e., such that for any $A \subset [m], a \in [m] - A$, we have $\Pr_\pi[\pi(a) < \min_{b \in A} \pi(b)] \leq \frac{2}{|A|+1}$. Such families exist and are constructible from only $O(\log^2 m)$ random bits [43].

Lemma 36 *There is a distribution of x so that, for any $0 < \zeta < 1$, if $0 < \kappa < 1 < \alpha = 1 + \epsilon' < 2$, then for a fixed pair (j,r) the probability of*

$$f_{i'(j,r)} < x\alpha^{k'(j,r)}(1+\kappa)$$

is at most $O(\kappa/\epsilon' \cdot \log m \cdot 1/\zeta + \zeta)$. Moreover, this fact holds even if π is chosen at random from a family of 2-wise functions.

Proof: For simplicity, we are going to omit the pair (j,r) in the notation below.

For a parameter β, define $I'(\beta) = |\{i : \beta \leq f_i\}|$, and $I'' = |\{i : \beta \leq f_i < \beta(1+\kappa)\}|$. Consider $\beta = x\alpha^{k'}$. Observe that the event we are concerned in this lemma occurs if $i' \in I''(\beta)$.

We choose $x = (1+\kappa)^s/\alpha$, where s is chosen uniformly at random from $\{0, \ldots, \log_{1+\kappa} \alpha\}$. Note that $\log_{1+\kappa} \alpha = \Theta(\epsilon'/\kappa)$.

Observe that the value of β ranges in $[f_{i^*}/\alpha, \ldots, f_{i^*}]$. Also, observe that each value in that interval is assumed at most once (that is, for only one value of x).

Claim 37 *For any $0 < \zeta < 1$, the number of different values of β such that $I''(\beta)/I'(\beta) \geq \zeta$ is at most $\log_{1+\zeta}(m+1) + 1 = O(\log m/\zeta)$.*

Proof: Assume this is not the case, and let $\beta_1, \ldots, \beta_t$, $t \geq \log_{1+\zeta}(m+1) + 1$ be the different values of β such that $I''(\beta)/I'(\beta) \geq \zeta$, in decreasing order.

Since $I'(\beta) = I''(\beta) + I'(\beta(1+\kappa))$, we have that for each β_i, $I'(\beta_i) \geq \frac{1}{1-\zeta} I'(\beta_i(1+\kappa)) \geq (1+\zeta)I'(\beta(1+\kappa))$. Moreover, the value of $I'(\beta)$ does not decrease as β decreases. It follows that $I'(\beta_t) \geq (1+\zeta)^{t-1} > m$, which is a contradiction. ■

Thus, for the value β induced by a random choice of x, the probability that $I''(\beta)/I'(\beta) \geq \zeta$ is at most $O(\kappa/\epsilon' \log_{1+\zeta} m)$. The probability that i' belongs to $I''(\beta)$ is at most ζ (if π is a truly random permutation) or at most 2ζ (if π is chosen from 2-approximate min-wise independent family). ■

The other part of the event $\mathcal{B}(j, r)$ can be handled in a similar way. By setting $\zeta = \sqrt{\kappa/\epsilon' \cdot \log m}$ we get,

Lemma 38 *The probability that* some *event $\mathcal{B}(j, r)$ does not hold is at most*

$$O(Rb\sqrt{\kappa/\epsilon' \cdot \log m})$$

which is $o(1)$ for small enough $\kappa = 1/(1/\epsilon' + \log m)^{O(1)}$.

3.6 Reducing the Randomness

It remains to show that the functions h_j^r used by our algorithm can be generated in small space. To this end, we will use Nisan's pseudorandom generator (PRG), as done in [44].

Specifically, observe that the state maintained by algorithm consists of several counters c (as in the CountSketch algorithm). Each counter is identified by indices j, r and i. Given a new element x, the counter performs the following operation: if $h_j^r(x) = 1$ and $g(x) = i$, then $c = c + Y_x$.

Therefore, we can use Lemma 3 of [44], to show that the random numbers $h_j^r(0), \ldots h_j^r(m-1)$ can be generated by a PRG using only $O(\log^2(nm))$ truly random bits as a seed. Thus, the total number of random bits we need to store is bounded by the total storage used by our 1-pass algorithm times $O(\log(nm))$.

Wrapping Up: At this point we have proven the correctness and efficiency of our 1-pass algorithm in the restricted cash register model where the input consists of pairs of the form $(x, 1)$. Note that the algorithm can easily be modified to handle the cash register model where the input consists of pairs of the form (x, z) where z is a positive integer. Indeed, instead of adding 1 to the appropriate counters, we simply add z. That is, CountSketchFilter is easily seen to be implementable in the (unrestricted) cash register model. In fact, F_k-Approximator is even correct in the turnstile model. This again follows from the fact that CountSketch and CountSketchFilter can be implemented in the turnstile model. That is, when seeing a pair $(x, z) \in [m] \times \mathcal{R}$, we find the $h_i(x)$-th counter for each $i \in [t]$. We then add to $c(h_i(x))$ the value $z \cdot s_i(x)$, for each value of i. The analysis proceeds as before.

Theorem 39 *There is a 1-pass $\tilde{O}(m^{1-2/p})\text{poly}(1/\epsilon)$-space streaming algorithm for $(1 \pm \epsilon)$-approximating the L_p norm of an m-dimensional vector presented as a data stream for any $p \geq 2$. This also holds for the frequency moments F_p.*

Chapter 4

Lower Bounds for L_p Distance and Frequency Moments

4.1 Communication Complexity

Here we review a few notions from communication complexity. We closely follow the presentation in the book by Kushilevitz and Nisan [53]. The interested reader may consult [53] for more detail.

Let $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ be a Boolean function. We will consider two parties, Alice and Bob, receiving x and y respectively, who try to compute $f(x, y)$. For non-trivial f, Alice and Bob will need to communicate with each other to evaluate $f(x, y)$. The communication is carried out according to some fixed protocol Π, which depends only on f.

In each round of the protocol, Π must determine whether the protocol terminates or if not, which player should speak next. If the protocol terminates, it must specify an answer (that is, $f(x, y)$). This information must depend only on the bits communicated thus far, as this is the only information common to both parties. Also, if it is a party's turn to speak, the protocol must specify what the party sends, and this must depend only on the communication thus far and the input of the party.

We are only interested in the amount of communication between Alice and Bob. We thus allow Alice and Bob to be computationally unbounded. The cost of a protocol Π on input (x, y) is the number of bits communicated by Π on (x, y). The cost of a protocol Π is the maximal cost of Π over all inputs (x, y). This is formalized as follows.

Definition 40 *A protocol Π over domain $X \times Y$ with range Z is a binary tree where each internal node v is labeled either by a function $a_v : X \to \{0,1\}$ or by a function $b_v : Y \to \{0,1\}$, and each leaf is labeled with an element $z \in Z$.*

The value of a protocol Π on input (x,y) is the label of the leaf reached by starting from the root, and walking on the tree. At each internal node v labeled by a_v walking left if $a_v(x) = 0$ and right if $a_v(x) = 1$, and at each internal node labeled by b_v walking left if $b_v(y) = 0$ and right if $b_v(y) = 1$. The cost of Π on input (x,y) is the length of the path taken on input (x,y). The cost of the protocol Π is the height of the tree.

Definition 41 *For a function $f : X \times Y \to Z$, the deterministic communication complexity of f, denoted $D(f)$, is the minimum cost of Π, over all protocols Π that compute f.*

Note that we always have $D(F) \leq \min(\log_2 |X|, \log_2 |Y|) + 1$.

We will be mostly interested in the setting where both parties have access to random coins. Here, Alice has access to a random string r_A and Bob has access to a random string r_B. Here the random strings are of arbitrary length, and are chosen independently according to some probability distribution. Now when we look at the protocol tree, Alice's node are labeled by arbitrary functions of x and r_A, while Bob's nodes are labeled by arbitrary functions of y and r_B. Every combination of x, y, r_A, and r_B determines a leaf of the protocol tree with a specified output. The difference now is that for some inputs (x,y) and some choices of r_A and r_B, the protocol may output the wrong answer. We say that a protocol Π computes a function f with δ-error if for every (x,y),

$$\Pr[\Pi(x,y) = f(x,y)] \geq 1 - \delta.$$

Our measure of communication cost is as follows.

Definition 42 *The worst case running time of a randomized protocol Π on input (x,y) is the maximum number of bits communicated for any choice of the random strings, r_A and r_B. The worst case cost of Π is the maximum, over all inputs (x,y) of the worst case running time of Π on (x,y).*

We note that it is also possible to define the cost with respect to average, rather than worst-case random strings, though we do not take this up here. We can now defined the randomized communication complexity of a function.

Definition 43 *Let $f : X \times Y \rightarrow \{0,1\}$ be a function. Then for $0 < \delta < 1/2$, $R_\delta(f)$ is the minimum worst case cost of a randomized protocol that computes f with error δ. We denote $R(f) = R_{1/3}(f)$.*

Note that for any two constants δ, δ' with $0 < \delta, \delta' < 1/2$, we have $R_\delta(f) = \Theta(R_{\delta'}(f))$, so it is w.l.o.g. that we fix $R(f) = R_{1/3}(f)$. Indeed, given a protocol with error probability $\delta > \delta'$, we may repeat it $O(\log 1/\delta')$ times, using independently chosen random strings each time, and take the majority output. This operation defines a new protocol which errs at most an δ' fraction of the time (for every input).

In many of the protocols we consider, we have an even simpler model. Namely, Alice computes some function $A(x)$ of x and sends the result to Bob. Bob then attempts to compute $f(x, y)$ from $A(x)$ and y. Here only one message is sent, and it is from Alice to Bob.

Definition 44 *Let $f : X \times Y \rightarrow \{0,1\}$ be a function. Then for $0 < \delta < 1/2$, the δ-error 1-way randomized communication complexity $R_\delta^{1-way}(f)$ is the minimum worst case cost of a randomized protocol that computes f with error δ, in which only a single message is sent from Alice to Bob. We denote $R^{1-way}(f) = R_{1/3}^{1-way}(f)$.*

As defined, in a randomized protocol Alice and Bob each have random strings but they do not have any random bits in common. We could have instead allowed the parties to have a "public" coin, so that both parties can see the results of a single series of random coin flips. More precisely, there is a common random string r (chosen according to some distribution) and in the protocol tree Alice's communication corresponds to function of x and r and Bob's communication corresponds to functions of y and r. This can be viewed as a distribution $\{\Pi_r\}_r$ over deterministic protocols.

Definition 45 *A public coin protocol is a probability distribution over deterministic protocols. The success probability of a public coin protocol on input (x, y) is the probability of choosing a deterministic protocol, according to the distribution of r, that computes $f(x, y)$ correctly. We let $R_\delta^{pub}(f)$ be the minimum cost of a public coin protocol that computes f with an error of at most δ on every input (x, y). We denote $R^{pub}(f) = R_{1/3}^{pub}(f)$.*

Clearly $R_\delta^{pub}(f) \leq R_\delta(f)$ since the parties can use a public coin to define their individual private coins. In fact, there is a close converse due to Newman.

Theorem 46 *([62]) Let $f : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ be a function. For every $\delta, \delta' > 0$, $R_{\delta+\delta'}(f) \leq R^{pub}_{\delta}(f) + O(\log m + \log 1/\delta')$. This continues to hold if both protocols are 1-way.*

Throughout we have been discussing protocols which, for every input (x, y), err with probability at most δ, where the probability is only over the random strings of the protocol. It is sometimes natural to look at protocols which err on a certain fraction of inputs.

Definition 47 *Let μ be a probability distribution on $X \times Y$. The (μ, δ)-distributional communication complexity of f, $D_{\mu,\delta}(f)$, is the cost of the best deterministic protocol that gives the correct answer for f on at least a $1-\delta$ fraction of all inputs in $X \times Y$, weighted by μ. We denote $D_\mu(f) = D_{\mu,1/3}(f)$.*

A famous theorem, known as Yao's Minimax Principle, relates $D_{\mu,\delta}(f)$ to $R^{pub}_{\delta}(f)$.

Theorem 48 *([71])*

$$R^{pub}_{\delta}(f) = \max_{\mu} D_{\mu,\delta}(f).$$

Furthermore, the same relationship holds for 1-way protocols.

One of these directions is quite easy to prove, as we now show.

Lemma 49 *$D_{\mu,\delta}(f) \leq R^{pub}_{\delta}(f)$ for any distribution μ, and this also holds if both protocols are 1-way.*

Proof: If Π is a randomized protocol realizing $R_\delta(f)$, then for all inputs x, y, we have $\Pr[\Pi(x, y) \neq f(x, y)] \leq \delta$. This means that $\Pr_{(x,y)\sim\mu}[\Pr_{\text{coins of }\Pi}[\Pi(x, y) \neq f(x, y)]] \leq \delta$, and by switching the order of summations we have

$$\Pr_{\text{coins of }\Pi}[\Pr_{(x,y)\sim\mu}[\Pi(x, y) \neq f(x, y)]] = \mathbf{E}_{\text{coins of }\Pi} \Pr_{(x,y)\sim\mu}[\Pi(x, y) \neq f(x, y)] \leq \delta.$$

So we can fix a certain random string of Π, making it a deterministic protocol realizing $D_{\mu,\delta}(f)$. ■

We will also need to extend the above definitions to handle relations rather than just functions. Here we have a relation $T \subseteq X \times Y \times Z$. The communication problem is the following: Alice is given $x \in X$ and Bob $y \in Y$, and their task is to find some $z \in Z$ for which $(x, y, z) \in T$.

Definition 50 *A protocol* Π *computes a relation* T *if for every legal input* $(x, y) \in X \times Y$*, the protocol reaches a leaf marked by a value* z *such that* $(x, y, z) \in T$*. The deterministic communication complexity of a relation* T*, denoted* $D(T)$*, is the number of bits sent on the worst case input (legal or illegal) by the best protocol that computes* T*. The definitions of* $R_\delta(T)$*,* $R_\delta^{1-way}(T)$*,* $R_\delta^{pub}(T)$*, and* $D_{\mu,\delta}(T)$ *are similar.*

We will mostly consider a special type of relation, known as a *promise problem.* Here $Z = \{0, 1\}$ and we have disjoint subsets P_{yes} and P_{no} of $X \times Y$. The relation T is such that if $(x, y) \in P_{yes}$, then $(x, y, 1) \in T$ but $(x, y, 0) \notin T$. If $(x, y) \in P_{no}$, then $(x, y, 0) \in T$ but $(x, y, 1) \notin T$. If $(x, y) \notin P_{yes} \cup P_{no}$, then both $(x, y, 0)$ and $(x, y, 1)$ are in T.

The following is our generalization of Lemma 49 to promise problems.

Lemma 51 *Let* T *be a promise problem with sets* P_{yes} *and* P_{no}*. Let* $g : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ *be any function for which for all* $(x, y) \in P_{yes}$*,* $g(x, y) = 1$*, and for all* $(x, y) \in P_{no}$*,* $g(x, y) = 0$*. For any distribution* μ *on* $\mathcal{X} \times \mathcal{Y}$*,*

$$R_\delta^{pub}(T) \geq D_{\mu,\delta'}(g),$$

where $\delta' = \delta + \mu(\bar{P}_{yes} \cap \bar{P}_{no})$*. This also holds if both protocols are 1-way.*

Proof: Let Π be a randomized protocol realizing $R_\delta(T)$. Then for all inputs $(x, y) \in P_{yes}$, $\Pr[\Pi(x, y) = 1] \geq 1 - \delta$, and for all inputs $(x, y) \in P_{no}$, $\Pr[\Pi(x, y) = 0] \geq 1 - \delta$. Thus,

$$\begin{aligned} & \Pr_{(x,y)\sim\mu}[\Pr_{\text{coins of }\Pi}[\Pi(x, y) \neq g(x, y)] \mid (x, y) \in P_{yes} \cup P_{no}] \\ = & \Pr_{(x,y)\sim\mu}[\Pr_{\text{coins of }\Pi}[(x, y, \Pi(x, y)) \notin T] \mid (x, y) \in P_{yes} \cup P_{no}] \\ \leq & \ \delta. \end{aligned}$$

Thus,

$$\begin{aligned} \Pr_{(x,y)\sim\mu}[\Pr_{\text{coins of }\Pi}[\Pi(x, y) \neq g(x, y)]] & \leq \Pr_{(x,y)\sim\mu}[\Pr_{\text{coins of }\Pi}[(x, y, \Pi(x, y)) \notin T] \mid (x, y) \in P_{yes} \cup P_{no}] \\ & + \Pr_{(x,y)\sim\mu}[(x, y) \notin P_{yes} \cup P_{no}] \\ & \leq \delta + \mu(\bar{P}_{yes} \cap \bar{P}_{no}). \end{aligned}$$

By switching the order of summations we have

$$\Pr_{\text{coins of }\Pi}[\Pr_{(x,y)\sim\mu}[\Pi(x,y) \neq g(x,y)]] = \mathbf{E}_{\text{coins of }\Pi} \Pr_{(x,y)\sim\mu}[\Pi(x,y) \neq g(x,y)] \leq \delta + \mu(\bar{P}_{yes} \cap \bar{P}_{no}).$$

We can fix a random string of Π, making it deterministic with $\Pr_{(x,y)\sim\mu}[\Pi(x,y) \neq g(x,y)] \leq \delta + \mu(\bar{P}_{yes} \cap \bar{P}_{no})$. ■

In everything that follows we will assume that protocols are public-coin, since this will only change the communication complexity by an additive $O(\log m)$, which will be negligible. We will omit the superscript *pub* for simplicity.

4.2 The Gap Hamming Problem

In our lower bounds, we are particularly concerned with the c-Gap-Hamming distance promise problem, denoted *c-GH*.

Definition 52 *In the c-GH problem* $X = \{0,1\}^m$, $Y = \{0,1\}^m$, $P_{yes} = \{(x,y) \mid \Delta(x,y) \geq m/2 + c\sqrt{m}\}$, *and* $P_{no} = \{(x,y) \mid \Delta(x,y) \leq m/2 - c\sqrt{m}\}$. *Here,* $c > 0$ *is allowed to depend on* m.

This relation captures the communication complexity of *approximating* the Hamming distance. Indeed, consider the following relation $T_\epsilon \subseteq \{0,1\}^m \times \{0,1\}^m \times \{0,1,2,\ldots,m\}$ defined as follows: $(x,y,z) \in T_\epsilon$ iff $(1-\epsilon)\Delta(x,y) \leq z \leq (1+\epsilon)\Delta(x,y)$.

Lemma 53 $D(T_\epsilon) \geq D(\epsilon\sqrt{m}\text{-}GH)$, $R(T_\epsilon) \geq R(\epsilon\sqrt{m}\text{-}GH)$, $R^{1-way}(T_\epsilon) \geq R^{1-way}(\epsilon\sqrt{m}\text{-}GH)$, $D_\mu(T_\epsilon) \geq D_\mu(\epsilon\sqrt{m}\text{-}GH)$.

Proof: Let Π be a protocol for the $\epsilon\sqrt{m}$-*GH* problem which behaves as follows. Let Π' be a protocol for T_ϵ. On input (x,y), Π invokes Π' to obtain an answer z. If $z > m/2$, $\Pi(x,y) = 1$, else $\Pi(x,y) = 0$. The (deterministic, randomized, randomized 1-way, distributional) communication cost of Π is that same as that of Π'.

To analyze the correctness of Π, observe that for the $\epsilon\sqrt{m}$-*GH* problem we have $P_{yes} = \{(x,y) \mid \Delta(x,y) \geq m/2 + \epsilon m\}$ and $P_{no} = \{(x,y) \mid \Delta(x,y) \leq m/2 - \epsilon m\}$. Suppose $(x,y) \in P_{yes}$. In this case $(x,y,z) \in T_\epsilon$ iff $(1-\epsilon)\Delta(x,y) \leq z \leq (1+\epsilon)\Delta(x,y)$. In particular $(1-\epsilon)(m/2 + \epsilon m) = m/2 + \epsilon m - \epsilon m/2 - \epsilon^2 m = m/2 + (1-2\epsilon)\epsilon m/2 > m/2$. Now suppose $(x,y) \in P_{no}$. Then if $(x,y,z) \in T_\epsilon$, $z \leq (1+\epsilon)\Delta(x,y) \leq (1+\epsilon)(m/2 - \epsilon m) =$

$m/2 - \epsilon m + \epsilon m/2 - \epsilon^2 m < m/2 - \epsilon m/2 < m/2$. Thus, Π is correct for $\epsilon\sqrt{m}$-GH whenever Π' is correct for T_ϵ. The lemma follows. ■

For general L_p norms for arbitrary real numbers $p \geq 0$, we may consider the relation $T_{p,\epsilon} \subseteq \{0,1,2,\dots r\}^m \times \{0,1,2,\dots,r\}^m \to \{0,1,2,\dots,\lceil mr^p \rceil\}$ defined as follows: $(x,y,z) \in T_{p,\epsilon}$ iff $(1-\epsilon)||x-y||_p^p \leq z \leq (1+\epsilon)||x-y||_p^p$. Observe that if x,y have coordinates either 0 or 1, then $||x-y||_p^p = \Delta(x,y)$. We thus have,

Lemma 54 *For all $p \geq 0$, $D(T_{p,\epsilon}) \geq D(\epsilon\sqrt{m}\text{-}GH)$, $R(T_{p,\epsilon}) \geq R(\epsilon\sqrt{m}\text{-}GH)$, $R^{1-way}(T_{p,\epsilon}) \geq R^{1-way}(\epsilon\sqrt{m}\text{-}GH)$, $D_\mu(T_{p,\epsilon}) \geq D_\mu(\epsilon\sqrt{m}\text{-}GH)$. Here we take $T_{0,\epsilon} = T_\epsilon$, i.e., we consider the Hamming distance when $p = 0$.*

Let $c > 0$ be an arbitrary constant. We need the following reduction from c-GH on inputs of length $m' = c^2/\epsilon^2$, denoted c-$GH_{m'}$ to $\epsilon\sqrt{m}$-GH on inputs of size m, denoted $\epsilon\sqrt{m}$-GH_m. Here, w.l.o.g., we assume that c^2/ϵ^2 is an integer. We will only be interested in the randomized 1-way communication complexity, though the statement also holds for the other notions of complexity we have considered.

Lemma 55 *For any $\epsilon \geq \frac{c}{\sqrt{m}}$, $R^{1-way}(\epsilon\sqrt{m}\text{-}GH_m) \geq R^{1-way}(1\text{-}GH_{m'})$.*

Proof: Given an input (x,y) to c-$GH_{m'}$, we create x' by replacing each bit x_i of x with a block of $\epsilon^2 m/c^2$ bits all of value x_i. Do the same operation to obtain y' from y. Then $|x'| = |y'| = m$. If $\Delta(x,y) \geq m'/2 + c\sqrt{m'}$, then $\Delta(x',y') \geq m/2 + \epsilon m$. If $\Delta(x,y) \leq m'/2 - c\sqrt{m'}$, then $\Delta(x',y') \leq m/2 - \epsilon m$. Thus, running a randomized 1-round protocol for $\epsilon\sqrt{m}$-GH_m and outputting whatever it outputs yields a randomized 1-round protocol for c-$GH_{m'}$ with the same properties. ■

Combining the previous two lemmas,

Corollary 56 *For any $\epsilon \geq \frac{c}{\sqrt{m}}$ and any $p \geq 0$, $R^{1-way}(T_{p,\epsilon}) \geq R^{1-way}(1\text{-}GH_{m'})$.*

We will later see that $R^{1-way}(1\text{-}GH_{m'}) = \Omega(1/\epsilon^2)$, and thus the restriction that $\epsilon \geq \frac{c}{\sqrt{m}}$ is necessary. Indeed, the trivial protocol in which Alice just sends her input to Bob has communication m.

Next we state some bounds on the communication complexity of c-GH.

Theorem 57 *For any constant $c > 0$, $D(c\text{-}GH) = \Omega(m)$.*

Proof: We reduce from the equality function $EQ : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ defined by $EQ(x,y) = 1$ iff $x = y$. We need the fact that there is a constant $\zeta > 0$ and an encoding $C : \{0,1\}^{m/6} \to \{0,1\}^{m/2+\zeta m/2}$ such that for any distinct $x, y \in \{0,1\}^{m/6}$, $\Delta(x,y) > \zeta m$ (see, e.g., [67]). We consider the setting $|x| = |y| = m/6$. It is known that in this case $D(EQ) = \Omega(m)$ (see, e.g., [53]).

Given an instance (x,y) of EQ with $|x| = |y| = m/6$, the parties compute $C(x)$ and $C(y)$, respectively, which are of length $m/2 + \zeta m/2$. Let $m' = m/2 - \zeta m/2$. Alice creates $x' \in \{0,1\}^m$ by padding x with m' trailing zeros. Bob creates $y' \in \{0,1\}^m$ by padding y with m' trailing ones. If $x = y$, then $\Delta(x', y') = m' = m/2 - \zeta m/2 < m/2 - c\sqrt{m}$ for any constant c. If $x \neq y$, then $\Delta(x', y') > m' + \zeta m = m/2 + \zeta m/2 > m/2 + c\sqrt{m}$ for any constant c. Thus, any deterministic protocol Π which solves c-GH for any constant c on inputs of size m, also solves EQ on inputs of size $m/6$. Thus, $D(c\text{-}GH) = \Omega(m)$. ■

Theorem 58 *([5]) For any constant $0 < c < 1/\sqrt{3}$, $R(c\text{-}GH) = \Omega(\sqrt{m})$.*

Proof: We reduce from the disjointness function $DIS : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ defined by $DIS(x,y) = 1$ iff there is an $i \in [m]$ for which $x_i = y_i = 1$. It is known [50, 64] that $R(DIS) = \Omega(m)$. Actually, it is known that even for the restriction to the case when $wt(x) = wt(y) = m/4$, the disjointness function has $\Omega(m)$ randomized complexity [64]. This is what we'll reduce from.

Let $m' = 2c\sqrt{m}$, which we assume is a multiple of 4. Given an instance (x,y) of DIS with $|x| = |y| = m'$ and $wt(x) = wt(y) = m'/4$, the parties do the following. Alice first replaces each bit x_i of x with a block of $\sqrt{m}/(2c) - 1$ bits, all of which equal x_i. She then pads this with a block of $2c\sqrt{m}$ zeros, to obtain x'. Similarly, Bob replaces each bit y_i of y with a block of $\sqrt{m}/(2c) - 1$ bits, all of which equal y_i. He then pads this with a block of $2c\sqrt{m}$ ones, to obtain y'. Note that $|x'| = |y'| = m$.

If $DIS(x,y) = 0$, then $\Delta(x', y') = (m'/2) \cdot (\sqrt{m}/(2c) - 1) + 2c\sqrt{m} = m/2 - c\sqrt{m} + 2c\sqrt{m} = m/2 + c\sqrt{m}$. If $DIS(x,y) = 1$, then $\Delta(x', y') = (m'/2 - 2) \cdot (\sqrt{m}/(2c) - 1) + 2c\sqrt{m} = m/2 + c\sqrt{m} - \sqrt{m}/c + 2 \leq m/2 + c\sqrt{m} - 3c\sqrt{m} + 2$ since $c < 1/\sqrt{3}$. The latter is at most $m/2 - c\sqrt{m}$ for sufficiently large m. Thus, any randomized protocol Π which solves c-GH on inputs of size m must have communication complexity $\Omega(\sqrt{m})$. ■

The main open question here is whether $R(c\text{-}GH) = \Omega(m)$ for constant c. We will prove this is true for 1-round protocols. This is of major importance to streaming algorithms, where the streaming algorithm is usually assumed to only have one pass over the data stream. Here we formalize the connection between this problem and streaming algorithms for approximating F_p, as discovered by the author [68] (and earlier for F_0 by Indyk and the author [46]). We show the lower bound holds even in the cash register model for data streams.

Theorem 59 *For all $p \neq 1$ and any constants $\epsilon, \delta > 0$, $S_{\epsilon',\delta}(F_p) \geq R^{1-way}(1\text{-}GH_{m'}) - O(\log 1/\epsilon)$, where $\epsilon' = \epsilon/(2^{p-1}-1)$ and $m' = 1/\epsilon^2$.*

Proof: We use an (ϵ', δ)-approximation algorithm for F_p in the streaming model to build a protocol for $1\text{-}GH_{m'}$. Alice chooses an arbitrary stream $\mathbf{a}_x$ with characteristic vector $x \in \{0,1\}^{m'}$, and Bob chooses an arbitrary stream $\mathbf{a}_y$ with characteristic vector $y \in \{0,1\}^{m'}$. Note that the universe the stream elements are drawn from is $[m']$, and the meaning is that i occurs in $\mathbf{a}_x$ iff $x_i = 1$ (and similarly for $\mathbf{a}_y$ and y_i).

Let M be an (ϵ, δ) F_p-approximation algorithm for some constant $p \neq 1$. Alice runs M on $\mathbf{a}_x$. When M terminates, she transmits the state S of M to Bob along with $wt(x)$. Bob feeds both S and $\mathbf{a}_y$ into his copy of M. Let $\tilde{F_p}$ be the output of M. The claim is that $\tilde{F_p}$ along with $wt(x)$ and $wt(y)$ can be used to solve $T_{p,\epsilon}$. We first decompose F_p:

$$\begin{aligned} F_p(\mathbf{a}_x \circ \mathbf{a}_y) &= \sum_{i \in [m]} f_i^p \\ &= 2^p wt(x \wedge y) + 1^p \Delta(x,y) \\ &= 2^{p-1}(wt(x) + wt(y) - \Delta(x,y)) + \Delta(x,y) \\ &= 2^{p-1}(wt(x) + wt(y)) + (1 - 2^{p-1})\Delta(x,y). \end{aligned}$$

and thus, for $p \neq 1$.

$$\Delta(x,y) = \frac{2^{p-1}}{2^{p-1}-1}(wt(x) + wt(y)) - \frac{F_p(\mathbf{a}_x \circ \mathbf{a}_y)}{2^{p-1}-1}.$$

For $p \neq 1$, define the quantity E to be

$$E = \frac{2^{p-1}(wt(x) + wt(y))}{1 - 2^{p-1}} - \frac{M(\mathbf{a}_x \circ \mathbf{a}_y)}{1 - 2^{p-1}}.$$

If $E > m'/2$, Bob decides that $\Delta(x,y) > m'/2 + \sqrt{m'}$, and otherwise Bob decides that $\Delta(x,y) < m'/2 - \sqrt{m'}$. We now analyze correctness. Suppose M outputs a $(1 \pm \epsilon')$ approximation to $F_p(\mathbf{a}_x \circ \mathbf{a}_y)$.

Case 1: Suppose $\Delta(x,y) > m'/2 + \sqrt{m'}$. Then

$$E \geq \frac{2^{p-1}(wt(x)+wt(y))}{1-2^{p-1}} - (1+\epsilon')\frac{F_p(\mathbf{a}_x \circ \mathbf{a}_y)}{2^{p-1}-1} = \Delta(x,y) - \epsilon' \frac{F_p(\mathbf{a}_x \circ \mathbf{a}_y)}{2^{p-1}-1}.$$

Now, $F_p(\mathbf{a}_x \circ \mathbf{a}_y) \geq \Delta(x,y)$, and thus

$$E \geq \Delta(x,y) - \epsilon' \frac{\Delta(x,y)}{2^{p-1}-1} = (1-\epsilon)\Delta(x,y) = \left(1 - \frac{1}{\sqrt{m'}}\right)\Delta(x,y) > \frac{m'}{2}.$$

Case 2: Suppose $\Delta(x,y) < m'/2 - \sqrt{m'}$. Then

$$E \leq \frac{2^{p-1}(wt(x)+wt(y))}{1-2^{p-1}} - (1-\epsilon')\frac{F_p(\mathbf{a}_x \circ \mathbf{a}_y)}{2^{p-1}-1} = \Delta(x,y) + \epsilon' \frac{F_p(\mathbf{a}_x \circ \mathbf{a}_y)}{2^{p-1}-1}.$$

Now, $F_p(\mathbf{a}_x \circ \mathbf{a}_y) \leq m'$, and thus

$$E \leq \Delta(x,y) + \epsilon' \frac{m'}{2^{p-1}-1} = \Delta(x,y) + \epsilon m' \leq \Delta(x,y) + \sqrt{m'} < \frac{m'}{2}.$$

It follows that the parties can decide $R_\delta^{1-way}(1\text{-}GH_{m'})$ with communication at most $S_{\epsilon',\delta}(F_p) + \lceil \log m' \rceil$, where the additive $O(\log m') = O(\log 1/\epsilon)$ is due to the transmission of $wt(x)$. ■

4.3 The Randomized 1-way Lower Bound

Here we prove that $R^{1-way}(1\text{-}GH) = \Omega(m)$. The original proofs of this fact are due to Indyk and the author [46, 68]. The proof presented here is based on a reduction from the function $IND : \{0,1\}^m \times [m] \to \{0,1\}$, where $IND(x,i) = x_i$. The proof is quite similar to the simplified proof due to Bar-Yossef, Jayram, Kumar, and Sivakumar [8]. We present this proof rather than the original proofs [46, 68] due to its simplicity. At the end we will discuss the original proofs since they establish a number of other results, and in the next section we will present a new proof due to the author, which is stronger in a certain sense.

Recall that we are assuming all randomized protocols are public-coin. By Theorem 46, this will change our bounds by at most an additive $O(\log m)$ factor, which will not matter. Also, assume w.l.o.g. that m is odd. The following is well-known, and can be found in [51].

Theorem 60 $R^{1-way}(IND) = \Omega(m)$.

We need the following lemma.

Lemma 61 *Let m be a sufficiently large odd integer. There is a constant $c > 0$ such that for i.i.d. Bernoulli$(1/2)$ random variables $B_1, \ldots, B_m$, for any i, $1 \leq i \leq m$,*

$$\Pr[MAJ(B_1, \ldots, B_m) = 1 \mid B_i = 1] > \frac{1}{2} + \frac{c}{\sqrt{m}},$$

where $MAJ(B_1, \ldots, B_m) = 1$ iff the majority of the B_i are 1.

Proof: Let $B = \sum_{i=1}^{m} B_i$. Then $\Pr[MAJ(B_1, \ldots, B_m) = 1 \mid B_i = 1] = \Pr[B > m/2 \mid B_i = 1]$. Using Stirling's approximation (see Section 2.9 of [30]), we derive that

$$\begin{aligned}
\Pr[B > \frac{m}{2} \mid B_i = 1] &= \sum_{k=\lceil \frac{m}{2} \rceil}^{m} \Pr[B = k \mid B_i = 1] \\
&= \sum_{k=\frac{m+1}{2}}^{m} \binom{m-1}{k-1} 2^{-(m-1)} = 2^{-(m-1)} \left[2^{m-2} + \binom{m-1}{\frac{m-1}{2}} \right] \\
&= 2^{-(m-1)} \left[2^{m-2} + 2^{m-1} \sqrt{\frac{2}{\pi(m-1)}} \left(1 + o(1)\right) \right], \\
&= \frac{1}{2} + \sqrt{\frac{2}{\pi m}} \left(1 + o(1)\right).
\end{aligned}$$

which, for sufficiently large m, is $1/2 + c/\sqrt{m}$ for any $0 < c < \sqrt{2/\pi}$. ■

We design a protocol Π for IND based on a protocol Π' for 1-GH, where we assume w.l.o.g. that Π' errs with probability at most $1/12$ on all inputs. Let $d = c^2/9$, where c is the constant in Lemma 61, and assume w.l.o.g. that dm is an integer. Alice is given $x \in \{0,1\}^{dm}$ and Bob is given $k \in [dm]$, that is, the parties are given an instance of IND when Alice's input is of size dm. Alice and Bob use a public coin to generate random $r_1, \ldots, r_{dm} \in \{0,1\}^m$. Alice then computes the string $s \in \{0,1\}^m$ as follows: for each $j \in [m]$, $s_j = MAJ(r_{i,j} \mid x_i = 1)$. Thus, s is just the coordinate-wise majority of the strings r_i for which $x_i = 1$. Alice and Bob then run $\Pi'(s, r_k)$. Bob outputs $\Pi(x,k) = 1 - \Pi'(s, r_k)$.

Lemma 62 *For all $x \in \{0,1\}^m$ and $k \in [m]$,*

$$\Pr[\Pi(x,k) = IND(x,k)] \geq \frac{2}{3},$$

where the probability is over the public coin.

Proof: Suppose $x_k = 1$. Then

$$\Pr[\Pi(x,k) = 1] = \Pr[\Pi'(s, r_k) = 0] \geq \Pr[\Delta(s, r_k) \leq \frac{m}{2} - \sqrt{m}] - \frac{1}{12}.$$

The equality follows by definition of Π' and the inequality follows from the fact that for every input (in particular, for (s, r_k)), Π' errs with probability at most $1/12$.

Let $Z_1, \ldots, Z_m$ be independent Bernoulli random variables defined as follows: $Z_i = \Pr[s_i = r_{k,i}]$. By Lemma 61, since there are at most dm coordinates j for which $x_j = 1$, we have $\Pr[Z_i = 1] > 1/2 + c/\sqrt{dm} = 1/2 + 3/\sqrt{m}$. Let $Z = \sum_{i=1}^m Z_i$. Then $\mathbf{E}[Z] = \sum_{i=1}^m \mathbf{E}[Z_i] > m/2 + 3\sqrt{m}$. By independence of the Z_is, $\mathbf{Var}[Z] = \sum_{i=1}^m \mathbf{Var}[Z_i] \leq \sum_{i=1}^m \mathbf{E}[Z_i] \leq m$. By Chebyshev's inequality,

$$\Pr\left[Z < \frac{m}{2} + \sqrt{m}\right] \leq \frac{\mathbf{Var}[Z]}{(3\sqrt{m} - \sqrt{m})^2} \leq \frac{1}{4}.$$

Thus, $\Pr[\Delta(s, r_k) < m/2] \geq 3/4$ and $\Pr[\Pi(x,k) = 1] \geq 3/4 - 1/12 = 2/3$. An analogous argument shows that if $x_k = 0$, then $\Pr[\Pi(x,k) = 0] \geq 2/3$. This proves the lemma. ■

As the communication of Π is just that of Π', from Lemma 62 we have,

Theorem 63 $R^{1-way}(1\text{-}GH) = \Omega(m)$.

It follows that for $\epsilon \geq \frac{1}{\sqrt{m}}$, we have $R^{1-way}(1\text{-}GH_{m'}) = \Omega(1/\epsilon^2)$. Thus, by Corollary 56,

Theorem 64 *For any $\epsilon \geq \frac{1}{\sqrt{m}}$ and any $p \geq 0$, $R^{1-way}(T_{p,\epsilon}) = \Omega(1/\epsilon^2)$.*

Using the connection to streaming algorithms given in Theorem 59,

Theorem 65 *For any $\epsilon \geq \frac{1}{\sqrt{m}}$ and any constants $p \neq 1$ and $\delta > 0$, $S_{\epsilon,\delta}(F_p) = \Omega(1/\epsilon^2)$.*

Remark 66 The major implications of this algorithm are for $p = 0$ and $p = 2$, corresponding to counting distinct elements in a data stream and computing Gini's index of homogeneity. In both cases, up to sub-logarithmic factors, there are matching 1-pass upper bounds. For $p > 2$, there are lower bounds of the form $\Omega(m^{1-2/p})$ [7, 6, 65, 20].

Remark 67 We will not present the original proof of the lower bound due to Indyk and the author [46], since it did not hold for all m and the proof in the previous section is simpler. For simplicity, we will also not present the original proof that held for all m due to the author [68]. That proof is a bit complicated, and uses an approach based on shatter coefficients [5]. It is worth noting that the more combinatorial approach in [68] established a few additional results on degree-constrained bipartite graphs. We refer the reader to that paper for the details.

4.4 A Lower Bound for the Uniform Distribution

We now give a new proof that there is a constant $c > 0$ for which $R^{1-way}(c\text{-}GH) = \Omega(m)$. Unlike the proof in the previous section, our proof uses Yao's minimax principle and goes through distributional complexity. One advantage of this proof is that it may extend to multiple rounds, whereas the proof in the previous section cannot. This is because the IND function in that reduction has a 2-round protocol with only $O(\log m)$ bits. To give further evidence that $R(c\text{-}GH) = \Omega(m)$, in the next section we adapt this argument to prove that if the protocol is a *linear multiround protocol* for deciding *c-GH*, then its randomized complexity is $\Omega(m)$. We define such protocols in that section.

Another advantage is the implication this new proof has in practice. Practitioners in the area may complain that the lower bound in the previous section is artificial in the sense that the inputs that are hard to approximate are not likely to occur in practice. In practice two entities may wish to mine their joint data with additional assumptions on the distribution of their data. A natural assumption is that the input data is uniformly distributed over some domain. We show that this *does not make the problem easier* by giving an $\Omega(m)$ bound for $D_\mu^{1-way}(c\text{-}GH)$ when μ is uniform on $\mathcal{X} \times \mathcal{Y}$.

Assume m is odd. Let μ be the uniform distribution on $\{0,1\}^m \times \{0,1\}^m$, and let X and Y be uniform on $\{0,1\}^m$. Thus, $\mu = X \times Y$ as distributions.

Lemma 68 *For any constant $d > 0$, for a sufficiently small choice of the parameter c,*

$$\Pr_{(x,y)\sim\mu}[|\Delta(x,y) - m/2| \geq c\sqrt{m}] > 1 - d.$$

Proof: Let $v = \binom{m}{(m+1)/2}/2^m$. There is a constant $b > 0$ for which $v < bm^{-1/2}$ by Stirling's formula [30]. So $\Pr[|\Delta(x,y) - n/2| \geq c\sqrt{m}] \geq 1 - 2vc\sqrt{m} > 1 - 2bc$. Choose c so that $2bc = d$. ■

Define the function $g : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ as follows: $g(x,y) = 1$ if and only if

$\Delta(x, y) > m/2$.

Corollary 69 *For a sufficiently small constant c and constant δ, $R_\delta(c - GH) \geq D_{\mu,2\delta}(g)$ and $R_\delta^{1-way}(c\text{-}GH) \geq D_{\mu,2\delta}^{1-way}(g)$.*

Proof: By Lemma 51, $R_\delta(c\text{-}GH) \geq D_{\mu,\delta'}(g)$, where $\delta' = \delta + \mu(\bar{P}_{yes} \cap \bar{P}_{no})$. By the previous lemma, for c small enough, $\mu(\bar{P}_{yes} \cap \bar{P}_{no}) \leq \delta$, so $\delta' \leq 2\delta$. Since Lemma 51 also holds if both protocols are one-way, the corollary follows. ■

Fix a 1-round protocol Π realizing $D_{\mu,2\delta}^{1-way}(g)$. We assume $k \overset{\text{def}}{=} D_{\mu,2\delta}^{1-way} = o(m)$, and derive a contradiction. Let M be the single message sent from Alice to Bob in Π. Let Alg be the (deterministic) algorithm run by Bob on M and Y. By the properties of Π,

$$\Pr_{(x,y)\sim\mu}[Alg(M, Y) = g(X, Y)] \geq 1 - 2\delta.$$

We need the following information-theoretic inequality, known as Fano's inequality.

Fact 70 *([27]) For any random variables $R, S \in \{0, 1\}$ and any function h,*

$$H(\Pr[h(R) \neq S]) \geq H(S \mid R).$$

Applying this inequality with $h = Alg$ and assuming that $\delta \leq 1/6$,

$$H(g(X, Y) \mid M, Y) \leq H(2\delta).$$

We will now lower bound $H(g(X, Y) \mid M, Y)$ in order to reach a contradiction.

For any $r \in \{0, 1\}^k$, let S_r be the set of $x \in \{0, 1\}^m$ for which $M = r$. Then $\mathbf{E}[|S_M|] = 2^{m-k}$. By a Markov argument, the number of different x contained in some S_r for which $|S_r| \leq 2^{m-k-1}$ is at most 2^{m-1}. Therefore, $\Pr_{(x,y)\sim\mu}[|S_M| \geq 2^{m-k-1}] \geq \frac{1}{2}$. Let us condition on the event $\mathcal{E} : |S_M| \geq 2^{m-k-1}$. By concavity of the entropy,

$$H(g(X, Y) \mid M, Y) \geq H(g(X, Y) \mid M, Y, \mathcal{E}) \Pr[\mathcal{E}] \geq H(g(X, Y) \mid M, Y, \mathcal{E})/2.$$

Now let $S \in \{0, 1\}^m$ be any set of size at least 2^{m-k-1}, and let X' be the uniform distribution on $x \in S$. For $y \in \{0, 1\}^m$, let $V_y = \Pr_{x\sim X'}[g(x, y) = 1]$. Then $\mathbf{E}_{y\sim Y}[V_y] = \frac{1}{2}$. For $u \in S$, let $C_u = 1$ if $g(u, Y) = 1$, and $C_u = 0$ otherwise. Then $V_u = \frac{1}{|S|} \sum_{u \in S} C_u$. We use the second-moment method (see, e.g., [3] for an introduction to this technique).

Consider

$$\mathbf{Var}_{y \sim Y}[V_y] = \frac{1}{|S|^2} \left(\sum_{u,v \in S} \mathbf{E}[C_u C_v] - \mathbf{E}[C_u]\mathbf{E}[C_v] \right).$$

Then $\mathbf{E}[C_u] = \frac{1}{2}$ for all $u \in S$. Moreover, $\mathbf{E}[C_u^2] = \mathbf{E}[C_u] = \frac{1}{2}$ for all $u \in S$. Thus,

$$\mathbf{Var}_{y \sim Y}[V_y] = \frac{1}{|S|^2} \left(\frac{|S|}{4} + \sum_{u \neq v} \left(\mathbf{E}[C_u C_v] - \frac{1}{4} \right) \right) = o(1) + \frac{1}{|S|^2} \sum_{u \neq v} \left(\mathbf{E}[C_u C_v] - \frac{1}{4} \right).$$

The difficulty is in bounding $\mathbf{E}[C_u C_v] = \Pr_y[g(u, y) = 1 \wedge g(v, y) = 1]$. The latter equals

$$\begin{aligned} \Pr_{u \oplus y}[wt(y) > m/2 \wedge g(u \oplus v, y) = 1] &= \Pr_y[g(u \oplus v, y) = 1 \wedge wt(y) > m/2] \\ &= \frac{1}{2} \Pr_y[g(u \oplus v, y) = 1 \mid wt(y) > m/2]. \end{aligned}$$

Now we use the fact that $|S|$ is large. The following is well-known.

Fact 71 *([67]) For any $u \in \{0,1\}^m$, the number of $v \in \{0,1\}^m$ for which $\Delta(u, v) < m/3$ or $\Delta(u, v) > 2m/3$ is at most $2 \cdot 2^{H(1/3)m}$.*

Now, $|S| > 2^{m-k-1}$. It follows that of the $\binom{|S|}{2}$ pairs of $u, v \in S$ with $u \neq v$, all but $2|S|2^{H(1/3)m}$ of them have Hamming distance at least $m/3$ and at most $2m/3$. Thus, at least half of the pairs have this property. Let α be the fraction of pairs with this property.

Then $\alpha \geq 1/2$.

$$\begin{aligned}
\mathbf{Var}_{y\sim Y}[V_y] &= o(1) + \frac{1}{|S|^2}\sum_{u\neq v}\left(\mathbf{E}[C_uC_v] - \frac{1}{4}\right) \\
&= o(1) + \frac{1}{|S|^2}\sum_{\Delta(u,v)\leq n/3 \text{ or } \Delta(u,v)\geq 2m/3}\left(\mathbf{E}[C_uC_v] - \frac{1}{4}\right) \\
&+ \frac{1}{|S|^2}\sum_{m/3<\Delta(u,v)<2m/3}\left(\mathbf{E}[C_uC_v] - \frac{1}{4}\right) \\
&\leq o(1) + \frac{(1-\alpha)}{4} + \frac{1}{|S|^2}\sum_{m/3<\Delta(u,v)<2m/3}\left(\mathbf{E}[C_uC_v] - \frac{1}{4}\right) \\
&= o(1) + \frac{(1-\alpha)}{4} \\
&+ \frac{1}{2|S|^2}\sum_{m/3<\Delta(u,v)<2m/3}\left(\Pr_y[g(u\oplus v, y) = 1 \mid wt(y) > m/2] - \frac{1}{2}\right).
\end{aligned}$$

For u, v with $m/3 < \Delta(u,v) < 2m/3$, we will upper bound $\Pr_y[g(u\oplus v, y) = 1 \mid wt(y) > m/2]$ by lower bounding $\Pr_y[g(u \oplus v, y) = 0 \mid wt(y) > m/2]$. Let $r = wt(u \oplus v) = \Delta(u, v)$. Then $\Delta(u \oplus v, y) = r + wt(y) - 2t$, where t is the number of coordinates which are 1 in both $u \oplus v$ and in y. Thus, $g(u \oplus v, y) = 0$ iff $\frac{r}{2} + \frac{wt(y)}{2} - \frac{m}{4} < t$.

We collect some standard facts about the binomial distribution.

Fact 72 *For any integer m, $\binom{m}{m/2+O(\sqrt{m})} = \Theta\left(\frac{2^m}{\sqrt{m}}\right)$.*

Proof: Using Stirling's approximation, for any $c = O(1)$,

$$\begin{aligned}
\binom{m}{m/2 + c\sqrt{m}} &= \frac{m!}{(m/2 + c\sqrt{m})!(m/2 - c\sqrt{m})!} \\
&= \frac{\Theta(m^{-1/2})}{(1/2 + cm^{-1/2})^{m/2+cm^{1/2}}(1/2 - cm^{-1/2})^{m/2-cm^{1/2}}} \\
&= \frac{2^m}{\Theta(\sqrt{m})(1 + 2cm^{-1/2})^{m/2+cm^{1/2}}(1 - 2cm^{-1/2})^{m/2-cm^{1/2}}} \\
&\geq \frac{2^m}{\Theta(\sqrt{m})e^{c\sqrt{m}+2c^2-c\sqrt{m}+2c^2}} = \frac{2^m}{\Theta(\sqrt{m})},
\end{aligned}$$

where we have used that $(1 + x)^y \leq e^{xy}$ for all x, y. But $\binom{m}{m/2+c\sqrt{m}} \leq \binom{m}{m/2} = \frac{m!}{((m/2)!)^2} = \frac{2^m}{\Theta(\sqrt{m})}$, and so $\binom{m}{m/2+c\sqrt{m}} = \frac{2^m}{\Theta(\sqrt{m})}$. ■

Fact 73 *For any constant $\beta > 0$, there is a constant $\eta > 0$ for which*

$$\Pr_{y \sim Y}[wt(y) < m/2 + \beta\sqrt{m} \mid wt(y) > m/2] \geq \eta.$$

Proof: By Fact 72, for any i, $m/2 < i \leq \beta\sqrt{m}$, $\Pr[wt(y) = i] = \Theta(1/\sqrt{m})$. Thus,

$$\Pr[wt(y) = i \mid wt(y) > m/2] = 2 \cdot \Theta(1/\sqrt{m}) = \Theta(1/\sqrt{m}).$$

Thus,

$$\Pr[wt(y) < m/2 + \beta\sqrt{m} \mid wt(y) > m/2] = \beta\sqrt{m}\Theta(1/\sqrt{m}) = \Omega(1) > \eta > 0$$

for a sufficiently small constant η. ■

Fact 74 *For any constant $\beta > 0$ and any i, $m/2 < i < m/2 + \beta\sqrt{m}$,*

$$\Pr_{y \sim Y}[wt(y) = i \mid m/2 < wt(y) < m/2 + \beta\sqrt{m}] = \Omega\left(\frac{1}{\sqrt{m}}\right).$$

Proof: By Fact 72, for any i, $m/2 < i \leq \beta\sqrt{m}$ there are $\Theta(2^m/\sqrt{m})$ strings of weight i. Thus, after conditioning on the event that $m/2 < wt(y) < m/2 + \beta\sqrt{m}$, these $\beta\sqrt{m}$ weight classes all have the same probability of occurring, up to a constant factor. ■

Returning to our distributional lower bound and using the facts above, for any constant $\beta > 0$, there is a constant $\eta > 0$ for which

$$\begin{aligned}
\Pr_y[g(u \oplus v, y) = 0 \mid wt(y) > m/2] &\geq \eta \Pr_y[g(u \oplus v, y) = 0 \mid m/2 < wt(y) < m/2 + \beta\sqrt{m}]. \\
&= \Omega\left(\frac{1}{\sqrt{m}}\right) \sum_{m/2 < i < m/2 + \beta\sqrt{m}} \Pr_y[g(u \oplus v, y) = 0 \mid wt(y) = i] \\
&= \Omega\left(\frac{1}{\sqrt{m}}\right) \sum_{m/2 < i < m/2 + \beta\sqrt{m}} \sum_{t > \frac{r}{2} + \frac{i}{2} - \frac{m}{4}}^{\min(r,i)} \binom{r}{t}\binom{m-r}{i-t} / \binom{m}{i}.
\end{aligned}$$

We claim that this expression is $\Omega(1)$. To see this, we first show that $\min(r,i) - \frac{r}{2} - \frac{i}{2} + \frac{m}{4} = \Omega(\sqrt{m})$. Indeed,

$$\min(r,i) - \frac{r}{2} - \frac{i}{2} + \frac{m}{4} \geq \min(r,i) - \frac{r}{2} \geq \min(r/2, i - r/2) \geq m/6,$$

since $m/3 \leq r \leq 2m/3$, and $i > m/2$. Next, we show that $\binom{r}{\frac{r}{2}+\frac{i}{2}-\frac{m}{4}+O(\sqrt{m})} = \Omega(2^r/\sqrt{m})$. This follows immediately from the fact that $m/2 < i < m/2 + \beta\sqrt{m}$, Fact 72, and the fact that $r = \Theta(m)$. Next we show that $\binom{m-r}{i-(\frac{r}{2}+\frac{i}{2}-\frac{m}{4})+O(\sqrt{m})} = \Omega(2^{m-r}/\sqrt{m})$. This again follows from the fact that $m/2 < i < m/2 + \beta\sqrt{m}$ and Fact 72. It follows that for every value of i, there are $\Omega(\sqrt{m})$ values of t for which $\binom{r}{t} \cdot \binom{m-r}{i-t}$ is $\Omega(2^m/m)$. Now by Fact 72, $\binom{m}{i} = \Theta(2^m/\sqrt{m})$ for every value of i. Thus, there are $\Omega(m)$ pairs of i and t for which $\binom{r}{t} \cdot \binom{m-r}{i-t}/\binom{m}{i} = \Omega(1/\sqrt{m})$. It follows that

$$\Pr_y[g(u \oplus v, y) = 0 \mid wt(y) > m/2] = \Omega(1).$$

Thus, $\Pr_y[g(u \oplus v, y) = 0 \mid wt(y) > m/2] = \Omega(1)$. Let $\gamma > 0$ be a constant such that for sufficiently large m, $\Pr_y[g(u \oplus v, y) = 0 \mid wt(y) > m/2] > \gamma$. Returning to our variance computation,

$$\begin{aligned}
\mathbf{Var}_{y \sim Y}[V_y] &\leq o(1) + \frac{(1-\alpha)}{4} \\
&+ \frac{1}{2|S|^2} \sum_{m/3 < \Delta(u,v) < 2m/3} \left(\Pr_y[g(u \oplus v, y) = 1 \mid wt(y) > m/2] - \frac{1}{2} \right) \\
&\leq o(1) + \frac{(1-\alpha)}{4} + \frac{1}{2|S|^2} \sum_{m/3 < \Delta(u,v) < 2m/3} \left(1 - \gamma - \frac{1}{2} \right) \\
&= o(1) + \frac{(1-\alpha)}{4} + \frac{\alpha(1-\gamma-\frac{1}{2})}{2}.
\end{aligned}$$

So, there is a constant $\zeta < 1/4$ for which for sufficiently large m,

$$\mathbf{Var}_{y \sim Y}[V_y] < \zeta.$$

Define the constant $\zeta' = \sqrt{\frac{\zeta}{2} + \frac{1}{8}}$, and note that $\zeta' < 1/2$. It follows by Chebyshev's inequality that,

$$\Pr_{y \sim Y}\left[\left| V_y - \frac{1}{2} \right| > \zeta' \right] < \frac{\zeta}{\frac{\zeta}{2} + \frac{1}{8}} < 1.$$

Thus, for an $\Omega(1)$ fraction of y, $|V_y - 1/2| \leq \zeta$. Let us return to bounding the conditional entropy. Consider the event

$$\mathcal{F} : |V_Y - 1/2| \leq \zeta',$$

where V_Y is a random variable that depends on Y and is defined with respect to the set of

x in S_M. Since X' and Y are independent, the above analysis implies that

$$\Pr_{(x,y)\sim\mu}[\mathcal{F} \mid \mathcal{E}] = \Omega(1).$$

Thus, $H(g(X,Y) \mid M, Y, \mathcal{E}) = \Omega(H(g(X,Y) \mid M, Y, \mathcal{E}, \mathcal{F}))$, where the constant in the $\Omega(\cdot)$ is absolute (i.e., independent of δ). But, by definition of V_Y, if $\mathcal{E} \cap \mathcal{F}$ occurs, then

$$1/2 - \zeta' \ \le \Pr_{x\sim X'}[g(x,Y) = 1] \le 1/2 + \zeta'.$$

Thus, $H(g(X,Y) \mid M, Y, \mathcal{E}, \mathcal{F}) = \Omega(1)$, where the constant is independent of δ. It follows that $H(g(X,Y) \mid M, Y) = \Omega(1)$.

On the other hand, we have shown that $H(g(X,Y) \mid M, Y) \le H(2\delta)$. This is a contradiction if $\delta > 0$ is a small enough constant. Thus, our assumption that $k = o(m)$ was false. We conclude,

Theorem 75 *For a small enough constant c, $R_\delta^{1-way}(c\text{-}GH) = \Omega(D_{\mu,2\delta}^{1-way}(g)) = \Omega(m)$.*

4.5 A Multiround Lower Bound for Linear Protocols

In this section we prove that for a class of protocols that we call linear, the multiround randomized communication complexity of the gap Hamming distance problem is $\Omega(m)$. This can be viewed as a first step in extending the lower bound of the previous section to hold for more than one round.

Again assume that m is odd and let $\mu = X \times Y$ be the uniform distribution on $\{0,1\}^m \times \{0,1\}^m$. By Corollary 69, for a small enough constant $c > 0$, $R_\delta(c\text{-}GH) \ge D_{\mu,2\delta}(g)$. By increasing the communication by at most a factor of 2, we may assume that each message sent between the players is a single bit and that the players take turns alternating messages with Bob outputting the answer.

Let Π be an arbitrary protocol realizing $D_{\mu,2\delta}(g)$. We think of Π as a binary decision tree with vertices v labeled by subsets S_v of $\{0,1\}^m$. At the root r, if $x \in S_r$, then Alice transmits a 0, and otherwise she transmits a 1. This corresponds to either going to the left child of r or the right child of r. Call the visited child w. Then Bob transmits either a 0 or 1, depending on whether or not $y \in S_w$, and in the decision tree the path goes to the corresponding child of w. This process repeats until Bob reaches a leaf vertex (we can assume Bob reaches the leaf vertex by increasing the depth by at most 1), at which point he outputs the label of the leaf. The cost of Π is the depth of this tree.

Definition 76 *Π is a linear protocol if each set S_v in the decision tree for Π is described by a linear function over $GF(2)$. That is, S_v is described by a vector $L_v \in \{0,1\}^m$. If it is Alice's turn, then if $\langle L_v, x\rangle \bmod 2 = 0$, the protocol branches left, otherwise it branches right. If it is Bob's turn, then if $\langle L_v, y\rangle \bmod 2 = 0$, the protocol branches left, otherwise it branches right.*

We note that the trivial protocol, in which the parties just exchange inputs, is linear since the sets S_v are described by the unit vectors in $\{0,1\}^m$. Protocols in which parties adaptively sample bits of their inputs are also captured here. Various other classes of protocols can be reduced to linear ones. For instance, if the sets S_v are described by affine functions, then by swapping certain branches in the decision tree, the protocol can be made linear.

Let Π be a protocol realizing $D_{\mu,2\delta}(g)$ and let T be the associated decision tree. We create a new tree T' as follows. By averaging, there exists a $y^* \in \{0,1\}^m$ for which

$$\Pr_{x\sim X}[\Pi(x,y^*) = g(x,y^*)] \geq 1 - 2\delta.$$

We claim that without loss of generality, we can assume $y^* = 0^m$. Indeed, consider the following procedure. Let $v_1, v_2, \ldots, v_r$ be a list of vertices in T that occur in a breath-first-order starting at the root. So, v_1 is the root of T, v_2 and v_3 are its children, v_4, v_5, v_6, v_7 its grandchildren, etc. For $v \in T$, let $\mathsf{Lchild}(v)$ be its left child, and $\mathsf{Rchild}(v)$ its right child.

1. For $i = 1$ to r,
   - If $L_{v_i}(y^*) = 1$ and v_i is not a leaf, then
     - $temp \leftarrow \mathsf{Lchild}(v_i)$.
     - $\mathsf{Lchild}(v_i) \leftarrow \mathsf{Rchild}(v_i)$.
     - $\mathsf{Rchild}(v_i) \leftarrow temp$.

Let T' be the resulting tree. For inputs x and y, let $T'(x,y)$ be the label of the leaf reached by x and y.

Lemma 77 *$T'(x,y) = \Pi(x \oplus y^*, y \oplus y^*)$.*

Proof: We argue inductively that the ith node w_i that we visit in T' given inputs x and y is the same as the ith node that we visit in T given inputs $x \oplus y^*$ and $y \oplus y^*$. When $i = 1$, this is true, since the roots of T and T' coincide. Assume, inductively, that this is true for some $i \geq 1$, and consider the $(i+1)$st node visited. If $L_{w_i}(y^*) = 0$, then the left and right children of w_i in T are the same as those in T'. If it is Alice's turn, then $L_{w_i}(x \oplus y^*) = L_{w_i}(x) \oplus L_{w_i}(y^*) = L_{w_i}(x)$, and thus we choose the left child of w_i in T' iff we choose the left child of w_i in T. Similarly, if it is Bob's turn, then $L_{w_i}(y \oplus y^*) = L_{w_i}(y) \oplus L_{w_i}(y^*) = L_{w_i}(y)$, and so we choose the left child of w_i in T' iff we choose the left child of w_i in T.

More interestingly, if $L_{w_i}(y^*) = 1$, then the children of w_i in T are swapped in T'. If it is Alice's turn, then $L_{w_i}(x \oplus y^*) = 1 \oplus L_{w_i}(x)$, and if it is Bob's turn then $L_{w_i}(y \oplus y^*) = 1 \oplus L_{w_i}(y)$. Thus, in both cases we choose the left child of w_i in T' iff we choose the right child of w_i in T. Since the children of w_i in T are swapped in T', it follows that the $(i+1)$st node that we visit in T given inputs $x \oplus y^*$ and $y \oplus y^*$ is the same as the $(i+1)$st node that we visit in T' given inputs x and y. This completes the inductive step, and the lemma follows since the leaf visited by $T'(x, y)$ is the same as that visited by $T(x \oplus y^*, y \oplus y^*)$. ■

Since for any $x, y, y^* \in \{0,1\}^m$ we have $g(x, y) = g(x \oplus y^*, y \oplus y^*)$, by the previous lemma

$$\begin{aligned}
\Pr_{x \sim X}[T'(x, 0^m) = g(x, 0^m)] &= \Pr_{x \sim X}[\Pi(x \oplus y^*, y^*) = g(x, 0^m)] \\
&= \Pr_{x \sim X}[\Pi(x \oplus y^*, y^*) = g(x \oplus y^*, y^*)] \\
&= \Pr_{x \sim X}[\Pi(x, y^*) = g(x, y^*)] \\
&\geq 1 - 2\delta,
\end{aligned}$$

so we may indeed assume that $y^* = 0^m$. In this case $g(x, 0^m) = 1$ iff $wt(x) > m/2$.

Now for each level in T' for which it is Bob's turn, we follow the path assuming Bob's input is 0^m. This results in collapsing levels for which it is Bob's turn. We are left with a tree T' in which Alice just follows a path of linear functions and outputs the label of the leaf at the end of the path. By the propertes of T',

$$\Pr_{x \sim X}[T'(x) = g(x, 0^m)] \geq 1 - 2\delta.$$

4.5.1 Linear-algebraic Lemmas

Consider the m-dimensional vector space V consisting of all linear combinations of the variables $X_1, X_2, \ldots, X_m$ with coefficients in $GF(2)$.

Let A be a set of linearly independent vectors in V. Suppose we extend A to a basis of V by adding as many X_i as possible.

Definition 78 *The extension number, denoted $ex(A)$, is the maximum number of X_i we may add to A in order to obtain a basis of V.*

Since the X_i are linearly independent, $ex(A) = m - |A|$. Say a set $B \subseteq \{X_1, \ldots, X_m\}$ *realizes* $ex(A)$ if the vectors in $A \cup B$ are linearly independent and $|B| = ex(A)$.

Lemma 79 *Let $A \subseteq V$ be a set of linearly independent vectors. Suppose $L \notin span(A)$. If B realizes $ex(A)$, then a subset of B realizes $ex(A \cup \{L\}) = ex(A) - 1$.*

Proof: Let B realize $ex(A)$, so that $A \cup B$ is a basis of V. Write $L = \sum_{a \in A'} a \oplus \sum_{b \in B'} b$ for unique $A' \subseteq A$ and $B' \subseteq B$.

Note that $B' \neq \emptyset$ since $L \notin \text{span}(A)$. Let $b' \in B'$. The claim is that the vectors $(A \cup \{L\}) \cup (B \setminus \{b'\})$ are linearly independent. Any non-trivial zero combination must have the form $\sum_{a \in A''} a \oplus L \oplus \sum_{b \in B''} b = 0$, for some $A'' \subseteq A$ and $B'' \subseteq B \setminus \{b'\}$. Using the definition of L, $\sum_{a \in A''} a \oplus \sum_{a \in A'} a \oplus \sum_{b \in B'} b \oplus \sum_{b \in B''} b = 0$. Rewriting, $\sum_{a \in A'' \Delta A'} a \oplus \sum_{b \in B'' \Delta B'} b = 0$. Since $A \cup B$ is a basis, $A'' \Delta A' = B'' \Delta B' = \emptyset$. But $b' \in B'' \Delta B'$, a contradiction. Thus, $B \setminus \{b'\}$ realizes $ex(A \cup \{L\})$. ■

Let $A \subseteq V$ be a set of linearly independent vectors, and let B realize $ex(A)$. Let $B' = \{X_1, \ldots, X_n\} \setminus B$. For $v \in V$, let $\rho(v)$ be the projection of v onto A, and let $\sigma(v)$ be the projection of v onto B.

Lemma 80 *The set of vectors $\{\rho(X_i) \mid X_i \in B'\}$ is a linearly independent set.*

Proof: By definition, $|B \cup B'| = m$. Thus the multiset $\{\rho(X_i) \mid X_i \in B'\} \cup B$ has size m. To prove the lemma, we will show that $\text{rank}(\{\rho(X_i) \mid X_i \in B'\} \cup B) = m$.

First note that for all $X_i \in B'$, $X_i \in \text{span}(A \cup B)$ since $A \cup B$ is a basis. Thus, for all $X_i \in B'$, $X_i = \rho(X_i) \oplus \sigma(X_i)$. It follows that for all $X_i \in B'$, $X_i \in \text{span}(\{\rho(X_i) \mid X_i \in B'\} \cup B)$. Thus for all i, $X_i \in \text{span}(\{\rho(X_i) \mid X_i \in B'\} \cup B)$. But $X_1, \ldots, X_m$ are linearly independent, so their rank is m, and thus $\text{rank}(\{\rho(X_i) \mid X_i \in B'\} \cup B) = m$. ■

4.5.2 The Lower Bound for Linear Protocols

For each node $v \in T'$, let A_v be the set of linear combinations queried from the root along the unique path to v in T', including the query made at v. We may assume, w.l.o.g., that the set of linear combinations queried along any path are linearly independent since any node with a dependent linear combination may be collapsed. Moreover, we can assume T' is a complete binary tree. This does not change the depth, which is the communication complexity. Let k be the depth of T', so that any path from root to leaf contains exactly $k+1$ nodes. We will derive a contradiction assuming $k = o(m)$.

Let B_v be a set of vectors realizing $ex(A_v)$. By Lemma 79, we may assume that for all $v, w \in T$ for which w is a child of v, $B_w \subseteq B_v$ and $|B_w| = |B_v| - 1$. Define $B'_v = \{X_1, \ldots, X_m\} \setminus B_v$ for each vertex v.

Bounding the Conditional Entropy

Let $L_0, L_1, \ldots, L_k$ be the linear combinations in T queried, and let $A = (L_0, L_1, \ldots, L_k)$ be the list of these combinations. Note that the L_i and A are random variables. Define the list U of evaluations of these random variables:

$$U = (L_0(X), L_1(X), \ldots, L_k(X)).$$

Note that U determines $A, L_0, L_1, \ldots, L_k$. Let Alg be the (deterministic) algorithm run to determine the label of a leaf. By the properties of T',

$$\Pr_{x \sim X}[Alg(U) = g(X, 0^m)] \geq 1 - 2\delta.$$

By Fact 70, $H(g(X, 0^m) \mid U) \leq H(2\delta)$. We will now lower bound $H(g(X, 0^m) \mid U)$ in order to reach a contradiction.

We define the events $\mathcal{E}_1$ and $\mathcal{E}_2$ as follows. Let B be the set of vectors realizing $ex(A)$. Note that there may be more than one such set. We choose $B = B_v$, where v is the leaf reached in T' on input X, and B_v is as defined above. Let $B' = \{X_1, \ldots, X_m\} \setminus B$. Note that B and B' are random variables and $B \cup B'$ is a basis of V. Since $k = o(m)$ and T' is non-empty, we have $|B|, |B'| > 0$.

Let W_1 be the number of different $X_i \in B$ for which $X_i = 1$. Define the event

$$\mathcal{E}_1 : \quad \left| W_1 - \frac{|B|}{2} \right| > c\sqrt{|B|}.$$

Recall that c is the parameter in the c-GH problem. Let W_2 be the number of different $X_i \in B'$ for which $X_i = 1$. Define the event

$$\mathcal{E}_2 : \quad \left| W_2 - \frac{|B'|}{2} \right| < \frac{1}{c}\sqrt{|B'|}.$$

Note that $\mathcal{E}_1$ and $\mathcal{E}_2$ are independent since $B \cup B'$ is a basis of V and $B \cap B' = \emptyset$. The crux of the argument is the following lemma.

Lemma 81 *For a sufficiently small choice of the parameter $c > 0$,*

$$\Pr[\mathcal{E}_1 \wedge \mathcal{E}_2] \geq 1 - \delta.$$

Before proving this, let us see how it implies a contradiction. By concavity of the entropy,

$$\begin{aligned} H(g(X, 0^m) \mid U) &\geq H(g(X, 0^m) \mid U, \mathcal{E}_1 \wedge \mathcal{E}_2) \Pr[\mathcal{E}_1 \wedge \mathcal{E}_2] \\ &\geq H(g(X, 0^m) \mid U, \mathcal{E}_1 \wedge \mathcal{E}_2)(1 - \delta). \end{aligned}$$

By the definition of conditional entropy,

$$H(g(X, 0^m) \mid U, \mathcal{E}_1 \wedge \mathcal{E}_2) = \sum_u \Pr[U = u \mid \mathcal{E}_1 \wedge \mathcal{E}_2] \cdot H(g(X, 0^m) \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2).$$

Let $\mathcal{F}$ be the event that $W_1 - \frac{|B|}{2} > c\sqrt{|B|}$. Fix any u for which $\Pr[U = u \mid \mathcal{E}_1 \wedge \mathcal{E}_2] > 0$. Conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2$ and $U = u$, $g(X, 0^m) = 1$ iff $\mathcal{F}$ occurs. Indeed, $g(X, 0^m) = 1$ iff $W_1 + W_2 > m/2$. Since $k = o(m)$ and B realizes $ex(A)$, $|B'| = o(m)$ and $|B| = m - o(m)$. By definition of events $\mathcal{E}_1$ and $\mathcal{E}_2$, conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2$, $W_1 + W_2 > m/2$ iff $\mathcal{F}$ occurs (for sufficiently large m). Thus,

$$H(g(X, 0^m) \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2) = H(\mathcal{F} \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2).$$

Let $p_u = \Pr_{x \sim X}[\mathcal{F} \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]$. By the definition of entropy,

$$H(\mathcal{F} \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2) = p_u \log_2 \frac{1}{p_u} + (1 - p_u) \log_2 \frac{1}{1 - p_u}.$$

Let $b_1, b_2, \ldots, b_{|B|}$ be the vectors in B, and let $b'_1, b'_2, \ldots, b'_{|B'|}$ be the vectors in B'. Let $S^+ \subseteq \{0,1\}^{|B|}$ be the vectors with weight more than $|B|/2 + c\sqrt{|B|}$, and let S^- be the vectors with weight less than $|B|/2 - c\sqrt{|B|}$ ones. Let $S = S^+ \cup S^-$. Let $S' \subseteq \{0,1\}^{|B'|}$ be the set of vectors of weight less than $|B'|/2 + \sqrt{|B'|}/c$ but weight at least $|B'|/2 - \sqrt{|B'|}/c$.

For $s \in \{0,1\}^{|B|}$, we use the notation $\vec{b} = s$ to indicate that $b_1(x) = s_1, b_2(x) = s_2, \ldots, b_{|B|}(x) = s_{|B|}$. Analogously, we use the notation $\vec{b'} = s'$ for $s' \in \{0,1\}^{|B'|}$ to indicate that $b'_1(x) = s'_1, b'_2(x) = s'_2, \ldots, b'_{|B'|}(x) = s_{|B'|}$.

In the following, for any events $\mathcal{G}_1$ and $\mathcal{G}_2$ we let $\Pr[\mathcal{G}_1 \mid \mathcal{G}_2] = 0$ whenever $\Pr[\mathcal{G}_2] = 0$.

Lemma 82

$$p_u = \sum_{s \in \{0,1\}^{|B|},\ t \in \{0,1\}^{|B'|}} \Pr_{x \sim X}[\mathcal{F} \mid U = u \wedge \vec{b} = s \wedge \vec{b'} = t] \Pr_{x \sim X}[\vec{b} = s \wedge \vec{b'} = t \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2].$$

Proof: By definition of $\mathcal{E}_1$ and $\mathcal{E}_2$,

$$\Pr[\mathcal{E}_1 \wedge \mathcal{E}_2] = \sum_{s,t} \Pr[\vec{b} = s \wedge \vec{b'} = t].$$

By Bayes' rule, and noting that $\Pr[U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2] > 0$ by our choice of u,

$$\Pr[\mathcal{F} \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2] = \frac{\Pr[\mathcal{F} \wedge U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]}{\Pr[U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]}.$$

Noting that the events $\mathcal{F} \wedge (U = u) \wedge (\vec{b} = s) \wedge (\vec{b'} = t)$ are disjoint for different s and t, via another application of Bayes' rule,

$$\begin{aligned}\frac{\Pr[\mathcal{F} \wedge U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]}{\Pr[U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]} &= \frac{\sum_{s,t} \Pr[\mathcal{F} \wedge U = u \wedge \vec{b} = s \wedge \vec{b'} = t]}{\Pr[U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]} \\ &= \frac{\sum_{s,t} \Pr[\mathcal{F} \mid U = u \wedge \vec{b} = s \wedge \vec{b'} = t] \Pr[U = u \wedge \vec{b} = s \wedge \vec{b'} = t]}{\Pr[U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]}.\end{aligned}$$

Since the events $\vec{b} = s$ and $\vec{b'} = t$ imply $\mathcal{E}_1 \wedge \mathcal{E}_2$ occurs, we can rewrite the above as

$$\begin{aligned} & \sum_{s,t} \Pr[\mathcal{F} \mid U = u \wedge \vec{b} = s \wedge \vec{b'} = t] \left(\frac{\Pr[U = u \wedge \vec{b} = s \wedge \vec{b'} = t \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]}{\Pr[U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2]} \right) \\ = & \sum_{s,t} \Pr[\mathcal{F} \mid U = u \wedge \vec{b} = s \wedge \vec{b'} = t] \Pr[\vec{b} = s \wedge \vec{b'} = t \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2], \end{aligned}$$

which completes the proof. ■

Now consider a term $\Pr[\mathcal{F} \mid U = u \wedge \vec{b} = s \wedge \vec{b'} = t]$. Note that $A \cup B$ is a basis for V, so the event $U = u \wedge \vec{b} = s$ uniquely determines the value $t(s)$ of $\vec{b'}$. It follows from the previous lemma that by dropping terms that are zero, we may write

$$p_u = \sum_{s \in \{0,1\}^{|B|}} \Pr[\mathcal{F} \mid U = u \wedge \vec{b} = s] \Pr[\vec{b} = s \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2].$$

Since $A \cup B$ is a basis of V, $\Pr[U = u \wedge \vec{b} = s] > 0$ for any values of u and s, and in fact this probability is independent of s. The occurrence of $\mathcal{F}$ is entirely determined from the event $\vec{b} = s$, and $\mathcal{F}$ occurs iff $wt(s) > |B|/2 + c\sqrt{|B|}$. Thus, we may rewrite p_u as

$$p_u = \sum_{s \in S^+} \Pr[\vec{b} = s \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2].$$

Since $A \cup B$ is a basis and $A \cap B = \emptyset$, for any $s \in S^+$ we have $\Pr[\vec{b} = s \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2] = \Pr[\vec{b} = s \mid \mathcal{E}_1]$. Thus,

$$p_u = \sum_{s \in S^+} \Pr[\vec{b} = s \mid \mathcal{E}_1].$$

By symmetry, $p_u = \Pr[\vec{b} \in S^+ \mid \mathcal{E}_1] = \Pr[\vec{b} \in S^- \mid \mathcal{E}_1] = 1/2$. It follows that $H(\mathcal{F} \mid U = u \wedge \mathcal{E}_1 \wedge \mathcal{E}_2) = 1$. At long last, it follows that $H(g(X, 0^m) \mid U) \geq 1 - \delta$.

For small enough $\delta > 0$, this contradicts our earlier observation that $H(g(X,Y) \mid U, Y) \leq H(2\delta)$. We conclude that $k = \Omega(m)$.

It remains to prove Lemma 81.

Proof of Lemma 81: : We argue that both $\Pr[\mathcal{E}_1] \geq 1 - \delta/2$ and $\Pr[\mathcal{E}_2] \geq 1 - \delta/2$ for a sufficiently small choice of the parameter c, which implies the lemma by a union bound (and actually, $\mathcal{E}_1$ and $\mathcal{E}_2$ are even independent). We have,

$$\begin{aligned}
\Pr[\mathcal{E}_1] &= \sum_u \Pr[\mathcal{E}_1 \mid U = u] \Pr[U = u] \quad \text{(law of total probability)} \\
&= \sum_u \Pr\left[\left|W_1 - \frac{|B|}{2}\right| > c\sqrt{|B|} \mid U = u\right] \Pr[U = u] \quad \text{(definition)} \\
&= \sum_u \Pr\left[\left|W_1 - \frac{|B|}{2}\right| > c\sqrt{|B|}\right] \Pr[U = u] \quad \text{(independence)} \\
&\geq \sum_u (1 - \delta/2) \Pr[U = u] \quad \text{(Lemma 68)} \\
&= 1 - \delta/2.
\end{aligned}$$

Now we bound the more difficult probability, $\Pr[\mathcal{E}_2] = \Pr\left[\left|W_2 - \frac{|B'|}{2}\right| < \frac{1}{c}\sqrt{|B'|}\right]$.

Let $Z_1, \ldots, Z_r$ be the r random variables in B'. Since for all nodes u and v in the decision tree for which v is a child of u, we have $B_v \subset B_u$, we have $B'_u \subset B'_v$. Moreover, since $|B_v| = |B_u| - 1$, $|B'_u| = |B'_v| - 1$. Therefore, we can think of the $Z_1, \ldots, Z_r$ as being added to the set B' one-by-one along the path taken in T'. Thus, Z_1 denotes the first random variable added to B', Z_2 the second, etc.

We claim that it suffices to show that the random variables $Z_1, \ldots, Z_r$ are i.i.d. Bernoulli(.5) random variables. Suppose for the moment that we could prove this. The following is a standard fact about the binomial distribution on N variables.

Fact 83 *Let $X_1, \ldots, X_N$ be i.i.d. Bernoulli(.5) random variables. For any constant $d > 0$ and for a sufficiently small choice of the parameter c (that depends on d),*

$$\Pr\left[\left|\sum_{i=1}^{N} X_i - \frac{N}{2}\right| < \frac{1}{c}\sqrt{N}\right] > 1 - \delta/2.$$

Applying the fact to the Z_i, we will have $\Pr[\mathcal{E}_2] > 1 - \delta/2$, as needed.

It remains to show that $Z_1, \ldots, Z_r$ are i.i.d. Bernoulli(.5) random variables.

For a node $u \in T'$ and for any vector $v \in V$, let $\rho^u(v)$ be the projection of v onto A_u. We inductively show that for any u visited along the random path in T', the set $\{\rho^u(Z_i) \mid Z_i \in B'_u\}$ is a set of i.i.d. Bernoulli(.5) random variables. Thus, when $i = r$ we will have that $\rho(Z_1), \ldots, \rho(Z_r)$ are i.i.d. Bernoull(.5) random variables. We will later show this implies that $Z_1, \ldots, Z_r$ are i.i.d. Bernoulli(.5) random variables.

Base case: Let u be the root of T', so $B'_u = \{Z_1\}$. Then $Z_1 = L_0 \oplus \sigma^u(Z_1)$, where $\sigma^u(Z_1)$ is the projection of Z_1 onto B_u. Thus, $\rho^u(Z_1) = L_0(X) = U_0$, which is Bernoulli(.5).

Inductive step: Assume that for the jth vertex visited along the path in T', call it u, $\{\rho^u(Z_i) \mid Z_i \in B'_u\}$ is a set of i.i.d. Bernoulli(.5) random variables. Suppose u' is the $j+1$st vertex visited. We need to show that the set $\{\rho^{u'}(Z_i) \mid Z_i \in B'_{u'}\}$ is a set of i.i.d. Bernoulli(.5) random variables. For all $Z_i \in B'_{u'}$, we have $Z_i \in \text{span}(A_{u'} \cup B_{u'})$.

Let Z be the unique element in $B'_{u'} \setminus B'_u = B_u \setminus B_{u'}$. Then $A_{u'} = A_u \cup \{L_{j+1}\}$ (since u' is the $(j+1)$st vertex visited in T'). For those $Z_i \in B'_u$ in $\text{span}(A_u \cup B_{u'})$, we have $\rho^{u'}(Z_i) = \rho^u(Z_i)$ since the vectors in $A_{u'} \cup B_{u'}$ are linearly independent and $A_u \cup B_{u'} \subset A_{u'} \cup B_{u'}$.

Now if Z_i is in B'_u but not in $\text{span}(A_u \cup B_{u'})$, then $\sigma^u(Z_i)$ has a non-zero coefficient in front of Z. It follows that $\rho^{u'}(Z_i) = \rho^u(Z_i) \oplus \rho^{u'}(Z)$. Since $Z \in \text{span}(A_u \cup \{L_{j+1}\} \cup B_{u'})$, but $Z \notin \text{span}(A_u \cup B_{u'})$ (since Z can be added to $B_{u'}$ to obtain B_u, and the vectors in $A_u \cup B_u$ are linearly independent), it follows that $\rho^{u'}(Z)$ has a non-zero coefficient in front of L_{j+1}. Since $L_{j+1}(X) = U_{j+1}$ and U_{j+1} is independent of $U_1, \ldots, U_j$, it follows that $\rho^{u'}(Z)$ is Bernoulli(.5) and independent of the random variable $(\rho^u(Z_1), \ldots, \rho^u(Z_j))$, which is determined by $U_1, \ldots, U_j$.

It follows by the inductive hypothesis and the above that the random variables in the set

$$\begin{aligned}
\{\rho^{u'}(Z_i) \mid Z_i \in B'_{u'}\} &= \{\rho^{u'}(Z)\} \cup \{\rho^{u'}(Z_i) \mid Z_i \in B'_u\} \\
&= \{\rho^{u'}(Z)\} \\
&\quad \cup \{\rho^u(Z_i) \mid Z_i \in B'_u \text{ and } Z_i \in \text{ span } (A_u \cup B_{u'})\} \\
&\quad \cup \{\rho^u(Z_i) \oplus \rho^{u'}(Z) \mid Z_i \in B'_u \text{ and } Z_i \notin \text{ span } (A_u \cup B_{u'})\},
\end{aligned}$$

are i.i.d. Bernoulli(.5).

Thus, we have shown that $\rho(Z_1), \ldots, \rho(Z_r)$ are i.i.d. Bernoulli(.5) random variables. It remains to show that this implies $Z_1, \ldots, Z_r$ are i.i.d. Bernoulli(.5) random variables. For each i, $1 \leq i \leq r$, we have $Z_i = \rho(Z_i) \oplus \sigma(Z_i)$. For any fixed assignment $\vec{b} = s$, we have $Z_i = \rho(Z_i) \oplus \beta_i$, where $\beta_i \in \{0, 1\}$ is the result of evaluating $\sigma(Z_i)$ on s. Now since the

$\rho(Z_i)$ are i.i.d. Bernoulli(.5), so are the $\rho(Z_i) \oplus \beta_i$. Thus, the Z_i are i.i.d. Bernoull(.5).

The proof of the lemma is now complete. ■

Theorem 84 *For a constant $c > 0$, for linear protocols $R_\delta(c\text{-}GH) \geq D_{\mu,2\delta}(g) = \Omega(m)$.*

The lower bound for unrestricted multiround randomized protocols is still open. Our conjecture is the following, which was made independently by Ravi Kumar [52]. This problem is listed as the 10th question in the list of open problems at the IITK Workshop on Algorithms for Data Streams (http://www.cse.iitk.ac.in/users/sganguly/data-stream-probs.pdf).

Conjecture 85 *For every constant $c > 0$, $R(c\text{-}GH) = \Omega(m)$.*

Chapter 5

Private Protocol for the Euclidean Distance

Recent years witnessed the explosive growth of the amount of available data. Large data sets, such as transaction data, astronomical data, the web, or network traffic, are in abundance. Much of the data is stored or made accessible in a distributed fashion. This neccessitates the development of efficient protocols that compute or approximate functions over such data (e.g. see [13]).

At the same time, the availability of this data has raised significant privacy concerns. It became apparent that one needs cryptographic techniques in order to control data access and prevent potential misuse. In principle, this task can be achieved using the general results of secure function evaluation (SFE) [70, 36]. However, in most cases the resulting private protocols are much less efficient than their non-private counterparts (an exception is the result of [60], who show how to obtain private and communication-efficient versions of non-private protocols, as long as the communication cost is logarithmic). Moreover, SFE applies only to algorithms that compute functions exactly, while for many problems, only efficient approximation algorithms are known or are possible. Indeed, while it is true that SFE can be used to privately implement any efficient algorithm, it is of little use applying it to an approximation algorithm when the approximation leaks more information about the inputs than the solution itself.

In a pioneering paper [29], the authors introduced a framework for secure computation of approximations. They also proposed an $\tilde{O}(\sqrt{m})$-communication[1] two-party protocol for approximating the Hamming distance between two binary vectors. This improves over the linear complexity of computing the distance exactly via SFE, but still does not achieve the polylogarithmic efficiency of a non-private protocol of [54]. Improving the aforementioned bound was one of the main problems left open in [29].

In this and the next chapter we provide several new results for secure computation of approximations. In this chapter we provide a $\tilde{O}(1)$-communication protocol for approximating the Euclidean (ℓ_2) distance between two vectors. This, in particular, solves the open problem of [29]. Since distance computation is a basic geometric primitive, we believe that our result could lead to other algorithms for secure approximations. Indeed, in [2] the authors show how to approximate the ℓ_2 distance using small space and/or short amount of communication, initiating a rich body of work on streaming algorithms.

5.1 Cryptographic Tools

We start by reviewing homomorphic encryption, oblivious transfer (OT), and secure function evaluation (SFE).

Homomorphic Encryption: An encryption scheme, $E : (G_1, +) \rightarrow (G_2, \cdot)$ is homomorphic if for all $a, b \in G_1$, $E(a + b) = E(a) \cdot E(b)$. For more background on this primitive see, for example, [37, 59]. We will make use of the Paillier homomorphic encryption scheme [63]. in some of our protocols and so we briefly repeat it here:

1. **Initialize:** Choose two primes, p and q and set $N = p \cdot q$. Let $\lambda = lcm(p-1, q-1)$. Let the public key $PK = (N, g)$ where the order of g is a multiple of N. Let the secret key, $SK = \lambda$.

2. **Encrypt:** Given a message $M \in Z_N$, choose a random value $x \in Z_N^*$. The encryption of M is, $E(M) = g^M x^N mod N^2$.

3. **Decrypt:** Let $L(u) = \frac{(u-1)}{N}$, where u is congruent to 1 modulo N.To recover M from $E(M)$ calculate, $\frac{L(E(M)^\lambda mod\, N^2)}{L(g^\lambda mod\, N^2)} mod\, N$.

[1] We write $f = \tilde{O}(g)$ if $f(m,k) = O\left(g(m,k) \log^{O(1)}(m) \text{poly}(k)\right)$, where k is a security parameter.

In [63] it's shown that the Paillier encryption scheme's semantic security is equivalent to the Decisional Composite Residuosity Assumption. The following shows homomorphy:

$$\begin{aligned} E(M_1) \cdot E(M_2) &= (g^{M_1} {x_1}^N \mod N^2) \cdot (g^{M_2} {x_2}^N \mod N^2) \\ &= g^{M_1+M_2} (x_1 x_2)^N \mod N^2 = E(M_1 + M_2). \end{aligned}$$

Oblivious Transfer and SPIR: Oblivious transfer is equivalent to the notion of symmetrically-private information retrieval (SPIR), where the latter usually refers to communication-efficient implementations of the former. SPIR was introduced in [34]. With each invocation of a SPIR protocol a user learns exactly one bit of a binary database of length N while giving the server no information about which bit was learned. We rely on single-server SPIR schemes in our protocols. Such schemes necessarily offer computational, rather than unconditional, security [24]. Applying the transformation of [61] to the PIR scheme of [17] give SPIR constructions with $\tilde{O}(N)$ server work and $\tilde{O}(1)$ communication.

One issue is that in some of our schemes, we actually perform OT on *records* rather than on bits. It is a simple matter to convert a binary OT scheme into an OT scheme on records by running r invocations of the binary scheme in parallel, where r is the record size. This gives us a 1-round, $\tilde{O}(r)$ communication, $\tilde{O}(Nr)$ server work OT protocol on records of size r. The dependence on r can be improved using techniques of [23].

Secure Function Evaluation: In [36, 70] it is shown how two parties holdings inputs x and y can privately evaluate any circuit C with communication $O(k(|C| + |x| + |y|))$, where k is a security parameter. In [16] it is shown how to do this in one round for the semi-honest case we consider. The time complexity is the same as the communication. We use such protocols as black boxes in our protocols.

5.2 Privacy

We assume both parties are computationally bounded and semi-honest, meaning they follow the protocol but may keep message histories in an attempt to learn more than is prescribed. In [36, 18, 60], it is shown how to transform a semi-honest protocol into a protocol secure in the malicious model. Further, [60] does this at a communication blowup of at most a factor of poly(k). Therefore, we assume parties are semi-honest in the remainder of the paper.

We briefly review the semi-honest model, referring the reader to [35, 55] for more details. Let $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ be a function, the first element denoted $f_1(x_1, x_2)$ and the second $f_2(x_1, x_2)$. Let π be a two-party protocol for computing f. The views of players P_1 and P_2 during an execution of $\pi(x_1, x_2)$, denoted $\texttt{View}_1^\pi(x_1, x_2)$ and $\texttt{View}_2^\pi(x_1, x_2)$ respectively, are:

$$\texttt{View}_1^\pi(x_1, x_2) = (x_1, r_1, m_{1,1}, \ldots, m_{1,t}), \texttt{View}_2^\pi(x_1, x_2) = (x_2, r_2, m_{2,1}, \ldots, m_{2,t}),$$

where r_i is the random input and $m_{i,j}$ the messages received by player i respectively. The outputs of P_1 and P_2 during an execution of $\pi(x_1, x_2)$ are denoted $\texttt{output}_1^\pi(x_1, x_2)$ and $\texttt{output}_2^\pi(x_1, x_2)$. We define $\texttt{output}^\pi(x_1, x_2)$ to be $(\texttt{output}_1^\pi(x_1, x_2), \texttt{output}_2^\pi(x_1, x_2))$. We say that π privately computes a function f if there exist PPT algorithms S_1, S_2 for which for $i \in \{1, 2\}$ we have the following indistinguishability

$$\{S_i(x_i, f_i(x_1, x_2)), f(x_1, x_2)\} \overset{c}{\equiv} \{\texttt{View}_i^\pi(x_1, x_2), \texttt{output}^\pi(x_1, x_2)\}.$$

This simplifies to $\{S_i(x_i, f_i(x_1, x_2))\} \overset{c}{\equiv} \{\texttt{View}_i^\pi(x_1, x_2)\}$ if either $f_1(x_1 x_2) = f_2(x_1, x_2)$ or if $f(x_1, x_2)$ is deterministic or equals a specific value with probability $1 - \text{negl}(k, N)$, for k a security parameter.

We need a standard composition theorem [35] concerning private subprotocols. An *oracle-aided protocol* (see [55]) is a protocol augmented with a pair of oracle tapes for each party and oracle-call steps. In an oracle-call step parties write to their oracle tape and the oracle responds to the requesting parties. An oracle-aided protocol uses the *oracle-functionality* $f = (f_1, f_2)$ if the oracle responds to query x, y with $(f_1(x, y), f_2(x, y))$, where f_1, f_2 denote first and second party's output respectively. An oracle-aided protocol *privately reduces* g to f if it privately computes g when using oracle-functionality f.

Theorem 86 *[35] If a function g is privately reducible to a function f, then the protocol g' derived from g by replacing oracle calls to f with a protocol for privately computing f, privately computes g.*

We now define the *functional privacy* of an approximation as in [29]. For our approximation protocols we will have $f_1(x, y) = f_2(x, y) = f(x, y)$.

Definition 87 *Let $f(x,y)$ be a function, and let $\hat{f}(x,y)$ be a randomized function. Then $\hat{f}(x,y)$ is functionally private for f if there is an efficient simulator S s.t. for every x, y, we have $\hat{f}(x,y) \overset{c}{\equiv} S(f(x,y))$.*

A *private approximation* of f privately computes a randomized function $\hat{f}$ that is functionally private for f.

Finally, we need the notion of a protocol for securely evaluating a circuit *with ROM.* In this setting, the ith party has a table $R_i \in (\{0,1\}^r)^s$ defined by his inputs. The circuit, in addition to the usual gates, is equipped with *lookup gates* which on inputs (i,j), output $R_i[j]$.

Theorem 88 *[60] If C is a circuit with ROM, then it can be securely computed with $\tilde{O}(|C|T(r,s))$ communication, where $T(r,s)$ is the communication of 1-out-of-s OT on words of size r.*

5.3 Private Euclidean Distance

Here we give a private approximation of the ℓ_2 distance. Alice is given a vector $a \in [N]^m$, and Bob a vector $b \in [N]^m$. Note that $\|a-b\|^2 \leq T_{max} \overset{\text{def}}{=} mN^2$. In addition, parameters ϵ, δ and k are specified. For simplicity, we assume that $k = \Omega(\log(Nm))$. The goal is for both parties to compute an estimate E such that $|E - \|x\|^2| \leq \epsilon \|x\|^2$ with probability at least $1-\delta$, for $x \overset{\text{def}}{=} a - b$. Further, we want E to be a private approximation of $\|x\|$. As discussed there, wlog we assume the parties are semi-honest. We set the parameter $B = \Theta(k)$; this notation means $B = ck$ for a large enough constant c independent from $k, n, M, \delta, \epsilon$. In our protocol we make the following cryptographic assumptions.

1. There exists a PRG G stretching polylog(m) bits to m bits secure against poly(m)-sized circuits.

2. There exists an OT scheme for communicating 1 of m bits with communication polylog(m).

At the end of the section we discuss the necessity and plausibility of these assumptions. Our protocol relies on the following fact and corollary.

Fact 89 *[56] Let A be a random $m \times m$ orthonormal matrix (i.e., A is picked from a distribution defined by the Haar measure). Then there is $c > 0$ such that for any $x \in \Re^m$, any $i = 1, \ldots, m$, and any $t > 1$,*

$$\Pr[|(Ax)_i| \geq \frac{\|x\|}{\sqrt{m}} t] \leq e^{-ct^2}.$$

Corollary 90 *Suppose we sample A as in Fact 89 but instead generate our randomness from G, rounding its entries to the nearest multiple of $2^{-\Theta(B)}$. Then,*

$$\forall x \in \Re^m, \quad \Pr[(1 - 2^{-B})\|x\|^2 \leq \|Ax\|^2 \leq \|x\|^2 \text{ and } \forall_i (Ax)_i^2 < \frac{\|x\|^2}{m} B] > 1 - \operatorname{neg}(k, m)$$

Proof: If there were an infinite sequence of $x \in [N]^m$ for which this did not hold, a circuit with x hardwired would contradict the pseudorandomness of G. ■

Protocol Overview: Before describing our protocol, it is instructive to look at some natural approaches and why they fail. We start with the easier case of approximating the Hamming distance, and suppose the parties share a common random string. Consider the following non-private protocol of [54] discussed in [29]: Alice and Bob agree upon a random $O(\log m) \times m$ binary matrix R where the ith row consists of m i.i.d. Bernoulli(β^i) entries, where β is a constant depending on ϵ. Alice and Bob exchange Ra, Rb, and compute $R(a - b) = Rx$. Then $\|x\|$ can be approximated by observing that $\Pr[(Ra)_i = (Rb)_i] \approx 1/2$ if $\|x\| \gg \beta^{-i}$, and $\Pr[(Ra)_i = (Rb)_i] \approx 1$ if $\|x\| \ll \beta^{-i}$. Let the output be E. The communication is $O(\log m)$, but it is not private since both parties learn Rx. Indeed, as mentioned in [29], if $a = 0$ and $b = e_i$, then Rx equals the ith column of R, which cannot be simulated without knowing i.

However, given only $\|x\|$, it is possible to simulate E. Therefore, as pointed out in [29], one natural approach to try to achieve privacy is to run an SFE with inputs Ra, Rb, and output E. But this also fails, since knowing E *together with the randomness* R may reveal additional information about the inputs. If E is a deterministic function of Ra, Rb, and if $a = 0$ and $b = e_i$, Alice may be able to find i from a and R.

In [29], two private protocols which each have $\Omega(m)$ communication for a worst-case choice of inputs, were cleverly combined to overcome these problems and to achieve $\tilde{O}(\sqrt{m})$ communication. The first protocol, High-Distance Estimator, works when $\|x\| > \sqrt{m}$. The idea is for the parties to obliviously sample random coordinates of x, and use these to estimate $\|x\|$. Since the sampling is oblivious, the views depend only on $\|x\|$, and since it is random, the estimate is good provided we take $\tilde{O}(\sqrt{m})$ samples.

The second protocol, Low-Distance Estimator, works when $\|x\| \leq \sqrt{m}$. Roughly, the idea is for the parties to perfectly hash their vectors into $\tilde{O}(\sqrt{m})$ buckets so that at most one coordinate j for which $a_j \neq b_j$ lies in any given bucket. The parties then run an SFE with their buckets as input, which can compute $\|x\|$ exactly by counting the number of buckets which differ.

Our protocol breaks this $O(\sqrt{m})$ communication barrier as follows. First, Alice and Bob agree upon a random *orthonormal* matrix A in $\mathbb{R}^{m \times m}$, and compute Aa and Ab. The point of this step is to uniformly spread the mass of the difference vector x over the m coordinates, as per Fact 89, while preserving the length. Since we plan to sample random coordinates of Ax to estimate $\|x\|$, it is crucial to spread out the mass of $\|x\|$, as otherwise we could not for instance, distinguish $x = 0$ from $x = e_i$. The matrix multiplication can be seen as an analogue to the perfect hashing in Low-Distance Estimator, and the coordinate sampling as an analogue to that in High-Distance Estimator.

To estimate $\|x\|$ from the samples, we need to be careful of a few things. First, the parties should not learn the sampled values $(Ax)_j$, since these can reveal too much information. Indeed, if $a = 0$, then $(Ax)_j = (Ab)_j$, which is not private. To this end, the parties run a secure circuit with ROM Aa and Ab, which privately obtains the samples.

Second, we need the circuit's output distribution E to depend only on $\|x\|$. It is not enough for $\mathbf{E}[E] = \|x\|^2$, since a polynomial number of samples from E may reveal non-simulatable information about x based on E's higher moments. To this end, the circuit uses the $(Ax)_j$ to independently generate r.v.s z_j from a Bernoulli distribution with success probability depending only on $\|x\|$. Hence, z_j depends only on $\|x\|$.

Third, we need to ensure that the z_j contain enough information to approximate $\|x\|$. We do this by maintaining a loop variable T which at any point in time is guaranteed to be an upper bound on $\|x\|^2$ with overwhelming probability. Using Corollary 90, for all j it holds that $q \stackrel{\text{def}}{=} m(Ax)_j^2/(TB) \leq 1$ for a parameter B, so we can generate the z_j from a Bernoulli(q) distribution. Since T is halved in each iteration, for some iteration $\mathbf{E}[\sum_j z_j]$ will be large enough to ensure that E is tightly concentrated.

We now describe the protocol in detail. Set $\ell = \Theta(B)(1/\epsilon^2 \log(Nm)\log(1/\delta) + k)$. In the following, if $q > 1$, then the distribution Bernoulli(q) means Bernoulli(1).

```
$\ell_2$-Approx $(a, b)$:

  1. Alice, Bob exchange a seed of $G$ and generate a random $A$ as in Corollary 90
  2. Set $T = T_{max}$
  3. Repeat:
       (a) {Assertion: $\|x\|^2 \leq T$ }
       (b) A secure circuit with ROM $Aa, Ab$ computes the following
             • Generate random coordinates $i_1, \ldots, i_\ell$ and compute $(Ax)^2_{i_1}, \ldots (Ax)^2_{i_\ell}$
             • For $j \in [\ell]$, independently generate $z_j$ from a Bernoulli$\Big(m(Ax)^2_{i_j}/(TB)\Big)$
               distribution
       (c) T = T/2
  4. Until $\sum_i z_i \geq \frac{\ell}{4B}$ or $T < 1$
  5. Output $E = \frac{2TB}{\ell}\sum_i z_i$ as an estimate of $\|x\|^2$
```

Note that the protocol can be implemented in $O(1)$ rounds by parallelizing the secure circuit invocations.

Analysis: To show the correctness and privacy of our protocol, we start with the following lemma.

Lemma 91 *The probability that assertion 3a holds in every iteration of step 3 is* $1 - \text{neg}(k, m)$*. Moreover, when the algorithm exits, with probability* $1 - \text{neg}(k, m)$ *it holds that* $\mathbf{E}[\sum_j z_j] \geq \ell/(3B)$*.*

Proof: By Corollary 90, $\Pr_A[(1 - 2^{-B})\|x\|^2 \leq \|Ax\|^2 \leq \|x\|^2 \text{ and } \forall_i (Ax)^2_i < \frac{\|x\|^2}{m}B] = 1 - \text{neg}(k, m)$, so we may condition on this occurring. If $\|x\|^2 = 0$, then $\Pr[Ax = 0] = 1 - \text{neg}(k, m)$, and thus $\Pr[E = 0] = 1 - \text{neg}(k, m)$. Otherwise, $\|x\|^2 \geq 1$. Consider the smallest j for which $T_{max}/2^j < ||x||^2$. We show for $T = T_{max}/2^{j-1} \geq \|x\|^2 \geq 1$ that $\Pr[\sum_j z_j <$

$\ell/(4B)] = \text{neg}(k,m)$. The assertion holds at the beginning of the jth iteration by our choice of T. Thus, $m(Ax)_i^2 \leq TB$ for all $i \in [m]$. So for all j, $\Pr[z_j = 1] = \frac{\|Ax\|^2}{TB} \geq (1-2^{-B})/(2B)$, and thus $\mathbf{E}[\sum_j z_j] \geq \ell/(3B)$. By a Chernoff bound, $\Pr[\sum_j z_j < \ell/(4B)] = \text{neg}(k,m)$, so if ever $T = T_{max}/2^{j-1}$, then this is the last iteration with overwhelming probability. ■

Correctness: We show $\Pr[|E - \|x\|^2| \leq \epsilon] \geq 1-\delta$. By Lemma 91, when the algorithm exits, with probability $1 - \text{neg}(k,m)$, $\mathbf{E}\left[\sum_i z_i\right] > \frac{\ell}{3B}$, so we assume this event occurs. By a Chernoff bound,

$$\Pr\left[\left|\sum_i z_i - E\left[\sum_i z_i\right]\right| \geq \frac{\epsilon}{2} E\left[\sum_i z_i\right] \mid \sum_i z_i \geq \frac{\ell}{4B}\right] \leq e^{-\Theta\left(\epsilon^2 \frac{\ell}{B}\right)} < \frac{\delta}{2}$$

By Lemma 91, assertion 3a holds, so that

$$\ell(1-2^{-B})\|x\|^2 \ \leq \ TB \cdot \mathbf{E}[\sum_i z_i] \ \leq \ell \ \|x\|^2$$

Setting $E = \frac{2TB}{\ell} \sum_i z_i$ (recall that T is halved in step 3c) shows that $\Pr[|E - \|x\|^2 \geq \epsilon\|x\|^2] \leq \delta$.

Privacy: We replace the secure circuit with ROM in step 3b of ℓ_2-Approx with an oracle. We construct a single simulator Sim, which given $\Delta \overset{\text{def}}{=} \|x\|^2$, satisfies $\mathsf{Sim}(\Delta) \overset{\text{c}}{\equiv} \mathtt{View}_{\mathtt{A}}^{\pi}(\mathtt{a},\mathtt{b})$ and $\mathsf{Sim}(\Delta) \overset{\text{c}}{\equiv} \mathtt{View}_{\mathtt{B}}^{\pi}(\mathtt{a},\mathtt{b})$, where $\mathtt{View}_{\mathtt{A}}^{\pi}(\mathtt{a},\mathtt{b}), \mathtt{View}_{\mathtt{B}}^{\pi}(\mathtt{a},\mathtt{b})$ are Alice, Bob's real views respectively. This, in particular, implies functional privacy. It will follow that ℓ_2-Approx is a private approximation of Δ.

```
Sim (Δ):
1. Generate a random seed of $G$
2. Set $T = T_{max}$
3. Repeat:
    (a) For $j \in [\ell]$, independently generate $z_j$ from a Bernoulli($\Delta/(TB)$) distribution
    (b) $T = T/2$
4. Until $\sum_i z_i \geq \frac{\ell}{4B}$ or $T < 1$
5. Output $E = \frac{2TB}{l} \sum_i z_i$
```

With probability $1 - \text{neg}(k, m)$, the matrix A satisfies the property in Corollary 90, so we assume this event occurs. In each iteration, the random variables z_j are independent in both the simulation and the protocol. Further, the probabilities that $z_j = 1$ in the simulated and real views differ only by a multiplicative factor of $(1 - 2^{-B})$ as long as $T \geq \Delta$. But the probability that, in either view, we encounter $T < \Delta$ is $\text{neg}(k, m)$.

Complexity. Given our cryptographic assumptions, we use $\tilde{O}(1)$ communication and $O(1)$ rounds.

Remark 92 Our cryptographic assumptions are fairly standard, and similar to the ones in [29]. There the authors make the weaker assumptions that PRGs stretching m^γ bits to m bits and OT with m^γ communication exist for any constant γ. In fact, the latter implies the former [41, 40]. If we were to instead use these assumptions, our communication would be $O(m^\gamma)$, still greatly improving upon the $O(m^{1/2+\gamma})$ communication of [29]. A candidate OT scheme satisfying our assumptions can be based on the Φ-Hiding Assumption [17], and can be derived by applying the PIR to OT transformation of [61] to the scheme in that paper.

Remark 93 For the special case of Hamming distance, we have an alternative protocol based on the following idea. Roughly, both parties apply the perfect hashing of the Low-Distance Estimator protocol of [29] for a logarithmic number of levels j, where the jth level contains $\tilde{O}(2^j)$ buckets. To overcome the $\tilde{O}(\sqrt{m})$ barrier of [29], instead of exchanging the buckets, the set of buckets is randomly and obliviously sampled. From the samples, an estimate of $\Delta(a, b)$ is output. For some j, $2^j \approx \Delta(a, b)$, so the estimate will be tightly concentrated, and for reasons similar to ℓ_2-Approx, will be simulatable. We omit the details, but note that two advantages of this alternative protocol are that the time complexity will be $\tilde{O}(m)$ instead of $\tilde{O}(m^2)$, and that we don't need the PRG G, as we may use k-wise independence for the hashing.

Chapter 6

Private Protocols for Efficient Matching

In this chapter, we look at secure computation of a *near neighbor* for a query point q (held by Alice) among n data points P (held by Bob) in $\{0,1\}^d$. We improve upon known results [28, 32] for this problem under various distance metrics, including ℓ_2, set difference, and Hamming distance over arbitrary alphabets. Our techniques also result in better communication for the *all-nearest-neighbors* problem, where Alice holds n different query points, resolving an open question of [32], and yield a binary inner product protocol with communication $d + O(k)$ in the common random string model.

However, all of our protocols for the near neighbor problem have the drawback of needing $\Omega(n)$ bits of communication, though the dependence on d is often optimal. Thus, we focus on what we term the *approximate near neighbor problem.* For this we introduce a new definition of secure computation of approximations for functions that return points (or sets of points) rather than values.

Let $P_t(q)$ be the set of points in P within distance t from q. In the *c-approximate near neighbor* problem, the protocol is required to report a point in $P_{cr}(q)$, as long as $P_r(q)$ is nonempty. We say that a protocol solving this problem is *c'-private* (or just *private* if $c' = c$) if Bob learns nothing, while Alice learns nothing except what can be deduced from the set $P_{c'r}(q)$. In our paper we always set $c' = c$.

We believe this to be a natural definition of privacy in the context of the approximate near neighbor problem. First, observe that if we insist that Alice learns only the set P_r (as opposed to P_{cr}), then the problem degenerates to the *exact* near neighbor problem. Indeed, even though the definition of correctness allows the protocol to output a point $p \in P_{cr} - P_r$, in general Alice cannot simulate this protocol given only the set P_r. Thus, in order to make use of the flexibility provided by the approximate definition of the problem, it seems necessary to relax the definition of privacy as well.

Within this framework, we give a protocol based on dimensionality reduction [54] with communication $\tilde{O}(n^{1/2} + d)$ for any constant $c > 1$. We show how the dependence on d can be made polylogarithmic if Alice just wants a coordinate of a point in P_{cr}. We also give a protocol bacsed on locality-sensitive hashing (LSH) [45], with communication $\tilde{O}(n^{1/2+1/(2c)} + d)$, but significantly less work (though still polynomial).

Finally, proceeding along the lines of [39], we say the protocol *leaks b bits of information* if it can be simulated given b extra bits which may depend arbitrarily on the input. With this definition, we give a protocol with $\tilde{O}(n^{1/3} + d)$ communication leaking only k bits, where k is a security parameter.

We also give an alternative protocol, based on locality-sensitive hashing (LSH) [45], with communication $\tilde{O}(n^{1/2+1/(2c)} + d)$, but significantly less work. That is, the work of the previous scheme is $O(nd) + n^{\rho(c-1)}$, where $\rho(x) = O(1/x^2 + \log(1+x)/(1+x))$. Although this is polynomial work for constant c, the computation time can be costly in practice, e.g., $\rho(1) \approx 12$, see [42], p. 34 for a plot of the function. In contrast, the time complexity of the LSH scheme is at most $O(n^2(d+n))$ for any c.

Here we consider the setting in which Alice has a point q, and Bob a set of n points P.

6.1 Exact Problems

6.1.1 Private Near Neighbor Problem

Suppose for some integer U, Alice has $q \in [U]^d$, Bob has $P = p_1, \ldots, p_n \in [U]^d$, and Alice should learn $\min_i f(q, p_i)$, where f is some distance function. In [28] protocols for ℓ_1, ℓ_2, Hamming distance over U-ary alphabets, set difference, and arbitrary distance functions $f(a,b) = \sum_{i=1}^{d} f_i(a_i, b_i)$ were proposed, using an untrusted third party. We improve the communication of these protocols and remove the third party using homomorphic encryption to implement polynomial evaluation as in [32], and various hashing tricks.

In [32], the authors consider the private all-near neighbors problem in which Alice has

n queries $q_1, \ldots, q_n \in [U]^d$ and wants all p_i for which $\Delta(p_i, q_j) \leq t < d$ for some j and parameter t. Our techniques improve the $\tilde{O}(n^2 d)$ communication of a generic SFE and the $\tilde{O}(n\binom{d}{t})$ communication of [32] for this problem to $\tilde{O}(nd^2 + n^2)$. Finally, in the common random string model we achieve $\lceil \log d \rceil + O(k)$ communication for the (exact) Hamming distance, and an inner product protocol with $d + O(k)$ communication.

6.1.2 Private Near Neighbor for ℓ_2 and Hamming Distance

Alice has $q \in [U]^d$, and Bob a set of points $P = p_1, \ldots, p_n$ in $[U]^d$. Alice should output $\text{argmin}_i \sum_j |p_{i,j} - q_j|^2$. The protocol is easily modified to return the p_i realizing the minimum. We assume a semantically secure homomorphic encryption scheme E such as Paillier encryption, that the message domain is isomorphic to $\mathbb{Z}_m$ for some m, and that m is large enough so that arithmetic is actually over $\mathbb{Z}$.

Exact-$\ell_2(q, P)$:

1. Alice generates (PK, SK) for E and sends PK, $E(q_1), \ldots, E(q_d)$ to Bob
2. For all i, Bob computes (by himself) $z_i = E(\langle q, p_i \rangle)$ and $v_i = \|p_i\|^2$
3. A secure circuit with inputs $q, SK, \{z_i\}_i$, and $\{v_i\}_i$ computes
   - $\langle q, p_i \rangle = D_{SK}(z_i)$ for all i
   - Return $\text{argmin}_i(v_i - 2\langle q, p_i \rangle)$

Using the homomorphy of E and the $\tilde{O}(n)$-sized circuit in step 3, we make the communication $\tilde{O}(n + d)$ rather than the $\tilde{O}(nd)$ of a generic SFE. The correctness is easy to verify. Using theorem 86 and the semantic security of E, privacy is just as easy to show. We note a natural extension to ℓ_p distances: Alice sends

$$\{E(q_{i_1})\}, \{E(q_{i_1} q_{i_2})\}, \ldots, \{E(q_{i_1} \cdots q_{i_{p-1}})\},$$

where $i_1, \ldots, i_{p-1}$ range over all of $[d]$. The communication is $\tilde{O}(n + d^{p-1})$, which is interesting for $d = O(n^{1/(p-2)})$.

6.1.3 Private Near Neighbor for Generic Distance Functions

Now Alice wants $\min_i f(q, p_i)$ for an arbitrary $f(a, b) = \sum_{i=1}^{d} f_i(a_i, b_i)$. We use homomorphic encryption to implement polynomial evaluation as in [32].

Exact-Generic(q, P):

1. Alice creates d degree-$(U-1)$ polynomials s_j by interpolating from $s_j(u) = f_j(p_j, u)$ for all $u \in [U]$
2. Alice generates (PK, SK) for E and sends the encrypted coefficients of the s_j and PK to Bob
3. Bob computes (by himself) $z_i = E(\sum_j s_j(p_{i,j})) = E(f(q, p_i))$ for all i
4. A secure circuit with inputs $SK, \{z_i\}_i$ outputs $\text{argmin}_i D_{SK}(z_i)$

The proofs are similar to those of the previous section and are omitted. The communication here is $\tilde{O}(dU+n)$, improving the $O(ndU)$ communication of [28]. A special case of the result in section 6.1.5 improves this to $\tilde{O}(d^2 + n)$ in case $f(a, b)$ is Hamming distance and $U > d$.

6.1.4 Private Near Neighbor for $n = 1$

We now show how Alice, holding $q \in \{0, 1\}^d$, and Bob, holding $p \in \{0, 1\}^d$ for some prime d, can privately compute $\Delta(q, p)$ with communication $d\lceil \log d \rceil + O(k)$. This extends to solve the private near neighbor problem for $n = 1$ with communication $2d\lceil \log d \rceil + \tilde{O}(k)$. The communication outperforms the $\Theta(dk)$ communication of SFE.

We assume both parties have access to the same uniformly random string. We need a homomorphic encryption whose message domain can be decoupled from its security parameter. Recall in Paillier encryption that if encryptions are k bits long, messages are about $k/2$ bits long. For low communication we want the domain to be very small, that is, roughly d elements instead of $2^{k/2}$. To do this, we use a Benaloh encryption scheme E [12], which is homomorphic and semantically secure assuming the prime residuousity assumption. The message domain is $\mathbb{Z}_d$ while encryptions are of size k.

Exact-1(q, p):

1. Alice generate (PK, SK) for E, and sends PK to Bob
2. Both parties interpret [1] the common random string R as d encryptions $E(z_i)$
3. Alice obtains the z_i by decrypting, and sends Bob $s_i = q_i - z_i \bmod d$ for all i
4. Bob computes (by himself) $E(z_i + q_i) = E(q_i)$ and $E(\sum_{i=1}^{d}(p_i + (-1)^{p_i} q_i)) = E(\Delta(p, q))$
5. Bob rerandomizes the $E(\Delta(p, q))$
6. Alice outputs $D_{SK}(E(\Delta(p, q))) = \Delta(x, y)$

The correctness of the protocol is straightforward. The key property for security is that if R is uniformly random, then for any PK, SK, the $E(z_1), \ldots, E(z_d)$ are independent uniformly random encryptions of random elements $z_1, \ldots, z_d \in [d]$.

To see complexity $d\lceil \log d \rceil + o(d)$, the list of s_i's that Alice sends has length $d\lceil \log d \rceil$. Also, $E(\Delta(q, p))$ has length k, the security parameter, which can be set to d^{ϵ} for any $\epsilon > 0$. Similar techniques give $d + O(k)$ communication for private inner product, using GM-encryption [37].

6.1.5 Private All-Near Neighbors

We consider the setting of [32], in which Alice and Bob have $Q = q_1, \ldots, q_n \in [U]^d$ and $P = p_1, \ldots, p_n \in [U]^d$ respectively, and Alice wants all p_j for which $\Delta(q_i, p_j) \leq t < d$ for some $i \in [n]$ and parameter t. We assume a semantically secure homomorphic encryption scheme E and OT with polylog(n) communication.

```
All-Near$(Q, P)$:

  1. The parties randomly permute their points
  2. Alice generates parameters $(PK, SK)$ of $E$ and sends Bob $PK$
  3. For $l = 1, \ldots, k$,
       • The parties choose a pairwise independent hash function $h : [U] \rightarrow [2d]$
       • For $i \in [n]$, Alice computes $\tilde{x}_i = h(x_i)$, where $h$ is applied coordinate-wise
       • Replace each entry $j$ of each $\tilde{x}_i$ with a length $2d$ unit vector with $r$th bit 1 iff $\tilde{x}_{i,j} = r$
       • Bob forms $\hat{y}_i$ similarly
       • Alice sends the coordinate-wise encryption of each vector for each coordinate of each $\tilde{x}_i$
       • Bob computes (by himself) $Z_{i,j,l} = E(\Delta(\tilde{x}_i, \tilde{y}_j))$ for all $i, j \in [n]$
  4. A secure circuit with inputs $SK, Z_{i,j,l}$ computes
       • $Z_{i,j} = \min_l D_{SK}(Z_{i,j,l})$
       • Output $Z = \{j \mid \exists i \text{ s.t. } Z_{i,j} \geq d - t\}$ to Alice
  5. Perform OT on records of size $d$ for Alice to retrieve $Y = \{y_j \mid j \in Z\}$
```

Theorem 94 *The above is a private all-near neighbors protocol with communication $\tilde{O}(nd^2 + n^2)$.*

Proof: We first argue correctness, which means showing $\Pr[Y = \{y_j \mid \exists i \text{ s.t. } \Delta(q_i, p_j) \leq t\}] = 1 - 2^{-\Omega(k)}$. We show for $i, j \in [n]$, $\Pr[\Delta(q_i, p_j) = n - Z_{i,j}] = 1 - 2^{-\Omega(k)}$. By a union bound, for any h,

$$\Pr[D(Z_{i,j}) = n - \Delta(q_i, p_j)] \geq T/2T = 1/2.$$

But $D(Z_{i,j}) \geq n - \Delta(q_i, p_j)$ since hashing only increases the number of agreements. Thus, $\Pr[\min_l D(Z_{i,j,l}) > n - \Delta(q_i, p_j)] < 2^{-\Omega(k)}$, so that $Z_{i,j} = n - \Delta(q_i, p_j)$ with the required probability.

For privacy, since the output assumes a specific value with probability $1 - 2^{-\Omega(k)}$, we just need to show each party's view is simulatable. As usual, we replace the SFE and OT by oracles. Alice's output from the SFE is a list of random indices, and her output from the OT is her protocol output. Hence, her simulator just outputs a list of $|Y|$ random indices. Bob's simulator chooses k random hash functions and $2d^2nk$ encryptions of 0 under E. By the semantic security of E and theorem 86, the protocol is secure.

To see that the communication is $\tilde{O}(nd^2+n^2)$, in each of k executions, Alice sends $O(nd^2)$ encryptions. Bob then inputs $O(n^2)$ encryptions to the SFE, which can be implemented with a circuit of size $\tilde{O}(n^2)$. Step 5 of the protocol can be done with $\tilde{O}(nd)$ communication using the best OT schemes (see [23, 17]). ■

Remark 95 A simple modification of the protocol gives the promised $\tilde{O}(d^2 + n)$ communication for Hamming distance in the setting of [28] for any U.

Remark 96 The protocol can be adapted to give $\tilde{O}(d+n)$ communication for set difference. In this case Alice has a single vector q. The idea is that Alice, Bob can hash their entries down to $2d$ values using h as in the protocol, and now Alice can homomorphically encrypt and send the coefficients of a degree-$(2d-1)$ polynomial pol, where pol is such that $pol(t) = 0$ if $t \in \{r \mid \exists i \text{ s.t. } r = h(q_i)\}$ and $pol(t) = 1$ otherwise. Bob can evaluate pol on each (hashed) coordinate of each p_i and use E's homomorphy to compute $E(f(\tilde{q}, \tilde{p_i}))$, f denoting set difference. We then repeat this k times over different h and take a maximum in the SFE. Since coordinate order is immaterial for set difference, we achieve $\tilde{O}(n + d)$ instead of $\tilde{O}(n + d^2)$ communication.

Although we have improved the communication of [32], one may worry about the work the parties need to perform. We have the following optimization:

Theorem 97 *The protocol can be implemented with total work $\tilde{O}(n^2 d^{2c-4})$, where $c \approx 2.376$ is the exponent of matrix multiplication.*

Proof: The work is dominated by step 3, in which Bob needs to compute encryptions of all pairwise Hamming distances. To reduce the work, we think of what Alice sends as an encrypted $n \times d^2$ matrix M_1, and that Bob has a $d^2 \times n$ matrix M_2 and needs an encrypted $M_1 M_2$. It is shown in [11] that even the best known matrix multiplication algorithm still works if one of the matrices is homomorphically encrypted. Thus Bob can perform $(n/d^2)^2$

fast multiplications of $d^2 \times d^2$ matrices, requiring $\tilde{O}((n/d^2)^2(d^2)^r) = \tilde{O}(n^2 d^{2r-4})$ work, which improves upon the $\tilde{O}(n^2d^2)$ work of a naive implementation. ■

6.2 Approximate Near Neighbor Problems

6.2.1 Private c-approximate Near Neighbor Problem

Suppose $q \in \{0,1\}^d$ and $p_i \in \{0,1\}^d$ for all i. Let $P_t = \{p \in P \mid \Delta(p,q) \leq t\}$, and $c > 1$ be a constant.

Definition 98 *A* **c-approximate NN protocol** *is correct if when $P_r \neq \emptyset$, Alice outputs a point $f(q,P) \in P_{cr}$ with probability $1 - 2^{-\Omega(k)}$. It is private if in the computational sense, Bob learns nothing, while Alice learns nothing except what follows from P_{cr}. Formally, Alice's privacy is implied by an efficient simulator Sim for which $\langle q, P, f(q,P)\rangle \overset{c}{\equiv} \langle q, P, Sim(1^n, P_{cr}, q)\rangle$ for* poly(d,n,k)*-time machines.*

Following [39], we say the protocol *leaks b bits of information* if there is a deterministic "hint" function $h : \{0,1\}^{(n+1)d} \to \{0,1\}^b$ such that the distributions $\langle q, P, f(q,P)\rangle$ and $\langle q, P, Sim(1^n, P_{cr}, q, h(P,q))\rangle$ are indistinguishable. We believe these to be natural extensions of private approximations in [29, 39] from values to sets of values.

We give a private c-approximate NN protocol with communication $\tilde{O}(\sqrt{n} + d)$ and a c-approximate NN protocol with communication $\tilde{O}(n^{1/3} + d)$ which leaks k bits of information. Both protocols are based on dimensionality reduction in the hypercube [54]. There it is shown that for an $O(\log n) \times d$ matrix A with entries i.i.d. Bernoulli$(1/d)$, there is an $\tau = \tau(r, cr)$ such that for all $p, q \in \{0,1\}^d$, the following event holds with probability at least $1 - 1/\text{poly}(n)$

$$\text{If } \Delta(p,q) \leq r, \text{ then } \Delta(Ap, Aq) \leq \tau, \text{ and if } \Delta(p,q) \geq cr, \text{ then } \Delta(Ap, Aq) > \tau.$$

Here, arithmetic occurs in $\mathbb{Z}_2$. We use this idea in the following helper protocol $\mathsf{DimReduce}(\tau, B, q, P)$. Let A be a random matrix as described above. Let $S = \{p \in P \mid \Delta(Ap, Aq) \leq \tau\}$. If $|S| > B$, replace S with the lexicographically first B elements of S. $\mathsf{DimReduce}$ outputs random shares of S.

DimReduce (τ, B, q, P):

1. Bob performs the following computation
   - Generate a matrix A as above, and initialize L to an empty list.
   - For each $v \in \{0,1\}^{O(\log n)}$, let $L(v)$ be the first B p_i for which $\Delta(Ap_i, v) \leq \tau$.
2. A secure circuit with ROM L performs the following computation on input (q, A),
   - Compute Aq.
   - Lookup Aq in L to obtain S. If $|S| < B$, pad S so that all S have the same length.
   - Output random shares (S^1, S^2) of S so that $S = S^1 \oplus S^2$.

It is an easy exercise to show the correctness and privacy of DimReduce.

Remark 99 As stated, the communication is $\tilde{O}(dB)$. The dependence on d can be improved to $\tilde{O}(d+B)$ using homomorphic encryption. Roughly, Alice sends $E(q_1), \ldots, E(q_d)$ to Bob, who sets $L(v)$ to be the first B different $E(\Delta(p_i, q))$ for which $\Delta(Ap_i, v) \leq \tau$. Note that $E(\Delta(p_i, q))$ is efficiently computable, and has size $\tilde{O}(1) \ll d$.

It will be useful to define the following event $\mathcal{H}(r_1, r_2, P)$ with $r_1 < r_2$. Suppose we run DimReduce independently k times with matrices A_i. Then $\mathcal{H}(r_1, r_2, P)$ is the event that at least $k/2$ different i satisfy

$$\forall p \in P_{r_1},\ \Delta(A_i p, A_i q) \leq \tau(r_1, r_2) \text{ and } \forall p \in P \setminus P_{r_2},\ \Delta(A_i p, A_i q) > \tau(r_1, r_2).$$

The next lemma follows from the properties of the A_i and standard Chernoff bounds:

Lemma 100 $\Pr[\mathcal{H}(r_1, r_2, P)] = 1 - 2^{-\Omega(k)}$.

6.2.2 Reducing the Dependence on d for Private c-approximate NN

Here we sketch how the communication of the protocol of section 6.2.3 can be reduced to $\tilde{O}(n^{1/2} + \text{polylog}(d))$ if Alice just wants to privately learn some coordinate of some element of P_{cr}.

Proof Sketch: The idea is to perform an approximation to the Hamming distance instead of using the $E(\Delta(p_i, q))$ in the current protocol (see, e.g., $\mathsf{DimReduce}$, and the following remark). The approximation we use is that given in [54], namely, the parties will agree upon random matrices A_i for some subset of i in $[n]$, and from the $A_i p_i$ and $A_i q$ will determine $(1 \pm \epsilon)$ approximations to the $\Delta(p_i, q)$ with probability $1 - 2^{-k}$. We don't need private approximations since the parties will not learn these values, but rather, they will input the $A_i p_i, A_i q$ into a secure circuit which makes decisions based on these approximations.

More precisely, Bob samples B of his vectors p_i, and in parallel agrees upon B matrices A_i and feeds the $A_i p_i$ into a secure circuit. Alice feeds in the $A_i q$. Let $c \geq 1 + 8\epsilon$. The circuit looks for an approximation of at most $r(1 + 6\epsilon)$. If such a value exists, the circuit gives Alice the corresponding index. Observe that if $|P_{r(1+4\epsilon)}| > \sqrt{n}$, then with probability $1 - 2^{-k}$ an index is returned to an element in P_{cr}, and that this distribution is simulatable. So assume $|P_{r(1+4\epsilon)}| \leq \sqrt{n}$.

The parties proceed by performing a variant of $\mathsf{DimReduce}(\tau(r, r(1+4\epsilon)), B, q, P)$, with the important difference being that the output no longer consists of shares of the $E(\Delta(p_i, q))$. Instead, for each entry $L(v)$, Bob pretends he is running the approximation of [54] with Alice's point q. That is, the parties agree on B different matrices A_i and Bob computes $A_i p$ for each $p \in L(v)$. A secure circuit obtains these products, and computes the approximations. It outputs an index to a random element with approximation at most $r(1 + 2\epsilon)$. If P_r is nonempty, such an index will exist with probability $1 - 2^{-k}$. Also, the probability that an index to an element outside of $P_{r(1+4\epsilon)}$ is returned is less than 2^{-k}, and so the distribution of the index returned is simulatable.

Finally, given the index of some element in P_{cr}, the parties perform OT and Alice obtains the desired coordinate, The communication is now $\tilde{O}(\sqrt{n})$. □

6.2.3 c-approximate NN Protocol

Protocol Overview: Our protocol is based on the following intuition. When $|P_{cr}|$ is large, a simple solution is to run a secure function evaluation with Alice's point q as input, together with a random sample P' of roughly a $k/|P_{cr}|$ fraction of Bob's points P. The circuit returns a random point of $P' \cap P_{cr}$, which is non-empty with overwhelming probability. The communication is $\tilde{O}(n/|P_{cr}|)$.

On the other hand, when $|P_{cr}|$ is small, if Alice and Bob run $\mathsf{DimReduce}(\tau(r, cr), |P_{cr}|, q, P)$ independently k times, then with overwhelming probability $P_r \subseteq \cup_i S_i$, where S_i denotes the (randomly shared) output in the ith execution. A secure function evaluation can then take in the random shares of the S_i and output a random point of P_r. The communication of this scheme is $\tilde{O}(|P_{cr}|)$.

Our protocol combines these two protocols to achieve $\tilde{O}(\sqrt{n})$ communication, by sampling roughly an $n^{-1/2}$ fraction of Bob's points in the first protocol, and by invoking $\mathsf{DimReduce}$ with parameter $B = \tilde{O}(\sqrt{n})$ in the second protocol. This approach is similar in spirit to the "high distance / low distance" approach used to privately approximate the Hamming distance in [29].

c-**Approx** (q, P):

1. Set $B = \tilde{O}(\sqrt{n})$.
2. Independently run $\mathsf{DimReduce}(\tau(r, cr), B, q, P)$ k times, generating shares (S_i^1, S_i^2).
3. Bob finds a random subset P' of P of size B.
4. A secure circuit performs the following computation on inputs q, S_i^1, S_i^2, P'.
    - Compute $S_i = S_i^1 \oplus S_i^2$ for all i.
    - Let $f(q, P)$ be a random point from $P_{cr} \cap P' \neq \emptyset$ if it is non-empty,
    - Else let $f(q, P)$ be a random point from $P_r \cap \cup_i S_i$ if it is non-empty, else set $f(q, P) = \emptyset$.
    - Output $(f(q, P), \mathsf{null})$.

Using the ideas in Remark 99, the communication is $\tilde{O}(d+B)$, since the SFE has size $\tilde{O}(B)$. Let $\mathcal{F}$ be the event that $P' \cap P_{cr} \neq \emptyset$, and put $\mathcal{H} = \mathcal{H}(r, cr, P)$.

Correctness: Suppose P_r is nonempty. The probability s of correctness is just the probability we don't output $\emptyset$. Thus $s \geq \Pr[\mathcal{F}] + \Pr[\neg\mathcal{F}] \Pr[f(q, P) \neq \emptyset \mid \neg\mathcal{F}]$.

Case $|P_{cr}| \geq \sqrt{n}$: For sufficiently large B, we have $s \geq \Pr[\mathcal{F}] = 1 - 2^{-\Omega(k)}$.

Case $|P_{cr}| < \sqrt{n}$: It suffices to show $\Pr[f(q,P) \neq \emptyset \mid \neg\mathcal{F}] = 1 - 2^{-\Omega(k)}$. But this probability is at least $\Pr[f(q,P) \neq \emptyset \mid \mathcal{H}, \neg\mathcal{F}]\Pr[\mathcal{H}]$, and if $\mathcal{H}$ occurs, then $f(q,P) \neq \emptyset$. By Lemma 100, $\Pr[\mathcal{H}] = 1 - 2^{-\Omega(k)}$.

Privacy Note that Bob gets no output, so Alice's privacy follows from the composition of of DimReduce and the secure circuit protocol of step 5. Similarly, if we can construct a simulator *Sim* with inputs $1^n, P_{cr}, q$ so that the distributions $\langle q, P, f(q,P)\rangle$ and $\langle q, P, Sim(1^n, P_{cr}, q)\rangle$ are statistically close, Bob's privacy will follow by that of DimReduce and the secure circuit protocol of step 5.

```
Sim ($1^n, P_{cr}, q$):

    1. Set $B = \tilde{O}(n^{1/2})$.

    2. With probability $1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, output a random element of $P_{cr}$,

    3. Else output a random element of $P_r$.
```

Let X denote the output of $Sim(1^n, P_{cr}, q)$. It suffices to show that for each $p \in P$, $|\Pr[f(q,P) = p] - \Pr[X = p]| = 2^{-\Omega(k)}$, since this also implies $|\Pr[f(q,P) = \emptyset] - \Pr[X = \emptyset]| = 2^{-\Omega(k)}$. We have

$$\begin{aligned}
\Pr[f(q,P) = p] &= \Pr[f(q,P) = p, \mathcal{F}] + \Pr[f(q,P) = p, \neg\mathcal{F}] \\
&= \Pr[f(q,P) = p, \mathcal{F}] + \Pr[f(q,P) = p, \neg\mathcal{F} \mid \mathcal{H}] \pm 2^{-\Omega(k)} \\
&= \Pr[\mathcal{F}]\,|P_{cr}|^{-1} + \Pr[\neg\mathcal{F}]\Pr[f(q,P) = p \mid \mathcal{H}, \neg\mathcal{F}] \pm 2^{-\Omega(k)},
\end{aligned}$$

where we have used Lemma 100. Since $\Pr[\mathcal{F}] = 1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, we have

$$|\Pr[f(q,P) = p] - \Pr[X = p]| \leq \Pr[\neg\mathcal{F}]\left|\Pr[f(q,P) = p \mid \mathcal{H}, \neg\mathcal{F}] - \delta(p \in P_r)|P_r|^{-1}\right| + 2^{-\Omega(k)}.$$

If $|P_{cr}| \geq \sqrt{n}$, then $\Pr[\neg\mathcal{F}] = 2^{-\Omega(k)}$. If $|P_{cr}| < \sqrt{n}$, then $\Pr[f(q,P) = p \mid \mathcal{H}, \neg\mathcal{F}] = \delta(p \in P_r)|P_r|^{-1}$.

Extensions: The way the current problem is stated, there is an $\Omega(d)$ lower bound. In appendix 6.2.2 we sketch how, if Alice just wants to learn some coordinate of an element of P_{cr}, this dependence can be made polylogarithmic. We also have a similar protocol based on locality-sensitive hashing (LSH), which only achieves $\tilde{O}(n^{1/2+1/(2c)}+d)$ communication, but has much smaller time complexity (though still polynomial). More precisely, the work of the LSH scheme is $n^{O(1)}$, whereas the work of c-**Approx** is $n^{O(1/(c-1)^2)}$, which is polynomial only for constant c. See Appendix 6.2.4 for the details.

6.2.4 Private c-approximate NN Based on Locality Sensitive Hashing

We give an alternative private c-approximate NN protocol, with slightly more communication than that in section 6.2.1, but less work (though still polynomial). It is based on locality sensitive hashing (LSH) [45]. The fact we need is that there is a family of functions $\mathcal{G} : \{0,1\}^d \to \{0,1\}^{\tilde{O}(1)}$ such that each $g \in \mathcal{G}$ has description size $\tilde{O}(1)$, and $\mathcal{G}$ is such that for all $p, q \in \{0,1\}^d$,

$$\Pr_{g \in \mathcal{G}}[g(p) = g(q)] = \Theta\left(n^{-\Delta(p,q)/cr}\right)$$

Recall that Alice has a point $q \in \{0,1\}^d$ and Bob has n points $P \subseteq \{0,1\}^d$. For correctness, Alice should learn a point of P_{cr} provided $P_r \neq \emptyset$. For privacy, her view should be simulatable given only P_{cr}.

Our protocol is similar to that in section 6.2.1. When $|P_{cr}|$ is large, one can run a secure function evaluation with Alice's point q as input, together with a random sample P' of roughly a $k/|P_{cr}|$ fraction of Bob's points P. The circuit returns a random point of $P' \cap P_{cr}$ which is non-empty with probabiity $1 - 2^{-\Omega(k)}$. The communication is $\tilde{O}(n/|P_{cr}|)$.

On the other hand, when $|P_{cr}|$ is small, if Alice and Bob exchange functions g_i independently $\tilde{O}(n^{1/c})$ times, then with overwhelming probability $P_r \subseteq \cup_i S_i$, where S_i denotes the subset of Bob's points p with $g_i(p) = g_i(q)$. Using a secure ciruit with ROM, we can obtain these sets S_i, and output a random point of P_r. The communication is $\tilde{O}(n^{1/c}|P_{cr}|)$.

Our protocol balances these approaches to achieve $\tilde{O}(n^{1/2+1/(2c)})$ communication.

There are a few technicalities dodged by this intuition. First, even though the parties exchange $\tilde{O}(n^{1/c})$ different g_i, and can thus guarantee that each p is in some S_i with probability $1 - 2^{-\Omega(k)}$, it may be that whenever $p \in S_i$, many points from $P \setminus P_{cr}$ also land in S_i, so that S_i is very large. Even though we only expect $|P \setminus P_{cr}|O(1/n) = O(1)$ points from

$P \setminus P_{cr}$ in S_i, since $\Pr[p \in S_i] = \Theta(n^{-1/c})$ is small, p may only be in S_i when S_i is large. Because the size of the S_i affects the communication of our protocol, we cannot always afford for the ROM to receive the whole S_i (sometimes we will truncate it). However, in the analysis, we show that the average S_i is small, and this will be enough to get by with low communication.

Second, we need to extend the notion of a lookup gate given earlier. Instead of just mapping inputs (i, j) to output $R_i[j]$, the jth entry in the ith party's ROM, we also allow j to be a key, so that the output is the record in R_i keyed by j. This can be done efficiently using [23], and Theorem 88 is unchanged, assuming the length of the keys is $\tilde{O}(1)$.

```
LSH $(q, P)$:

1. Set $B = \tilde{O}(n^{1/2+1/(2c)})$ and $C = \tilde{O}(n^{1/c})$.
2. Bob finds a random subset $P'$ of $P$ of size $B$ .
3. For $i = 1$ to $k$,
    (a) Alice and Bob agree upon $C$ random $g_{i,j} \in \mathcal{G}$.
    (b) Bob creates a ROM $L$ with entries $L(v)$ containing the points $p$ for which $g(p) = v$.
    (c) A secure circuit with ROM $L$ performs the following computation on input $(q, \{g_{i,j}\})$,
        • Compute $v_{i,j} = g_{i,j}(q)$ for each $j$.
        • Lookup the $L(v_{i,j})$ one by one for the different $v_{i,j}$ until the communication exceeds $dB$. If it is less, make dummy queries so that it is exactly $dB$.
        • Output shares $S_i^1, S_i^2$ so that $S_i^1 \oplus S_i^2$ is the (possibly truncated) set of sets $L(v_j)$.
4. A secure circuit with inputs $P'$, $S_i^1, S_i^2$,
    • Compute the set $S_i = S_i^1 \oplus S_i^2 = \cup_j L(v_j)$ for all $i$.
    • Let $f(q, P)$ be random in $P_{cr} \cap P'$ if it is non-empty.
    • Else let $f(q, P)$ be random in $P_r \cap \cup_i S_i$ if it is non-empty, else set $f(q, P) = \emptyset$.
    • Output $(f(q, P), \mathsf{null})$.
```

The communication is $\tilde{O}(dB)$. By using homomorphic encryption, one can reduce the dependence on d, as per remark 99. Let $\mathcal{E}$ be the event that $P_r \subseteq \cup_i S_i$, and let $\mathcal{F}$ be the event that $P_{cr} \cap P'$ is non-empty.

Correctness: Suppose $P_r \neq \emptyset$. The probability s of correctness is just the probability we don't output $\emptyset$. Thus $s \geq \Pr[\mathcal{F}] + \Pr[\neg\mathcal{F}] \Pr[f(q, P) \neq \emptyset \mid \neg\mathcal{F}]$.

Case $|P_{cr}| \geq n^{1/2-1/(2c)}$: For sufficiently large B, we have $s \geq \Pr[\mathcal{F}] = 1 - 2^{-\Omega(k)}$.

Case $|P_{cr}| < n^{1/2-1/(2c)}$: It is enough to show $\Pr[f(q,P) \neq \emptyset \mid \neg\mathcal{F}] = 1 - 2^{-\Omega(k)}$. Fix i. Put $Y = \sum_j |L(v_{i,j})|$, where $|L(v_{i,j})|$ denotes the number of points in $L(v_{i,j})$. The expected number of points in $P \setminus P_{cr}$ that are in $L(v_{i,j})$ is at most $n \cdot O(1/n) = O(1)$. Since $|P_{cr}| < n^{1/2-1/(2c)}$, $\mathbf{E}[L(v_{i,j})] < n^{1/2-1/(2c)} + O(1)$. Thus $\mathbf{E}[Y] \leq B/3$ for large enough B, so $\Pr[Y > B] \leq 1/3$ by Markov's inequality. Thus, with probability $1 - 2^{-\Omega(k)}$, for at least half of the i, S_i is not truncated in step 3c. Moreover, for large enough B, any i, and any $p \in P_r$, $\Pr[p \in S_i] = 1 - 2^{-\Omega(k)}$ for large enough C. By a few union bounds then, $\Pr[P_r \subseteq \cup_i S_i] = \Pr[\mathcal{E}] = 1 - 2^{-\Omega(k)}$. Thus,

$$\Pr[f(q,P) \neq \emptyset \mid \neg\mathcal{F}] \geq \Pr[f(q,P) \neq \emptyset,\, \mathcal{E} \mid \neg\mathcal{F}] = \Pr[f(q,P) \neq \emptyset \mid \mathcal{E},\, \neg\mathcal{F}]\Pr[\mathcal{E}] \geq 1-2^{-\Omega(k)}.$$

Privacy: Note that Bob gets no output, so Alice's privacy follows from that of the secure circuit protocol. We construct a simulator $Sim(1^n, P_{cr}, q)$ so that the distributions $\langle q, P, f(q,P)\rangle$ and $\langle q, P, Sim(1^n, P_{cr}, q)\rangle$ are statistically close. Bob's privacy then follows by the composition with the secure circuit protocol.

```
Sim ($1^n, P_{cr}, q$):

  1. Set $B = \tilde{O}(n^{1/2+1/(2c)})$.

  2. With probabiity $1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, output a random element of $P_{cr}$.

  3. Else output a random element of $P_r$.
```

Let X denote the output of $Sim(1^n, P_{cr}, q)$. It suffices to show that for each $p \in P$, $|\Pr[f(q,P) = p] - \Pr[X = p]| = 2^{-\Omega(k)}$, since this also implies $|\Pr[f(q,P) = \emptyset] - \Pr[X = \emptyset]| = 2^{-\Omega(k)}$. We have

$$\begin{aligned}
\Pr\left[f(q,P) = p\right] &= \Pr\left[f(q,P) = p, \mathcal{F}\right] + \Pr\left[f(q,P) = p, \neg\mathcal{F}\right] \\
&= \Pr\left[\mathcal{F}\right] |P_{cr}|^{-1} + \Pr\left[f(q,P) = p, \neg\mathcal{F}\right]
\end{aligned}$$

Note that $\Pr[\mathcal{F}] = 1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$. Therefore,

$$|\Pr[f(q,P) = p] - \Pr[X = p]| = \Pr[\neg\mathcal{F}] |\Pr\left[f(q,P) = p \mid \neg\mathcal{F}\right] - \delta(p \in P_r)|P_r|^{-1}|.$$

If $|P_{cr}| \geq n^{1/2-1/(2c)}$, this is $2^{-\Omega(k)}$, since then $\Pr[\neg\mathcal{F}] = 2^{-\Omega(k)}$. Otherwise, $|P_{cr}| < n^{1/2-1/(2c)}$, and as shown in the proof of correctness, we have $\Pr[\mathcal{E}] = \Pr[P_r \subseteq \cup_i S_i] = 1 - 2^{-\Omega(k)}$. Thus

$$\Pr[f(q,P) = p \mid \neg\mathcal{F}] = \Pr[f(q,P) = p \mid \mathcal{E},\ \neg\mathcal{F}]\Pr[\mathcal{E}] \pm 2^{-\Omega(k)} = \delta(p \in P_r)|P_r|^{-1} \pm 2^{-\Omega(k)},$$

which completes the proof.

6.2.5 c-approximate NN Protocol Leaking k Bits

Protocol Overview: We consider three balls $P_r \subseteq P_{br} \subseteq P_{cr}$, where $c - b, b - 1 \in \Theta(1)$. We start by trying to use dimensionality reduction to separate P_r from $P \setminus P_{br}$, and to output a random point of P_r. If this fails, we try to sample and output a random point of P_{cr}. If this also fails, then it will likely hold that $n^{1/3} \leq |P_{br}| \leq |P_{cr}| \leq n^{2/3}$. We then sample down the pointset P by a factor of $n^{-1/3}$, obtaining $\tilde{P}$ with survivors $\tilde{P}_{br}, \tilde{P}_{cr}$ of P_{br}, P_{cr} respectively. It will now likely hold that we can use dimensionality reduction to separate $\tilde{P}_{br}$ from $\tilde{P} \setminus \tilde{P}_{cr}$ to obtain and output a random point of $\tilde{P}_{br}$. The hint function will encode the probability, to the nearest multiple of 2^{-k}, that the first dimensionality reduction fails, which may be a non-negligible function of $P \setminus P_{cr}$. This hint will be enough to simulate the entire protocol.

***c*-ApproxWithHelp** (q, P):

1. Set $B = \tilde{O}(n^{1/3})$.
2. Independently run $\mathsf{DimReduce}(\tau(r, br), B, q, P)$ k times, generating shares (S_i^1, S_i^2).
3. Bob finds random subsets $P', \tilde{P}$ of P of respective sizes B and $n^{2/3}$.
4. Independently run $\mathsf{DimReduce}(\tau(br, cr), B, q, \tilde{P})$ k times, generating shares $(\tilde{S}_i^1, \tilde{S}_i^2)$.
5. A secure circuit performs the following computation on inputs $q, S_i^1, S_i^2, P', \tilde{S}_i^1, \tilde{S}_i^2$.
   - Compute $S_i = S_i^1 \oplus S_i^2$ and $\tilde{S}_i = \tilde{S}_i^1 \oplus \tilde{S}_i^2$ for all i.
   - If for most i, $|S_i| < B$, let $f(q, P)$ be a random point in $P_r \cap \cup_i S_i$, or $\emptyset$ if it is empty.
   - Else if $P_{cr} \cap P' \neq \emptyset$, let $f(q, P)$ be a random point in $P_{cr} \cap P'$.
   - Else let $f(q, P)$ be a random point in $P_{br} \cap \cup_i \tilde{S}_i$ if it is non-empty, otherwise set $f(q, P) = \emptyset$.
   - Output $(f(q, P), \mathsf{null})$.

The protocol can be implemented in polynomial time with communication $\tilde{O}(B + d) = \tilde{O}(n^{1/3} + d)$.

To prove correctness and privacy, we introduce some notation. Let $\mathcal{E}_1$ be the event that the majority of the $|S_i|$ are less than B, and $\mathcal{E}_2$ the event that $P_r \subseteq \cup_i S_i$. Let $\mathcal{F}$ be the event that $P' \cap P_{cr} \neq \emptyset$. Let $\mathcal{G}_1$ be the event that $1 \leq \tilde{P}_{br} \leq \tilde{P}_{cr} \leq B$ and $\mathcal{G}_2$ the event that $\tilde{P}_{br} \subseteq \cup_i \tilde{S}_i$. Finally, let $\mathcal{H}_1 = \mathcal{H}(r, br, P)$ and $\mathcal{H}_2 = \mathcal{H}(br, cr, \tilde{P})$. Note that $\Pr[\mathcal{H}_1], \Pr[\mathcal{H}_2]$ are $1 - 2^{-\Omega(k)}$ by Lemma 100. We need two lemmas:

Lemma 101 $\Pr[\mathcal{E}_2 \mid \mathcal{E}_1] = 1 - 2^{-\Omega(k)}$.

Proof: If $\mathcal{H}_1$ and $\mathcal{E}_1$ occur, then there is an i for which $P_r \subseteq S_i$, so $\mathcal{E}_2$ occurs. ■

Lemma 102 $\Pr[\mathcal{G}_2 \mid \mathcal{G}_1] = 1 - 2^{-\Omega(k)}$.

Proof: If $\mathcal{H}_2$ and $\mathcal{E}_2$ occur, then the majority of the $\tilde{S}_i$ contain $\tilde{P}_{br}$, so $\mathcal{G}_2$ occurs. ■

Correctness: We may assume $P_r \neq \emptyset$. The probability s of correctness is just the probability the algorithm doesn't return $\emptyset$. Since $\mathcal{F}, \mathcal{E}_1$, and $\mathcal{G}_1$ are independent,

$$s \geq \Pr[\mathcal{E}_1] \Pr[\mathcal{E}_2 \mid \mathcal{E}_1] + \Pr[\neg \mathcal{E}_1](\Pr[\mathcal{F}] + \Pr[\neg \mathcal{F}] \Pr[\mathcal{G}_1] \Pr[\mathcal{G}_2 \mid \mathcal{G}_1]).$$

Case $|P_{br}| < B$: $\mathcal{H}_1$ implies $\mathcal{E}_1$ since $|P_{br}| < B$, and using Lemma 101, $s \geq \Pr[\mathcal{E}_1] \Pr[\mathcal{E}_2 \mid \mathcal{E}_1] = 1 - 2^{-\Omega(k)}$.

Case $|P_{br}| \geq B$: Since $\Pr[\mathcal{E}_2 \mid \mathcal{E}_1] = 1 - 2^{-\Omega(k)}$ by Lemma 101, we just need to show that $\Pr[\mathcal{F}] + \Pr[\neg \mathcal{F}] \Pr[\mathcal{G}_1] \Pr[\mathcal{G}_2 \mid \mathcal{G}_1] = 1 - 2^{-\Omega(k)}$. If $|P_{cr}| > n^{2/3}$, it suffices to show $\Pr[\mathcal{F}] = 1 - 2^{-\Omega(k)}$. This holds for large enough $B = \tilde{O}(n^{1/3})$. Otherwise, if $|P_{cr}| \leq n^{2/3}$, then it suffices to show $\Pr[\mathcal{G}_1] \Pr[\mathcal{G}_2 \mid \mathcal{G}_1] = 1 - 2^{-\Omega(k)}$. By assumption, $B \leq |P_{br}| \leq |P_{cr}| \leq n^{2/3}$. Therefore, for large enough B, $\Pr[\mathcal{G}_1] = 1 - 2^{-\Omega(k)}$, and thus by Lemma 102, $\Pr[\mathcal{G}_1] \Pr[\mathcal{G}_2 \mid \mathcal{G}_1] = 1 - 2^{-\Omega(k)}$.

Privacy: Note that Bob gets no output, so Alice's privacy follows from the composition of DimReduce and the secure circuit protocol of step 5. Similarly, if we can construct a simulator Sim with inputs $1^n, P_{cr}, q, h(P_{cr}, q)$ so that the distributions $\langle q, P, f(q, P) \rangle$ and $\langle q, P, Sim(1^n, P_{cr}, q, h(P_{cr}, q)) \rangle$ are statistically close, Bob's privacy will follow by that of DimReduce and the secure circuit of step 5.

We define the hint function $h(P_{cr}, q)$ to output the nearest multiple of 2^{-k} to $\Pr[\mathcal{E}_1]$. In the analysis we may assume that Sim knows $\Pr[\mathcal{E}_1]$ exactly, since its output distribution in this case will be statistically close to its real output distribution.

```
Sim $(1^n, P_{cr}, q, \Pr[\mathcal{E}_1])$:

    1. Set $B = \tilde{O}(n^{1/3})$.

    2. With probabiity $\Pr[\mathcal{E}_1]$, output a random element of $P_r$, or output $\emptyset$ if $P_r = \emptyset$.

    3. Else with probability $1 - \binom{n-|P_{cr}|}{B} \binom{n}{B}^{-1}$, output a random element of $P_{cr}$,

    4. Else output a random element of $P_{br}$.
```

Let X denote the output of $Sim(1^n, P_{cr}, q, \Pr[\mathcal{E}_1])$. It suffices to show that for each $p \in P$,

$$|\Pr[f(q,P) = p] - \Pr[X = p]| = 2^{-\Omega(k)},$$

since then we have $|\Pr[f(q,P) = \emptyset] - \Pr[X = \emptyset]| = 2^{-\Omega(k)}$. Using the independence of $\mathcal{F}, \mathcal{E}_1, \mathcal{G}_1$, and Lemmas 101, 102, we bound $\Pr[f(q,P) = p]$ as follows

$$\begin{aligned}
& \Pr[f(q,P) = p] = \Pr[\mathcal{E}_1, f(q,P) = p] + \Pr[\neg\mathcal{E}_1, f(q,P) = p] \\
= \;& \Pr[\mathcal{E}_1]\Pr[f(q,P) = p \mid \mathcal{E}_2\mathcal{E}_1] \pm 2^{-\Omega(k)} + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{F}]\Pr[f(q,P) = p \mid F, \neg\mathcal{E}_1] \\
+ \;& \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[f(q,P) = p \mid \neg F, \neg\mathcal{E}_1] \\
= \;& \Pr[\mathcal{E}_1]|P_r|^{-1}\delta(p \in P_r) \pm 2^{-\Omega(k)} + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{F}]|P_{cr}|^{-1} \\
+ \;& \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\mathcal{G}_1]\Pr[f(q,P) = p \mid \mathcal{G}_1\mathcal{G}_2\neg\mathcal{F}\neg\mathcal{E}_1] \pm 2^{-\Omega(k)} \\
+ \;& \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\neg\mathcal{G}_1]\Pr[f(q,P) = p \mid \neg\mathcal{G}_1\neg\mathcal{F}\neg\mathcal{E}_1] \\
= \;& \Pr[\mathcal{E}_1]|P_r|^{-1}\delta(p \in P_r) + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{F}]|P_{cr}|^{-1} + \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\mathcal{G}_1]|P_{br}|^{-1}\delta(p \in P_{br}) \\
+ \;& \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\neg\mathcal{G}_1]\Pr[f(q,P) = p \mid \neg\mathcal{E}_1\neg\mathcal{F}\neg\mathcal{G}_1] \pm 2^{-\Omega(k)}.
\end{aligned}$$

On the other hand, since $\Pr[\mathcal{F}] = 1 - \binom{n-|P_{cr}|}{B}\binom{n}{B}^{-1}$, we have

$$\Pr[X = p] = \Pr[\mathcal{E}_1]|P_r|^{-1}\delta(p \in P_r) + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{F}]|P_{cr}|^{-1} + \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]|P_{br}|^{-1}\delta(p \in P_{br}),$$

so that

$$|\Pr[f(q,P) = p] - \Pr[X = p]| \le \Pr[\neg\mathcal{E}_1]\Pr[\neg\mathcal{F}]\Pr[\neg\mathcal{G}_1]\Pr[f(q,P) = p \mid \neg\mathcal{E}_1\neg\mathcal{F}\neg\mathcal{G}_1] + 2^{-\Omega(k)}.$$

If $|P_{br}| < B$, $\Pr[\neg\mathcal{E}_1] = 2^{-\Omega(k)}$. If $|P_{cr}| \ge n^{2/3}$, $\Pr[\neg\mathcal{F}] = 2^{-\Omega(k)}$. Otherwise $B \le |P_{br}| \le |P_{cr}| \le n^{2/3}$, and as shown for correctness, $\Pr[\neg\mathcal{G}_1] = 2^{-\Omega(k)}$, which shows $|\Pr[f(q,P) = p] - \Pr[X = p]| = 2^{-\Omega(k)}$.

Bibliography

[1] A. Akella, A. Bharambe, M. Reiter, and S. Seshan. Detecting ddos attacks on isp networks, 2003.

[2] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *STOC*, pages 20–29, 1996.

[3] Noga Alon and Joel Spencer. *The Probabilistic Method.* John Wiley, 1992.

[4] Alexandr Andoni, Piotr Indyk, and Mihai Patrascu. On the optimality of the dimensionality reduction method. In *FOCS*, pages 449–458, 2006.

[5] Z. Bar-Yossef. The complexity of massive data set computations, 2002.

[6] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *FOCS*, pages 209–218, 2002.

[7] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. Information theory methods in communication complexity. In *IEEE Conference on Computational Complexity*, pages 93–102, 2002.

[8] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An easy $\omega(n)$ lower bound for a gap hamming distance problem, 2004.

[9] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *RANDOM*, pages 1–10, 2002.

[10] Amos Beimel, Renen Hallak, and Kobbi Nissim. Private approximation of clustering and vertex cover. In *TCC*, pages 383–403, 2007.

[11] Amos Beimel, Yuval Ishai, and Tal Malkin. Reducing the servers computation in private information retrieval: Pir with preprocessing. In *CRYPTO*, pages 55–73, 2000.

[12] J. D. C. Benaloh. Verifiable secret-ballot elections., 1987.

[13] Krishna Bharat and Andrei Z. Broder. A technique for measuring the relative size and overlap of public web search engines. *Computer Networks*, 30(1-7):379–388, 1998.

[14] Lakshminath Bhuvanagiri and Sumit Ganguly. Estimating entropy over data streams. In *ESA*, pages 148–159, 2006.

[15] Lakshminath Bhuvanagiri, Sumit Ganguly, Deepanjan Kesh, and Chandan Saha. Simpler algorithm for estimating frequency moments of data streams. In *SODA*, pages 708–713, 2006.

[16] Christian Cachin, Jan Camenisch, Joe Kilian, and Joy Müller. One-round secure computation and secure autonomous mobile agents. In *ICALP*, pages 512–523, 2000.

[17] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally private information retrieval with polylogarithmic communication. In *EUROCRYPT*, pages 402–414, 1999.

[18] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503, 2002.

[19] Amit Chakrabarti, Graham Cormode, and Andrew McGregor. A near-optimal algorithm for computing the entropy of a stream. In *SODA*, pages 328–335, 2007.

[20] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *IEEE Conference on Computational Complexity*, pages 107–117, 2003.

[21] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *ICALP*, pages 693–703, 2002.

[22] Chmielewski and Hoepman. Fuzzy private matching. In *Manuscript*, 2006.

[23] B. Chor, N. Gilboa, and M. Naor. Private information retrieval by keywords, 1997.

[24] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *FOCS*, pages 41–50, 1995.

[25] Don Coppersmith and Ravi Kumar. An improved data stream algorithm for frequency moments. In *SODA*, pages 151–156, 2004.

[26] Graham Cormode and S. Muthukrishnan. Summarizing and mining skewed data streams. In *SDM*, 2005.

[27] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991.

[28] W. Du and M. Atallah. Protocols for secure remote database access with approximate matching, 2000.

[29] Joan Feigenbaum, Yuval Ishai, Tal Malkin, Kobbi Nissim, Martin J. Strauss, and Rebecca N. Wright. Secure multiparty computation of approximations. *ACM Transactions on Algorithms*, 2(3):435–472, 2006.

[30] W. Feller. *An Introduction to Probability Theory and its Applications.* John Wiley, 1968.

[31] Philippe Flajolet and G. Nigel Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

[32] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *EUROCRYPT*, pages 1–19, 2004.

[33] Sumit Ganguly. Estimating frequency moments of data streams using random linear combinations. In *APPROX-RANDOM*, pages 369–380, 2004.

[34] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *STOC*, pages 151–160, 1998.

[35] Oded Goldreich. Secure multi-party computation. Working Draft, 2000.

[36] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.

[37] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[38] I.J. Good. Surprise indexes and p-values. *J. of Statistical Computation and Simulation*, 32:90–92, 1989.

[39] Shai Halevi, Robert Krauthgamer, Eyal Kushilevitz, and Kobbi Nissim. Private approximation of np-hard functions. In *STOC*, pages 550–559, 2001.

[40] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[41] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235, 1989.

[42] P. Indyk. High-dimensional computational geometry, 2000.

[43] Piotr Indyk. A small approximately min-wise independent family of hash functions. *J. Algorithms*, 38(1):84–90, 2001.

[44] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323, 2006.

[45] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613, 1998.

[46] Piotr Indyk and David P. Woodruff. Tight lower bounds for the distinct elements problem. In *FOCS*, pages 283–, 2003.

[47] Piotr Indyk and David P. Woodruff. Optimal approximations of the frequency moments of data streams. In *STOC*, pages 202–208, 2005.

[48] Piotr Indyk and David P. Woodruff. Polylogarithmic private approximations and efficient matching. In *TCC*, pages 245–264, 2006.

[49] W. Johnson and J. Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[50] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discrete Math.*, 5(4):545–557, 1992.

[51] Ilan Kremer, Noam Nisan, and Dana Ron. Errata for: ”on randomized one-round communication complexity”. *Computational Complexity*, 10(4):314–315, 2001.

[52] R. Kumar. Story of distinct elements, 2006.

[53] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[54] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *STOC*, pages 614–623, 1998.

[55] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. *Lecture Notes in Computer Science*, 1880:36–??, 2000.

[56] V.D. Milman and G. Schechtman. Asymptotic theory of finite dimensional normed spaces. *Lecture Notes in Mathematics*, 1200, 1986.

[57] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge Univ. Press, 1995.

[58] S. Muthukrishnan. Data streams: algorithms and applications, 2003.

[59] David Naccache and Jacques Stern. A new public-key cryptosystem. In *EUROCRYPT*, pages 27–36, 1997.

[60] Moni Naor and Kobbi Nissim. Communication preserving protocols for secure function evaluation. In *STOC*, pages 590–599, 2001.

[61] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *STOC*, pages 245–254, 1999.

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

[63] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.

[64] Alexander A. Razborov. On the distributional complexity of disjointness. *Theor. Comput. Sci.*, 106(2):385–390, 1992.

[65] Michael E. Saks and Xiaodong Sun. Space lower bounds for distance approximation in the data stream model. In *STOC*, pages 360–369, 2002.

[66] Strauss and Zheng. Private approximate heavy hitters. In *Manuscript*, 2007.

[67] J. H. van Lint. *An Introduction to Coding Theory.* New York: Springer-Verlag, 1992.

[68] David P. Woodruff. Optimal space lower bounds for all frequency moments. In *SODA*, pages 167–175, 2004.

[69] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *STOC*, pages 209–213, 1979.

[70] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *FOCS*, pages 160–164, 1982.

[71] Andrew Chi-Chih Yao. Lower bounds by probabilistic arguments (extended abstract). In *FOCS*, pages 420–428, 1983.