The Complexity of Massive Data Set Computations

by

Ziv Bar-Yossef

B.S. (Hebrew University of Jerusalem) 1997
M.S. (Hebrew University of Jerusalem) 1999

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Christos H. Papadimitriou, Chair
Professor Philip Stark
Professor Luca Trevisan

Fall 2002

The dissertation of Ziv Bar-Yossef is approved:

Chair Date

Date

Date

University of California at Berkeley

Fall 2002

The Complexity of Massive Data Set Computations

Abstract

The Complexity of Massive Data Set Computations

by

Ziv Bar-Yossef
Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor Christos H. Papadimitriou, Chair

Numerous *massive data sets*, ranging from flows of Internet traffic to logs of supermarket transactions, have emerged during the past few years. Their overwhelming size and the typically restricted access to them call for new computational models. This thesis studies three such models: sampling computations, data stream computations, and sketch computations.

While most of the previous work focused on designing algorithms in the new models, this thesis revolves around the *limitations* of the models. We develop a suite of lower bound techniques that characterize the complexity of functions in these models, indicating which problems can be solved efficiently in them. We derive specific bounds for a multitude of practical problems, arising from applications in database, networking, and information retrieval, such as frequency statistics, selection functions, statistical moments, and distance estimation.

We present general, powerful, and easy to use lower bound techniques for the sampling model. The techniques apply to all functions and address both oblivious and adaptive sampling. They frequently produce optimal bounds for a wide range of functions. They are stated in terms of new combinatorial and statistical properties of functions, which are easy to calculate.

We obtain lower bounds for the data stream and sketch models through one-way and simultaneous communication complexity. We develop lower bounds for the latter via a new information-theoretic view of communication complexity. A highlight of this work is an optimal simultaneous communication complexity lower bound for the important multi-party set-disjointness problem.

Finally, we present a powerful method for proving lower bounds for *general* communication complexity. The method is based on a direct sum property of a new measure of complexity for communication complexity protocols and on a novel statistical view of communication complexity. We use the technique to obtain improved communication complexity and data stream lower bounds for several problems, including multi-party set-disjointness, frequency moments, and L_p distance estimation. These results solve open problems of Alon, Matias, and Szegedy and of Saks and Sun.

Professor Christos H. Papadimitriou
Dissertation Committee Chair

To Inbal and Almog

Contents

Acknowledgements

I thank my advisor, Christos Papadimitriou, whose support, encouragement, and advice have always been inspiring and stimulating. I thank the members of my dissertation committee, Philip Stark and Luca Trevisan, for taking the effort to read and comment on this thesis. I thank the faculty, post-docs, visitors, and especially my fellow graduate students at the theory group for enabling me to spend such a lovely time at graduate school in Berkeley. Special thanks to Kris Hildrum, an office-mate, a collaborator, and a close friend. I thank the Berkeley staff for their dedicated assistance.

Mostly I am indebted to my three closest collaborators and friends, with whom most of the material presented in this thesis is joint work: T.S. Jayram, Ravi Kumar, and D. Sivakumar. I learned enormously from their comprehensive knowledge and their diligent academic style. I am especially grateful to Siva, who was a mentor to me both in research and outside.

Special thanks to Sridhar Rajagopalan, who introduced me to the area of massive data sets, and to David Zuckerman, from whom I learned about the Hellinger distance and its uses. I thank the many people, with whom I had discussions about material covered in this thesis.

I thank my collaborators in work outside the scope of this thesis: Alex Berg, Steve Chien, Jittat Fakcharoenphol, Kris Hildrum, Omer Reingold, Ronen Shaltiel, Luca Trevisan, Dror Weitz, and Felix Wu. I thank Luca for many useful discussions about the topics of this thesis and others.

I thank my parents, Tami and Yigal, whose everlasting care and love are the source of my strength. Thanks Dad for coming to visit us so frequently; each of your visits fueled us with vital energy. Thanks Mom for helping so much with Almog; undoubtedly writing this thesis would have taken much longer without your help. I thank my brother, Dror, and my sister, Adi, for visiting us, and keeping the spirits up from afar.

I thank my father and mother in law, Yuval and Rivka, for their constant care and attention. Rivka, your love, care, and especially your humor filled us with joy that enabled us to cope with the difficulties of living abroad and that will be cherished with me forever.

Finally, I thank my two most precious treasures, Inbal and Almog. Thank you Almog for giving me such an enjoyable time (albeit, not that productive...) while writing this thesis. And to you, Inbal, I want to express my deepest gratitude for sharing with me

the happy moments and the more difficult times, for always being on my side helping me navigate my way, and for offering valuable advice that enabled me to get to this moment. I love you so much.

This work was supported by a UC Regents Fellowship, NSF grant CCR-9820897, and NSF ITR grant CCR-0121555. Part of this work was done while visiting the IBM Almaden Research Center.

Chapter 1

Introduction

How many uniform samples of web pages are needed to get a good estimate of the distribution of languages on the web? Could one use fewer samples by sampling non-uniformly? How much storage does an Internet router need in order to track the number of distinct packet flows it routes every day? How succinctly can one store a database of documents, and yet be able to answer whether a new document is "similar" to any of the documents in the database? Questions of this nature are the main focus of this thesis.

During the past few decades, information has emerged as the main building block in the evolution of our economy, science, and culture. With increasing technological capabilities, the scope of the information gathered is constantly expanding. While in the past most of the available information was authored by humans, nowadays much of the data is generated automatically. Machines are now recording a large number of human and natural activities, ranging from credit card transactions to movements of heavenly bodies.

The explosion of information coupled with the dramatic increase in storage capacity have resulted in the emergence of gigantic bodies of raw data, ever increasing in volume, and constantly changing in nature. These are referred to as *massive data sets*. Some prominent examples are the Web, Internet traffic, stock market transactions, telephone calls, the human genome, weather data, and astronomical sky surveys.

The numerous challenges posed by the difficulty to analyze massive data sets and to extract valuable information from them have sparked off new fields of applied research, such as data mining and information retrieval, engaging computer scientists, statisticians, engineers, and others. The recently published Handbook of Massive Data Sets [1] gives a comprehensive overview of the state of the art of research in these areas.

While applied research provides technological solutions to problems, the theory of computation lays the mathematical foundations necessary to understand what types of solutions are viable and how to forge these solutions. As the branch of computer science that is concerned with the mathematical study of algorithms and computational complexity, the theory of computation is the most suitable medium for exploring the power and the limitations of new computational environments, such as that of massive data sets.

From a theoretical perspective, computations over massive data sets raise many open questions. Traditional models of computation and the standard practice of algorithm design are inadequate for handling massive data sets. The access to massive data sets is usually very costly and severely constrained, since they are typically stored on slow, sometimes distant, storage devices; traditional algorithms, however, assume access is cheap and that any access pattern is legitimate. Conventional algorithms are deemed *efficient* if they run in polynomial time; for massive data sets, even linear running time may be prohibitive, thereby necessitating algorithms that are not allowed to read their whole input. Finally, the space (i.e., memory) used for computations over massive data sets can be only a tiny fraction of the total size of the data; most of the standard algorithmic techniques assume that linear or even polynomial space are acceptable.

In response to the new needs, there has been a surge in the study of novel computational models, which are more suitable for the special requirements and constraints of computing over massive data sets. Some of the prominent examples are sketch and lossy compression schemes [20, 50, 24], data stream computations [4, 64, 46], external memory algorithms [124], property testing [112, 55], and traditional sampling computations. Most of the work done was on the development of new algorithmic methodologies pertinent to these models, and on designing algorithms to solve specific problems.

It is standard practice in the theory of computation that given a computational problem, to seek at finding the best algorithm (upper bound) to solve the problem and match it against the best lower bound for the problem. The ultimate goal is to obtain matching bounds, implying that the found algorithm is *optimal*. Prior to our work, lower bounds for computations over massive data sets were scarce and ad hoc—they were tailored to the problem at hand.

This thesis explores lower bound methodologies for computations over massive data sets. Specifically, we develop general techniques that give us recipes to easily derive lower bounds for a variety of functions in three of the aforementioned computational models: sampling computations, data stream computations, and sketch computations. We apply these techniques to obtain bounds for several practical problems that arise in actual applications, such as statistical and frequency moments, selection functions, and distance estimation.

1.1 Computational models for massive data sets

The severe constraints posed by the computational models for massive data sets coupled with the high requirements of performance rule out the possibility of using either exact computations or deterministic computations. Thus, most algorithms for massive data sets are both randomized and approximate. Indeed, in most situations one is not particularly interested in the exact value of the quantity to be computed; a good estimation of it is perfectly sufficient.

We consider three models for computing on massive data sets: sampling computations, data stream computations, and sketch computations.

1.1.1 Sampling computations

Sampling is the oldest and most pervasive model of computation over massive data sets, arising from statistics. A sampling algorithm queries only a few locations of the input data, and based on the outcomes of these queries it produces an output, which is supposed to approximate the wanted quantity. The crux of the algorithm is in the way it chooses the locations to be queried. The computer science setting departs from the statistical setting by allowing very strong query policies. Statistical sampling computations usually draw independent and identically distributed ("i.i.d.") samples from the data. In computer science the queries can depend on each other, they can be drawn according to different distributions, and they may be *adaptive*—that is, the distribution of each query can depend on the outcomes of previous queries.

The main measure of cost for sampling algorithms is the number of queries they perform. We ignore other resources like running time or space. Ideally, for a given function, we would like to find a sampling algorithm that approximates the function with low probability of error and using a number of samples which is independent of the size of the input. The number of samples may depend on the quality of the approximation and on the error probability. Clearly, such an algorithm is ideal for massive data set applications, because the running time and the space of such an algorithm (which are typically proportional to its number of samples) are small, regardless of the size of the data. Unfortunately, not all functions can be computed or approximated by sampling algorithms that use a small number of samples. "Sensitive" functions like the parity need the whole input to be queried in order to be computed correctly.

Sampling computations have numerous applications in almost all sciences. The most classical application is in statistical estimation theory (cf. [121]), in which one wishes to estimate some unknown parameter of the data (such as its mean or its variance) based on samples from the data. Statistical learning theory [122] and PAC learning [120, 76] are variants of sampling computations, in which the input data is the description (or "truth table") of the function/concept to be learned. Sampling is also a common approximation technique in computer science: one prominent example is the recent fully-polynomial randomized approximation scheme (fpras) for the permanent [72]. The model of *property testing* [112, 55] introduced in the context of probabilistically checkable proofs (PCPs) is yet another variant of sampling computations. Zuckerman [129] proved that extractors and dispersers—fundamental derandomization primitives—are sampling algorithms that estimate the mean and the maximum, respectively. Finally, randomized decision trees are a restricted class of sampling computations, which exactly compute Boolean functions.

1.1.2 Data stream computations

A data stream algorithm receives its input as a one-way stream; it can thus read each item of the data exactly once in the order given by the stream. The algorithm is furthermore limited to stringent storage constraints, implying it cannot store much of the input stream in memory. The algorithm has no control over the order in which the data items arrive, and it is required to output an approximation of the wanted quantity with low probability of error, irrespective of the arrival order.

Data stream computations are an appropriate model both for *static data sets* and for *online data sets*. Static data sets are ones that are stored in their entirety on a storage device, usually in the form of one or more database relations. Due to their large size, these data sets are typically stored on sequential magnetic devices, like tapes and disks, for which one-way access is either the only or the most efficient access method. Online data sets are ones whose data items are generated continuously on the fly, and are not necessarily stored anywhere (for example, Internet packets flowing through a router). Data stream computations are basically the only method to compute over them.

Data stream algorithms are usually measured in terms of the space they consume and the amount of time they spend processing each data item in the stream. Ideally, we would like algorithms whose space and running time per data item are poly-logarithmic in the size of the stream.

Several remarkably efficient data stream algorithms were designed in the last few years, targeting many "real-life" practical problems. These include algorithms for estimating frequency statistics [4, 47, 52, 49, 18, 16, 37, 75], L_p distance between vectors for any $p \in [1, 2]$ [46, 48, 68], histograms [54, 88, 51, 59], quantiles [58, 86, 87], clustering [60], number of inversions in a permutation [3], and number of triangles in a graph [18]. The data stream model also arises in computational complexity. Ordered Binary Decision Diagrams (OBDDs) [26, 126] are a special class of data stream computations, which exactly compute Boolean functions. Space-bounded randomized algorithms are restricted to access their random bits in a one-way fashion, and therefore can be viewed as data stream computations with respect to their random string inputs (cf. [13, 19]). A recent survey by Babcock *et al.* [11] gives a comprehensive overview of the data stream model and its applications.

1.1.3 Sketch computations

A sketch algorithm computes not from the input data itself, but rather from a "sketch" of the data. A sketch is a compact representation of the data, which encapsulates their properties that are necessary for computing some function of the data. The exact properties required depend on the function. In general, sketches are produced by a lossy compression scheme, but for some functions a lossless compression may be required (e.g., the identity function). The goal is to design a sketching scheme that compresses the data as much as possible, yet still enables the algorithm to output an approximation of the function with low probability of error.

In almost all uses of sketch computations, the input is partitioned into two or more pieces, each one of which is compressed into a sketch separately. The most popular scenario is as follows. The data is partitioned into two pieces; one is very large and static, while the other is small and constantly changing. The sketch algorithm is required to update the value of the computed function after each change to the second piece. In the most basic instantiation of this scenario, the large piece is a database and the small piece is a query. The algorithm is supposed to produce approximate answers to a sequence of queries to the database. Using a sketch algorithm this task can be accomplished while storing only a small sketch of the database. This paradigm has many instances, such as approximate nearest neighbor schemes [78, 82, 69, 67], synopsis data structures [50], data squashing [44], syntactic clustering [25, 24, 35], histogram maintenance [54, 88, 51, 59], pattern matching on compressed text files [5], and operations on compressed video sequences [34].

Sketch computations are also suitable for computing over massive data sets that are stored distributively on distant sites. A sketching procedure running on each such site produces a sketch of the piece of the data stored on that site, and sends the sketch to the sketch algorithm. Compact sketches are necessary here to minimize the amount of communication needed to perform the computation. Comparing patterns of Internet traffic in different routers is one application of this approach [46]. This approach also shows that sketch computations are a generalization of the model of simultaneous communication complexity [127], which traditionally dealt with exact computation of Boolean functions.

Numerous sketching techniques are used in the above applications and others, including Singular Value Decomposition (SVD), wavelet and Fourier transforms, statistical regression, histograms, clustering, sampling, hashing, and low distortion embeddings. See [20] for a (slightly out-of-date) survey of sketching techniques.

1.2 Our contributions

This thesis studies the limitations of computing over massive data sets. The main goal is to understand what kind of functions can be computed efficiently on massive data sets. We seek quantitative characterizations of the complexity of functions under the three prominent models for massive data sets: sampling computations, data stream computations, and sketch computations.

We achieve these goals by developing a suite of general lower bound techniques for the aforementioned models. We then apply these techniques to derive lower bounds for interesting problems, arising from practical applications, including statistical moments, frequency statistics, selection functions, and distance estimation. Most of our lower bounds are stated in terms of simple-to-calculate properties of the functions, yielding recipes for obtaining bounds for concrete examples. These recipes should greatly facilitate the derivation of lower bounds for new problems.

1.2.1 Sampling lower bounds

We present three lower bound techniques for sampling computations: a combinatorial technique, a statistical technique, and an information-theoretic technique.

The combinatorial technique is based on a generalization of the notion of *block sensitivity* [94]. Its greatest advantage is its level of generality: it applies to all functions and to all types of sampling computations. It also gives lower bounds on the expected query complexity, not just on the worst-case query complexity. The main drawback of this technique is that the bounds it produces can be rather weak.

The statistical technique is based on a reduction from the setting of statistical hypothesis testing. It gives lower bounds only for *oblivious* (i.e., non-adaptive) sampling computations. Such bounds are still very interesting for two reasons. First, oblivious sampling is the most common form of sampling, and thus studying the oblivious query complexity of functions is desirable. Second, we prove that for a wide class of functions—the symmetric functions[1]—oblivious sampling computations are as strong as general adaptive sampling computations; hence, the statistical technique yields general query complexity lower bounds for these functions.

The statistical technique frequently results in optimal lower bounds for many symmetric functions. We were able to use it to obtain new bounds as well as known bounds whose previous proofs are considerably more complicated:

1. New lower bounds for approximation of the frequency moments. (The k-th frequency moment, F_k, takes as input a sequence of n integers in $[m]$. The frequency of $j \in [m]$, f_j, is the number of times j occurs in this sequence. $\mathrm{F}_k = \sum_j f_j^k$). We show that these bounds are tight, by presenting a matching upper bound.

2. Lower bounds for approximation of the median and other selection functions, including the minimum and the maximum.

3. Optimal lower bound on the expected query complexity of additive approximation of the mean of n real numbers in the interval $[0, 1]$ (previously proved by Canetti, Even,

[1] A symmetric function is one that is invariant under permutations of the input data items (e.g., the mean, the median, etc.)

and Goldreich [31] for the worst-case query complexity and by Radhakrishnan and Ta-Shma [104] for the expected query complexity). The same bound holds for higher statistical moments as well.

4. Using Zuckerman's connection [129] between extractors and sampling computations, we derive new and simple proofs of the seed length lower bounds for extractors and dispersers (which were proved previously by Radhakrishnan and Ta-Shma [104] using a very involved argument).

5. A lower bound for the estimation of the number of distinct elements in a sequence (previously proved by Charikar, Chaudhuri, Motwani, and Narasayya [36]).

The main idea of the statistical technique is the following. Every sampling computation that accurately approximates a function f forms a *statistical test* that can distinguish, with low probability of error, between the sample distribution corresponding to an input x and the sample distribution corresponding to an input y, for any x, y for which $f(x)$ and $f(y)$ are very "different" from each other. If we find two such inputs that are "hard" to distinguish (i.e., require any test to use many samples) we can derive a lower bound on the query cost of the algorithm. We then apply sampling lower bounds for statistical tests from the statistics literature, which are stated in terms of various measures of statistical distance between the sample distributions, most notably the *Hellinger distance* and the *Kullback-Leibler (KL) divergence*.

The information-theoretic technique is an extension of the statistical technique, building on the observation that a sampling computation can be used to form a statistical test that distinguishes between more than two sample distributions: for any set $S = \{x_1, \ldots, x_k\}$ of inputs for which $f(x_1), \ldots, f(x_k)$ are very different from each other, there is a test that simultaneously distinguishes between all of them. Using an information-theoretic view of statistical testing, we obtain a query complexity lower bound in terms of the *Jensen-Shannon divergence*—a symmetric version of the KL divergence. This bound has better dependence on $|S|$—the number of mutually "different" inputs—which is significant in some applications. We use this technique to give a simple proof of the lower bound for the estimation of the frequency distribution of a sequence (previously proved by Batu *et al.* [21]).

1.2.2 Lower bounds for one-way and simultaneous communication complexity

A standard methodology for proving lower bounds for data stream and sketch computations is through the model of communication complexity [127]. Furthermore, lower bounds in restricted models of communication complexity suffice for this purpose: *one-way* (multi-party) communication lower bounds yield space lower bounds for data stream computations and *simultaneous* (multi-party) communication lower bounds yield compression lower bounds for sketch computations.

In one-way communication complexity the players send messages sequentially: the first player sends a message to the second, the second to the third, and so forth up to the last player, who announces the output. In simultaneous communication complexity, all the players simultaneously send a message to a "referee"; the referee computes and announces the output based on these messages. Motivated by the data stream and sketch applications, we study lower bound techniques for these models.

We present a lower bound technique for one way communication complexity, through the notion of *shatter coefficients*, strengthening previous results of Kremer, Nisan, and Ron [79] and Ablayev [2], which used the notion of VC-dimension. We show that the technique gives tight bounds for multi-party protocols, if the input to the function is chosen according to a product distribution. This generalizes a result of Kremer *et al.* [79], who proved it only for two-party protocols. We apply the technique to derive data stream space lower bounds for the median and other selection functions and for the problem of counting distinct elements in a sequence.

We show that for product input distributions, the one-way and the simultaneous models are equivalent, for an arbitrary number of players. However, for non-product distributions, simultaneous communication can be exponentially weaker than one-way communication, even for two players. This disproves a statement by Kremer *et al.* [79].

Finally, we prove an optimal $\Omega(n/t)$ lower bound on the simultaneous communication complexity of the t-party set-disjointness problem. (In t-party set-disjointness, each of the t players gets a subset S_i of a universe of size n. The sets are guaranteed to be either mutually and uniquely interesting (a Yes instance) or pairwise disjoint (a No instance)). This bound improves over the $\Omega(n/t^4)$ lower bound of Alon, Matias, and Szegedy [4], though the latter holds for general communication complexity. We use the new lower bound to derive compression lower bounds for sketch computation of the frequency moments.

Methodologically, all of our proofs are based on a novel view of communication complexity in the context of information theory. Using this view, we are able to apply tools from information theory, such as Fano's inequality and the Maximum Likelihood Estimator principle, to derive simple and intuitive proofs for communication complexity results.

1.2.3 Lower bounds for general communication complexity

Lower bounds for one-way and simultaneous communication complexity are sufficient for the data stream and sketch applications but cannot be used to derive lower bounds for stronger models. Even simple generalizations of the data stream model, such as k-pass data stream computations[2], require stronger communication complexity lower bounds. This motivates our study of lower bounds for general, interactive, communication complexity.

We develop a novel and powerful method for obtaining lower bounds for randomized communication complexity. We use this method to derive lower bounds for communication complexity problems that arise in the data stream context.

We show that the randomized communication complexity of the t-party set disjointness problem is $\Omega(n/t^2)$. This is an improvement over the $\Omega(n/t^4)$ bound of Alon, Matias, and Szegedy [4]. (A special case of this bound is a new proof of the famous $\Omega(n)$ lower bound for the two-party set-disjointness problem, originally due to Kalyanasundaram and Schnitger [73] with simplifications of Razborov [109].) In the one-way model we show a nearly optimal lower bound of $\Omega(n/t^{1+\epsilon})$ for arbitrarily small ϵ.

A consequence of the lower bound for the one-way model is the following: we show the first polynomial space lower bounds for approximating the k-th frequency moment F_k for any real $k > 2$ in the data stream model. This completely resolves the conjecture of Alon, Matias, and Szegedy [4], who showed an $\Omega(m^{1-5/k})$ space lower bound for constant factor approximation of F_k for $k > 5$. We show that approximating F_k for $k > 2$ to within constant factors requires $\Omega(m^{1-2/k})$ space. The best known space upper bound for F_k is $\tilde{O}(m^{1-1/k})$ [4]. Since our lower bound is essentially optimal for the communication problem, closing this gap would require either a better algorithm or a different lower bound method for the data stream model.

In the L_∞ *promise problem*, Alice is given $\vec{x} \in [m]^n$ and Bob is given $\vec{y} \in [m]^n$.

[2] A k-pass data stream computation can have k sequential passes over the data, rather than just one.

The promise is that either $|\vec{x} - \vec{y}|_\infty \leq 1$ (Yes instances) or $|\vec{x} - \vec{y}|_\infty \geq \theta$ (No instances). We show that the randomized communication complexity of this problem is $\Omega(n/\theta^2)$. This solves the open problem of Saks and Sun [113], who showed this bound for the restricted one-way communication model.

A consequence of this result is a lower bound for approximating L_p distances for $p > 2$: approximating the L_p distance between n-dimensional vectors to within a factor of n^ϵ requires $\Omega(n^{1-4\epsilon-2/p})$ space in the data stream model for any constant number of passes over the input. The communication complexity lower bound of [113] gives a similar bound for the one-pass data stream model. This bound is optimal for $p = \infty$.

Our new lower bound method is based on a direct sum theorem for *conditional information complexity*—a new information-theoretic measure of complexity for communication complexity protocols. The direct sum theorem allows us to reduce the task of proving lower bounds for "complex" multi-coordinate problems to that of obtaining lower bounds for "simple" single-coordinate problems. For example, the lower bound for set-disjointness reduces to a lower bound for the And of two bits.

In order to derive lower bounds for single-coordinate functions, we exploit connections between information complexity and various statistical distance measures, most notably the Hellinger distance, as well as new properties of these measures pertinent to the distributions that come up in communication complexity protocols.

1.3 Notes for the Reader

1.3.1 Prerequisites

The thesis assumes the reader is comfortable with basic terminology and notation used in theoretical computer science, such as asymptotic analysis of complexity of algorithms (big O and big Ω notation), Turing machines, and reductions. The textbooks of Cormen, Leiserson, Rivest, and Stein [39] and Papadimitriou [100] provide thorough discussions of these topics.

The reader is also assumed to have background in basic probability theory. No prior knowledge is assumed in information theory, statistics, or advanced computational complexity.

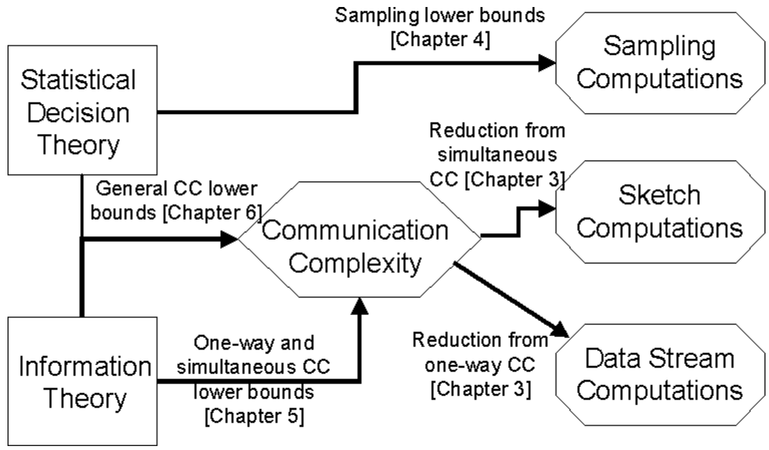

Figure 1.1: The thesis blueprint

1.3.2 Thesis organization

Figure 1.1 gives a bird's eye view of the structure of the thesis:

Chapter 2 is an overview of the three fundamental areas on which this thesis builds: communication complexity, information theory, and statistical decision theory. It also includes a list of the standard notations used throughout the subsequent chapters.

Chapter 3 gives rigorous definitions of the three computational models we study: sampling computations, data stream computations, and sketch computations. It also outlines the reductions from communication complexity to the data stream and sketch models, and reviews the basic connections among the three models.

Chapter 4 contains the sampling lower bounds.

Chapter 5 includes the lower bound methods for one-way and simultaneous communication complexity.

Chapter 6 discusses the direct sum lower bound technique for general communication complexity.

1.3.3 Bibliographic notes

The material covered in Chapters 2 and 3 is a survey of known results from the literature. The exposition and some of the definitions in Chapter 3 (such as the general sketch model), though, are original.

The sampling lower bounds described in Chapter 4 are partly joint work with Ravi Kumar and D. Sivakumar [17] and partly yet unpublished [12].

The one-way and simultaneous communication complexity lower bounds, appearing in Chapter 5, are joint work with T.S. Jayram, Ravi Kumar, and D. Sivakumar [15].

The general communication complexity lower bounds, discussed in Chapter 6, are joint work with T.S. Jayram, Ravi Kumar, and D. Sivakumar [14].

Chapter 2

Preliminaries and Notation

In this chapter we review the basic tools that underly the new techniques developed in this thesis. The tools come from three areas: communication complexity, information theory, and statistical decision theory. We start the chapter with the standard notation used throughout the thesis.

2.1 Standard notation

Standard characters We denote sets by capital Calligraphic letters (e.g., $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$), elements of sets by lower case letters (e.g., x, y, z), random variables by capital letters (e.g., X, Y, Z), and distributions by lower case Greek letters (e.g., μ, ν, λ). A ceiling arrow denotes a vector (e.g., $\vec{\mathcal{X}}, \vec{y}, \vec{Z}, \vec{\mu}$). For a d-dimensional vector $\vec{X}$ and an index $1 \leq j \leq d$, we denote by $\vec{X}^{-j}$ the $(d-1)$-dimensional vector obtained from $\vec{X}$ by deleting its j-th entry.

Probability Throughout the thesis, we deal only with finite discrete probability spaces. For a distribution μ on a domain $\mathcal{X}$, we denote by $\mu(x)$ the probability of an element $x \in \mathcal{X}$ under μ. We denote by either $\mu(A)$ or $\Pr_\mu(A)$ the probability of an event $A \subseteq \mathcal{X}$ under μ. When μ is understood from the context, we simply write $\Pr(A)$.

Distributions For a vector $\vec{\mu}$ of distributions $\mu_1, \ldots, \mu_n$, we say that $\vec{\mu}$ is product if $\mu_1, \ldots, \mu_n$ are mutually independent. For two distributions μ and ν we denote by (μ, ν) the joint distribution of μ and ν and by $\mu \times \nu$ the product distribution of μ and ν (i.e., $(\mu \times \nu)(x, y) = \mu(x) \cdot \nu(y)$, for all x, y). For a distribution μ and an event A, we denote by $\mu|A$ the conditional distribution of μ given the event A. We denote the probability of an element x under $\mu|A$ either by $(\mu|A)(x)$ or by $\mu(x|A)$. Similarly, the probability of an event B under $\mu|A$ is denoted either by $(\mu|A)(B)$ or by $\mu(B|A)$. For a distribution μ on $\mathcal{X}$ and a function $f : \mathcal{X} \to \mathcal{Y}$, we denote by $f(\mu)$ the distribution induced by μ and f on $\mathcal{Y}$.

Random variables $X \sim \mu$ means the probability distribution of the random variable X is μ. The probability of an event A which is given as a Boolean statement $\phi(X)$ about a random variable X (e.g., "$X \geq 5$") is denoted $\Pr_{X\sim\mu}(\phi(X))$. When μ is understood from the context, we simply write $\Pr(\phi(X))$. The expectation of X is denoted either by $\mathrm{E}_\mu[X]$ or simply by $\mathrm{E}[X]$. The expectation of a function $f(X)$ of X is denoted $\mathrm{E}_{X\sim\mu}[f(X)]$. Similar notations are used for the variance, $\textsc{Var}$. $\{X = x\}$ denotes the event $X^{-1}(x)$.

A vector of random variables $\vec{X} = (X_1, \ldots, X_n)$ is called product, if $X_1, \ldots, X_n$ are mutually independent. For a random variable $X \sim \mu$ and for an event A, $X|A$ denotes the random variable whose probability distribution is $\mu|A$.

Miscellaneous All logarithms, unless stated otherwise, are to base 2. The symbol ln denotes the natural logarithm. The symbol $[n]$ denotes the set of integers $\{1, \ldots, n\}$.

2.2 Communication complexity

The communication complexity model, introduced by Yao [127], has emerged as one of the most useful and intriguing models in computational complexity. Its main merit is its simple and well understood structure, which has led to a variety of strong lower bound techniques. The great importance of communication complexity is due to its role as an engine of lower bounds, which through reductions yield lower bounds for many other different models of computation. Some of the outstanding beneficiaries of this technique include circuit complexity [74, 107, 106], time-space tradeoffs [10], VLSI theory [84], decision tree complexity [95], data structure complexity [89, 90], pseudorandom generators for logspace [10, 66], data stream computations [4, 113, 14], and recently even computational economics [96, 42]. The textbook of Kushilevitz and Nisan [81] provides an excellent review of the model and its applications. In the following we briefly outline the model and its fundamental properties of which we make use throughout the thesis.

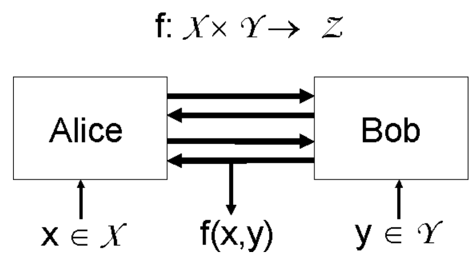

Figure 2.1: The communication complexity model

2.2.1 Deterministic communication complexity

In the classical version of the communication complexity model (see Figure 2.1) two players, Alice and Bob, are required to compute a bivariate function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ ($\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ are some arbitrary sets). Alice is given one argument of the input $x \in \mathcal{X}$ and Bob is given the other argument $y \in \mathcal{Y}$. Since neither Alice nor Bob know the input in its entirety, neither can compute $f(x, y)$ on his/her own. Alice and Bob thus need to engage in a "conversation" in order to jointly compute $f(x, y)$. The complexity measure of concern in this model is the amount of data Alice and Bob have to communicate in order to compute $f(x, y)$ correctly. Alice and Bob are assumed to be computationally all powerful—they are not restricted by any computational resource, such as running time or storage space.

The conversation between Alice and Bob is governed by a *protocol*. The protocol determines at each step whose turn it is to speak, and what message should that party transmit based on her input and on the transcript of the conversation so far. The protocol also specifies how an observer who monitors the communication between Alice and Bob can deduce the function's value from the transcript of messages. Formally, a protocol is defined as follows:

Definition 2.1 (Communication complexity protocol) *A* communication complexity protocol Π *is a quadruple* $\langle N, A, B, O\rangle$. $N : \{0,1\}^* \to \{\mathrm{Alice}, \mathrm{Bob}, \mathrm{End}\}$ *is a function that determines which player should send the next bit of communication based on the bits communicated so far. It gives the value "*End*" when the communication is over.* $A : \mathcal{X} \times \{0,1\}^* \to \{0,1\}$ *is a function that determines the next message bit of Alice, based on her input and on the bits communicated so far. Similarly,* $B : \mathcal{Y} \times \{0,1\}^* \to \{0,1\}$ *determines Bob's next message bit. Finally,* $O : \{0,1\}^* \to \mathcal{Z}$ *determines the output of the protocol based on the bits communicated.*

We denote by $\Pi(x, y)$ *the transcript of messages communicated in the protocol* Π *when running on inputs* x *and* y*. We denote by* $\Pi^{out}(x, y)$ *the output of this protocol on* (x, y)*.* Π *is said to* compute $f(x, y)$ correctly*, if for all input pairs* (x, y)*,* $\Pi^{out}(x, y) = f(x, y)$*. The* communication cost of Π on (x, y) *is* $|\Pi(x, y)|$*—the number of bits communicated between Alice and Bob on* (x, y)*. The* communication cost of Π*, denoted* $CC(\Pi)$*, is its worst-case communication cost over all possible input pairs* $(x, y) \in \mathcal{X} \times \mathcal{Y}$*.*

The (deterministic) communication complexity *of* f*, denoted* $D(f)$*, is the communication cost of the best protocol that computes* f*.*

Further conventions Throughout this thesis we make a couple of further assumptions about the model, which make it more convenient to analyze, but do not modify it qualitatively.

As defined above, the model computes functions over the *full domain* (that is, every pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is a legal input). However, the model perfectly fits also functions over a partial domain: $f : \mathcal{T} \to \mathcal{Z}$, where $\mathcal{T} \subseteq \mathcal{X} \times \mathcal{Y}$. Our first assumption is that any communication complexity protocol for a partial function f is well defined for any pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, even if this pair is not a legal input (i.e., $(x, y) \notin \mathcal{T}$). This can be easily achieved by requiring the first player to be unable to continue a proper execution of the protocol to broadcast the special symbol '*' and halt the protocol.

Our second assumption is that the protocol always produces transcripts of the same length, because it it does not, we can modify it by requiring the player who sends the last message to pad his message with an appropriate number of 0's.

Rectangle property For a string $\tau \in \{0, 1\}^*$ and for a protocol Π, we denote by $\Pi^{-1}(\tau)$ the set of inputs for which τ is the transcript $\Pi(x, y)$. The most fundamental property of deterministic communication complexity protocols is the *rectangle property* (also known as the *cut & paste property*):

Proposition 2.2 *For any deterministic protocol* Π *for a function* $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ *and for any string* $\tau \in \{0, 1\}^*$*, the set* $\Pi^{-1}(\tau)$ *is a combinatorial rectangle: that is,* $\Pi^{-1}(\tau) = \mathcal{S}_1 \times \mathcal{S}_2$*, where* $\mathcal{S}_1 \subseteq \mathcal{X}$ *and* $\mathcal{S}_2 \subseteq \mathcal{Y}$*.*

The proof can be found in [81], Chapter 1. Proposition 2.2 implies that if $\Pi(x, y) = \tau$ and $\Pi(w, z) = \tau$, then also $\Pi(x, z) = \tau$ and $\Pi(w, y) = \tau$ (see Figure 2.2).

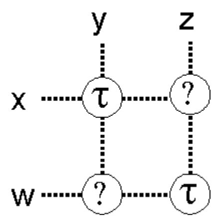

Figure 2.2: The rectangle (cut & paste) property of communication complexity

Examples Consider the AVG problem: Alice holds a subset $x \subseteq [n]$, Bob holds another subset $y \subseteq [n]$ and their goal is to compute the average of the multi-set $x \cup y$. An efficient straightforward protocol is the following: Alice computes the sum of the elements in x as well as their number and sends them to Bob. Bob is then able to know the sum of all the elements in $x \cup y$ and their number, and therefore can compute $\textsc{Avg}(x \cup y)$. Bob sends the result to Alice. The protocol requires $O(\log n)$ bits of communication.

Consider, on the other hand, the EQ problem: Alice and Bob hold binary strings $x, y \in \{0,1\}^n$ and would like to determine whether the strings are equal or not. The straightforward protocol is for Alice to send her input in its entirety to Bob; Bob can then check whether the strings are equal and send the result to Alice. This protocol requires $n+1$ bits of communication. This highly costly protocol is in fact the best possible: if there is a protocol Π for EQ whose communication cost is less than n, then by the pigeonhole principle there are two strings $\alpha \neq \beta \in \{0,1\}^n$ such that $\Pi(\alpha, \alpha) = \Pi(\beta, \beta)$. However, by the cut & paste property, also $\Pi(\alpha, \beta) = \Pi(\alpha, \alpha)$. This is impossible, since $\textsc{Eq}(\alpha, \alpha) = 1$ and $\textsc{Eq}(\alpha, \beta) = 0$. Thus, Π has to have 2^n different transcripts on the inputs on which EQ is 1. It has to have at least one additional different transcript for inputs on which EQ is 0. Therefore, $D(\textsc{Eq}) = n + 1$.

2.2.2 Randomized communication complexity

In this thesis we mainly concentrate on *randomized protocols*. In such protocols Alice and Bob are allowed to use random coin tosses to determine the messages they send. Formally, they are defined as follows:

Definition 2.3 (Randomized protocol) *A* randomized communication complexity protocol Π *is identical to a deterministic protocol, with the following difference: the message function A takes an additional argument r_A from a domain $\mathcal{R}_A$ and the message function*

B takes and additional argument r_B from a domain $\mathcal{R}_B$. r_A and r_B are assigned independent random values R_A and R_B, which are uniformly distributed on $\mathcal{R}_A$ and $\mathcal{R}_B$, respectively. r_A and r_B are called private random strings*, since Alice and Bob have no a priori information about the random strings of each other.*

We denote by $\Pi(x, y, r_A, r_B)$ and by $\Pi^{out}(x, y, r_A, r_B)$ the transcript of messages communicated and the output, respectively, of Π on input (x, y) and random strings (r_A, r_B). Π is said to compute f with error probability δ*, if* $\Pr\left(\Pi^{out}(x, y, R_A, R_B) = f(x, y)\right) \geq 1 - \delta$*, for all input pairs (x, y). The* communication cost of Π on (x, y) *is the maximum length of $\Pi(x, y, r_A, r_B)$ over all possible choices of private random strings (r_A, r_B). The* communication cost of Π*, denoted $CC(\Pi)$, is its worst-case communication cost over all possible input pairs (x, y).*

The δ-error randomized communication complexity *of a function f, denoted $R_\delta(f)$, is the communication cost of the best protocol that computes f with error probability δ.*

Example Randomized protocols can do much better than deterministic ones, as demonstrated by perhaps the most famous communication complexity protocol: the $O(\log n)$ communication protocol for EQ due to Rabin and Yao. In this protocol, Alice and Bob view their inputs as n-bit integers. Alice chooses a random prime p among the first n^2 primes, and sends to Bob p and $x \bmod p$. Bob checks whether $y \bmod p = x \bmod p$, and sends the answer 1 if and only if they are equal. This protocol requires only $O(\log n)$ bits of communication, since the prime p is polynomial in n. It is clear the protocol is always correct on inputs for which EQ is 1. If $x \neq y$, then the protocol is erroneous only if p divides $|x - y|$. Since there are at most n primes dividing $|x - y|$, the protocol is correct with probability at least $1 - 1/n$.

Public coins A wider class of randomized protocols are ones in which Alice and Bob are allowed to use *public coins*, on top of their private random strings. In such protocols Alice and Bob have both free access to a shared public random string, which they can use in order to determine their messages. Formally, we augment each of the message functions A and B with yet another input argument r from a domain $\mathcal{R}$. r is assigned in both functions the same random variable R, which is uniformly distributed uniformly on $\mathcal{R}$ and is independent of R_A and R_B. We correspondingly denote by $\Pi(x, y, r, r_A, r_B)$ and $\Pi^{out}(x, y, r, r_A, r_B)$ the transcript and the output, respectively, of Π on input (x, y), public randomness r, and private randomness (r_A, r_B). The definitions of correct computation and communication cost of public-coin randomized protocols are the same as for private-coin protocols; the only change is that now the probability is both over the private random strings and the public random string. We denote by $R_\delta^{pub}(f)$ the δ-error public-coin randomized communication complexity of f.

An equivalent useful characterization of public-coin protocols is as a distribution over private-coin protocols. The proof is trivial: every fixing r of the public random string induces a protocol Π_r that uses private randomness only. The expected error probability of Π_R over the choice of R is exactly δ.

By definition, every private-coin protocol is also a public-coin one, implying that $R_\delta^{pub}(f) \leq R_\delta(f)$ for any function f. Newman [92] showed that the difference between the two cannot be too large:

Theorem 2.4 (Newman [92]) *For any function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ and for every $\delta, \delta' > 0$,*

$$R_{\delta+\delta'}(f) \ \leq \ R_\delta^{pub}(f) + O(\log\log(|\mathcal{X}| \cdot |\mathcal{Y}|) + \log(1/\delta')).$$

Distributional communication complexity Public-coin randomized communication complexity is interesting, since it has a complete characterization in terms of deterministic *distributional communication complexity.*

Definition 2.5 (Distributional communication complexity) *Let μ be a distribution over the input domain $\mathcal{X} \times \mathcal{Y}$, and let $(X, Y) \sim \mu$. For a deterministic protocol Π, we say that Π computes f with error probability δ on μ, if $\Pr\left(\Pi^{out}(X, Y) = f(X, Y)\right) \geq 1 - \delta$, where the probability is over the choice of (X, Y) according to μ. The δ-error μ-distributional communication complexity of f, denoted $D_{\mu,\delta}(f)$, is the communication cost of the best deterministic protocol that computes f with error probability δ on μ.*

Yao's Lemma uses the minimax theorem of game theory to give the following characterization of public-coin randomized communication complexity in terms of distributional communication complexity:

Lemma 2.6 (Yao's Lemma [128]) *For any function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, and for any $\delta > 0$,*

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

Yao's Lemma is perhaps the most important tool in proving lower bounds on randomized communication complexity. It implies that in order to give a lower bound on $R_\delta(f)$ it is sufficient to exhibit some input distribution μ of our choice, and prove a lower bound on $D_{\mu,\delta}(f)$. This is easier to do since: (1) we can choose μ as we like, and (2) we have to deal with deterministic protocols that are simpler and easier to analyze than randomized ones.

A special class of input distributions that are particularly convenient to analyze are *product distributions*:

Definition 2.7 (Product/rectangular distribution) *An input distribution $\mu = (\mu_X, \mu_Y)$ is called* product *(also known as* rectangular*), if $\mu = \mu_X \times \mu_Y$.*

2.2.3 Multi-party communication complexity

In multi-party communication complexity (see Figure 2.3) t players are required to compute a function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \mathcal{Z}$. We will concentrate on the *number-in-the-hand* version of the model, in which the j-th player receives x_j.[1] The players engage in a "discussion" according to a fixed protocol, in the end of which they jointly compute $f(x_1, \ldots, x_t)$. We concentrate on broadcast communication—the messages of each player are simultaneously received by all other players. A convenient way to visualize this is the following: the players all sit in a classroom in front of a blackboard; each player, in her turn, writes her message on the blackboard, so all the other players can see it. She does not erase previous messages from the blackboard. In the end of the interaction, the teacher enters the classroom, reads what is written on the blackboard, and can deduce $f(x_1, \ldots, x_t)$. The formal definition of multi-party protocols is as follows:

Definition 2.8 (Multi-party protocol) *A t*-party communication complexity protocol Π *is a triple $\langle N, \vec{P}, O\rangle$. $N : \{0,1\}^* \to [t] \cup \{End\}$ is a function that determines the player to send the next bit of communication, based on the bits transmitted so far. N gives the value "End" when the communication is over. $\vec{P} = (P_1, \ldots, P_t)$ is the vector of message functions. $P_j : \mathcal{X}_j \times \{0,1\}^* \to \{0,1\}$ determines the next bit the j-th player is to send, based on her input and on the transcript of bits communicated so far. Finally, $O : \{0,1\}^* \to \mathcal{Z}$ determines $f(x_1, \ldots, x_t)$ from the transcript of messages in the end of the communication.*

[1] There are other methods to distribute the input arguments to the players, most notably the *number-on-the-forehead* method [33], in which the j-th player gets all the input arguments except for x_j.

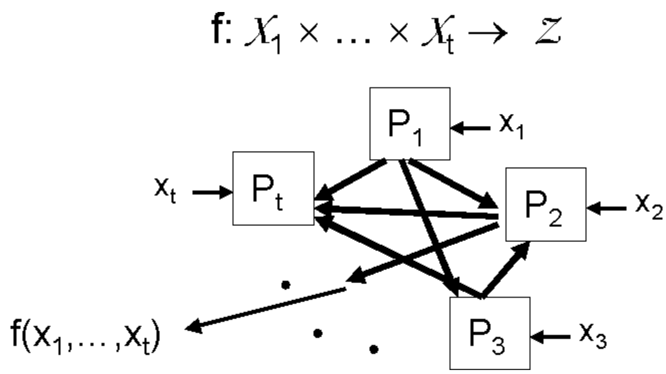

Figure 2.3: Multi-party communication complexity

The definitions of correct computation and communication cost of a protocol are as for the two-party case.

Multi-party protocols preserve the rectangle property of communication complexity:

Proposition 2.9 *For any deterministic t party protocol Π for $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \mathcal{Z}$ and for any string $\tau \in \{0,1\}^*$, the set $\Pi^{-1}(\tau)$ is a combinatorial rectangle: that is, $\Pi^{-1}(\tau) = \mathcal{S}_1 \times \cdots \times \mathcal{S}_t$, where $\mathcal{S}_j \subseteq \mathcal{X}_j$ for all $j \in [t]$.*

Randomization is incorporated into multi-party protocols exactly in the same way it is done in two-party protocols. In particular, Newman's Theorem and Yao's Lemma hold in the multi-party case as well.

2.2.4 One-way communication complexity

The one-way communication model [101, 2, 79, 93] is a restricted model of communication complexity. It will be of interest to us, since it is sufficient to derive space lower bounds for the data stream model.

In the t-party version of the model (see Figure 2.4) each of the t players sends exactly one message throughout the protocol. The communication starts with the first player sending a message to the second player; the second player then sends a message to the third player; and so on. The t-th player sends the last message to the first player (this last message typically consists of the output $f(x_1, \ldots, x_t)$).

The above description is not exactly compatible with our definition of multi-party protocols, in which messages are always broadcast and cannot be addressed to a specific

Figure 2.4: One-way communication complexity

player. In order to formally define the model, we assume messages are broadcast, but each player can use only the last message sent to determine her next message (in the analogy to the blackboard scenario, each player erases everything written on the blackboard before writing her message). The formal definition of one-way communication is as follows:

Definition 2.10 (One-way communication protocol) *A t-party* one-way communication complexity protocol Π *is a t-party protocol with the following restrictions: (1) The next-player function N is restricted to enforce the players to speak in order: first, the first player broadcasts all the bits she wishes to communicate, then the second player, the third, and so forth, up to the t-th player. (2) The message functions $P_1, \ldots, P_t$ are restricted so P_j depends only on the input of the j-th player (x_j) and the part of the transcript that was transmitted by the $j-1$-st player. We denote the* one-way communication complexity *of a function f by $D^{\rightarrow}(f)$.*

For two party one-way communication, we will denote by $D^{A\rightarrow B}(f)$ the one-way communication complexity of f, when Alice is the first player and Bob is the second; we will denote by $D^{B\rightarrow A}(f)$ the one-way communication complexity of f, when Bob is the first player and Alice is the second.

Randomization is incorporated into the model in the standard way. We denote by $R_\delta^{\rightarrow}(f)$ the δ-error one-way randomized communication complexity of f, and similarly by $R_\delta^{A\rightarrow B}(f)$ and $R_\delta^{B\rightarrow A}(f)$ the $A \rightarrow B$ and $B \rightarrow A$ one-way randomized communication complexity of a bivariate f.

All the properties of deterministic and randomized communication complexity mentioned for general communication hold also for one-way communication.

Example One-way communication complexity can be exponentially weaker than general communication. Consider, for example the Ind function: Alice gets a string $x \in \{0,1\}^n$ and Bob gets an index $i \in [n]$, and they would like to find x_i. Clearly, $D(f) = O(\log n)$, since Bob can simply send i to Alice who responds with x_i. The one-way communication

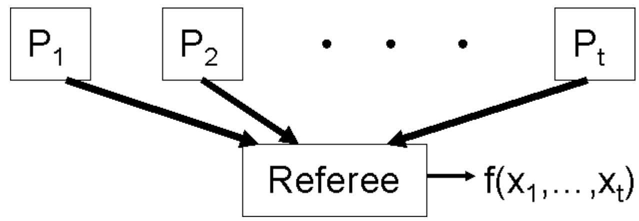

Figure 2.5: Simultaneous communication complexity

(and more specifically, the $A \to B$ communication) of IND is $\Omega(n)$, even for randomized protocols.

2.2.5 Simultaneous communication complexity

The simultaneous communication model [127, 79, 8, 93] (see also the recent survey by Babai [6]) is another restricted model of communication, which is even weaker than the one-way model. It is of interest to us due to its application to compression lower bounds in the sketch model.

In the t-party version of the model (see Figure 2.5), t "regular" players get input arguments (as usual) and another auxiliary player, called the *referee*, does not get to see any part of the input. The regular players send simultaneously one message to the referee, who is then supposed to compute and announce the output.

As in the one-way model, the above description is not compatible with our formal definition of multi-party protocols. We thus formally define simultaneous protocols as follows:

Definition 2.11 (Simultaneous communication protocol) *A t* party simultaneous communication complexity protocol Π *is a t-party protocol with the following restrictions: (1) The next-player function N is restricted to enforce the players to send their messages one after the other in some fixed order (for example, in the order of their indices). (2) The message functions $P_1, \ldots, P_t$ are restricted so P_j depends only on x_j and not on the transcript. We denote the* simultaneous communication complexity *of a function f by $D^{||}(f)$.*

Note that in the formal definition we do not need to explicitly mention the referee, since his job is carried out by the output function O.

Randomization is incorporated into the model in the standard way. We denote by

$R_\delta^{||}(f)$ the δ-error simultaneous randomized communication complexity of f. Most of the properties of deterministic and randomized communication complexity mentioned above hold also for simultaneous communication. The main property which does not hold is Newman's Theorem: his reduction from private coins to public coins does not work in simultaneous protocols. Indeed, in simultaneous communication complexity the gap between public coins and private coins may be unbounded: the EQ function, for example, has an $O(1)$ public-coin simultaneous protocol, but requires $\Omega(\sqrt{n})$ communication in the private-coin model [93, 9].

Example Simultaneous communication complexity is weaker than one-way communication complexity (see discussion in Chapter 5). An example in which there is an exponential gap between the two is the EQ function: while the Rabin-Yao $O(\log n)$ communication protocol can be carried out in the one-way model, an $\Omega(\sqrt{n})$ lower bound holds for the problem in the simultaneous model [93, 9].

2.3 Information theory

Information theory (cf. [40]) is the theory of efficient and reliable transmission of information over a network channel. Its applications, however, are far reaching, ranging from statistics and physics to economics and computer science. In this thesis we make fundamental use of information theory to study two computational models: sampling computations and communication complexity. In the following we survey some of the central notions of information theory and their basic properties, of which we make use in this thesis. For the sake of brevity, we do not provide proofs. Unless stated otherwise, proofs for all the results mentioned can be found in the classical textbook of Cover and Thomas [40].

2.3.1 Entropy and mutual information

Entropy The most important and well-known concept of information theory is the *entropy.* Intuitively, the entropy of a random variable X taking values in a set $\mathcal{X}$ is the amount of "uncertainty" in the choice of $x \in \mathcal{X}$ according to X, before this choice is made. This uncertainty is usually measured in bits (simple generalizations work for larger alphabets too), and it captures the number of bits necessary to encode the unknown choice of X. Two extreme examples, which may help the reader grasp the notion, are the following:

1. X is a constant random variable; in this case there is no uncertainty in X, and thus its entropy is 0.

2. X is uniformly distributed on $\mathcal{X}$. In this case, $\log|\mathcal{X}|$ bits are needed to encode the chosen element, implying the entropy of X is $\log|\mathcal{X}|$.

Formally, the entropy is defined as follows:

Definition 2.12 (Entropy) *Let μ be a distribution on a finite set $\mathcal{X}$, and let $X \sim \mu$. The* (binary) entropy *of X is:*

$$H(X) \stackrel{\text{def}}{=} \sum_{x \in \mathcal{X}} \mu(x) \cdot \log \frac{1}{\mu(x)}.$$

We also write $H(\mu)$ for the above quantity.

Some basic properties of the entropy are the following:

Proposition 2.13 (Basic properties of entropy)

1. $0 \leq H(X)$. *Equality if and only if X is constant.*

2. $H(X) \leq \log|\mathcal{X}|$. *Equality if and only if X is uniformly distributed on $\mathcal{X}$.*

3. $H(\mu)$ *is a concave function. That is, for any two distributions μ, ν on $\mathcal{X}$ and for any $0 \leq \lambda \leq 1$, $H(\lambda\mu + (1-\lambda)\nu) \geq \lambda H(\mu) + (1-\lambda) H(\nu)$.*

Binary entropy function One particular random variable for which the entropy is very useful is the Bernoulli random variable: $B(p)$ is a 0-1 valued random variable, with probability p for 1. We will denote by $H_2(p) \stackrel{\text{def}}{=} H(B(p))$ the *binary entropy function*. Note that in this case the entropy takes the following simple form: $H_2(p) = p\log(1/p) + (1-p)\log(1/(1-p))$. Figure 2.6 illustrates the graph of the binary entropy function, which demonstrates it basic properties: it is a concave function that takes values between 0 and 1, with minimum at $p = 0, 1$ and maximum at $p = 1/2$. It is instructive to see that for small values of p, $H_2(p) \approx p\log(1/p) \approx p$.

Joint entropy Let X and Y be two random variables. The *joint entropy* of X and Y is the uncertainty in the pair of random variables (X, Y):

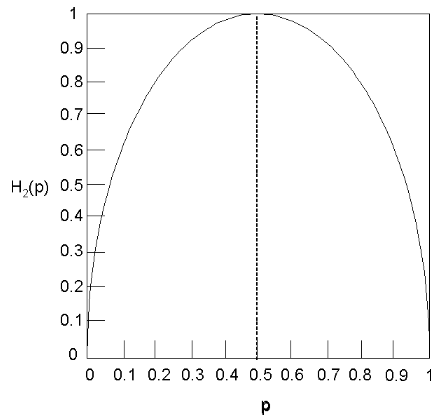

Figure 2.6: The graph of the binary entropy function

Definition 2.14 (Joint entropy) *The* joint entropy *of a pair of random variables X and Y with a joint distribution $\mu = (\mu_X, \mu_Y)$ on a set $\mathcal{X} \times \mathcal{Y}$ is:*

$$H(X,Y) \stackrel{\text{def}}{=} H((X,Y)) = \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} \mu(x,y) \log \frac{1}{\mu(x,y)}.$$

The most basic property of joint entropy is sub-additivity:

Proposition 2.15 (Sub-additivity of entropy) $H(X,Y) \le H(X) + H(Y)$. *Equality if and only if X and Y are independent.*

The definition of joint entropy and the sub-additivity property easily extend to an arbitrary number of random variables.

A generalization of the sub-additivity of entropy is what is known as the *entropy subset inequality* and is sometimes attributed to Han [63] (see proof in [40], page 491):

Theorem 2.16 (Entropy subset inequality) *For n random variables $X_1, \ldots, X_n$ and a set $S \subseteq [n]$, define $X_S = \{X_i \mid i \in S\}$. Define for each $k = 1, \ldots, n$,*

$$H_k^{(n)} \stackrel{\text{def}}{=} \frac{1}{\binom{n}{k}} \sum_{S:|S|=k} \frac{H(X_S)}{k}.$$

Then,

$$H_1^{(n)} \ge \cdots \ge H_n^{(n)}.$$

Note that $H_1^{(n)} = \frac{1}{n}\sum_{i=1}^{n} H(X_i)$ and $H_n^{(n)} = \frac{1}{n} H(X_1, \ldots, X_n)$. Therefore, $H_1^{(n)} \ge H_n^{(n)}$ is just the entropy sub-additivity.

Conditional entropy For an event A, the *conditional entropy* of X given A is the uncertainty left in X after knowing that the chosen element belongs to A:

Definition 2.17 (Conditional entropy given an event) *The* conditional entropy *of a random variable X with a distribution μ on $\mathcal{X}$ given an event $A \subseteq \mathcal{X}$ is:*

$$H(X \mid A) \stackrel{\text{def}}{=} H((X|A)) = \sum_{x \in \mathcal{X}} \mu(x|A) \log \frac{1}{\mu(x|A)}.$$

The *conditional entropy* of a random variable X given a random variable Y is the uncertainty left in X after knowing Y. Intuitively, if Y is very related to X, knowing Y might decrease the uncertainty in X. If, however, Y is independent of X, then the uncertainty in X will not change even given Y. Formally,

Definition 2.18 (Conditional entropy given a random variable) *Let X and Y be a pair of random variables with joint distribution $\mu = (\mu_X, \mu_Y)$ on $\mathcal{X} \times \mathcal{Y}$. The* conditional entropy *of X given Y is:*

$$\begin{aligned} H(X \mid Y) &\stackrel{\text{def}}{=} \sum_{y \in \mathcal{Y}} \mu_Y(y) H(X \mid Y = y) \\ &= \sum_{y \in \mathcal{Y}} \mu_Y(y) \sum_{x \in \mathcal{X}} \mu_X(x|Y=y) \log \frac{1}{\mu_X(x|Y=y)} \end{aligned}$$

The most basic property of conditional entropy is the following:

Proposition 2.19 (Conditioning reduces entropy) *For any two random variables X, Y, $H(X \mid Y) \leq H(X)$. Equality if and only if X and Y are independent.*

Conditional entropy is always non-negative:

Proposition 2.20 (Non-negativity of conditional entropy) $H(X \mid Y) \geq 0$. *Equality if and only if $X = f(Y)$ for some function f.*

The following chain rule for entropy allows one to express the joint entropy in terms of entropy and conditional entropy:

Proposition 2.21 (Chain rule for entropy) *For any two random variables X, Y,*

$$H(X, Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y).$$

And in general, for any random variables $X_1, \ldots, X_n$,

$$H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i \mid X_1, \ldots, X_{i-1}).$$

We note that most of the properties of entropy work also for conditional entropy (such as the sub-additivity of entropy and the entropy subset inequality). In some cases one has to be more cautious. For example,

Proposition 2.22 *For any three random variables X, Y, Z, for which X, Y are jointly independent of Z (i.e., the pair of random variables (X, Y) is independent of Z), then*

$$H(X \mid Y, Z) \;=\; H(X \mid Y).$$

Mutual information Mutual information is another central concept in information theory, measuring the amount of information one random variable has about another. It is formally defined as follows:

Definition 2.23 (Mutual information) *The* mutual information *between two random variables X, Y is:*

$$I(X;\, Y) \;\stackrel{\text{def}}{=}\; H(X) - H(X \mid Y) \;=\; H(Y) - H(Y \mid X).$$

Thus, the mutual information between X and Y is the reduction in uncertainty in X resulting from knowing Y, which happens to be (cf. the chain rule for entropy) the same as the reduction in uncertainty in Y resulting from knowing X.

A particularly useful trick for memorizing the complicated connections among entropy, joint entropy, conditional entropy, and mutual information is the Venn Diagram view (see Figure 2.7). In this view the entropy of each random variable is associated with a set in the diagram, the mutual information is the intersection of the sets, the joint entropy is the union of the sets, and the conditional entropy is the subtraction of the sets. We should warn the reader that the Venn diagram view is effective in deriving identities among information quantities; it is misleading when trying to derive inequalities.

The basic properties of mutual information are the following:

Proposition 2.24 (Basic properties of mutual information)

1. $I(X;\, Y) \geq 0$. *Equality if and only if X and Y are independent.*
2. $I(X;\, Y) = I(Y;\, X)$.
3. $I(X;\, X) = H(X)$.

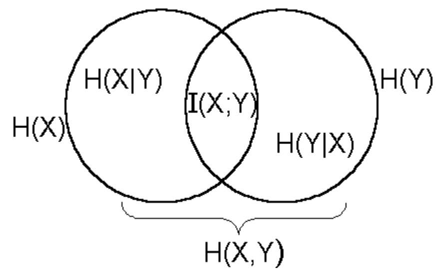

Figure 2.7: Venn diagram view of entropy and mutual information

The *conditional mutual information* measures the amount of information between two random variables after knowing another random variable or event:

Definition 2.25 (Conditional mutual information) *The* conditional mutual information *between two random variables X, Y given an event A is:*

$$I(X;\ Y \mid A) \stackrel{\text{def}}{=} H(X \mid A) - H(X \mid Y, A) \ = \ I(X|A;\ Y|A).$$

For three random variables X, Y, Z with joint distribution $\mu = (\mu_X, \mu_Y, \mu_Z)$ on $\mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, the conditional mutual information *between X, Y given Z is:*

$$I(X;\ Y \mid Z) \stackrel{\text{def}}{=} H(X \mid Z) - H(X \mid Y, Z) \ = \ \sum_{z \in \mathcal{Z}} \mu_Z(z) \cdot I(X;\ Y \mid Z = z).$$

The chain rule for mutual information allows one to rewrite the mutual information between ordered pairs of random variables as simple mutual information and conditional mutual information:

Proposition 2.26 (Chain rule for mutual information) *For any X, Y, Z,*

$$I(X, Y;\ Z) \ = \ I(X;\ Z) + I(Y;\ Z \mid X) \ = \ I(Y;\ Z) + I(X;\ Z \mid Y).$$

And in general, for any random variables $X_1, \ldots, X_n, Y$,

$$I(X_1, \ldots, X_n;\ Y) = \sum_{i=1}^{n} I(X_i;\ Y \mid X_1, \ldots, X_{i-1}).$$

One of the most important properties of mutual information is the *data processing inequality*:

Theorem 2.27 (Data processing inequality) *For any random variables X, Y, Z for which X, Z are conditionally independent given Y (i.e., for every choice of y, the conditional random variables $X|\{Y = y\}$ and $Z|\{Y = y\}$ are independent),*

$$I(X;\ Y) \ \geq \ I(X;\ Z).$$

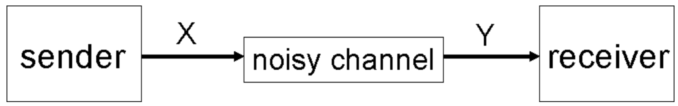

Figure 2.8: The classical information theory scenario: communication over a noisy channel

Intuitively, the interpretation of this theorem is as follows: Since X and Z are independent given Y, one can think of Z as the product of some "processing" made to Y. Therefore, the information Y has about X surely subsumes the information Z has about X. This inequality has several immediate useful consequences:

Corollary 2.28 (Consequences of the data processing inequality)

1. *For any random variables X, Y and function $f(Y)$, $I(X;\ Y) \geq I(X;\ f(Y))$.*

2. *If X, Z are conditionally independent given Y, then $I(X;\ Y \mid Z) \leq I(X;\ Y)$.*

Note that the second consequence is interesting, since in general conditioning does not reduce mutual information.

2.3.2 Fano's inequality and the MLE principle

The classical scenario studied in information theory is depicted in Figure 2.8: a sender wishes to communicate a message x to a receiver through a noisy channel. x is drawn from a domain $\mathcal{X}$ according to some distribution μ_X. The message received y is drawn from a domain $\mathcal{Y}$ according to a distribution μ_Y, which depends on x and on the random noise in the channel. The goal of the receiver is to recover x from y. In order to do that, the receiver applies some reconstruction function $g : \mathcal{Y} \to \mathcal{X}$ to y. Clearly, the reconstruction may incur some error: $\delta_g \stackrel{\text{def}}{=} \Pr(g(Y) \neq X)$ (where $X \sim \mu_X$ and $Y \sim \mu_Y$). Fano's inequality and the MLE principle are two central results in information theory giving lower and upper bounds on the reconstruction error δ_g.

Fano's inequality [45] (cf. [40], pages 38–40) gives a lower bound on the error of any reconstruction function the receiver may apply in terms of the conditional entropy of the sent message X given the received message Y:

Theorem 2.29 (Fano's inequality) *For any reconstruction function $g : \mathcal{Y} \to \mathcal{X}$,*

$$H_2(\delta_g) + \delta_g \log(|\mathcal{X}| - 1) \quad \geq \quad H(X \mid Y).$$

Note that in the special case that $\mathcal{X}$ is binary, we have a more explicit lower bound on the error probability (recall that $H_2(\delta_g) \approx \delta_g$ for small δ_g):

$$H_2(\delta_g) \geq H(X \mid Y).$$

The intuition behind the proof is clear: no matter what the receiver does, he cannot recover the information lost about X in the channel. This loss is measured in terms of the uncertainty in the message X given the received message Y.

The MLE principle (cf. [61], Chapter 17 and [43], pages 26–27) gives a specific reconstruction function—the *Maximum Likelihood Estimator (MLE)*—which almost matches the error lower bound given by Fano's inequality. The MLE function is simple: $MLE(y) = x$ for the x that maximizes the conditional probability $\mu_X(x|Y = y)$. Ties can be broken arbitrarily.

Theorem 2.30 (The MLE principle) $\delta_{MLE} \leq 1 - \frac{1}{2^{H(X|Y)}}$.

Note that in the case $\mathcal{X}$ is binary, $\delta_{MLE} \leq H(X \mid Y)$, which almost matches Fano's inequality, modulo the difference between δ_{MLE} and $H_2(\delta_{MLE})$.

2.4 Statistical decision theory

Statistical decision theory (cf. [22]) is one of the core areas of theoretical statistics. It is concerned with the problem of making successful decisions in an environment of uncertainty. The theory has applications in almost all sciences, including medicine, biology, economics, engineering, physics, and others. We will concentrate on three concepts from this area, which play a crucial role in the results of this thesis: statistical distance measures, statistical hypothesis testing, and statistical sequential analysis.

2.4.1 Statistical distance measures

The statistics literature discusses a wide variety of measures of distance between distributions. Each measure has its own merits that make it desirable to use in certain circumstances. In statistical decision theory these measures are used to quantify the loss incurred by unsuccessful decisions. The usefulness of these measures, however, extends far beyond their application in statistical decision theory.

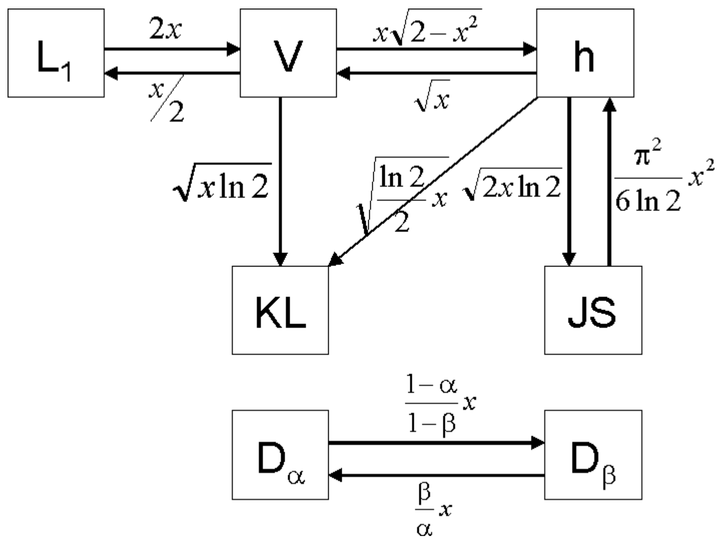

Figure 2.9: Statistical distance measures and the relations among them. An arrow from one distance d to another distance d' labeled by a function f, means that $d \leq f(d')$.

In the following we review some prominent distance measures, of which we make use in the thesis. We outline their special properties, and highlight the relations among them. A recent survey by Gibbs and Su [53] gives a good overview of these distance measures and others. Other references include [110, 83]. We restrict our discussion to distributions over finite probability spaces. μ and ν will denote two distributions over the same space $\mathcal{X}$.

Figure 2.9 illustrates the distance measures we discuss in this section and the relations among them.

Total variation distance Perhaps the best known (at least to computer scientists) statistical distance measure is the *total variation distance* (also known as *variational distance* or simply *statistical distance*), which measures the largest difference in probability of an event between the two distributions:

Definition 2.31 (Total variation distance)

$$V(\mu, \nu) \stackrel{\text{def}}{=} \max_{A \subseteq \mathcal{X}} |\mu(A) - \nu(A)| .$$

Total variation distance takes values in the interval $[0, 1]$. It has a convenient characterization as half of the L_1 distance between μ and ν (the proof is trivial):

Proposition 2.32 $V(\mu,\nu) \;=\; \frac{1}{2}L_1(\mu,\nu) \;=\; \frac{1}{2}\sum_{x\in\mathcal{X}}|\mu(x)-\nu(x)|\,.$

This characterization shows that the total variation distance is a metric, and in particular it is symmetric and respects the triangle inequality.

The following lemma (whose proof can be found in [110], Appendix 3) bounds the total variation distance between product distributions in terms of the total variation distance between the individual distributions:

Lemma 2.33 *For any distributions μ_1,ν_1 on $\mathcal{X}_1$ and μ_2,ν_2 on $\mathcal{X}_2$,*

$$V(\mu_1\times\nu_1,\mu_2\times\nu_2) \;\le\; V(\mu_1,\mu_2)+V(\nu_1,\nu_2)\,.$$

In some situations, the above bound is too weak. The Hellinger distance, discussed below, provides a tighter connection between a product distribution and its constituent distributions.

Applying a fixed function to two distributions can only reduce the distance between them:

Proposition 2.34 *For any distributions μ,ν on $\mathcal{X}$ and for any function $f:\mathcal{X}\to\mathcal{Y}$,*

$$V(f(\mu),f(\nu)) \;\le\; V(\mu,\nu)\,.$$

Hellinger distance A particularly useful measure is the *Hellinger distance* (cf., e.g., [83]), which is both a true metric and possesses a strong decomposition property with respect to product distributions:

Definition 2.35 (Hellinger distance) $h(\mu,\nu) \;\stackrel{\text{def}}{=}\; \sqrt{\frac{1}{2}\sum_{x\in\mathcal{X}}\left(\sqrt{\mu(x)}-\sqrt{\nu(x)}\right)^2}.$

The Hellinger distance takes values in the interval $[0,1]$. It has the following equivalent characterization (the proof is trivial):

Proposition 2.36

$$h(\mu,\nu) \;=\; \sqrt{1-\sum_{x\in\mathcal{X}}\sqrt{\mu(x)}\sqrt{\nu(x)}}.$$

The quantity $1-h^2(\mu,\nu)$ is called the *Hellinger affinity*, and it satisfies the following decomposition property (the proof appears in Section A.1.1 of the Appendix):

Proposition 2.37 *For any distributions μ_1,ν_1 on $\mathcal{X}_1$ and μ_2,ν_2 on $\mathcal{X}_2$,*

$$1-h^2(\mu_1\times\mu_2,\nu_1\times\nu_2) \;=\; (1-h^2(\mu_1,\nu_1))\cdot(1-h^2(\mu_2,\nu_2)).$$

The property easily extends by induction to products of an arbitrary number of distributions. The following relation holds between the Hellinger and the total variation distances (the proof appears in Section A.1.1 of the Appendix):

Proposition 2.38 (cf. [83])

$$h^2(\mu,\nu) \;\leq\; V(\mu,\nu) \;\leq\; h(\mu,\nu)\sqrt{2-h^2(\mu,\nu)}.$$

Equivalently, we can state the bounds as follows:

Corollary 2.39

$$1-\sqrt{1-V^2(\mu,\nu)} \;\leq\; h^2(\mu,\nu) \;\leq\; V(\mu,\nu).$$

KL distance The *Kullback-Leibler divergence* [80], or simply the *KL distance* (also known as *relative entropy*, *cross entropy*, and *information divergence*) is one of the most widely used measures of distance in statistics and information theory:

Definition 2.40 (KL distance)

$$D_{KL}(\mu \parallel \nu) \;\stackrel{\text{def}}{=}\; \sum_{x\in\mathcal{X}} \mu(x)\log\frac{\mu(x)}{\nu(x)}.$$

KL distance takes values in the interval $[0,\infty]$. It is infinite if the support of μ contains values that are not in the support of ν. It is not a true metric, since it is not symmetric and does not satisfy the triangle inequality. However, it is always non-negative, and it is 0 if and only if $\mu=\nu$ (cf. [40]).

In spite of its obvious deficiencies, the KL distance possesses many useful properties, some of which we review below. Unless stated otherwise, proofs of these properties can be found in [40], Chapter 2.

The mutual information between two distribution μ and ν is fully characterized as the KL distance between the joint distribution of μ and ν and their product distribution:

Proposition 2.41 $I(\mu;\,\nu) \;=\; D_{KL}((\mu,\nu)\parallel\mu\times\nu)$.

Conditional KL distance is defined as follows:

Definition 2.42 (Conditional KL distance) *For joint distributions* $\mu=(\mu_X,\mu_Y)$ *and* $\nu=(\nu_X,\nu_Y)$ *on* $\mathcal{X}\times\mathcal{Y}$*, let* $(X,Y)\sim\mu$ *and* $(W,Z)\sim\nu$*. Then,*

$$D_{KL}(\mu_X|\mu_Y \parallel \nu_X|\nu_Y) \;\stackrel{\text{def}}{=}\; \sum_{y\in\mathcal{Y}}\mu_Y(y)\sum_{x\in\mathcal{X}}\mu_X(x|Y=y)\log\frac{\mu_X(x|Y=y)}{\nu_X(x|Z=y)}.$$

The chain rule for KL distance allows us to expand the KL distance between two joint distributions of two random variables as the sum of KL distance and conditional KL distance:

Proposition 2.43 (Chain rule for KL distance) *For joint distributions $\mu = (\mu_X, \mu_Y)$ and $\nu = (\nu_X, \nu_Y)$ on $\mathcal{X} \times \mathcal{Y}$,*

$$D_{KL}(\mu \parallel \nu) \;=\; D_{KL}(\mu_Y \parallel \nu_Y) + D_{KL}(\mu_X|\mu_Y \parallel \nu_X|\nu_Y).$$

The KL distance is a convex function:

Proposition 2.44 (Convexity of the KL distance) *For any four distributions μ_1, μ_2, ν_1, ν_2 on $\mathcal{X}$ and for any $0 \le \lambda \le 1$,*

$$\begin{aligned} D_{KL}(\lambda\mu_1 + (1-\lambda)\mu_2 \parallel \lambda\nu_1 + (1-\lambda)\nu_2) \;\le \\ \lambda \cdot D_{KL}(\mu_1 \parallel \nu_1) + (1-\lambda) \cdot D_{KL}(\mu_2 \parallel \nu_2). \end{aligned}$$

The KL distance is bounded from below by the Hellinger distance and the total variation distance (the latter bound is known as Pinsker's inequality; the proof appears in Section A.1.2 of the Appendix):

Proposition 2.45 (cf. [110, 40])

$$D_{KL}(\mu \parallel \nu) \;\ge\; \frac{2}{\ln 2}\, h^2(\mu, \nu) \;\ge\; \frac{1}{\ln 2}\, V^2(\mu, \nu).$$

Rényi α-divergence The Rényi α-divergence [111] is a family of distance measures that interpolate between the Hellinger distance and the KL distance:

Definition 2.46 (Rényi α-divergence) *For $0 < \alpha < 1$,*

$$D_\alpha(\mu \parallel \nu) \;\stackrel{\text{def}}{=}\; 1 - \sum_{x \in \mathcal{X}} \mu(x)^\alpha \nu(x)^{1-\alpha}.$$

By Hölder's inequality, the α-divergence takes values in the interval $[0,1]$. By definition, $D_{\frac{1}{2}}$ is the square of the Hellinger distance. $\frac{1}{\alpha(1-\alpha)} D_\alpha(\mu \parallel \nu)$ tends to $D_{KL}(\mu \parallel \nu)$ as $\alpha \to 1$. Like the KL distance, the α-divergence is not a true metric, since it is not symmetric and does not satisfy the triangle inequality. The only value of α for which D_α is a square of a metric is $\frac{1}{2}$. The advantage of the α-divergence over the KL distance is that it is always bounded.

The following connection holds between D_α and $D_{1-\alpha}$ (the proof is trivial):

Proposition 2.47 $D_\alpha(\mu \parallel \nu) = D_{1-\alpha}(\nu \parallel \mu)$.

D_α and D_β for $\alpha \neq \beta$ satisfy the following relation (the proof appears in Section A.1.3 of the Appendix):

Proposition 2.48 $\forall \alpha < \beta$,

$$\frac{\alpha}{\beta} D_\beta(\mu \parallel \nu) \;\leq\; D_\alpha(\mu \parallel \nu) \;\leq\; \frac{1-\alpha}{1-\beta} D_\beta(\mu \parallel \nu).$$

The value $1 - D_\alpha(\mu \parallel \nu)$ is called the α-*affinity*. Like the Hellinger affinity, it satisfies a decomposition property for product distributions (the proof is a straightforward generalization of the proof of Proposition 2.37):

Proposition 2.49 *For any distributions μ_1, ν_1 on $\mathcal{X}_1$ and μ_2, ν_2 on $\mathcal{X}_2$,*

$$1 - D_\alpha(\mu_1 \times \mu_2 \parallel \nu_1 \times \nu_2) \;=\; (1 - D_\alpha(\mu_1 \parallel \nu_1)) \cdot (1 - D_\alpha(\mu_2 \parallel \nu_2)).$$

Jensen-Shannon divergence The Jensen-Shannon divergence [29, 30, 85] is a bounded symmetric version of the KL distance:

Definition 2.50 (Jensen-Shannon divergence)

$$JS(\mu, \nu) \;\stackrel{\text{def}}{=}\; \frac{1}{2}\left(D_{KL}\left(\mu \parallel \frac{\mu+\nu}{2}\right) + D_{KL}\left(\nu \parallel \frac{\mu+\nu}{2}\right)\right).$$

The Jensen-Shannon divergence is symmetric by definition. It is bounded due to the following connection to the Hellinger distance (the upper bound is by Jayram [70]; the lower bound is implicit in [85]; a proof of slightly weaker bounds appears in Section A.1.4 of the Appendix):

Proposition 2.51 ([85, 70]) $h^2(\mu, \nu) \;\leq\; JS(\mu, \nu) \;\leq\; \ln 2 \cdot h^2(\mu, \nu)$.

The Jensen-Shannon divergence can be generalized to measure the mutual distance among more than two distributions, and with non-uniform weights:

Definition 2.52 (Generalized Jensen-Shannon divergence) *For a distribution λ on $[n]$, and for distributions $\mu_1, \ldots, \mu_n$ on $\mathcal{X}$, we define $\mu_\lambda \stackrel{\text{def}}{=} \sum_{i=1}^n \lambda(i) \cdot \mu_i$ to be the* λ-weighted average distribution. *The* λ-generalized Jensen-Shannon divergence *of $\mu_1, \ldots, \mu_n$ is:*

$$JS_\lambda(\mu_1, \ldots, \mu_n) \;\stackrel{\text{def}}{=}\; \mathrm{E}_{I \sim \lambda}[D_{KL}(\mu_I \parallel \mu_\lambda)] \;=\; \sum_{i=1}^n \lambda(i) \cdot D_{KL}(\mu_i \parallel \mu_\lambda).$$

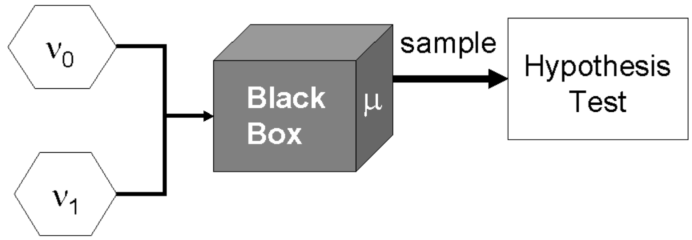

Figure 2.10: Simple statistical hypothesis testing

The Jensen-Shannon divergence is important due to the following connection to mutual information (the proof appears in Section A.1.4 of the Appendix):

Proposition 2.53 *Let μ be a distribution on $\mathcal{X} = \{x_1, \ldots, x_n\}$, let ν be an independent distribution on $\mathcal{Y}$, and let $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function. Then,*

$$I(\mu;\ f(\mu, \nu)) \;=\; JS_\mu\left(f(x_1, \nu), \ldots, f(x_n, \nu)\right).$$

The convexity of the KL distance implies the following relation between the generalized Jensen-Shannon divergence and the KL distance (the proof is trivial):

Proposition 2.54 $JS_\lambda(\mu_1, \ldots, \mu_n) \;\le\; \mathrm{E}_{(X,Y)\sim\lambda\times\lambda}\left[D_{KL}(\mu_X \parallel \mu_Y)\right]$.

2.4.2 Statistical hypothesis testing

The simplest version of statistical hypothesis testing (see Figure 2.10) is the following: a "black box" contains a distribution μ, which is guaranteed to be one of two distributions: either ν_0 or ν_1. A hypothesis test is a procedure, which based on a sample from the distribution μ, has to decide, with small probability of error, which of the two holds: either $H_0 : \mu = \nu_0$ (the *null hypothesis*) or $H_1 : \mu = \nu_1$ (the *alternative hypothesis*). Formally, a simple hypothesis test is defined as follows:

Definition 2.55 (Simple hypothesis test) *A δ-error* simple hypothesis test[2] *for two distributions ν_0, ν_1 on a domain $\mathcal{X}$ is a Boolean function $T : \mathcal{X} \to \{0, 1\}$, which satisfies for $b = 0, 1$,*

$$\Pr_{X\sim\nu_b}(T(X) = b) \;>\; 1 - \delta.$$

[2] In statistics one distinguishes between two types of error the test can make: either it decides H_0, when in fact H_1 is true, or it decides H_1, when H_0 is true. In this thesis, however, there will be no semantic difference between the two types of error, and therefore we relate just one error parameter to a hypothesis test.

Hypothesis testing is closely related to the total variation distance. The following proposition shows that the total variation distance between ν_0 and ν_1 is exactly the error probability of the best hypothesis test for ν_0, ν_1:

Proposition 2.56 *For two distributions ν_0, ν_1 on the same domain $\mathcal{X}$, let T^* be the simple hypothesis test for ν_0, ν_1 with the minimum error among all tests for ν_0, ν_1. Let δ^* be the error probability of T^*. Then, $\delta^* = (1 - V(\nu_0, \nu_1))/2$.*

Proof: Consider the event $A \stackrel{\text{def}}{=} \{x \in \mathcal{X} \mid T^*(x) = 0\}$. A is the set of instances on which the test decides H_0. Since T^* is erroneous with probability at most δ^*, then $\nu_0(A) \geq 1 - \delta^*$ and $\nu_1(A) \leq \delta^*$. Now, by the definition of total variation distance,

$$V(\nu_0, \nu_1) \;\geq\; |\nu_0(A) - \nu_1(A)| \;\geq\; 1 - 2\delta^*.$$

For the other direction, let A' be the event for which $V(\nu_0, \nu_1) = |\nu_0(A') - \nu_1(A')|$. Assume, without loss of generality, that $\nu_1(A') \geq \nu_0(A')$ (otherwise, replace A' by its complement). Define a hypothesis test T' which is the characteristic function of A': $T'(x) = 1$ if and only if $x \in A'$. If $\mu = \nu_0$, the error probability of T' is $\nu_0(A')$, and if $\mu = \nu_1$, its error probability is $\nu_1(A'^c) = 1 - \nu_1(A')$. From the optimality of T^*, we have, $\nu_0(A') + 1 - \nu_1(A') \geq 2\delta^*$. Therefore,

$$V(\nu_0, \nu_1) = \nu_1(A') - \nu_0(A') \;\leq\; 1 - 2\delta^*.$$

□

We will sometimes deal with *randomized* hypothesis tests, which may use internal random coins to make their decision:

Definition 2.57 (Simple randomized hypothesis test) *A* δ-error simple randomized hypothesis test *for two distributions ν_0, ν_1 on a domain $\mathcal{X}$ is a function $T : \mathcal{X} \times \mathcal{R} \to \{0, 1\}$, which satisfies for $b = 0, 1$,*

$$\Pr(T(X, R) = b) \;>\; 1 - \delta,$$

where $X \sim \nu_b$ and R is uniformly distributed on $\mathcal{R}$.

The following proposition shows that randomized hypothesis tests are not different qualitatively from deterministic ones:

Proposition 2.58 *For two distributions ν_0, ν_1 on the same domain $\mathcal{X}$, let T^* be the randomized hypothesis test for ν_0, ν_1 with the minimum error among all randomized tests for ν_0, ν_1. Let δ^* be the error probability of T^*. Then, $\delta^* = (1 - V(\nu_0, \nu_1))/2$.*

Proof: The direction $\delta^* \leq (1 - V(\nu_0, \nu_1))/2$ follows from Proposition 2.56, because every deterministic hypothesis test is a degenerate randomized one.

For the other direction, we view T^* as a deterministic hypothesis test for the distributions $\nu_0 \times \rho, \nu_1 \times \rho$, where ρ is the distribution of random coins R. By Proposition 2.56, $V(\nu_0 \times \rho, \nu_1 \times \rho) \geq 1 - 2\delta^*$. We now use Lemma 2.33, and obtain:

$$1 - 2\delta^* \;\leq\; V(\nu_0 \times \rho, \nu_1 \times \rho) \;\leq\; V(\nu_0, \nu_1) + V(\rho, \rho) \;=\; V(\nu_0, \nu_1).$$

$\square$

Assume now we are given two distributions ν_0, ν_1, and an arbitrary parameter $\delta > 0$, and we would like to design a δ-error hypothesis test for ν_0, ν_1. Propositions 2.56 and 2.58 above pose a limit on the performance of any simple test for ν_0, ν_1: it must err with probability at least $(1 - V(\nu_0, \nu_1))/2$. Thus, in order to design hypothesis tests with smaller error probabilities, we need to allow the test to draw more samples from the distribution μ:

Definition 2.59 (Hypothesis test) *A δ-error* hypothesis test *for two distributions ν_0, ν_1 on a domain $\mathcal{X}$ is a function $T : \mathcal{X}^k \to \{0, 1\}$, which satisfies for $b = 0, 1$,*

$$\Pr_{\vec{X} \sim \nu_b^k}\left(T(\vec{X}) = b\right) \;>\; 1 - \delta,$$

where ν_b^k is the k-fold product distribution of ν_b. $S(T) = k$ is called the sample cost *of T.*

Thus, a general hypothesis test draws k independent samples from μ, and in the end determines which hypothesis holds with error probability δ. A general test for ν_0, ν_1 can be viewed equivalently as a simple test for ν_0^k, ν_1^k. Note that since the total variation distance decreases with the number of samples (Lemma 2.33), then by taking k to be large enough we can guarantee that there is a δ-error test for ν_0, ν_1. The exact dependence of k on δ is one of the issues discussed in our sampling lower bounds (see Section 4.2.2).

A general randomized hypothesis test is defined in a similar way. Hypothesis testing can be generalized to test multiple hypotheses:

Definition 2.60 (Multiple hypotheses test) *A δ-error* multiple hypotheses test *for a family of distributions $\{\nu_b\}_{b \in \mathcal{B}}$ on a domain $\mathcal{X}$ is a function $T : \mathcal{X}^k \to \mathcal{B}$, which satisfies for all $b \in \mathcal{B}$,*

$$\Pr_{\vec{X} \sim \nu_b^k}\left(T(\vec{X}) = b\right) \;>\; 1 - \delta.$$

2.4.3 Sequential analysis

In the previous section a hypothesis test was defined such that it always uses a fixed number of samples (i.e., k samples). This definition requires one to choose k large enough to guarantee a small probability of error in the worst possible scenario. However, in some situations, we typically need far less samples than k in order to get to the right decision. Consider, for example, a test for the two following distributions ν_0, ν_1 on $\{0,1\}$: $\nu_0(0) = 1$, and $\nu_1(0) = 1 - \epsilon$ for some small $\epsilon > 0$. In the worst-case $k = O(1/\epsilon \cdot \log(1/\delta))$ samples are required for a δ-error test for ν_0, ν_1. However, note that if a test draws the samples sequentially, and at some point it sees a sample whose value is 1, it can immediately stop sampling and declare (with full confidence) that the distribution is ν_1.

The above example is one simple motivation for *sequential hypothesis tests*, defined as follows:

Definition 2.61 (Sequential hypothesis test) *A* δ-error sequential hypothesis test *for a family of distributions* $\{\nu_b\}_{b\in\mathcal{B}}$ *on a domain* $\mathcal{X}$ *is an oracle procedure* T^{μ}*, which satisfies for all* $b \in \mathcal{B}$*,*

$$\Pr_{\vec{X}\sim\nu_b^*}\left(T^{\vec{X}} = b\right) \; > \; 1-\delta.$$

Each oracle call of T^{μ} *produces a sample from* μ *independent from all previous samples. We define* $S\left(T^{\vec{X}}\right)$ *to be the* sample cost *of* T*—this is a random variable denoting the number of oracle calls made by* T *throughout its execution.* $S^{E}_{\nu_b}(T) = \mathrm{E}_{\vec{X}\sim\nu_b^*}\left[S\left(T^{\vec{X}}\right)\right]$ *is called the* expected sample cost *of* T *on* ν_b*, and* $S^{W}_{\nu_b}(T) = \max_{\vec{X}\in\mathrm{SUPP}(\nu_b^*)} S\left(T^{\vec{X}}\right)$ *is called the* worst-case sample cost *of* T *on* ν_b*.*

Randomized sequential hypothesis tests can use internal random coins to make their decision. For such tests, the expected and the worst-case sample costs are the expectation and the maximum, respectively, over both the samples from ν_b and over the internal random coin tosses.

A particularly useful fact in analyzing probabilistic sequential procedures is the famous "Wald's identity" (proof can be found, e.g., in [115], Section 2.2):

Theorem 2.62 (Wald's identity) *Let* $X_1, X_2, \ldots$ *be an infinite sequence of independent and identically distributed random variables with mean* μ*. Let* S *be a random variable on* $\{0,1,2,\ldots\}$*, for which the event* $\{S = k\}$ *is independent of* $X_{k+1}, X_{k+2}, \ldots$ *for all* k *(*S *is* *called a* stopping time random variable*). We further assume* $\mathrm{E}[S] < \infty$*. Then,*

$$\mathrm{E}\left[\sum_{i=1}^{S} X_i\right] = \mu \cdot \mathrm{E}[S].$$

Chapter 3

The Models and the Basic Reductions

In this chapter we define the three computational models we study in this thesis: sampling computations, data stream computations, and sketch computations. Aside from formal definitions of the models and their associated complexity measures, we discuss their concrete, non-uniform counterparts: randomized decision trees for sampling computations, randomized oblivious read-once branching programs for data stream computations, and public-coin multi-party simultaneous communication complexity for sketch computations. The non-uniform models are stronger than the original models—they are able to compute a larger class of functions—but they are more amenable to lower bound techniques. Thus the strategy for proving lower bounds for the original models would be to prove lower bounds for the non-uniform models. This strategy is pervasive in computational complexity; proving running time lower bounds via circuit complexity is perhaps the most well-known example.

A non-uniform algorithm for a function $f : \mathcal{X}^* \to \mathcal{Z}$ is in fact a countably infinite family of algorithms, the n-th member of which is responsible for computing $f_n : \mathcal{X}^n \to \mathcal{Z}$, the restriction of f to inputs of length n. Such an infinite specification is of course unacceptable as a realistic computational model, but for the purpose of proving lower bounds it is actually convenient. We can restrict our attention to an arbitrary fixed input length n, and argue about lower bounds for f_n as a function of n.

All the three computational models—sampling, data stream, and sketch—produce approximations rather than exact computations. We therefore begin the chapter (Section

3.1) by presenting an abstraction of the notion of approximation, which uniformly captures all types of approximation (e.g., relative, additive, etc.). This abstraction allows us to argue uniformly about functions computed in these models, regardless of the type of approximation involved.

In Sections 3.2, 3.3, and 3.4 we formally define, respectively, the sampling model, the data stream model, and the sketch model. In Sections 3.3 and 3.4 we also describe the reductions from one-way and simultaneous communication complexity to, respectively, data stream and sketch computations. We end the chapter (Section 3.5) with an outline of some elementary relations among the three models.

3.1 Abstraction of approximation

An ϵ-approximation of a function $f : \mathcal{X} \to \mathcal{Z}$ ($\mathcal{X}, \mathcal{Z}$ are some arbitrary sets) is a mapping that assigns to every input $x \in \mathcal{X}$ a subset of range values $C \subseteq \mathcal{Z}$. Intuitively, the elements of C are exactly the values that are considered a "good" approximation (i.e., ϵ-approximation) to $f(x)$. Different values of the error parameter ϵ induce different mappings. However, these mappings must satisfy two natural requirements, as specified in the following definition:

Definition 3.1 (Approximation) *An* approximation *for a function $f : \mathcal{X} \to \mathcal{Z}$ is a family of functions $\{C_{f,\epsilon} : \mathcal{X} \to 2^{\mathcal{Z}}\}_{\epsilon \geq 0}$, parameterized by an error parameter ϵ that satisfies the following two conditions:*

(1) Containment of the truth: $\forall x \in \mathcal{X}, \quad C_{f,0}(x) = \{f(x)\}$.

(2) Nesting: $\forall x \in \mathcal{X},\ \forall \epsilon > \epsilon' \geq 0, \quad C_{f,\epsilon}(x) \supseteq C_{f,\epsilon'}(x)$.

We call $C_{f,\epsilon}(x)$ the ϵ-approximation set of the input x, and each value $z \in C_{f,\epsilon}(x)$ an ϵ-approximation of $f(x)$.

One might wonder why we define $C_{f,\epsilon}$ as a function of the inputs rather than as a function of the output values, i.e., why doesn't $C_{f,\epsilon}$ determine for each value in $\mathcal{Z}$ which other values in $\mathcal{Z}$ are an ϵ-approximation to it? The rank and property testing approximations described below show that in some cases the approximation is indeed a function of the inputs: there can be two different inputs with the same output value but with different approximation sets.

Throughout this thesis, for simplicity of notation, we implicitly assume that every function f is associated with a single approximation $C_{f,\epsilon}$ (the "natural" approximation for f).

Following are a few examples of popular approximation notions expressed in the form of the definition above. In all of them we define $C_{f,0}(x) \stackrel{\text{def}}{=} \{f(x)\}$.

1. *Additive approximation:* $(\mathcal{Z}, d_{\mathcal{Z}})$ is a metric space.
   $C_{f,\epsilon}(x) \;=\; \{z \in \mathcal{Z} \;\mid\; d_{\mathcal{Z}}(z, f(x)) < \epsilon\}$ for $\epsilon > 0$.

2. *Relative approximation:* $\mathcal{Z} = \mathbb{R}$.
   $C_{f,\epsilon}(x) \;=\; \{z \in \mathcal{Z} \;\mid\; (1-\epsilon)f(x) < z < (1+\epsilon)f(x)\}$ for $0 < \epsilon \leq 1$.

3. *Ratio approximation:* $\mathcal{Z} = \mathbb{R}$.
   $C_{f,\epsilon}(x) \;=\; \{z \in \mathcal{Z} \;\mid\; (1-\epsilon)f(x) < z < (1/(1-\epsilon))f(x)\}$ for $0 < \epsilon < 1$.

4. *Rank approximation:* An approximation notion for selection functions, like the median. The input for $\textsc{Select}_p$, the selection function of order p ($0 \leq p \leq 1$), is a set of n real numbers $x_1, \ldots, x_n$. If we order these numbers from smallest to largest: $x_{i_0} \leq \cdots \leq x_{i_{n-1}}$, then $\textsc{Select}_p(x_1, \ldots, x_n) = x_{i_{\lfloor p(n-1) \rfloor}}$. For example, the median is $\textsc{Select}_{\frac{1}{2}}$, minimum is $\textsc{Select}_0$, and maximum is $\textsc{Select}_1$. The approximation for $0 < \epsilon \leq 1$ is: $C_{\textsc{Select}_{p,\epsilon}}(x_1, \ldots, x_n) = \{x_{i_j} \mid j \in (\lfloor p'(n-1) \rfloor, \lfloor p''(n-1) \rfloor), p' = \max(p-\epsilon, 0), p'' = \min(p+\epsilon, 1)\}$.

5. *Property testing approximation* [112, 55]: An approximation notion for functions with a Boolean range. We assume some distance measure $d_{\mathcal{X}}$ on the input domain. The approximation for $\epsilon > 0$ is: $C_{f,\epsilon}(x) = \{1\}$, if $d_{\mathcal{X}}(x, y) < \epsilon/2$ for some input y with $f(y) = 1$; $C_{f,\epsilon}(x) = \{0\}$, if $d_{\mathcal{X}}(x, y) > \epsilon$ for all inputs y with $f(y) = 1$; and $C_{f,\epsilon}(x) = \{0, 1\}$ otherwise.

3.2 Sampling computations

In this section we formally define sampling computations. We start with the definition of the model and associated complexity measures. We then describe randomized decision trees (and their variant, oblivious decision trees), which is the non-uniform model through which we prove lower bounds for sampling computations.

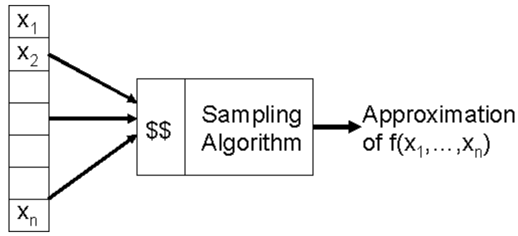

Figure 3.1: A sampling algorithm

3.2.1 The model

Loosely speaking, a sampling algorithm is a randomized algorithm that queries its input at random locations, and based on the outcomes of the queries decides the output (see Figure 3.1). This output is typically an approximation of the function's value on the input. The algorithm is allowed to err (i.e., to output a value which is not a good approximation) with some small fixed probability. The main point is that a sampling algorithm is not required to read the whole input, in order to determine the output. Ideally, the number of queries performed is independent of the input size, and depends only on the approximation accuracy and on the error probability. Formally,

Definition 3.2 (Sampling algorithm) *A* sampling algorithm *is a randomized oracle Turing machine A^x with the following features:*

1. *It gets an accuracy parameter $\epsilon \geq 0$, a confidence parameter $0 < \delta < 1$, and an input length n on a random access input tape.*[1]

2. *It gets an oracle access to an input $x \in \mathcal{X}^n$. To execute a query, the algorithm writes $i \in [n]$ on the query tape, and gets back x_i on the answer tape at the next step.*

3. *It gets a random string r on a random tape. r is drawn from the uniform distribution ρ on a domain $\mathcal{R}$.*

Note that fixing $\epsilon, \delta, n, x, r$ fully determines the execution of A. We thus denote by $A^x(\epsilon, \delta, n, r)$ the output of A when running on the input parameters ϵ, δ, n, the random string r, and having oracle access to x.

[1]To avoid precision issues, we assume from now on that ϵ and δ are rational and their representation requires only poly-log(n) bits.

A sampling algorithm may be in general *adaptive*, meaning that it may use outcomes of previous queries to decide which location of x to query next. An *oblivious* sampling algorithm is one which is non-adaptive; that is, it determines all the locations it is going to query at the beginning of the execution, before it actually makes any oracle calls. Oblivious sampling algorithms will be important to us in developing the sampling lower bound techniques.

Sampling algorithms are required to produce randomized approximations:

Definition 3.3 (Sampling approximation of a function) *A sampling algorithm A is said to* approximate a function $f : \mathcal{X}^* \to \mathcal{Z}$, *if for every parameters ϵ, δ, n and for every input $x \in \mathcal{X}^*$,*

$$\Pr_{R \sim \rho}\left(A^x(\epsilon, \delta, n, R) \in C_{f,\epsilon}(x)\right) \;>\; 1 - \delta.$$

The main measure of cost for sampling algorithms is the *query cost*—the number of queries (i.e., oracle calls) the algorithm performs. Note that the algorithm may perform a varying number of oracle calls on any input, depending on its random string r. We therefore discuss two measures of query cost: the *worst-case query cost* and the *expected query cost*:

Definition 3.4 (Query cost)

- *The* query cost of A on $(\epsilon, \delta, n, x, r)$, *denoted $Q(A^x(\epsilon, \delta, n, r))$, is the number of queries A performs when it gets as input (ϵ, δ, n), gets an oracle access to x, and gets a random string r.*

- *The* worst-case query cost of A on (ϵ, δ, n, x), *denoted $Q^W(A^x(\epsilon, \delta, n))$, is the maximum of $Q(A^x(\epsilon, \delta, n, r))$ over all $r \in \mathcal{R}$.*

- *The* expected query cost of A on (ϵ, δ, n, x), *denoted $Q^E(A^x(\epsilon, \delta, n))$, is the expectation of $Q(A^x(\epsilon, \delta, n, R))$, where $R \sim \rho$.*

- *The* worst-case query cost of A, *denoted $Q^W(A)$, is at most $q(\epsilon, \delta, n)$, if for all ϵ, δ, n, x, $Q^W(A^x(\epsilon, \delta, n))$ is at most $q(\epsilon, \delta, n)$.*

- *The* expected query cost of A, *denoted $Q^E(A)$, is at most $q(\epsilon, \delta, n)$, if for all ϵ, δ, n, x, $Q^E(A^x(\epsilon, \delta, n))$ is at most $q(\epsilon, \delta, n)$.*

Definition 3.5 (Query complexity)

- *The* worst-case query complexity of a function $f : \mathcal{X}^* \to \mathcal{Z}$*, denoted* $Q^W(f)$*, is at most* $q(\epsilon, \delta, n)$*, if there exists a sampling algorithm* A *that approximates* f*, for which* $Q^W(A)$ *is at most* $q(\epsilon, \delta, n)$*.*

- *The* expected query complexity of a function $f : \mathcal{X}^* \to \mathcal{Z}$*, denoted* $Q^E(f)$*, is at most* $q(\epsilon, \delta, n)$*, if there exists a sampling algorithm* A *that approximates* f*, for which* $Q^E(A)$ *is at most* $q(\epsilon, \delta, n)$*.*

Deterministic sampling algorithms A *deterministic sampling algorithm* is the same as a randomized sampling algorithm, except for two differences: (1) it does not get the confidence δ as an input parameter, and (2) it does not have access to a random string. The execution of such an algorithm is fully determined by ϵ, n, and x. The definitions of approximation of a function, query cost, and query complexity change in the obvious way when dealing with such algorithms:

- A deterministic algorithm is required to output an ϵ-approximation of $f(x)$ always.

- The deterministic query complexity of a function f is denoted $Q^D(f)$.

3.2.2 Randomized decision trees

Randomized decision trees are the concrete model with which we prove lower bounds for sampling computations. The randomized decision trees we discuss generalize Boolean decision trees, which were heavily studied in the theoretical computer science literature. (See a survey by Buhrman and de Wolf [27] for the state of the art in Boolean decision tree complexity). They can also simulate, with the same efficiency, various other models of decision trees, including comparison and algebraic decision trees.

Since randomized decision trees are a non-uniform model, it is enough to define how they compute a function over inputs of a fixed length n. The error parameter ϵ and confidence δ are also implicitly hard-coded into the decision tree by the virtue that it is designed to produce an ϵ-approximation with confidence δ, for this specific choice of ϵ and δ.

Informally, a randomized decision tree is a rooted labeled tree (not necessarily binary), whose internal nodes are associated with either queries of input variables or with random coin tosses and whose leaves are associated with outputs. Given an input x, we use the queries and the random coin tosses to determine a path from the root to one of the leaves. This leaf specifies the output on x. Formally,

Definition 3.6 (Randomized decision tree) *A* randomized decision tree *T over input domain $\mathcal{X}^n$ and output range $\mathcal{Z}$ is a rooted labeled tree. For every node v we denote by* $\deg(v)$ *the number of children it has and by $Q(v)$ the number of query nodes on the path from the root to v (including v itself). An internal node v (not a leaf) is labeled in one of the following ways:*

- (i_v, N_v)*, where $i_v \in [n]$ is an input variable and $N_v : \mathcal{X}^{Q(v)} \to [\deg(v)]$ is a child selection function. In this case v is called a* query node*.*
- *The special character* \$*. In this case v is called a* random coin node*.*

If v is a leaf, it is called an output node*, and it is labeled by an output function $O_v : \mathcal{X}^{Q(v)} \to \mathcal{Z}$.*

The computation of T on an input $x \in \mathcal{X}^n$ is done by inductively constructing a computation path *from the root to a leaf. Upon reaching a node v, the next node on the computation path is chosen as follows:*

- *If v is a query node, the next node is v's child whose index is $N_v(x_{i_1}, \ldots, x_{i_{Q(v)}})$, where $i_1, \ldots, i_{Q(v)}$ are the queries made along the path leading to v.*
- *If v is a random coin node, the next node is chosen from the children of v uniformly at random.*
- *If v is an output node, the computation path construction ends, and the output of the tree is $O_v(x_{i_1}, \ldots, x_{i_{Q(v)}})$, where $i_1, \ldots, i_{Q(v)}$ are the queries made along the path leading to v.*

ρ will denote the distribution of choices of the random coin nodes of the tree (all over tree, not only on the computation path). We denote by $\mathcal{R}$ the domain of ρ. Note that by fixing $r \in \mathcal{R}$ and an input $x \in \mathcal{X}^n$, the computation path is fully determined. We denote by $T(x, r)$ the output produced by this computation path. Any input x may be associated with several possible computation paths, depending on the choice of r. Thus, ρ induces a distribution over the computation paths corresponding to x and over the outputs the tree produces on x.

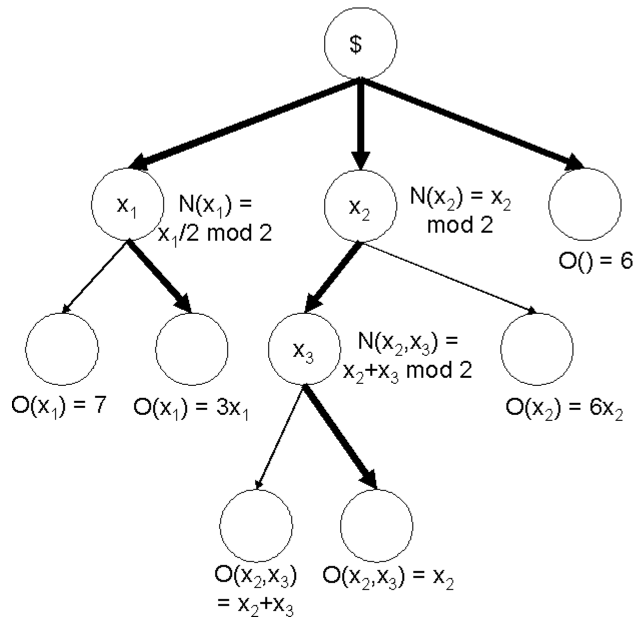

Figure 3.2: A randomized decision tree

Figure 3.2 shows an example of a randomized decision tree and all the computation paths from the root to leaves of the tree corresponding to the input $x = (2, 4, 3)$. The tree outputs on x the value 6 with probability 2/3 and the value 4 with probability 1/3.

Randomized decision trees are supposed to produce an (ϵ, δ)-approximation for a specific choice of ϵ and δ:

Definition 3.7 ((ϵ, δ)-approximation) *A randomized decision tree T is said to (ϵ, δ)-*approximate a function *$f : \mathcal{X}^n \to \mathcal{Z}$, if for every input $x \in \mathcal{X}^n$,*

$$\Pr_{R \sim \rho}\left(T(x, R) \in C_{f,\epsilon}(x)\right) \;>\; 1 - \delta.$$

The query cost of a decision tree is defined as follows:

Definition 3.8 (Query cost)

- *The* query cost of T on input $x \in \mathcal{X}^n$ and random choices $r \in \mathcal{R}$, *denoted, $Q(T(x, r))$, is the number of query nodes on the computation path in T associated with x and r.*

- *The* worst-case query cost of T on x, *denoted $Q^W(T(x))$, is $\max_{r \in \mathcal{R}} Q(T(x, r))$.*

- *The* expected query cost of T on x, *denoted $Q^E(T(x))$, is $\mathrm{E}_{R \sim \rho}[Q(T(x, R))]$.*

- *The* worst-case query cost *of T, denoted $Q^W(T)$, is* $\max_{x \in \mathcal{X}^n} Q^W(T(x))$.
- *The* expected query cost *of T, denoted $Q^E(T)$, is* $\max_{x \in \mathcal{X}^n} Q^E(T(x))$.

The worst-case query cost of the decision tree depicted in Figure 3.2 is 2. Its expected query cost is only 1.

Definition 3.9 (**(ϵ, δ)-query complexity**)

- *The* (ϵ, δ)-worst-case query complexity of a function *$f : \mathcal{X}^n \to \mathcal{Z}$, denoted $Q^W_{\epsilon,\delta}(f)$, is $\min_T Q^W(T)$, where the minimum is over all randomized decision trees T that (ϵ, δ)-approximate f.*
- *The* (ϵ, δ)-expected query complexity of a function *$f : \mathcal{X}^n \to \mathcal{Z}$, denoted $Q^E_{\epsilon,\delta}(f)$, is $\min_T Q^E(T)$, where the minimum is over all randomized decision trees T that (ϵ, δ)-approximate f.*

Simulation of sampling computations It is straightforward to verify that randomized decision trees indeed simulate sampling computations with the same query cost:

Proposition 3.10 *Let A be a sampling algorithm that approximates a function $f : \mathcal{X}^* \to \mathcal{Z}$ and has worst-case query cost $q(\epsilon, \delta, n)$. Then, there exists a family of randomized decision trees $\{T_{\epsilon,\delta,n}\}_{\epsilon,\delta,n}$, such that for every ϵ, δ, n, $T_{\epsilon,\delta,n}$ (ϵ, δ)-approximates $f_n : \mathcal{X}^n \to \mathcal{Z}$ (the restriction of f to inputs of length n) and has worst-case query cost of at most $q(\epsilon, \delta, n)$.*

An identical proposition holds for the expected query cost.

Thus, in order to prove a lower bound of $\Omega(q(\epsilon, \delta, n))$ on the worst-case query complexity of f, it is sufficient to show that for some ϵ and δ and for infinitely many choices of n, the (ϵ, δ)-query complexity of f_n is at least $\Omega(q(\epsilon, \delta, n))$.

Deterministic decision trees A deterministic decision tree is one that has no random coin nodes. In such trees each input x is associated with a single computation path from the root to a leaf. We denote the ϵ-deterministic query complexity of a function f by $Q^D_\epsilon(f)$.

Distributional view of randomized decision trees Every fixing of the random choices r of a randomized decision tree T induces a deterministic decision tree T_r. Thus, a randomized decision tree T can be viewed as a distribution μ over deterministic decision trees $\{T_r\}_{r\in\mathcal{R}}$. Note that $\mu = g(\rho)$, where $g(r) = T_r$. Thus, with this view, $Q^W(T) = \max_{r\in\mathcal{R}} Q^D(T_r)$ and $Q^E(T) = \mathrm{E}_{R\sim\rho}\left[Q^D(T_R)\right]$. Also, if T (ϵ,δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$, then $\forall x \in \mathcal{X}^n$, $\Pr_{R\sim\rho}\left(T_R(x) \in C_{f,\epsilon}(x)\right) > 1-\delta$. This characterization turns out to be very useful in the analysis of randomized decision trees.

3.2.3 Oblivious decision trees

In this section we introduce a restricted class of decision trees, called *oblivious decision trees*, in which the queries are chosen in advance according to a fixed query distribution, independently of the input and of the random choices of the tree.

In order to formally define oblivious decision trees, we first introduce the following transformation on decision trees:

Definition 3.11 (I-relabeling) *Let T be a decision tree of worst-case query cost q, and let $I = (i_1,\ldots,i_q)$ be a sequence of q input variables. The I-*relabeling *of T, denoted T^I, is a decision tree obtained from T by relabeling all its query nodes of depth j (that is, $Q(v) = j$), $j = 1,\ldots,q$, by i_j.*

We can now define oblivious decision trees:

Definition 3.12 (Oblivious decision tree) *An* oblivious decision tree *is a pair (T,Δ_q) where T is a decision tree of worst-case query cost q and Δ_q is a distribution on $[n]^q$, called the* query distribution*. The computation of an oblivious decision tree is done in two steps: (1) it chooses q variables $I = (i_1,\ldots,i_q)$ according to the distribution Δ_q; (2) it runs T^I on the input x.*

We call an oblivious decision tree with a query distribution Δ_q a Δ_q-decision tree. Two examples of query distributions we will use are the following:

1. U_q – *Uniform queries with replacement*: picks q variables from $[n]$ independently and uniformly at random.

2. W_q – *Uniform queries without replacement*: picks a uniform subset of $[n]$ of size q.

Oblivious decision trees are indeed a sub-class of standard randomized decision trees, as long as the query distribution Δ_q consists only of rational probabilities, because the choice of the query variables according to Δ_q can be simulated by random coin nodes.

The definitions of (ϵ, δ) approximation and expected/worst-case query cost of oblivious decision trees are the same as for randomized decision trees with the difference that now the probability is over both the random coins of the tree and the query distribution.

Note that, as opposed to standard decision trees, a query node v of an oblivious decision tree is not associated with a single query variable i_v. Several different query variables may be "plugged-in" into v, depending on the choices made by Δ_q. In particular, the identity of this query variable is not apparent from the query node. We therefore consider two variants of oblivious decision trees:

- Variable-aware: the child selection function of each query node v in T and the output function of each output node u in T are provided not only with the outcomes of the queries made along the path leading to v or u, but also with the identities of the variables queried. That is, $N_v : ([n] \times \mathcal{X})^{Q(v)} \to [\deg(v)]$ and $O_u : ([n] \times \mathcal{X})^{Q(u)} \to \mathcal{Z}$.

- Variable-oblivious: the child selection functions of query nodes and the output functions of output nodes are not provided with the identities of the variables queried. That is, $N_v : \mathcal{X}^{Q(v)} \to [\deg(v)]$ and $O_u : \mathcal{X}^{Q(u)} \to \mathcal{Z}$.

A variable-aware Δ_q-decision tree T for a function $f : \mathcal{X}^n \to \mathcal{Z}$ can be viewed as a mapping $T(I, x|_I, r)$ that maps the chosen query sequence $I = (i_1, \ldots, i_q)$, the outcomes of the queries $x|_I$, and the random choices of the tree r into an output value in $\mathcal{Z}$. A variable-oblivious Δ_q-decision tree is a mapping $T(x|_I, r)$ that maps only the outcomes of the queries and the random choices into a value in $\mathcal{Z}$. The inputs of these mappings distribute as follows: I distributes according to the query distribution Δ_q, $x|_I$ distributes according to the outcome distribution $\Delta_{q,x}$ (the distribution on $\mathcal{X}^q$ induced by Δ_q and x), and r distributes according to ρ. We call the joint distribution of I and $x|_I$ the joint query-outcome distribution and denote it by $\tilde{\Delta}_{q,x}$. Thus, variable-aware decision trees have access to a sample from the joint query-outcome distribution, while variable-oblivious decision trees have access only to a sample from the outcome distribution.

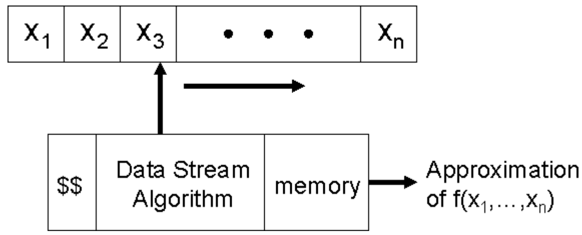

Figure 3.3: A data stream algorithm

3.3 Data stream computations

In this section we formally define data stream computations and the non-uniform model of randomized oblivious read-once branching programs, with which we prove space lower bounds for data stream computations. We then present a general reduction from one-way communication complexity to oblivious read-once branching programs. This reduction will allow us to obtain space lower bounds for the data stream model through communication complexity lower bounds for the one-way communication model.

3.3.1 The model

A data stream algorithm (see Figure 3.3) receives as input a stream of data items. The algorithm has only one-way, forward, access to the items in the stream; in particular, it cannot go backwards on the stream or have random access to the data. Moreover, the input data stream is usually much larger than the storage capabilities of the algorithm. Thus, the algorithm cannot store the stream, and then use standard random access to it.

The two main measures of complexity for data stream computations are the *space*—the amount of memory used by the algorithm and the *processing time per data item*—the amount of time spent processing each data item in the stream. Ideally, we would like the space and the processing time per item to be logarithmic or poly-logarithmic in the length of the stream and in the size of the domain from which the data items are drawn. Since exact computations and deterministic computations are usually hopelessly inefficient in the data stream model, most data stream algorithms are randomized and approximate.

Another common requirement from data stream algorithms is that they work irrespective of the order of the input data items. That is, if the input $x = x_1, \ldots, x_n$ arrives as $(\pi(1), x_{\pi(1)}), (\pi(2), x_{\pi(2)}), \ldots, (\pi(n), x_{\pi(n)})$, where π is some permutation in S_n, then the algorithm is still required to output a good approximation of $f(x)$ with high probability.

Formally a data stream algorithm is defined as follows:

Definition 3.13 (Data stream algorithm) *A data stream algorithm is a randomized Turing machine A with the following features:*

1. *It gets its input $x \in \mathcal{X}^n$ on a one-way read-only input tape. The input is represented by a sequence of pairs $(\pi(1), x_{\pi(1)}), \ldots, (\pi(n), x_{\pi(n)})$, where π is an arbitrary permutation.*
2. *It gets an approximation parameter $\epsilon \geq 0$ and a confidence parameter $0 < \delta < 1$ on a separate random-access input tape.*
3. *It gets a random string r on a random tape. r is drawn from a uniform distribution ρ over a domain $\mathcal{R}$.*
4. *It has a random-access work-tape.*

A variant of the model allows the algorithm to get also the input length n as a third parameter on the random-access input tape.

Note that fixing $\epsilon, \delta, x, \pi, r$ fully determines the execution of A. We thus denote by $A(\pi(x), \epsilon, \delta, r)$ the output of A when running on the input x given according to π, the parameters ϵ and δ, and the random string r.

Data stream algorithms are required to produce randomized approximations:

Definition 3.14 (Data stream approximation of a function) *A data stream algorithm A is said to* approximate a function *$f : \mathcal{X}^* \to \mathcal{Z}$, if for every parameters ϵ, δ, for every input $x \in \mathcal{X}^*$, and for every permutation $\pi \in S_{|x|}$,*

$$\Pr_{R \sim \rho} \left(A(\pi(x), \epsilon, \delta, R) \in C_{f,\epsilon}(x)\right) \;>\; 1 - \delta.$$

In this thesis we will concentrate only on the space complexity of data stream computations:

Definition 3.15 (Space of a data stream algorithm) *The* space *of A, denoted $SP(A)$, is at most $s(\epsilon, \delta, n, |\mathcal{X}|)$, if for all x of length n, for all permutations $\pi \in S_n$, and for all ϵ, δ, the logarithm of the total number of configurations A can have when running on $(\pi(x), \epsilon, \delta)$, is at most $s(\epsilon, \delta, n, |\mathcal{X}|)$. A configuration consists of the contents of the work-tape, the locations of the tape heads, and the state of the machine.*

Definition 3.16 (Data stream space complexity) *The* data stream space complexity of a function $f : \mathcal{X}^* \to \mathcal{Z}$, *denoted* $DS(f)$, *is at most* $s(\epsilon, \delta, n, |\mathcal{X}|)$, *if there exists a data stream algorithm* A *that approximates* f, *and whose space is at most* $s(\epsilon, \delta, n, |\mathcal{X}|)$.

A generalization of the data stream model [64] allows the algorithm to have several one-way, forward, passes over the data. This may allow computing efficiently functions that are intractable to compute just in a single pass. The number of passes becomes then a third measure of complexity for data stream algorithms. The definitions stated above easily extend to deal with this scenario as well. We will denote the k-pass data stream space complexity of a function f by $DS^k(f)$.

3.3.2 Randomized oblivious read-once branching programs

Randomized oblivious read-once branching programs are the concrete model with which we prove lower bounds for data stream computations. This model generalizes the well studied Boolean randomized oblivious read-once branching programs, also known as *Ordered Binary Decision Diagrams (OBDDs)* [26, 126]. Branching programs are a standard model for investigating the space complexity of computations. See [108] for a survey of known results in this model.

The definition of a randomized branching program is very similar to the definition of a randomized decision tree with two important distinctions: (1) a branching program can be a general directed acyclic graph, and not necessarily a tree, and (2) the child selection function of a node v in a branching program can use only the outcome of the query made at v (and not outcomes of previous queries). Formally,

Definition 3.17 (Randomized branching program) *A* randomized branching program *B with input domain $\mathcal{X}^n$ and output range $\mathcal{Z}$ is a labeled, leveled, directed acyclic graph. The nodes of the graph are partitioned into disjoint "levels". Nodes in the i-th level have edges only to nodes in the $i+1$-st level. The first level has only one node, called the* start state. *Nodes in the last level have no outgoing edges, and they are called* output states.

For each node v in the graph, we denote by $\deg(v)$ *the number of children it has (i.e., its out-degree). The levels of the graph, except for the last one, are classified into two types: (1)* query levels *and (2)* random coin levels. *Each node v in a query level is labeled by an input variable $i_v \in [n]$ and by a child selection function $N_v : \mathcal{X} \to [\deg(v)]$. Each*

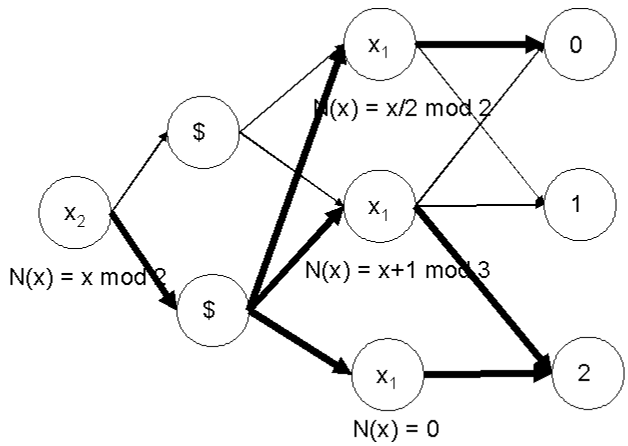

Figure 3.4: A randomized oblivious read-once branching program

node in a random coin level is labeled by the special character \$. *Output nodes are labeled by values from the range* $\mathcal{Z}$.

The computation of the branching program is done by inductively constructing a computation path *from the start state to an output state. Upon reaching a node* v, *the construction of the computation path proceeds as follows:*

- *If* v *is a query node, the next node on the computation path is the child of* v *whose index is* $N_v(x_{i_v})$, *where* x_{i_v} *is the outcome of the query made at* v.
- *If* v *is a random coin node, the next node on the computation path is chosen uniformly at random among the children of* v.
- *If* v *is an output node, the computation path construction halts, and the output is the value labeling* v.

We denote by ρ the distribution of the choices made at the random coin nodes of the branching program, and its domain by $\mathcal{R}$. Note that by fixing $r \in \mathcal{R}$ and an input $x \in \mathcal{X}^n$, the computation path is fully determined. We thus denote by $B(x, r)$ the output of B on x and r. In general, an input x is associated with a distribution over computation paths induced by ρ. The output of B on x is thus a random variable $B(x, R)$, where $R \sim \rho$.

Figure 3.4 shows an example of a branching program and all the computation paths in it corresponding to the input $x = (2, 3)$. Note that the branching program outputs the value 0 on x with probability 1/3 and the value 2 with probability 2/3.

Randomized branching programs are supposed to (ϵ, δ)-approximate the function for a specific choice of ϵ and δ:

Definition 3.18 ((ϵ, δ)-approximation) *A randomized branching program B is said to* (ϵ, δ)-approximate a function *$f : \mathcal{X}^n \to \mathcal{Z}$, if for every input $x \in \mathcal{X}^n$,*

$$\Pr_{R \sim \rho} \left(B(x, R) \in C_{f,\epsilon}(x) \right) \;>\; 1 - \delta.$$

The following is the main measure of cost for branching programs:

Definition 3.19 (Space of a branching program) *The* width *of a branching program B is the size of its largest level. The* space *of B, denoted $SP(B)$, is the logarithm of its width.*

In the example of Figure 3.4 the width of the branching program is 3 and its space is 2.

So far, we discussed general branching programs, which have random access to the input and can query each input item as many times as they wish. In order to define a model that captures the data stream scenario, we need to further restrict the way branching programs query their input.

Definition 3.20 (Oblivious branching program) *A branching program B is called* oblivious*, if all nodes in each query level ℓ of B are labeled by the same input variable i_ℓ.*

In an oblivious branching program the sequence of queries is fixed, and is independent of the outcomes of queries or random coin tosses.

Definition 3.21 (Oblivious read-once branching program) *An oblivious branching program is called* read-once*, if each of the input variables is queried at most once.*

One can associate with an oblivious read-once branching program a permutation $\pi \in S_n$, where the nodes in the i-th query level of the branching program query $x_{\pi(i)}$. We thus call such a branching program a π-branching program.

The branching program depicted in Figure 3.4 is oblivious and read-once. The permutation associated with it is $(2, 1)$.

Definition 3.22 ((ϵ, δ, π)-branching program space complexity) *The* (ϵ, δ, π) branching program space complexity of a function *$f : \mathcal{X}^n \to \mathcal{Z}$, denoted $BSP_{\epsilon,\delta,\pi}(f)$, is the minimum $SP(B)$ over all π-branching programs B that (ϵ, δ)-approximate f.*

Simulation of data stream computations It is straightforward to verify that randomized oblivious read-once branching programs can simulate data stream computations with the same space:

Proposition 3.23 *Let A be a data stream algorithm that approximates a function $f : \mathcal{X}^* \to \mathcal{Z}$ and uses $s(\epsilon, \delta, n, |\mathcal{X}|)$ space. Then, there exists a family of randomized oblivious read-once branching programs $\{B_{\epsilon,\delta,n,\pi}\}_{\epsilon,\delta,n,\pi}$, such that for every ϵ, δ, n, π, $B_{\epsilon,\delta,n,\pi}$ is a π-branching program that (ϵ, δ)-approximates $f_n : \mathcal{X}^n \to \mathcal{Z}$ (the restriction of f to inputs of length n) and uses at most $s(\epsilon, \delta, n, |\mathcal{X}|)$ space.*

Thus, in order to prove a lower bound of $\Omega(s(\epsilon, \delta, n, |\mathcal{X}|))$ on the data stream space complexity of f, it is sufficient to show that for some ϵ, δ and for infinitely many n, there is a permutation $\pi \in S_n$, such that the (ϵ, δ, π)-branching program space complexity of f_n is at least $\Omega(s(\epsilon, \delta, n, |\mathcal{X}|))$.

Read-k-times branching programs An oblivious read-k-times branching program is an oblivious branching program in which the n input variables are read k times in the same order. That is, the first n query levels query the input according to some permutation π. The next n query levels query the input according to the same permutation π. And so on, until the last n query levels. We define the (ϵ, δ, π) k-pass branching program space complexity of a function f to be $BSP^k_{\epsilon,\delta,\pi}(f) \stackrel{\text{def}}{=} \min_B SP(B)$, where the minimum is over all oblivious read-k-times π-branching programs that (ϵ, δ)-approximate f. A similar proposition to Proposition 3.23 shows that any k-pass data stream algorithm that uses space s can be simulated by a family of oblivious read-k-times branching programs that also use space s.

3.3.3 Reduction from one-way communication complexity

We next present the general framework of reductions from the model of one-way communication complexity (see Section 2.2.4) to oblivious read-once branching programs. (The framework is probably attributed to folklore; several versions of it can be found in [81, 64, 4]). The reduction uses a given randomized oblivious read-once branching program B for a function f to construct a randomized one-way communication complexity protocol Π for a function g, where the amount of communication used by Π is related to the space used by B. Using such a reduction, we can derive a space lower bound for data stream computation of f from a communication lower bound for g.

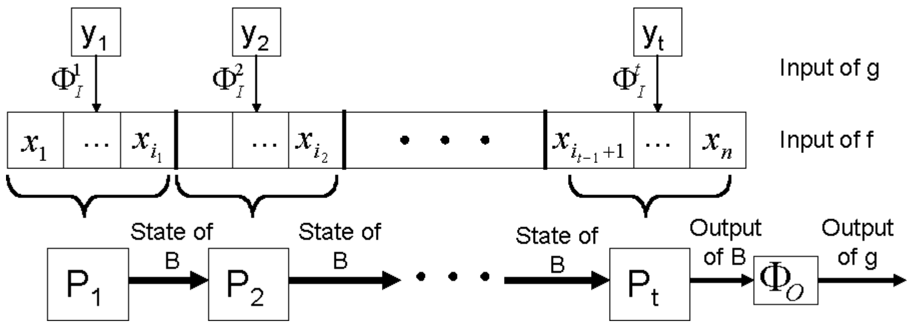

Figure 3.5: Reduction from one-way communication to oblivious read-once branching programs

Let $f : \mathcal{X}^n \to \mathcal{Z}$ be a function for which we have a randomized oblivious read-once branching program B that uses space S, and whose associated permutation is π. Let $g : \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_t \to \mathcal{W}$ be a function for which we wish to construct a randomized one-way communication complexity protocol using B. Figure 3.5 shows how the protocol Π for g resulting from the reduction works: the j-th player translates her input $y_j \in \mathcal{Y}_j$ into a sequence of data items $x_{i_j}, \ldots, x_{i_{j+1}-1} \in \mathcal{X}$. The concatenation of these data items forms a permutation $\pi(x)$ of an input $x \in \mathcal{X}^n$ for f. The players simulate the branching program B on this input x. The simulation is carried out in turns: the first player starts the simulation on her part of the input; when she is done, she sends the current state of B to the second player, who resumes the simulation on her part of the permuted input, and so on. The last player finishes the simulation, and gets the output, which with high probability is $f(x)$. She then translates this output back into $g(y)$.

Formally, a reduction is defined as follows:

Definition 3.24 ($(\pi, \epsilon, \epsilon')$-reduction) *A* $(\pi, \epsilon, \epsilon')$-reduction *from g to f is a pair $\phi = \langle \vec{\phi}_I, \phi_O \rangle$, where:*

- *$\vec{\phi}_I$ is a t-dimensional vector $(\phi_I^1, \ldots, \phi_I^t)$. The j-th entry in this vector is a function $\phi_I^j : \mathcal{Y}_j \to \mathcal{X}^{s_j}$, which determines how the j-th player translates her input into a sequence of s_j data items for f. $s_1, \ldots, s_t$ satisfy: $s_1 + \cdots + s_t = n$.*

- *$\phi_O : \mathcal{Z} \to \mathcal{W}$ determines how to translate an output of f into an output of g.*

- *For every input $y = (y_1, \ldots, y_t) \in \mathcal{Y}_1 \times \cdots \mathcal{Y}_t$,*

$$z \in C_{f,\epsilon}(\pi^{-1}\left(\phi_I^1(y_1) \circ \cdots \circ \phi_I^t(y_t)\right)) \implies \phi_O(z) \in C_{g,\epsilon'}(y).$$

Using such a reduction ϕ, the protocol Π for g works as follows: each player, P_j, computes $\phi_I^j(y_j)$, and obtains the part of the permuted input $\pi(x)$ she is supposed to run B on. The first player, P_1, starts the simulation of B at the start state. She simulates B for the first s_1 query levels, using the data items she generated as outcomes for these queries (note that there may be random coin levels interleaved with the query levels; upon reaching such a random coin state, P_1 simply chooses one of its children uniformly at random). After processing the s_1-th query, P_1 sends the currently visited state to P_2. P_2 now continues the simulation for the next s_2 query levels, using the data items she generated as outcomes for the queries. The simulation continues in this fashion, until one of the players, P_j gets to an output state. If $j < t$ (that is, P_j is not the last player), P_j sends this output state to the next player. The last player, P_t, upon receiving the output state (or getting to it herself in the simulation), extracts the value z output at this state, and announces $\phi_O(z)$ as the output of the protocol.

It is now straightforward to verify the following:

Proposition 3.25 *If B is a randomized oblivious read-once branching program that (ϵ, δ)-approximates f, then Π is a one-way protocol that (ϵ', δ)-approximates g. Furthermore, if B uses S space, then Π uses $S(t-1) + \log|\mathcal{W}|$ communication.*

A similar reduction works from *k-round one-way communication complexity* to oblivious read-k-times branching programs. The k-round one-way model is a generalization of one-way communication complexity, in which the communication is still one-way (i.e., P_i sends messages only to P_{i+1}), but there are k rounds of communication (that is, each player sends exactly k messages throughout the protocol). Using such a reduction, we can show:

Proposition 3.26 *If B is a randomized oblivious read-k-times branching program (ϵ, δ)-approximating f, then Π is a k-round one-way protocol (ϵ', δ)-approximating g. Furthermore, if B uses S space, then Π uses $S(tk-1) + \log|\mathcal{W}|$ communication.*

3.4 Sketch computations

In this section we define sketch computations and show that their non-uniform counterpart is public-coin simultaneous communication complexity.

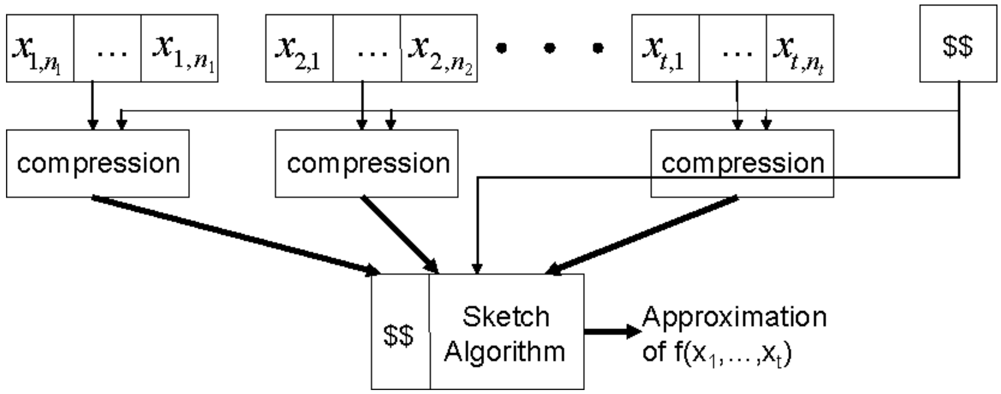

Figure 3.6: A sketch algorithm

3.4.1 The model

In the sketch model of computation (see Figure 3.6) the input x is distributed over t storage sites. The sketch algorithm runs on a distant site, and does not have direct access to any of the input parts. Each storage site runs a compression procedure that compresses the data stored on the site into a small "sketch", and sends this sketch to the sketch algorithm. We note that the compression used is in general lossy and may be randomized. The algorithm has to compute the function's value on x based just on these sketches. The compression procedures are not allowed to communicate with each other, but they are allowed to share randomness. That is, they can use the same random bits to compute the sketches; such random bits may be used, for example, to pick some random hash function, which would then be applied to the data to be compressed.

The main measure of complexity for sketch algorithms is the sum of the lengths of the sketches. Ideally, one would like sketches of length poly-logarithmic in the input length and in the size of the domain from which the data items are drawn.

Definition 3.27 (Sketch algorithm) *A* sketch algorithm *A is a collection of $t+1$ randomized Turing machines $T_c, T_1, \ldots, T_t$ with the following features:*

1. *Each of $T_1, \ldots, T_t$ has its own private random-access input tape, on which it receives some input $x_i \in \mathcal{X}_i^*$.*

2. *Each of $T_1, \ldots, T_t$ has its own output tape, on which it writes its output $y_i \in \mathcal{Y}_i^*$.*

3. *The input of T_c is the output tapes of $T_1, \ldots, T_t$. It has random-access to these tapes. The output of A is given by the output of T_c.*

4. *$T_c, T_1, \ldots, T_t$ have a shared public random tape. In addition, each of them has its own private random tape.*

5. *$T_c, T_1, \ldots, T_t$ have an additional shared public input tape, on which they receive the approximation parameter ϵ and the confidence parameter δ.*

We denote the distribution of the private random strings of $T_1, \ldots, T_t$ by $\vec{\rho} = (\rho_1, \ldots, \rho_t)$ and the distribution of the public random string by ρ_p. $\rho_1, \ldots, \rho_t, \rho_p$ are uniform on domains $\mathcal{R}_1, \ldots, \mathcal{R}_t, \mathcal{R}_p$, respectively.

Note that fixing ϵ, δ, $\vec{x} = (x_1, \ldots, x_t)$, $\vec{r} = (r_1, \ldots, r_t)$, and r_p fully determines the execution of A. We thus denote by $A(\vec{x}, \epsilon, \delta, \vec{r}, r_p)$ the output of A when running on these parameters.

Sketch algorithms are required to produce randomized approximations:

Definition 3.28 (Sketch approximation of a function) *A sketch algorithm A is said to* approximate a function *$f : \mathcal{X}_1^* \times \cdots \times \mathcal{X}_t^* \to \mathcal{Z}$, if for every parameters ϵ, δ, and for every input $\vec{x} \in \mathcal{X}_1^* \times \cdots \times \mathcal{X}_t^*$,*

$$\Pr_{\vec{R} \sim \vec{\rho}, R_p \sim \rho_p} \left(A\left(\vec{x}, \epsilon, \delta, \vec{R}, R_p\right) \in C_{f,\epsilon}(\vec{x}) \right) \;>\; 1 - \delta.$$

The following is the main measure of cost for sketch computations:

Definition 3.29 (Sketch cost)

- *The* sketch cost of *A* on *$(\vec{x}, \epsilon, \delta, \vec{r}, r_p)$, denoted $SC(A(\vec{x}, \epsilon, \delta, \vec{r}, r_p))$, is the sum of the lengths of the outputs of $T_1, \ldots, T_t$ when running on $\vec{x}, \epsilon, \delta, \vec{r}, r_p$.*

- *The* sketch cost of *A* on *$(\vec{x}, \epsilon, \delta)$, denoted $SC(A(\vec{x}, \epsilon, \delta))$, is $\max_{\vec{r}, r_p} SC(A(\vec{x}, \epsilon, \delta, \vec{r}, r_p))$.*

- *The* sketch cost of *A, denoted $SC(A)$, is at most $s(\epsilon, \delta, n, \max_i |\mathcal{X}_i|)$, if for all ϵ, δ, n, and for all $\vec{x}$ whose total length is n, $SC(A(\vec{x}, \epsilon, \delta))$ is at most $s(\epsilon, \delta, n, \max_i |\mathcal{X}_i|)$.*

Definition 3.30 (Sketch complexity) *The* sketch complexity of a function *$f : \mathcal{X}_1^* \times \cdots \times \mathcal{X}_t^* \to \mathcal{Z}$, denoted $SC(f)$, is at most $s(\epsilon, \delta, n, \max_i |\mathcal{X}_i|)$, if there exists a sketch algorithm A that approximates f, and whose sketch cost is at most $s(\epsilon, \delta, n, \max_i |\mathcal{X}_i|)$.*

3.4.2 Public-coin simultaneous communication complexity

Examining the sketch model closely reveals that its non-uniform version is simply the *t-party public-coin simultaneous communication complexity* (see Section 2.2.5). The compression machines $T_1, \ldots, T_t$ correspond to the players in the communication complexity protocol. The central algorithm T_c corresponds to the referee. The players share randomness through the public coins. The messages sent by the players are the "sketches", and the sum of the lengths of the sketches is exactly the communication cost of the protocol. Note that unlike traditional study of communication complexity protocols, which concentrated on exact computations, here the protocol is supposed to produce approximations.

We therefore use simultaneous communication complexity to define non-uniform sketch complexity:

Definition 3.31 ((ϵ, δ)-sketch complexity) *The* (ϵ, δ)-sketch complexity of a function $f : \mathcal{X}_1^{n_1} \times \cdots \times \mathcal{X}_t^{n_t} \to \mathcal{Z}$ *, denoted* $SC_{\epsilon,\delta}(f)$*, is* $\min_\Pi CC(\Pi)$*, where the minimum is over all public-coin randomized t-party simultaneous communication complexity protocols* Π *that* (ϵ, δ)*-approximate* f*.*

It is straightforward to verify that simultaneous protocols indeed simulate sketch algorithms with the same sketch cost:

Proposition 3.32 *Let A be a sketch algorithm that approximates a function $f : \mathcal{X}_1^* \times \cdots \mathcal{X}_t^* \to \mathcal{Z}$ and whose sketch cost is at most $s(\epsilon, \delta, n, \max_i |\mathcal{X}_i|)$. Then, there exists a family of t-party public-coin simultaneous communication complexity protocols $\left\{\Pi_{\epsilon,\delta,\vec{n}}\right\}_{\epsilon,\delta,\vec{n}}$, such that for every $\epsilon, \delta, \vec{n} = (n_1, \ldots, n_t)$, $\Pi_{\epsilon,\delta,\vec{n}}$ is a protocol that (ϵ, δ)-approximates $f_{\vec{n}} : \mathcal{X}_1^{n_1} \times \cdots \times \mathcal{X}_t^{n_t} \to \mathcal{Z}$ (the restriction of f to inputs whose i-th argument is of length n_i) and whose communication cost is at most $s(\epsilon, \delta, n, \max_i |\mathcal{X}_i|)$ (where $n = n_1 + \cdots + n_t$).*

3.5 Relations among the models

In this section we review some elementary relations among the models. Figure 3.7 summarizes these relations. An arrow from a model M to a model N implies that M can be simulated efficiently by N. A dashed arrow means that N simulates M under some restrictions on M.

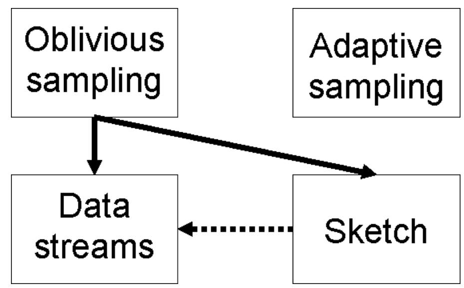

Figure 3.7: Relations among the models

3.5.1 Sampling vs. data streams

We show that:

- Oblivious sampling computations can be simulated efficiently by data stream computations, but adaptive sampling computations cannot.
- Data stream computations cannot be simulated efficiently by sampling computations, whether adaptive or oblivious.

These results are folklore; they can be found, e.g., in [17].

Proposition 3.33 *If there is an oblivious sampling algorithm A that approximates $f : \mathcal{X}^* \to \mathcal{Z}$ and whose worst-case query cost is q, then there is a data stream algorithm B that approximates f and uses $q \cdot O(\log |\mathcal{X}| + \log n)$ space.*

Proof: Since A is oblivious, its execution can be broken into two consecutive phases: (1) choose the query locations $i_1, \ldots, i_q$, and (2) perform the queries and use the outcomes to determine an output. B starts by simulating the first phase of A; it stores the chosen queries $i_1, \ldots, i_q$ in memory. B then streams through the input and stores in memory the data items $x_{i_1}, \ldots, x_{i_q}$. Finally, B executes the second phase of A with the stored data items as the answers to the queries. $\square$

Adaptive sampling, however, cannot be simulated efficiently by data stream computations: consider the index function: $\textsc{Ind}(x, i) = x_i$. By the one-way communication lower bound for $\textsc{Ind}$ (see example in Section 2.2.4), any data stream computation of $\textsc{Ind}$ would require space $\Omega(n)$. An adaptive sampling computation would require, however, only $\log n + 1$ queries: it queries i and then x_i.

An example of a function that has low data stream space complexity and high query complexity is the parity: $\textsc{Parity}(x) = \bigoplus_{i=1}^{n} x_i$. A simple data stream algorithm, which just records the parity of the bits seen so far, requires 1 bit of space. Any sampling computation, however, adaptive or oblivious, would need n queries, since the function is sensitive to all its variables.

3.5.2 Data streams vs. sketch

We show that:

- Sketch computations whose compression procedures have low data stream space complexity can be simulated efficiently by data stream computations.
- Data stream computations cannot be simulated efficiently by sketch computations.

These results are due to Feigenbaum, Kannan, Strauss, and Viswanathan [46].

Proposition 3.34 *If there is a sketch algorithm $A = (T_c, T_1, \ldots, T_t)$ that approximates a function $f : \mathcal{X}_1^* \times \cdots \times \mathcal{X}_t^* \to \mathcal{Z}$ and: (1) its sketch cost is s_1, (2) each of its compression procedures T_i is a data stream computation with space s_2, (3) the central algorithm T_c uses space s_3, then there is a data stream algorithm B that approximates f and uses $s_1 + ts_2 + s_3$ space.*

Proof: B first simulates the t compression procedures $T_1, \ldots, T_t$ simultaneously. Upon receiving an item (j, x_j) in the data stream, B uses j to determine which of the compression procedures should process it. After finishing reading the input stream, B completes the simulation of $T_1, \ldots, T_t$, and stores the sketches they output in memory. B then simulates T_c on the computed sketches. □

An example of a function that has low data stream complexity and high sketch complexity is the following version of $\textsc{Ind}$: $\textsc{Ind}(i, x) = x_i$. A data stream algorithm can compute this function with space $O(\log n)$, while any sketch computation would have sketch cost of $\Omega(n)$.

3.5.3 Sampling vs. sketch

We show that:

- Oblivious sampling computations can be simulated efficiently by sketch computations, but adaptive sampling computations cannot.

- Sketch computations cannot be simulated efficiently by sampling computations (oblivious or adaptive).

Proposition 3.35 *If there is an oblivious sampling algorithm A that approximates a function $f : \mathcal{X}_1^* \times \cdots \times \mathcal{X}_t^* \to \mathcal{Z}$ and whose worst-case query cost is q, then there is a sketch algorithm $B = (T_c, T_1, \ldots, T_t)$ that approximates f and whose sketch cost is $q \cdot \max_i \log |\mathcal{X}_i|$.*

Proof: Recall that since A is oblivious, its execution can be broken into two phases: selection of queries and computation. B simulates A as follows. The compression procedures $T_1, \ldots, T_t$ use the shared random tape to simulate the first phase of A, in which it chooses the queries $i_1, \ldots, i_q$. Note that the answer to each such query can be given by exactly one of the compression procedures. Thus, for each $j = 1, \ldots, t$, the j-th sketch consists of the outcomes of the queries T_j can answer. The central algorithm, T_c, now simulates the second part of A using the outcomes of the queries received from the compression procedures. $\square$

The index function mentioned above shows that adaptive sampling cannot be simulated efficiently by sketch computations. The parity function has low sketch complexity, but high query complexity, and therefore sketch computations cannot be simulated efficiently by sampling computations.

Chapter 4

Sampling Lower Bounds

In this chapter we present a suite of lower bound techniques for sampling computations (see definitions and discussion about the model in Section 3.2). The bounds apply to a wide range of functions, and are stated in terms of simple to compute properties of these functions. Thus, the techniques can be viewed as recipes or templates for obtaining query complexity lower bounds for specific functions. We demonstrate the strength and the ease of use of these techniques by several applications, including lower bounds for estimation of statistical and frequency moments, selection functions, and frequency distributions. We are also able to obtain new, simple, proofs for extractor and disperser lower bounds, which were the main technical contribution of Radhakrishnan and Ta-Shma [104].

The framework All the lower bounds are stated and proven for non-uniform randomized decision trees (see definitions in Section 3.2.2). Thus, we fix the input length n, the approximation error ϵ, and the confidence parameter δ, and prove lower bounds for randomized decision trees that (ϵ, δ)-approximate functions of the form $f : \mathcal{X}^n \to \mathcal{Z}$, where $\mathcal{X}$ and $\mathcal{Z}$ are arbitrary sets. The resulting query complexity bounds are functions of f, n, ϵ, δ, and sometimes also of $|\mathcal{X}|$.

Unless stated otherwise, the lower bounds do not assume anything about the function, about its domain, or about its range. Thus, the bounds apply to any function, even ones with infinite domain or range (e.g., $\mathbb{R}$). This is a significant departure from the classical study of sampling computations in theoretical computer science, which concentrated on functions over Boolean domain and Boolean range[1]. In this sense our framework is very

[1] Note that the usual considerations of precision do not play any role here, because the only measure of

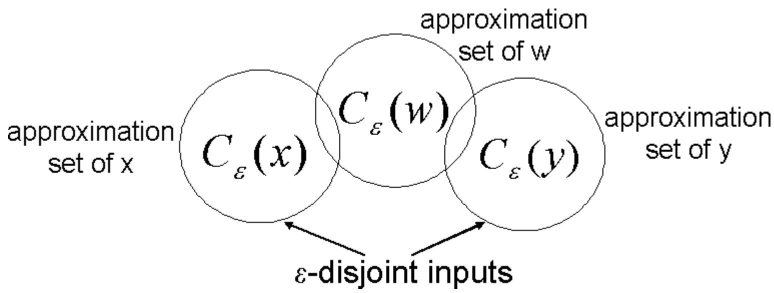

Figure 4.1: ϵ-disjoint inputs

similar to the one studied in statistics. The main difference, however, is that in statistics the sampling computations are usually assumed to be *oblivious*, and to have "i.i.d." (independent and identically distributed) queries, while the sampling computations we consider are more general: they may be adaptive and can perform dependent and not necessarily identically distributed queries.

The core idea The lower bound techniques are based on a variety of tools from combinatorics, statistics, and information theory. Nevertheless, all the bounds have one basic core idea, which we describe next.

Given a function $f : \mathcal{X}^n \to \mathcal{Z}$ with an associated notion of approximation $C_{f,\epsilon}$ (see definition in Section 3.1), we define the following:

Definition 4.1 (ϵ-disjoint inputs) *Two inputs $x, y \in \mathcal{X}^n$ are said to be ϵ-*disjoint *with respect to approximation $C_{f,\epsilon}$, if $C_{f,\epsilon}(x) \cap C_{f,\epsilon}(y) = \emptyset$.*

Figure 4.1 gives an illustration of ϵ-disjoint inputs. In the given example x and y are ϵ-disjoint, but x, w, and y, w are not.

Consider now any sampling algorithm T that (ϵ, δ)-approximates f. By definition, when T runs on x it has to output a value $z \in C_{f,\epsilon}(x)$ with probability at least $1 - \delta$, and when it runs on y it has to output a value $z' \in C_{f,\epsilon}(y)$ with probability at least $1 - \delta$. However, if x, y are ϵ-disjoint, no value $z \in \mathcal{Z}$ belongs both to $C_{f,\epsilon}(x)$ and to $C_{f,\epsilon}(y)$. It follows immediately that the distribution of outputs of T when running on x is very "far" from the distribution of outputs of T when running on y. Thus, T can be used to form a "distinguisher" between x and y for any pair of ϵ-disjoint inputs x, y. If we manage to

complexity we are considering is the number of queries to input arguments (thus, we assume, e.g., that one can read an infinite precision real number in a single query).

find such a pair of inputs x, y, which are not "easily" distinguishable, we obtain a lower bound on the number of queries T has to perform in order to work properly. The exact quantification of the "distance" between the output distributions and of the "difficulty" of distinguishing between x and y is the crux of each of the lower bound techniques.

The lower bound techniques We classify our lower bounds into three major categories: combinatorial bounds, statistical bounds, and information-theoretic bounds.

The combinatorial bound, discussed in Section 4.1, is very general—it applies to any function and works for all possible sampling computations. Moreover, it gives a bound on the *expected query complexity*, and not only on the worst-case query complexity. Its main drawback, however, is that it is weak—the specific bounds it produces tend not be tight. The bound is based on a generalization of the notion of *block sensitivity*, originally introduced by Nisan [94] for Boolean functions.

The statistical bounds, discussed in Section 4.2, apply to all functions, but work only for the restricted class of *oblivious sampling computations* (see definitions in Section 3.2.3). These bounds are still interesting for two reasons. First, oblivious sampling is perhaps the most popular form of sampling, and thus studying the oblivious query complexity of functions is important. Second, we prove that for a wide class of functions—the *symmetric functions*—oblivious sampling is equivalent to general sampling. Thus, the oblivious sampling lower bounds imply general sampling lower bounds for these functions.

The statistical bounds are based on a reduction from the framework of statistical hypothesis testing (see Section 2.4.2) to oblivious sampling computations. We then exploit several known sample complexity lower bounds for hypothesis testing to derive query complexity lower bounds for oblivious sampling computations. We obtain several bounds, each one working best for a different setting of the parameters. In particular, we prove bounds both on the worst-case query complexity and on the expected query complexity. The latter are slightly weaker than the former.

The information-theoretic bound, discussed in Section 4.3, is similar to the statistical bounds in the sense that it works only for oblivious sampling computations. This bound has the advantage of having a better dependence on the domain size $|\mathcal{X}|$, which is crucial in some applications (e.g., frequency distribution estimation). The bound is based on an information-theoretic view of hypothesis testing.

In Section 4.4 we present a variety of examples, which demonstrate how one can easily use the recipes given by our lower bound techniques to obtain query complexity lower bounds for specific functions. Some of these specific bounds are brand new, and others are known, but existing proofs are rather complicated and especially-tailored to the specific functions.

In Section 4.5 we discuss the tightness of our bounds and the relations among them. In particular, we show that the statistical and the information-theoretic bounds are almost optimal, by presenting a matching upper bound.

In Section 4.6 we review some of the previous work in theoretical computer science, in statistical decision theory, and in statistical and computational learning theory that is relevant to our work. We conclude in Section 4.7 with some open problems.

4.1 A combinatorial lower bound

In this section we generalize the notion of block sensitivity (defined by Nisan [94] for Boolean functions) and obtain a lower bound on the expected query complexity of functions in terms of their block sensitivity.

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be sensitive to a subset (a "block") $I \subseteq [n]$ on input x, if f flips its value when flipping all the bits in I. The block sensitivity of f on x, $bs(f,x)$, is the maximum number t of pairwise disjoint subsets $I_1, \ldots, I_t \subseteq [n]$, to which f is sensitive on x. The block sensitivity of f, $bs(f)$, is the maximum, over all inputs x, of $bs(f,x)$.

For general domains and ranges, the basic intuition we carry over from the Boolean case is that a function is sensitive to a block of variables if a change to the corresponding input arguments results in a significant change to the function value. We denote the input obtained from x by changing the arguments in $I \subseteq [n]$ to the values in $a \in \mathcal{X}^{|I|}$ by $x^{(I \leftarrow a)}$. Figure 4.2 shows an example of an input x and the input $x^{(I \leftarrow a)}$ obtained from x by changing the arguments corresponding to $I = \{2,3,4\}$ to the assignment $a = (a_1, a_2, a_3)$.

Definition 4.2 (Block sensitivity)

- *f is ϵ-*sensitive to a subset of variables *$I \subseteq [n]$* on input *x, if there exists $a \in \mathcal{X}^{|I|}$ such that x and $x^{(I \leftarrow a)}$ are ϵ-disjoint.*

- *$bs_\epsilon(f,x)$, the ϵ-*block sensitivity of *f* on *x, is the maximum number t of pairwise disjoint subsets $I_1, \ldots, I_t \subseteq [n]$, such that f is ϵ-sensitive to each of them on x.*

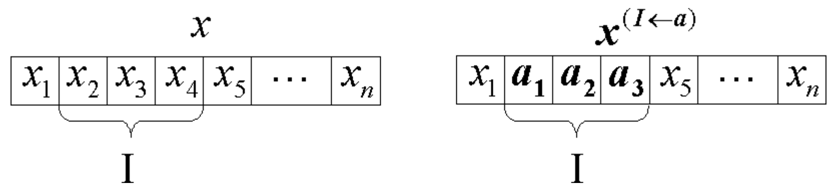

Figure 4.2: Block sensitivity

- $bs_\epsilon(f)$, *the* ϵ-block sensitivity of f, *is* $\max_{x\in\mathcal{X}^n} bs_\epsilon(f,x)$.

Nisan [94] proved that for Boolean functions f, $Q^E_{0,\delta}(f) \geq (1-2\delta)bs(f)$. We generalize this theorem to any function and to any approximation error:

Theorem 4.3 *For every* $f : \mathcal{X}^n \to \mathcal{Z}$, $\epsilon \geq 0$, *and* $0 \leq \delta \leq 1/2$,

$$Q^E_{\epsilon,\delta}(f) \;>\; (1-2\delta)\cdot bs_\epsilon(f).$$

Proof: Consider any decision tree T that (ϵ,δ)-approximates f. We will use the view of T as a distribution over deterministic decision trees $\{T_r\}_{r\in\mathcal{R}}$ induced by the choices of the random coins r according to ρ (see Section 3.2.2). We call a tree T_r bad for an input x, if $T_r(x) \notin C_{f,\epsilon}(x)$. Note that for every input x, $\Pr_{R\sim\rho}(T_R \text{ is bad for } x) < \delta$.

Let x be the input for which $bs_\epsilon(f) = bs_\epsilon(f,x)$, and let $I_1,\ldots,I_t \subseteq [n]$ be the disjoint variable blocks to which f is ϵ-sensitive on x (note that $t = bs_\epsilon(f)$). Finally, let $a_1,\ldots,a_t$ be the assignments to these blocks, for which x and $x^{(I_j\leftarrow a_j)}$ are ϵ-disjoint, $j = 1,\ldots,t$.

Consider a tree T_r in which the computation path corresponding to x does not query any of the variables in I_j for some $j\in[t]$. This means that the same computation path will correspond to $x^{(I_j\leftarrow a_j)}$. Moreover, since all the values queried on this computation path are the same at x and at $x^{(I_j\leftarrow a_j)}$, the leaf will output the same value z for both. However, since $C_{f,\epsilon}(x)\cap C_{f,\epsilon}(x^{(I_j\leftarrow a_j)}) = \emptyset$ either $z\notin C_{f,\epsilon}(x)$ or $z \notin C_{f,\epsilon}(x^{(I_j\leftarrow a_j)})$. So T_r is bad either for x or for $x^{(I_j\leftarrow a_j)}$. Therefore,

$$\begin{aligned}&\Pr_{R\sim\rho}(T_R \text{ does not query } I_j \text{ on } x)\\ &\quad\leq\; \Pr_{R\sim\rho}(T_R \text{ is bad for } x) + \Pr_{R\sim\rho}\left(T_R \text{ is bad for } x^{(I_j\leftarrow a_j)}\right) \;<\; 2\delta.\end{aligned}$$

Recall that $Q(T(x,R))$ is a random variable specifying the number of queries T performs on x. Define for every $j\in[t]$, a random variable X_j which counts the number of these queries that belong to I_j. Since $I_1, \ldots, I_t$ are pairwise disjoint, $Q(T(x,R)) \geq \sum_{j=1}^{t} X_j$. What we proved above implies that $\Pr(X_j \geq 1) > 1-2\delta$, implying that also $\mathrm{E}[X_j] > 1-2\delta$. Therefore, by the linearity of expectation,

$$\begin{aligned} Q^E_{\epsilon,\delta}(f) &\geq \mathrm{E}[Q(T(x,R))] \geq \mathrm{E}\left[\sum_{j=1}^{t} X_j\right] = \sum_{j=1}^{t} E(X_j) \\ &> t \cdot (1-2\delta) = (1-2\delta) \cdot bs_\epsilon(f). \end{aligned}$$

□

Tightness of the bound Let us consider an application of this lower bound to obtaining a query complexity lower bound for additive approximation of the mean: $\mu_1 : [0,1]^n \to [0,1]$. The best upper bound is $Q^W_{\epsilon,\delta}(\mu_1) = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$. Using Theorem 4.3 we prove the following:

Proposition 4.4 $Q^E_{\epsilon,\delta}(\mu_1) \geq \Omega(\frac{1}{\epsilon}(1-2\delta))$.

Proof: Let $x = 1^n$, and consider any subset $I \subseteq [n]$ of size $2\epsilon n$. Define an assignment $a = 0^{|I|}$. Since $\mu_1(x) = 1$ and $\mu_1(x^{(I \leftarrow a)}) = 1 - 2\epsilon$, x and $x^{(I \leftarrow a)}$ are ϵ-disjoint, implying μ_1 is ϵ-sensitive to I on x. This argument works for any subset I of size $2\epsilon n$; thus, the maximum number of pairwise disjoint subsets to which μ_1 is ϵ-sensitive on x is $n/|I| = 1/2\epsilon$. Applying Theorem 4.3 completes the proof. □

Proposition 4.4 shows that the block sensitivity lower bound does not yield tight bounds in general. In the case of the mean, the bound produced is not tight not in terms of the error ϵ, and not in terms of the confidence δ. In Section 4.4 we show how our statistical lower bound technique gives a tight lower bound for the mean.

Query complexity vs. block sensitivity Nisan [94] proved that for Boolean functions, block sensitivity and (deterministic/randomized) query complexity are polynomially related. A particular corollary of this is that for Boolean functions, deterministic and randomized query complexities are polynomially related. It is natural to ask if block sensitivity also gives a (polynomial) characterization of randomized query complexity for approximating general functions, but it turns out not to be the case. While the ϵ-approximate median problem has a randomized sampling algorithm that makes $O(1/\epsilon^2)$ queries (independent of n), it is easy to show that any deterministic algorithm needs at least $\Omega(n(1-\epsilon))$ queries (see Section 4.2.6).

4.2 Statistical lower bounds

The query complexity lower bound via block sensitivity is very general but is also weak. In this section we obtain stronger lower bounds that work only for the restricted class of *oblivious sampling computations* (see definitions in Section 3.2.3). These lower bounds are important on their own, since oblivious sampling is probably the most popular form of sampling. Moreover, we prove that for a wide class of functions—the *symmetric functions*—the lower bounds carry over to arbitrary sampling computations. The latter is proven via a reduction, showing that any sampling algorithm approximating a symmetric function can be simulated, with almost the same number of queries, by an oblivious sampling algorithm whose queries are chosen uniformly at random.

Reduction from hypothesis testing Our lower bounds are obtained via a reduction from statistical hypothesis testing (see Section 2.4.2) to oblivious sampling. Specifically, we prove that for any pair of ϵ-disjoint inputs x, y of a function f, any oblivious sampling algorithm, which uses q queries to (ϵ, δ)-approximate f and whose queries are chosen according to a fixed distribution Δ_q, derives a hypothesis test that uses q samples to distinguish, with probability of error δ, between Δ_x and Δ_y—the distributions of query outcomes on x and on y, respectively. We then apply known lower bounds on the number of samples required for a hypothesis test to achieve a prescribed error probability to derive lower bounds on the query complexity of the sampling computation. These bounds are stated in terms of various notions of distance between the distributions Δ_x and Δ_y, such as the *Hellinger distance* and the *KL distance* (see Section 2.4.1 for definitions).

Oblivious sampling lower bounds Some of the bounds we prove apply only to special kinds of oblivious sampling algorithms. Specifically, we consider two restrictions on oblivious sampling algorithms:

1. The queries are "i.i.d." (independent and identically distributed), meaning that the queries are all drawn from the same distribution and are independent of each other. In other words, $\Delta_q = \Delta^q$, where Δ is some distribution on $[n]$.

2. The algorithm is *variable-oblivious* (see definition in Section 3.2.3). That is, the algorithm uses only the outcomes of the queries, and is indifferent to the identity of the variables queried.

Restrictions	Type	Bound
i.i.d. & var-oblivious	worst-case	$\Omega\left(\frac{1}{h^2(\Delta_x,\Delta_y)} \cdot \log\frac{1}{\delta}\right)$
i.i.d.	worst-case	$\Omega\left(\frac{1}{V\left(\tilde{\Delta}_x,\tilde{\Delta}_y\right)} \cdot \log\frac{1}{\delta}\right)$
i.i.d. & var-oblivious	expected	$\Omega\left(\frac{1}{\min\{D_{KL}(\Delta_x\\|\Delta_y),D_{KL}(\Delta_y\\|\Delta_x)\}} \cdot \log\frac{1}{\delta}\right)$
var-oblivious	worst-case	$\Omega\left(\frac{1}{h_M^2(\Delta_{q,x},\Delta_{q,y})} \cdot \log\frac{1}{\delta}\right)$
none	worst-case	$\Omega\left(\frac{1}{h_M^2\left(\tilde{\Delta}_{q,x},\tilde{\Delta}_{q,y}\right)} \cdot \log\frac{1}{\delta}\right)$

Table 4.1: Lower bounds for oblivious sampling.

Oblivious sampling algorithms with i.i.d. queries are the most commonly used and analyzed. In fact, most of the statistics literature is concerned with this case. Variable-oblivious sampling algorithms are the ones obtained in the reduction for symmetric functions.

Table 4.1 lists the lower bounds for oblivious sampling algorithms. The table specifies for each bound what restrictions it assumes the algorithms satisfy, and whether the bound holds only for the worst-case query complexity, or also for the expected query complexity. The bounds are stated in terms of distances between Δ_x and Δ_y—the distributions of outcomes of queries on x and y— and between $\tilde{\Delta}_x$ and $\tilde{\Delta}_y$—the joint distributions of the query variables and the corresponding query outcomes on x and y (see exact definitions in Section 3.2.3). Here, x and y are any pair of ϵ-disjoint inputs of the function f, for which we prove the lower bound.

The distance measures used are the Hellinger distance, h, the KL distance, D_{KL}, and the total variation distance, V. The bound for arbitrary oblivious sampling algorithm is stated in terms of the *marginal Hellinger distance*, h_M^2, which is defined in Section 4.2.4. In general, lower bounds in terms of the Hellinger distance and the marginal Hellinger distance are optimal. Lower bounds in terms of the KL distance are frequently tight, but can be as bad as 0. Lower bounds in terms of the total variation distance are usually not tight. Section 4.5 has a more elaborate discussion of the tightness of the bounds.

These results give the following simple recipe for obtaining lower bounds for specific functions: given a function f, we need to find a pair of ϵ-disjoint inputs x and y, and then calculate the distance between Δ_x and Δ_y or between $\tilde{\Delta}_x$ and $\tilde{\Delta}_y$ under the appropriate distance measure. Computing these distances, with the exception of the marginal Hellinger distance, is usually very easy. Any pair of ϵ-disjoint inputs gives a bound, but clearly if one selects the pair of ϵ-disjoint inputs that has the minimum distance among all pairs of ϵ-disjoint inputs, then the resulting bound is best. The lower bound techniques do not specify how to find this minimum distance pair; in practice, much of the "art" of applying the bounds to specific functions is in the selection of appropriate pairs of ϵ-disjoint inputs.

Lower bounds for symmetric functions After proving the lower bounds for oblivious sampling algorithms, we show how they can be used to derive lower bounds for arbitrary sampling algorithms that approximate *symmetric functions*. Symmetric functions are ones that are invariant under permutations of the input arguments:

Definition 4.5 (Symmetric functions) *A function $f : \mathcal{X}^n \to \mathcal{Z}$ is ϵ-*symmetric *if for every input $x \in \mathcal{X}^n$ and every permutation $\pi \in S_n$, $C_{f,\epsilon}(x) = C_{f,\epsilon}(\pi(x))$, where $\pi(x)_i = x_{\pi^{-1}(i)}$.*

Many basic functions, like the mean and higher statistical moments, frequency moments, selection functions, and norms are symmetric.

For symmetric functions, we present two reductions from oblivious sampling to general sampling. The first reduction shows how to simulate any sampling algorithm that approximates a symmetric function by an oblivious sampling algorithm that uses exactly the same number of queries and whose queries are chosen uniformly at random *without replacement* from $[n]$. The resulting algorithm is also *variable-oblivious*. The second reduction shows how to simulate oblivious sampling algorithms that have a uniform without replacement query distribution by oblivious sampling algorithms whose query distribution is uniform *with replacement*. We have two versions of the latter reduction: one that is variable-oblivious, but works only if the simulated algorithm uses at most $O(\sqrt{n})$ queries, and another that is variable-aware, but works for algorithms that use at most $n/2$ queries. Combining the two reductions gives a simulation of any sampling algorithm for a symmetric function by an oblivious sampling algorithm that queries uniformly at random with replacement. Uniform sampling with replacement is easier to analyze than uniform sampling without replacement, because its queries are i.i.d.

Table 4.2 lists the lower bounds for symmetric functions. For each bound, the table specifies restrictions on the kind of functions or approximations it applies to ($q \leq B$ means that it works only if the function has query complexity at most B; $\delta > p$ means that it works only if the confidence is at least p) and whether the bound holds only for the

Restrictions	Type	Bound
$q \leq O(\sqrt{n})$, $\delta \geq \Omega\left(\frac{1}{n}\right)$	worst-case	$\Omega\left(\frac{1}{h^2(U_x,U_y)} \cdot \log\frac{1}{\delta}\right)$
$q \leq \frac{n}{2} - o(n)$, $\delta \geq \Omega\left(\frac{1}{n}\right)$	worst-case	$\Omega\left(\frac{1}{V(U_x,U_y)} \cdot \log\frac{1}{\delta}\right)$
none	worst-case	$\Omega\left(\frac{1}{h_M^2(W_{q,x},W_{q,y})} \cdot \log\frac{1}{\delta}\right)$
$q \leq O(\sqrt{n})$, $\delta \geq \Omega\left(\sqrt[3]{\frac{1}{n}}\right)$	expected	$\Omega\left(\frac{1}{\min\{D_{KL}(U_x\|U_y),D_{KL}(U_y\|U_x)\}} \cdot \log\frac{1}{\delta}\right)$

Table 4.2: Lower bounds for symmetric functions

worst-case query complexity or also for the expected query complexity.

Qualitatively, the best bound is the one in terms of the marginal Hellinger distance. However, this bound is difficult to use, because computing the marginal Hellinger distance is not easy. The Hellinger distance bound is optimal, but works only if the query complexity is low (at most $O(\sqrt{n})$). The total variation distance bound is weak, but works for almost all functions. Identical lower bounds to the first three stated in the table hold for the expected query complexity; the only difference is that these bounds miss the $O(\log(1/\delta))$ factor, and are thus not tight with respect to δ. The bound via KL distance stated in the table works for the expected query complexity and has a tight dependence on δ. It applies, however, only to functions with expected query complexity at most $O(\sqrt{n})$.

Deterministic query complexity vs. randomized query complexity We also present an $n(1 - \textsc{Ham}(x,y))$ lower bound on the deterministic query complexity of symmetric functions. Here, x and y are any pair of ϵ-disjoint inputs and $\textsc{Ham}(\cdot,\cdot)$ is the *Hamming distance* between strings: $\textsc{Ham}(x,y) = |\{i \mid x_i \neq y_i\}|$. This bound shows that in general there may be an unbounded gap between the deterministic query complexity and the randomized query complexity. This is an outstanding contrast to the situation for Boolean functions, for which it is known [94] that the deterministic and the randomized query complexities are polynomially related.

This section is organized as follows: we start (Section 4.2.1) with the reduction from hypothesis testing to oblivious sampling. We then state and prove lower bounds on the sample complexity of hypothesis tests (Section 4.2.2). In Section 4.2.3 we present our lower bounds for oblivious sampling algorithms that have i.i.d. queries, and in Section 4.2.4 our lower bounds for general oblivious sampling algorithms. In Section 4.2.5 we present the reduction from uniform oblivious sampling to general sampling for symmetric functions. Finally, in Section 4.2.6 we present our query complexity lower bounds for symmetric functions.

4.2.1 From hypothesis testing to oblivious sampling

The main lemma of this section proves that any Δ_q-decision tree that (ϵ, δ)-approximates a function f can be used as a *hypothesis test* (see Section 2.4.2) that distinguishes between the joint query-outcome distributions $\tilde{\Delta}_{q,x}$ and $\tilde{\Delta}_{q,y}$ corresponding to any two ϵ-disjoint inputs $x, y \in \mathcal{X}^n$. Moreover, if the tree happens to be variable-oblivious, it can be used also to design a test that distinguishes between the outcome distributions $\Delta_{q,x}$ and $\Delta_{q,y}$.

Lemma 4.6 (Reduction from hypothesis testing to oblivious sampling) *Let T be a Δ_q-decision tree of worst-case query cost q and expected query cost q_E that (ϵ, δ) approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$. Then, for any two ϵ-disjoint inputs x, y,*

1. *There is a δ-error simple randomized hypothesis test for $\tilde{\Delta}_{q,x}, \tilde{\Delta}_{q,y}$.*

2. *If $\Delta_q = \Delta^q$, then there is a δ-error randomized hypothesis test for $\tilde{\Delta}_x, \tilde{\Delta}_y$, whose sample cost is q.*

3. *If $\Delta_q = \Delta^q$, then there is a δ-error sequential randomized hypothesis test for $\tilde{\Delta}_x, \tilde{\Delta}_y$, whose expected sample cost on both $\tilde{\Delta}_x$ and $\tilde{\Delta}_y$ is at most q_E.*

Furthermore, if T is variable-oblivious, then there are such tests also for $\Delta_{q,x}, \Delta_{q,y}$ and for Δ_x, Δ_y.

Proof: We start by presenting a simple randomized hypothesis test H for $\tilde{\Delta}_{q,x}, \tilde{\Delta}_{q,y}$. H draws a sample from the unknown distribution μ (which is guaranteed to be either $\tilde{\Delta}_{q,x}$ or $\tilde{\Delta}_{q,y}$), and runs the decision tree T on the chosen queries and their outcomes. If the output of T belongs to $C_{f,\epsilon}(x)$, H decides that $\mu = \tilde{\Delta}_{q,x}$, and otherwise it decides $\mu = \tilde{\Delta}_{q,y}$.

Note that H is randomized, since T is randomized. We need to prove H errs with probability at most δ. Assume, first, that $\mu = \tilde{\Delta}_{q,x}$. Then, the probability that T outputs a "correct" answer (i.e., a value in $C_{f,\epsilon}(x)$) is at least $1 - \delta$, implying that H is correct with at least that much probability. Assume, then, that $\mu = \tilde{\Delta}_{q,y}$. T now outputs a value in $C_{f,\epsilon}(y)$ with probability at least $1 - \delta$. However, since x, y are ϵ-disjoint (meaning that $C_{f,\epsilon}(x) \cap C_{f,\epsilon}(y) = \emptyset$), when T outputs a value in $C_{f,\epsilon}(y)$, then H decides $\mu = \tilde{\Delta}_{q,y}$. Therefore, also in this case H is correct with probability at least $1 - \delta$.

If $\Delta_q = \Delta^q$, then also $\tilde{\Delta}_{q,x} = \tilde{\Delta}_x^q$ and $\tilde{\Delta}_{q,y} = \tilde{\Delta}_y^q$. Therefore, a simple test for $\tilde{\Delta}_{q,x}, \tilde{\Delta}_{q,y}$ is equivalent to a general test for $\tilde{\Delta}_x, \tilde{\Delta}_y$ whose sample cost is q. We thus immediately derive the second part of the lemma from its first part.

Finally, we design a sequential randomized test for $\tilde{\Delta}_x, \tilde{\Delta}_y$. The test simply simulates the decision tree T. Each time the simulation requires a query, the test draws a sample from the unknown distribution μ (which is guaranteed to be either $\tilde{\Delta}_x$ or $\tilde{\Delta}_y$) and feeds the simulation with this sample as the query variable and the corresponding outcome. Note that the number of samples used by the test is always identical to the number of queries made by the tree. Therefore, the expected sample cost of the test on $\tilde{\Delta}_z$ ($z = x, y$) is at most the expected query cost of T on z, $Q^E(T(z)) \leq q_E$. An identical argument to the above shows the test errs with probability at most δ.

When T is variable-oblivious, we can simulate it using only query outcomes (without the corresponding query variables). Thus, identical tests to the ones described above, which always feed the simulation of T samples from $\Delta_{q,x}, \Delta_{q,y}$ (instead of $\tilde{\Delta}_{q,x}, \tilde{\Delta}_{q,y}$) or Δ_x, Δ_y (instead of $\tilde{\Delta}_x, \tilde{\Delta}_y$) will succeed with probability at least $1 - \delta$. We obtain as a result tests for $\Delta_{q,x}, \Delta_{q,y}$ and Δ_x, Δ_y. $\square$

4.2.2 Hypothesis testing lower bounds

The main aim of this section is to obtain lower bounds on the number of samples required for a hypothesis test for two distributions ν_0, ν_1. We present two lower bounds from the statistical decision theory literature: one works for standard tests, which always draw a fixed number of samples, and the other works for sequential tests. The lower bounds are stated in terms of various notions of distance between the distributions tested.

We start with a lower bound for standard tests, which was pointed out to us by David Zuckerman [130], but is probably common knowledge in the statistics community (we could not find an explicit reference). The lower bound is stated in terms of the Hellinger distance (see Section 2.4.1), and fundamentally uses its nice behavior under product distributions:

Theorem 4.7 (Sampling lower bound for standard hypothesis tests) *For every $0 < \delta \leq 1/4$ and for any δ-error (possibly randomized) hypothesis test H for two distribu-*

tions ν_0, ν_1 on a domain $\mathcal{X}$ with $h^2(\nu_0, \nu_1) \leq 1/2$,

$$S(H) \;>\; \frac{1}{4 \cdot h^2(\nu_0, \nu_1)} \ln \frac{1}{4\delta}.$$

Proof: Let $k = S(H)$. We can view H as a simple δ-error test for ν_0^k, ν_1^k. Using the connection between simple hypothesis testing and total variation distance (Propositions 2.56 and 2.58), we have: $V\left(\nu_0^k, \nu_1^k\right) > 1 - 2\delta$.

One can now apply the upper bound on the total variation distance of product distributions (Lemma 2.33) to obtain a $(1/V(\nu_0, \nu_1))(1 - 2\delta)$ lower bound on the sample cost k. However, this bound is weak, and its weakness results from the slack in Lemma 2.33. We thus use the Hellinger distance, which has a much tighter characterization for the distance between product distributions.

In order to move to Hellinger distance, we first employ the upper bound it gives on the total variation distance (Corollary 2.39):

$$\begin{aligned} h^2\left(\nu_0^k, \nu_1^k\right) \;&\geq\; 1 - \sqrt{1 - V^2\left(\nu_0^k, \nu_1^k\right)} \;>\; 1 - \sqrt{1 - (1 - 2\delta)^2} \\ &=\; 1 - \sqrt{4\delta(1 - \delta)} \;>\; 1 - \sqrt{4\delta}. \end{aligned}$$

Proposition 2.37 allows us to rewrite the left-hand-side as follows:

$$1 - \left(1 - h^2(\nu_0, \nu_1)\right)^k \;>\; 1 - \sqrt{4\delta}.$$

It follows that:

$$k \;>\; \frac{\ln \frac{1}{\sqrt{4\delta}}}{\ln \frac{1}{1 - h^2(\nu_0, \nu_1)}} \;=\; \frac{\ln \frac{1}{4\delta}}{2 \cdot \ln \frac{1}{1 - h^2(\nu_0, \nu_1)}} \;\geq\; \frac{1}{4 \cdot h^2(\nu_0, \nu_1)} \ln \frac{1}{4\delta},$$

where in the last inequality, we used the fact that $1 - x \geq e^{-2x}$ for $0 \leq x \leq 1/2$. □

We next present a lower bound on the expected sample complexity of sequential hypothesis tests. This bound is known in the statistical sequential analysis literature as the *optimality of the sequential probability ratio test*. Proofs can be found, e.g., in [115], Section 2.4, or in [41]. For completeness, we repeat the proof here:

Theorem 4.8 (Lower bound for deterministic sequential tests) *For any δ-error deterministic sequential hypothesis test T for two distributions ν_0, ν_1 on a domain $\mathcal{X}$,*

$$S^E(T) \;>\; \frac{1}{\min\left\{D_{KL}(\nu_0 \parallel \nu_1), D_{KL}(\nu_1 \parallel \nu_0)\right\}} \cdot (1 - \delta) \cdot \log\left(\frac{1}{\delta} - 1\right).$$

Proof: T draws independent samples from the unknown distribution μ, and after a certain number of steps S stops and outputs a decision. S can be viewed as a random variable $S : \mathcal{X}^\infty \to \{0, 1, 2, \ldots\}$. Our goal is to give a lower bound on $S^E_{\nu_0}(T) = \mathrm{E}_{\nu_0}[S]$ and $S^E_{\nu_1}(T) = \mathrm{E}_{\nu_1}[S]$. We will show the lower bound on $\mathrm{E}_{\nu_0}[S]$. An identical argument shows the equivalent bound for $\mathrm{E}_{\nu_1}[S]$.

Let $C_0 \subseteq \mathcal{X}^*$ denote the set of sample sequences on which T decides "ν_0", and let $C_1 \subseteq \mathcal{X}^*$ denote the set of sample sequences on which T decides "ν_1". Let $C \stackrel{\text{def}}{=} C_0 \cup C_1$. Since T errs with probability at most δ, $\mathrm{Pr}_{\nu_0}(C_0) > 1 - \delta$ and $\mathrm{Pr}_{\nu_1}(C_0) < \delta$.

We define L to be a random variable on the probability space $\mathcal{X}$, measuring the "log-likelihood ratio" of ν_0 and ν_1:

$$L(x) \stackrel{\text{def}}{=} \ln\left(\frac{\nu_0(x)}{\nu_1(x)}\right).$$

We define R to be a random variable on the probability space C, measuring the log-likelihood ratio of the distributions of sample sequences obtained under ν_0 and under ν_1:

$$R(x_1, \ldots, x_s) \stackrel{\text{def}}{=} \sum_{i=1}^{s} L(x_i).$$

Using Wald's identity (Theorem 2.62), we can write the expectation of R as:

$$\mathrm{E}_{\nu_0}[R] = \mathrm{E}_{\nu_0}[S] \cdot \mathrm{E}_{\nu_0}[L]. \tag{4.1}$$

Note that $\mathrm{E}_{\nu_0}[L] = \ln 2 \cdot D_{KL}(\nu_0 \parallel \nu_1)$. We can lower bound $\mathrm{E}_{\nu_0}[R]$ as follows:

$$\begin{aligned} \mathrm{E}_{\nu_0}[R] &= \sum_{\vec{x} \in C} R(\vec{x}) \cdot \Pr_{\nu_0}(\vec{x}) \geq \sum_{\vec{x} \in C_0} R(\vec{x}) \cdot \Pr_{\nu_0}(\vec{x}) \\ &= \sum_{\vec{x} \in C_0} \ln\left(\frac{\Pi_{i=1}^{|\vec{x}|}\nu_0(x_i)}{\Pi_{i=1}^{|\vec{x}|}\nu_1(x_i)}\right) \cdot \Pi_{i=1}^{|\vec{x}|}\nu_0(x_i). \end{aligned} \tag{4.2}$$

We next use the "log sum inequality" (cf. [40], Chapter 2), which states the following:

Theorem 4.9 (Log sum inequality) *For non-negative $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$,*

$$\sum_{i=1}^{n} a_i \ln \frac{a_i}{b_i} \geq \left(\sum_{i=1}^{n} a_i\right) \ln \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}.$$

Using this inequality, we obtain in Equation 4.2:

$$\begin{aligned} \sum_{\vec{x} \in C_0} \ln\left(\frac{\Pi_{i=1}^{|\vec{x}|}\nu_0(x_i)}{\Pi_{i=1}^{|\vec{x}|}\nu_1(x_i)}\right) \cdot \Pi_{i=1}^{|\vec{x}|}\nu_0(x_i) &\geq \left(\sum_{\vec{x} \in C_0} \Pi_{i=1}^{|\vec{x}|}\nu_0(x_i)\right) \cdot \ln\left(\frac{\sum_{\vec{x} \in C_0} \Pi_{i=1}^{|\vec{x}|}\nu_0(x_i)}{\sum_{x \vec{\in} C_0} \Pi_{i=1}^{|\vec{x}|}\nu_1(x_i)}\right) \\ &= \Pr_{\nu_0}(C_0) \cdot \ln \frac{\mathrm{Pr}_{\nu_0}(C_0)}{\mathrm{Pr}_{\nu_1}(C_0)} > (1 - \delta) \cdot \ln \frac{1 - \delta}{\delta}. \end{aligned}$$

Combining this lower bound on $\mathrm{E}_{\nu_0}[R]$ and Equation 4.1, we obtain the following lower bound on $\mathrm{E}_{\nu_0}[S]$:

$$\begin{aligned} \mathrm{E}_{\nu_0}[S] &> \frac{1}{\ln 2 \cdot D_{KL}(\nu_0 \parallel \nu_1)} \cdot (1-\delta) \cdot \ln \frac{1-\delta}{\delta} \\ &= \frac{1}{D_{KL}(\nu_0 \parallel \nu_1)} \cdot (1-\delta) \cdot \log\left(\frac{1}{\delta} - 1\right). \end{aligned}$$

An identical argument shows that:

$$\mathrm{E}_{\nu_1}[S] > \frac{1}{D_{KL}(\nu_1 \parallel \nu_0)} \cdot (1-\delta) \cdot \log\left(\frac{1}{\delta} - 1\right).$$

□

The above lower bound holds only for deterministic sequential tests. An easy argument extends it to randomized tests too:

Corollary 4.10 (Lower bound for randomized sequential tests) *For every* $0 < \delta < 1/4$, *and for any* δ*-error randomized sequential hypothesis test* T *for two distributions* ν_0, ν_1 *on a domain* $\mathcal{X}$,

$$S^E(T) > \frac{1}{4 \cdot \min\{D_{KL}(\nu_0 \parallel \nu_1), D_{KL}(\nu_1 \parallel \nu_0)\}} \cdot (1-4\delta) \cdot \log\left(\frac{1}{4\delta} - 1\right).$$

Proof: For $b = 0, 1$, let $k_b = S^E_{\nu_b}(T)$. We view the randomized test T as a distribution $\{T_r\}_{r \in \mathcal{R}}$ over deterministic tests induced by the choice of the random coins r according to ρ. For $b = 0, 1$, let $\delta_{b,r}$ denote the error of T_r on ν_b, and let $k_{b,r}$ denote the expected sample cost of T_r on ν_b. We have for $b = 0, 1$,

$$\mathrm{E}_{R\sim\rho}[\delta_{b,R}] \leq \delta, \qquad \mathrm{E}_{R\sim\rho}[k_{b,R}] = k_b.$$

By Markov's inequality, less than 1/4 of the tests have error $\delta_{b,r} > 4\delta$, and less than 1/4 of the tests have expected sample cost $k_{b,r} > 4k_b$. Therefore, there is at least one test T_r that has both error $\delta_{b,r} \leq 4\delta$ and expected sample cost $k_{b,r} \leq 4k_b$ for $b = 0, 1$. Applying now Theorem 4.8 to T_r, we obtain:

$$4k_b > \frac{1}{D_{KL}(\nu_b \parallel \nu_{1-b})} \cdot (1-4\delta) \cdot \log\left(\frac{1}{4\delta} - 1\right).$$

□

4.2.3 Lower bounds for oblivious sampling with i.i.d. queries

Combining the connection between oblivious sampling and hypothesis testing and the sampling lower bounds for hypothesis testing, we obtain query complexity lower bounds for functions approximated by oblivious decision trees that use i.i.d. queries.

Lemma 4.6 (part (2)) and Theorem 4.7 immediately imply the following lower bound on the worst-case query cost of variable-oblivious Δ_q-decision trees:

Theorem 4.11 (Worst-case query cost of variable-oblivious sampling) *For every $\epsilon \geq 0$, every $0 < \delta \leq 1/4$, every $\Delta_q = \Delta^q$, every variable-oblivious Δ_q-decision tree T that (ϵ, δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$, and every ϵ-disjoint inputs x, y with $h^2(\Delta_x, \Delta_y) \leq 1/2$,*

$$Q^W(T) \;>\; \frac{1}{4 \cdot h^2(\Delta_x, \Delta_y)} \ln \frac{1}{4\delta}.$$

For variable-aware Δ_q-decision trees we can prove only a weaker bound:

Theorem 4.12 (Worst-case query cost of variable-aware sampling) *For every $\epsilon \geq 0$, every $0 < \delta \leq 1/4$, every $\Delta_q = \Delta^q$, every Δ_q-decision tree T that (ϵ, δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$, and every ϵ-disjoint inputs x, y for which $\Delta(\{i \mid x_i \neq y_i\}) \leq 1/2$,*

$$Q^W(T) \;>\; \frac{1}{4 \cdot V\left(\tilde{\Delta}_x, \tilde{\Delta}_y\right)} \ln \frac{1}{4\delta} \;=\; \frac{1}{4 \cdot \Delta(\{i \mid x_i \neq y_i\})} \ln \frac{1}{4\delta}.$$

Proof: We first prove the following fact:

Claim 4.12.1 $h^2\left(\tilde{\Delta}_x, \tilde{\Delta}_y\right) \;=\; V\left(\tilde{\Delta}_x, \tilde{\Delta}_y\right) \;=\; \Delta(\{i \mid x_i \neq y_i\})$

Proof: From Definition 2.35 and Proposition 2.32, we can write the Hellinger distance and the total variation distance as follows:

$$h^2\left(\tilde{\Delta}_x, \tilde{\Delta}_y\right) \;=\; \frac{1}{2} \sum_{i \in [n], a \in \mathcal{X}} \left(\sqrt{\tilde{\Delta}_x(i,a)} - \sqrt{\tilde{\Delta}_y(i,a)}\right)^2.$$

$$V\left(\tilde{\Delta}_x, \tilde{\Delta}_y\right) \;=\; \frac{1}{2} \sum_{i \in [n], a \in \mathcal{X}} \left|\tilde{\Delta}_x(i,a) - \tilde{\Delta}_y(i,a)\right|.$$

Every term in each of the above sums corresponds to a pair $(i, a) \in [n] \times \mathcal{X}$. In both sums, this term is $\Delta(i)$ if and only if exactly one of x_i, y_i equals a, and is 0 otherwise. Thus, each variable $i \in [n]$ for which $x_i \neq y_i$ contributes $2\Delta(i)$ to each of the two sums

($\Delta(i)$ in the term (i, x_i) and $\Delta(i)$ in the term (i, y_i)), while each variable i for which $x_i = y_i$ contributes 0. It follows that:

$$h^2\left(\tilde{\Delta}_x, \tilde{\Delta}_y\right) = V\left(\tilde{\Delta}_x, \tilde{\Delta}_y\right) = \Delta\left(\{i \mid x_i \neq y_i\}\right).$$

□

It follows from Claim 4.12.1 and the fact that $\Delta\left(\{i \mid x_i \neq y_i\}\right) \leq 1/2$ that also $h^2\left(\tilde{\Delta}_x, \tilde{\Delta}_y\right) \leq 1/2$. The theorem now follows from Lemma 4.6 and Theorem 4.7. □

Lemma 4.6 (part(3)) and Corollary 4.10 immediately imply the following lower bound on the expected query cost of variable-oblivious sampling. This bound is strong, but not as strong as the one on the worst-case query cost, because the KL distance may be larger than the Hellinger distance.

Theorem 4.13 (Expected query cost of variable-oblivious sampling) *For every $\epsilon \geq 0$, every $0 < \delta < 1/4$, every $\Delta_q = \Delta^q$, every variable-oblivious Δ_q-decision tree T that (ϵ, δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$, and every ϵ-disjoint inputs x, y,*

$$Q^E(T) \quad > \quad \frac{1}{4 \cdot \min\left\{D_{KL}\left(\Delta_x \parallel \Delta_y\right), D_{KL}\left(\Delta_y \parallel \Delta_x\right)\right\}} \cdot (1 - 4\delta) \cdot \log\left(\frac{1}{4\delta} - 1\right).$$

We cannot use Corollary 4.10 to obtain a lower bound on the expected query cost of variable-aware sampling, because $D_{KL}\left(\tilde{\Delta}_x \parallel \tilde{\Delta}_y\right) = D_{KL}\left(\tilde{\Delta}_y \parallel \tilde{\Delta}_x\right) = \infty$ for all $x \neq y$.

4.2.4 Lower bounds for general oblivious sampling

In this section we present a direct lower bound technique for oblivious sampling, which does not use the hypothesis testing sampling lower bounds. This technique is more difficult to use, but works for arbitrary oblivious decision trees, even ones whose queries are not independent or identically distributed.

This technique too uses the Hellinger distance. However, since the query distributions are not independent, we cannot use the multiplicativity of Hellinger distance for product distributions (Proposition 2.37) directly. We start with a short detour, in which we generalize Proposition 2.37 to deal with non-product distributions.

For a distribution μ on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_k$, for $j \in [k]$, we define $\textsc{Supp}_j(\mu)$ to be the projection of the support of μ on its first j coordinates. For any $(x_1, \ldots, x_{j-1}) \in \textsc{Supp}_{j-1}(\mu)$, we denote by $\mu^{j, x_1, \ldots, x_{j-1}}$ the marginal distribution of μ on $\mathcal{X}_j$, conditioned on the first $j-1$ coordinates of μ being $x_1,\ldots,x_{j-1}$. By the definition of conditional probability, for any such μ and for any $x_1,\ldots,x_k$,

$$\mu\left(x_1,\ldots,x_k\right) \;=\; \prod_{j=1}^{k} \mu^{j,x_1,\ldots,x_{j-1}}\left(x_j\right).$$

We use this characterization to define the following notion of distance between distributions:

Definition 4.14 (Marginal Hellinger distance) *Let μ,ν be two distributions on $\mathcal{X}_1\times\cdots\times\mathcal{X}_k$. For $j\in[k]$, the j-*marginal Hellinger distance of μ and ν *is:*

$$h^2_{M,j}\left(\mu,\nu\right) \stackrel{\text{def}}{=} \max\left\{h^2\left(\mu^{j,x_1,\ldots,x_{j-1}},\nu^{j,x_1,\ldots,x_{j-1}}\right) \mid \left(x_1,\ldots,x_{j-1}\right)\in \textsc{Supp}_{j-1}\left(\mu\right)\cap\textsc{Supp}_{j-1}\left(\nu\right)\right\}.$$

The marginal Hellinger distance of μ and ν *is:*

$$h^2_M\left(\mu,\nu\right) \stackrel{\text{def}}{=} \max_{j\in[k]} h^2_{M,j}\left(\mu,\nu\right).$$

Using the multiplicativity of Hellinger distance we can prove the following:

Proposition 4.15 *For any two distributions μ,ν on $\mathcal{X}_1\times\cdots\times\mathcal{X}_k$,*

$$1-h^2\left(\mu,\nu\right) \;\geq\; \prod_{j=1}^{k}\left(1-h^2_{M,j}\left(\mu,\nu\right)\right).$$

Note that when $\mu=\rho^k$ and $\nu=\lambda^k$, then $h^2_{M,j}\left(\mu,\nu\right)=h^2\left(\rho,\lambda\right)$ for all j, and therefore Proposition 2.37 is a (stronger) special case of Proposition 4.15.

Proof: In the sequel we use the following notation. $\vec{x}^j$ denotes a vector $\left(x_1,\ldots,x_j\right)$. $\textsc{Supp}_j\left(\mu,\nu\right)$ denotes the intersection $\textsc{Supp}_j\left(\mu\right)\cap\textsc{Supp}_j\left(\nu\right)$.

$$
\begin{aligned}
& 1 - h^2(\mu, \nu) \\
&\quad = \sum_{\vec{x}^k \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_k} \sqrt{\mu(x_1, \ldots, x_k)\, \nu(x_1, \ldots, x_k)} \\
&\quad = \sum_{\vec{x}^k \in \mathrm{SUPP}_k(\mu,\nu)} \sqrt{\prod_{j=1}^{k} \mu^{j,x_1,\ldots,x_{j-1}}(x_j) \cdot \prod_{j=1}^{k} \nu^{j,x_1,\ldots,x_{j-1}}(x_j)} \\
&\quad = \sum_{x_1 \in \mathrm{SUPP}_1(\mu,\nu)} \sqrt{\mu^1(x_1)\, \nu^1(x_1)} \sum_{x_2 \,:\, \vec{x}^2 \in \mathrm{SUPP}_2(\mu,\nu)} \sqrt{\mu^{2,x_1}(x_2)\, \nu^{2,x_1}(x_2)} \ldots \\
&\qquad\quad \ldots \sum_{x_k \,:\, \vec{x}^k \in \mathrm{SUPP}_k(\mu,\nu)} \sqrt{\mu^{k,x_1,\ldots,x_{k-1}}(x_k)\, \nu^{k,x_1,\ldots,x_{k-1}}(x_k)} \\
&\quad \geq \prod_{j=1}^{k} \left(1 - h_{M,j}^2(\mu,\nu)\right).
\end{aligned}
$$

$\square$

Theorem 4.16 (Lower bound for general oblivious sampling) *For every $\epsilon \geq 0$, every $0 < \delta \leq 1/4$, every Δ_q, every variable-oblivious Δ_q-decision tree T that (ϵ, δ) approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$, and every ϵ-disjoint inputs x, y with $h_M^2(\Delta_{q,x}, \Delta_{q,y}) \leq 1/2$,*

$$
Q^W(T) \;>\; \frac{1}{4 \cdot h_M^2(\Delta_{q,x}, \Delta_{q,y})} \ln \frac{1}{4\delta}.
$$

Moreover, the same bound holds for a variable-aware T, except that $h_M^2(\Delta_{q,x}, \Delta_{q,y})$ is replaced by $h_M^2\left(\tilde{\Delta}_{q,x}, \tilde{\Delta}_{q,y}\right)$.

Note that this lower bound is harder to use than the lower bounds for oblivious sampling with i.i.d. queries, because in order to apply it to a specific function we need to find two ϵ-disjoint inputs x, y, for which we can prove a uniform upper bound on $h^2\left(\Delta_{q,x}^{j,x_1,\ldots,x_{j-1}}, \Delta_{q,y}^{j,x_1,\ldots,x_{j-1}}\right)$ for all j and for all choices of $x_1, \ldots, x_{j-1}$.

Note that the lower bound is also not in closed form, because both $Q^W(T) = q$ and $h_M^2(\Delta_{q,x}, \Delta_{q,y})$ depends on q. The easiest way to get an explicit bound, is to derive an upper bound on $h_M^2(\Delta_{q,x}, \Delta_{q,y})$, which is independent of q (let's say for a small enough q).

Proof: By Lemma 4.6 (part (1)), T derives a simple δ-error hypothesis test H for $\Delta_{q,x}, \Delta_{q,y}$. Using the connection between simple hypothesis testing and total variation distance (Propositions 2.56 and 2.58), we have:

$$V\left(\Delta_{q,x}, \Delta_{q,y}\right) \;\geq\; 1 - 2\delta.$$

Moving to Hellinger distance (Corollary 2.39), we have:

$$h^2\left(\Delta_{q,x}, \Delta_{q,y}\right) \;\geq\; 1 - 2\sqrt{\delta}.$$

Applying Proposition 4.15, we have:

$$\prod_{j=1}^{q}\left(1 - h^2_{M,j}\left(\Delta_{q,x}, \Delta_{q,y}\right)\right) \;\leq\; 2\sqrt{\delta}.$$

By the definition of $h^2_M\left(\Delta_{q,x}, \Delta_{q,y}\right)$,

$$\left(1 - h^2_M\left(\Delta_{q,x}, \Delta_{q,y}\right)\right)^q \;\leq\; 2\sqrt{\delta}.$$

Therefore,

$$q \;\geq\; \frac{\ln \frac{1}{2\sqrt{\delta}}}{\ln \frac{1}{1 - h^2_M\left(\Delta_{q,x}, \Delta_{q,y}\right)}}.$$

Since $h^2_M\left(\Delta_{q,x}, \Delta_{q,y}\right) \leq 1/2$, we can use the inequality $1 - x \geq e^{-2x}$ for $0 \leq x \leq 1/2$, and obtain the following lower bound:

$$q \;\geq\; \frac{1}{2 \cdot h^2_M\left(\Delta_{q,x}, \Delta_{q,y}\right)} \ln \frac{1}{2\sqrt{\delta}} \;=\; \frac{1}{4 \cdot h^2_M\left(\Delta_{q,x}, \Delta_{q,y}\right)} \ln \frac{1}{4\delta}.$$

The same proof works for variable-aware decision trees, except for replacing $\Delta_{q,x}, \Delta_{q,y}$ by $\tilde{\Delta}_{q,x}, \tilde{\Delta}_{q,y}$. □

4.2.5 Generality of oblivious sampling for symmetric functions

The aim of this section is to show (Theorem 4.20) that for symmetric functions the restricted class of U_q-decision tree (i.e., ones that query only uniformly at random with replacement) is almost as good as general decision trees. We initially prove (Lemma 4.17) that any decision tree for a symmetric function can be simulated by a variable-oblivious W_q-decision tree (i.e., one that queries uniformly at random without replacement) with the same query cost; we then present (Lemma 4.19) two simulations of variable-oblivious W_q-decision trees by U_q-decision trees: a variable-oblivious simulation that works only for $q \leq O(\sqrt{n})$, and a variable-aware simulation that works for $q \leq n/2 - o(n)$.

Lemma 4.17 (Simulation by sampling without replacement) *For every $\epsilon \geq 0$ and $0 < \delta < 1$, and for every randomized decision tree T of worst-case query cost q and expected query cost q_E that (ϵ, δ)-approximates an ϵ-symmetric function $f : \mathcal{X}^n \to \mathcal{Z}$, there exists a variable-oblivious W_q-decision tree T' of worst-case query cost q and expected query cost q_E that (ϵ, δ)-approximates f.*

Proof: T' simulates T and has the same expected and worst-case query costs. Roughly, T' works as follows: given an input x, it chooses a random permutation $\pi \in S_n$ and runs T on $\pi(x)$. T is then likely to output an ϵ-approximation of $f(\pi(x))$, which by the symmetry of f is also an ϵ-approximation of $f(x)$. The only thing left to show is that T' is indeed a W_q-decision tree. Intuitively, this should be case, because no matter according to which distribution T queries its input or how adaptive it is, the queries T' performs are uniform without replacement, because they go through the filter of a random permutation. Making the latter argument precise requires some work, which forms the technical part of this proof.

Without loss of generality, we assume that T never queries the same variable more than once. T' works as follows: it chooses $I = (i_1, \ldots, i_q)$ according to W_q and then runs T^I, the I-relabeling of T (see Definition 3.11 in Section 3.2.3) on x. Since T is a standard decision tree, its child selection functions and its output functions depend only on query outcomes and not on the query variables. Therefore, T' is variable-oblivious. Clearly, T' has worst-case query cost q. We are left to prove that T' (ϵ, δ)-approximates f, and that its expected query cost is q_E. In order to do that, we first take a short detour to investigate the properties of the I-relabeling transformation.

Consider the computation of a deterministic decision tree T_D of query cost q on an input x. Since T_D is deterministic the computation path of x in T_D is fully determined. We therefore denote by $CP(T_D(x)) = (v_1, \ldots, v_k, u)$ $(k \leq q)$ the computation path of x in T_D and by $QS(T_D(x)) = (j_1, \ldots, j_k)$ the query sequence of T_D on x—the sequence of variables queried along the computation path.

For a sequence of q distinct input variables $I = (i_1, \ldots, i_q)$, let k be the number of queries made by T_D^I, the I-relabeling of T_D, on x. In other words, $QS(T_D^I(x)) = (i_1, \ldots, i_k)$. Let $CP(T_D^I(x)) = (v_1, \ldots, v_k, u)$, and let $(j_1, \ldots, j_k)$ be the variables labeling $v_1, \ldots, v_k$ in T_D. We define σ_x^I to be the partial permutation induced by x and I: $\sigma_x^I(i_1) = j_1, \ldots, \sigma_x^I(i_k) = j_k$. We define $S(\sigma_x^I)$ to be the set of permutations in S_n consistent with σ_x^I.

We next prove several properties of these partial permutations:

Proposition 4.18

1. *For all x and I, $\pi \in S(\sigma_x^I)$ if and only if π satisfies the following two properties:*

$$CP(T_D(\pi(x)) = CP(T_D^I(x)). \tag{4.3}$$

$$QS(T_D(x)) = (\pi(i_1), \ldots, \pi(i_k)). \tag{4.4}$$

2. *For all x and all $\pi \in S_n$, there exists some I such that $\pi \in S(\sigma_x^I)$.*

3. *For all x and all $I \neq I'$, either $\sigma_x^I = \sigma_x^{I'}$ or $S(\sigma_x^I) \cap S(\sigma_x^{I'}) = \emptyset$.*

Proof: We start by proving that any permutation π consistent with σ_x^I satisfies the two properties (4.3) and (4.4). We prove by induction on ℓ that for any $\ell = 1, \ldots, k+1$, the computation paths of $\pi(x)$ in T_D and of x in T_D^I share the first ℓ nodes.

The induction base, $\ell = 1$, is easy, since the two computation paths share the same first node—the root. Assume, then, that the computation paths of $\pi(x)$ and x share the first ℓ nodes. Recall that the computation path of x in T_D^I is $v_1, \ldots, v_k, u$ and the corresponding variable labels in T_D are $j_1, \ldots, j_k$. Since π is consistent with σ_x^I, then $\pi(i_1) = j_1, \ldots, \pi(i_k) = j_k$. If $\ell \leq k$, the ℓ-th node on the shared segment of $CP(T_D(\pi(x)))$ and $CP(T_D^I(x))$ is v_ℓ. The item T_D queries in v_ℓ on $\pi(x)$ is $\pi(x)_{j_\ell} = \pi(x)_{\pi(i_\ell)} = x_{i_\ell}$, which is exactly the same item T_D^I queries in v_ℓ on x. Therefore, the outcomes of these two queries are identical, implying that also the next node on the two computation paths are the same. If $\ell = k+1$, then it means the two computation paths are fully identical, and we are done.

The induction argument shows π has to satisfy property (4.3). It follows that $CP(T_D(\pi(x))) = (v_1, \ldots, v_k, u)$, implying $QS(T_D(\pi(x))) = (j_1, \ldots, j_k) = (\pi(i_1), \ldots, \pi(i_k))$, since π is consistent with σ_x^I. Hence, π also satisfies property (4.4).

For the other direction, assume π satisfies the two properties (4.3) and (4.4). From property (4.3) we know $CP(T_D(\pi(x))) = (v_1, \ldots, v_k, u)$, and therefore $QS(T_D(\pi(x))) = (j_1, \ldots, j_k)$. But now, from property (4.4), $\pi(i_1) = j_1, \ldots, \pi(i_k) = j_k$, implying π is consistent with σ_x^I. This completes the proof of the first part of the proposition.

For the second part, consider any $\pi \in S_n$, and let $QS(T_D(\pi(x))) = (j_1, \ldots, j_k)$, $(k \leq q)$. We choose $I = (i_1, \ldots, i_q)$ as follows: $i_1 = \pi^{-1}(j_1), \ldots, i_k = \pi^{-1}(j_k)$ and $i_{k+1}, \ldots, i_q$ are arbitrary distinct variables different from $i_1, \ldots, i_k$. We will show that π satisfies (4.3) and (4.4), with respect to I, implying by the first part that π is consistent with σ_x^I.

Assume, to the contradiction, π does not satisfy (4.3), and let $v_1, \ldots, v_\ell$ be the longest prefix common to $CP(T_D(\pi(x)))$ and $CP(T_D^I(x))$. Note that the item queried by T in v_ℓ when running on $\pi(x)$ is $\pi(x)_{j_\ell} = \pi(x)_{\pi(i_\ell)} = x_{i_\ell}$. However, the same item is also queried by T_D^I in v_ℓ when running on x. Therefore the next node on the computation paths of both $\pi(x)$ and x should be the same. Contradiction.

It follows now that in particular $QS(T_D^I(x)) = (i_1, \ldots, i_k)$ and therefore, by the definition of I, π satisfies also property (4.4). This completes the proof of the second part.

Finally, consider $I \neq I'$, $I = (i_1, \ldots, i_q)$ and $I' = (i'_1, \ldots, i'_q)$. Define $CP(T_D^I(x)) = (v_1, \ldots, v_k, u)$, $QS(T_D^I(x)) = (i_1, \ldots, i_k)$,$CP(T_D^{I'}(x)) = (v'_1, \ldots, v'_{k'}, u')$ and $QS(T_D^{I'}(x)) = (i'_1, \ldots, i'_{k'})$. If $k = k'$ and $(i_1, \ldots, i_k) = (i'_1, \ldots, i'_{k'})$, then $\sigma_x^I = \sigma_x^{I'}$, and we are done. Otherwise, $(i_1, \ldots, i_k)$ and $(i'_1, \ldots, i'_{k'})$ are different. Note that one cannot be a prefix of the other, and therefore there is some index ℓ such that $i_\ell \neq i'_\ell$. Assume ℓ is chosen to be the first such index.

Since I and I' share a prefix of length $\ell - 1$, $CP(T_D^I(x))$ and $CP(T_D^{I'}(x))$ share the first ℓ nodes. That is, $v_1 = v'_1, \ldots, v_\ell = v'_\ell$. Let j_ℓ be the label of $v_\ell = v'_\ell$ in T_D. By definition, $\sigma_x^I(j_\ell) = i_\ell$, while $\sigma_x^{I'}(j_\ell) = i'_\ell$, and therefore $\sigma_x^I(j_\ell) \neq \sigma_x^{I'}(j_\ell)$. This implies there is no permutation that is consistent with both σ_x^I and $\sigma_x^{I'}$. $\square$

There are two consequences of Proposition 4.18 which we are going to use. The first one is that for any permutation $\pi \in S(\sigma_x^I)$, the output of T_D on $\pi(x)$, $T_D(\pi(x))$, is identical to the output of T_D^I on x, $T_D^I(x)$. This follows from the fact that $\pi(x)$ and x have identical computation paths (and in particular the same output node), and moreover the data items queried on both are $x_{i_1}, \ldots, x_{i_k}$. Therefore, the outputs computed should also be the same.

The second consequence is that if one picks I according to W_q and then picks π uniformly at random from $S(\sigma_x^I)$, then the resulting distribution on S_n is uniform. To prove this, let $\sigma_1, \ldots, \sigma_m$ be the set of all partial permutations obtained as σ_x^I over all choices of I. For a partial permutation σ, denote by $U(\sigma)$ the uniform distribution on $S(\sigma)$. Now, for any $\tau \in S_n$,

$$\begin{aligned}
\Pr_{I \sim W_q, \Pi \sim U(\sigma_x^I)} (\Pi = \tau) &= \sum_J \Pr_{\Pi \sim U(\sigma_x^J)} (\Pi = \tau) \cdot \Pr_{I \sim W_q} (I = J) \\
&= \sum_{j=1}^{m} \Pr_{\Pi \sim U(\sigma_j)} (\Pi = \tau) \cdot \Pr_{I \sim W_q} \left(\sigma_x^I = \sigma_j\right).
\end{aligned}$$

Now, by the second and third parts of Proposition 4.18, for each permutation τ there exists exactly one partial permutation σ_j for which $\tau \in S(\sigma_j)$. Assume that σ_j is defined on k variables $(i_1, \ldots, i_k)$. It means that the size of $S(\sigma_j)$ is $(n-k)!$. The number of variable sequences I for which $\sigma_x^I = \sigma_j$ is $(n-k)!/(n-q)!$. The probability of each such variable sequence is $(n-q)!/n!$. Therefore,

$$\Pr_{\Pi \sim U(\sigma_j)} (\Pi = \tau) \cdot \Pr_{I \sim W_q} \left(\sigma_x^I = \sigma_j\right) \;=\; \frac{1}{(n-k)!} \cdot \frac{(n-k)!}{(n-q)!} \cdot \frac{(n-q)!}{n!} \;=\; \frac{1}{n!}.$$

We now use these two consequences to complete the proof of the lemma. We will use the view of T as a distribution over deterministic decision trees $\{T_r\}_{r \in \mathcal{R}}$, induced by the choice of the random coins r according to ρ (see Section 3.2.2). Under this view, T' first picks I, then chooses r according to ρ, and finally runs T_r^I on x.

From the first consequence of Proposition 4.18, for any input x, any choice of r, and any query sequence I, there is a partial permutation $\sigma_{x,r}^I$, such that for all $\pi \in S(\sigma_{x,r}^I)$, $T_r^I(x) = T_r(\pi(x))$. Denote by $U(\sigma_{x,r}^I)$ the uniform distribution on $S(\sigma_{x,r}^I)$. We obtain:

$$\begin{aligned}
\Pr_{\rho, W_q} \left(T'(x) \in C_{f,\epsilon}(x)\right) &= \Pr_{R \sim \rho, I \sim W_q} \left(T_R^I(x) \in C_{f,\epsilon}(x)\right) \\
&= \Pr_{R \sim \rho, I \sim W_q, \Pi \sim U(\sigma_{x,R}^I)} \left(T_R(\Pi(x)) \in C_{f,\epsilon}(x)\right).
\end{aligned}$$

Since f is ϵ-symmetric, $C_{f,\epsilon}(x) = C_{f,\epsilon}(\Pi(x))$. Therefore,

$$\begin{aligned}
\Pr_{\rho, W_q} \left(T'(x) \in C_{f,\epsilon}(x)\right) &= \Pr_{R \sim \rho, I \sim W_q, \Pi \sim U(\sigma_{x,R}^I)} \left(T_R(\Pi(x)) \in C_{f,\epsilon}(\Pi(x))\right) \\
&= \mathrm{E}_{R \sim \rho} \left[\Pr_{I \sim W_q, \Pi \sim U(\sigma_{x,R}^I)} \left(T_R(\Pi(x)) \in C_{f,\epsilon}(\Pi(x))\right) \right].
\end{aligned}$$

From the second consequence of Proposition 4.18, for every fixed input x and random coins r, the distribution of Π when choosing I according to W_q and then Π uniformly at random from $S(\sigma_{x,r}^I)$ is uniform in S_n. Denote by $U(S_n)$ the uniform distribution on S_n. We thus obtain:

$$\begin{aligned}
\Pr_{\rho, W_q} \left(T'(x) \in C_{f,\epsilon}(x)\right) &= \mathrm{E}_{R \sim \rho} \left[\Pr_{\Pi \sim U(S_n)} \left(T_R(\Pi(x)) \in C_{f,\epsilon}(\Pi(x))\right) \right] \\
&= \mathrm{E}_{\Pi \sim U(S_n)} \left[\Pr_{R \sim \rho} \left(T_R(\Pi(x)) \in C_{f,\epsilon}(\Pi(x))\right) \right] \;>\; 1 - \delta,
\end{aligned}$$

where the last inequality follows from the fact T (ϵ, δ)-approximates f. This proves that T' (ϵ, δ)-approximates f.

By Proposition 4.18 for any choice of r, any choice of I, and any input x, for all permutations $\pi \in S(\sigma^I_{x,r})$, $Q\left(T^I_r(x)\right) = Q\left(T_r(\pi(x))\right)$. Therefore,

$$\begin{aligned} Q^E(T') &= \max_x \mathrm{E}_{R\sim\rho, I\sim W_q}\left[Q\left(T^I_R(x)\right)\right] = \max_x \mathrm{E}_{R\sim\rho, I\sim W_q, \Pi\sim U(\sigma^I_{x,R})}\left[Q\left(T_R(\Pi(x))\right)\right] \\ &= \max_x \mathrm{E}_{R\sim\rho, \Pi\sim U(S_n)}\left[Q\left(T_R(\Pi(x))\right)\right] \le \max_x \max_{\pi\in S_n} \mathrm{E}_{R\sim\rho}\left[Q\left(T_R(\pi(x))\right)\right] = q_E. \end{aligned}$$

□

The following lemma presents two simulations of variable-oblivious W_q-decision trees by U_q-decision trees with almost the same query cost. The first one works only for decision trees of small enough query cost ($q \le O(\sqrt{n})$), but can be carried out variable-obliviously; the second one works for almost any query cost ($q \le n/2 - o(n)$), but is variable-aware.

Lemma 4.19 (From sampling without replacement to with replacement) *For every $\epsilon \ge 0$ and for every variable-oblivious W_q-decision tree T of worst-case query cost q and expected query cost q_E that (ϵ, δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$,*

1. *If $\frac{1}{2n} < \delta < \frac{1}{2}$ and $q \le \sqrt{2\delta n}$, there exists a variable-oblivious U_q-decision tree T' of worst-case query cost q and expected query cost $(1-\delta)q_E + \delta q$ that $(\epsilon, 2\delta)$-approximates f. Furthermore, if $q \le \sqrt[3]{2n \cdot q_E}$, then T' has expected query cost at most $2q_E$.*

2. *If $\frac{2}{n} < \delta < \frac{1}{2}$ and $q \le \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$, there exists a variable-aware U_{2q}-decision tree T'' of worst-case query cost $2q$ and expected query cost $2q_E$ that $(\epsilon, 2\delta)$-approximates f.*

Proof: We view T as a distribution over deterministic decision trees $\{T_r\}_{r\in\mathcal{R}}$ induced by the choice of the random coins r according to ρ. Given input x, T chooses I according to W_q and r according to ρ, and runs T^I_r on x.

We start with the proof of the first part. Given input x, T' picks $I = i_1, \ldots, i_q$ uniformly at random with replacement (i.e., according to U_q) and r according to ρ, and runs T^I_r on x. Clearly, T' is a U_q-decision tree whose worst-case query cost is q. Since T is variable-oblivious then so is T'. We are left to prove that if $q \le \sqrt{2\delta n}$, then T' $(\epsilon, 2\delta)$-approximates f and that its expected query cost is $(1-\delta)q_E + \delta q$.

Let E denote the event that $i_1, \ldots, i_q$ are distinct, and let $\overline{E}$ denote its complement. Note that conditioning on E, the distribution of $i_1, \ldots, i_q$ is uniform without replacement.

Thus, for any input x,

$$
\begin{aligned}
&\Pr_{I\sim U_q, R\sim\rho}\left(T_R^I(x) \notin C_{f,\epsilon}(x)\right) = \\
&= \Pr_{I\sim U_q, R\sim\rho}\left(T_R^I(x) \notin C_{f,\epsilon}(x) \ \mid\ E\right) \cdot \Pr_{I\sim U_q}(E) + \\
&\qquad\qquad \Pr_{I\sim U_q, R\sim\rho}\left(T_R^I(x) \notin C_{f,\epsilon}(x) \ \mid\ \overline{E}\right) \cdot \Pr_{I\sim U_q}\left(\overline{E}\right) \\
&\leq \Pr_{I\sim W_q, R\sim\rho}\left(T_R^I(x) \notin C_{f,\epsilon}(x)\right) + \Pr_{I\sim U_q}\left(\overline{E}\right) \\
&< \delta + \Pr_{I\sim U_q}\left(\overline{E}\right).
\end{aligned}
$$

We next show that $\Pr\left(\overline{E}\right) \leq \delta$, implying that T' $(\epsilon, 2\delta)$-approximates f. Define $\binom{q}{2}$ indicator random variables X_{jl}, such that $X_{jl} = 1$ iff $i_j = i_l$. $i_1, \ldots, i_q$ are not all distinct if and only if $\sum_{j,l} X_{jl} \geq 1$. The expectation of $\sum_{j,l} X_{jl}$ is $\binom{q}{2}\frac{1}{n} \leq \frac{q^2}{2n}$. Therefore, using Markov's inequality, $\Pr\left(\overline{E}\right) \leq \frac{q^2}{2n} \leq \delta$.

We finish the proof of the first part by proving that $Q^E(T') \leq (1-\delta)q_E + \delta q$:

$$
\begin{aligned}
Q^E(T') &= \max_x \mathrm{E}_{I\sim U_q, R\sim\rho}\left[Q(T_R^I(x))\right] \\
&= \max_x \left(\mathrm{E}_{I\sim U_q, R\sim\rho}\left[Q(T_R^I(x)) \mid E\right] \Pr_{I\sim U_q}(E) + \mathrm{E}_{I\sim U_q, R\sim\rho}\left[Q(T_R^I(x)) \mid \overline{E}\right] \Pr_{I\sim U_q}\left(\overline{E}\right)\right) \\
&\leq \max_x \left(\mathrm{E}_{I\sim W_q, R\sim\rho}\left[Q(T_R^I(x))\right]\right) \cdot \left(1 - \Pr\left(\overline{E}\right)\right) + q \cdot \Pr\left(\overline{E}\right) \ \leq \ q_E(1-\delta) + q\delta.
\end{aligned}
$$

Note that if $q \leq \sqrt[3]{2n \cdot q_E}$, the $q \cdot \Pr\left(\overline{E}\right) \leq q^3/(2n) \leq q_E$. Therefore, in this case, $Q^E(T') \leq q_E(1-\delta) + q_E < 2q_E$.

Turning to the proof of the second part, we construct T'' as a U_{2q}-decision tree that uses a transformation $\hat{T}$ of T. Each query node of T is replaced by a sub-tree that continuously performs queries until either obtaining a variable that was not queried before or until reaching a total of $2q$ queries. If the latter happens, T'' outputs an arbitrary value and stops. Note that this sub-tree is deterministic (i.e., does not consist of random coin nodes), and therefore when viewing $\hat{T}$ as a distribution over deterministic decision trees $\{\hat{T}_r\}_{r\in\mathcal{R}}$, r is distributed according to ρ and each $\hat{T}_r$ is obtained from T_r by the same transformation described above.

Given input x, T'' picks $2q$ variables $I = i_1, \ldots, i_{2q}$ uniformly at random with replacement and r according to ρ, and runs $\hat{T}_r^I$ on x. T'' is clearly a U_{2q}-decision tree of worst-case query cost $2q$. T'' is variable-aware, since it uses the query variables to determine whether each new query variable occurred already before or not. We next show that if $q \leq n/2 - \sqrt{n/(2\delta)}$, then T'' $(\epsilon, 2\delta)$-approximates f and that $Q^E(T'') \leq 2q_E$.

Again, we define E to be the event that at least q of the $2q$ variables chosen are distinct and $\overline{E}$ to be its complement. For a variable sequence $I \in [n]^{2q}$ that has at least q distinct variables, denote by $\hat{I}$ the sub-sequence of I consisting of the first q distinct variables. Note that conditioning on E, the distribution of $\hat{I}$ is uniform without replacement. Note also that for I with at least q distinct variables and for every r, $\hat{T}_r^{\,I}(x) = T_r^{\hat{I}}(x)$. Therefore,

$$\begin{aligned}
&\Pr_{I\sim U_{2q},R\sim\rho}\left(\hat{T}_R^I(x) \notin C_{f,\epsilon}(x)\right) = \\
&= \Pr_{I\sim U_{2q},R\sim\rho}\left(\hat{T}_R^I(x) \notin C_{f,\epsilon}(x) \;\middle|\; E\right) \cdot \Pr_{I\sim U_{2q}}(E) + \\
&\qquad \Pr_{I\sim U_{2q},R\sim\rho}\left(\hat{T}_R^I(x) \notin C_{f,\epsilon}(x) \;\middle|\; \overline{E}\right) \cdot \Pr_{I\sim U_{2q}}\left(\overline{E}\right) \\
&\leq \Pr_{\hat{I}\in W_q,R\sim\rho}\left(T_R^{\hat{I}}(x) \notin C_{f,\epsilon}(x)\right) + \Pr_{I\sim U_{2q}}\left(\overline{E}\right) \;<\; \delta + \Pr_{I\sim U_{2q}}\left(\overline{E}\right).
\end{aligned}$$

It thus suffices to prove that $\Pr_{I\sim U_{2q}}\left(\overline{E}\right) \leq \delta$. In order to do that, we define as before $\binom{2q}{2}$ indicator random variables X_{jl}, such that $X_{jl} = 1$ iff $i_j = i_l$. If the list $i_1, \ldots, i_{2q}$ contains at most q distinct values, then $\sum_{j,l} X_{jl} \geq q$ (because the way to create the least number of collisions is to have two occurrences for each of the q values). On the other hand, the expectation of $\sum_{j,l} X_{jl}$ is $\binom{2q}{2}\frac{1}{n} \leq \frac{2q^2}{n}$. Using the pairwise independence of X_{jl} and Chebyshev's inequality we obtain:

$$\begin{aligned}
\Pr\left(\overline{E}\right) &\leq \Pr\left(\sum_{j,l} X_{jl} \geq q\right) \leq \Pr\left(\left|\sum_{j,l} X_{jl} - \mathrm{E}\left[\sum_{j,l} X_{jl}\right]\right| \geq q - \frac{2q^2}{n}\right) \\
&\leq \frac{\textsc{Var}\left[\sum_{j,l} X_{jl}\right]}{q^2\left(1-\frac{2q}{n}\right)^2} = \frac{\binom{2q}{2}\frac{1}{n}\frac{n-1}{n}}{q^2\left(1-\frac{2q}{n}\right)^2} \leq \frac{\frac{2q^2}{n}}{q^2(1-\frac{2q}{n})^2} = \frac{2}{n\left(1-\frac{2q}{n}\right)^2}.
\end{aligned}$$

The last expression is at most δ for $q \leq n/2 - \sqrt{n/(2\delta)}$.

We now show that $Q^E(T'') \leq 2q_E$. The main observation here is that the expected number of U-queries needed to get each W-query (i.e., a variable that did not occur before) is at most 2, because there are at least $2q$ possible variables, and T uses at most q of them. Therefore,

$$\begin{aligned}
Q^E(T'') &= \max_x \mathrm{E}_{I\sim U_{2q},R\sim\rho}\left[Q\left(\hat{T}_R^I(x)\right)\right] = \max_x \mathrm{E}_{R\sim\rho}\left[\mathrm{E}_{I\in U_{2q}}\left[Q\left(\hat{T}_R^I(x)\right)\right]\right] \\
&\leq \max_x \mathrm{E}_{R\sim\rho}\left[2\cdot \mathrm{E}_{\hat{I}\sim W_q}\left[Q\left(T_R^{\hat{I}}(x)\right)\right]\right] = 2\cdot\max_x \mathrm{E}_{\hat{I}\sim W_q,R\sim\rho}\left[Q\left(T_R^{\hat{I}}(x)\right)\right] \\
&= 2q_E.
\end{aligned}$$

In the inequality we used Wald's identity (Theorem 2.62). In our case, X_i, the i-th element of the sequence of random variables, is the number of U-queries needed in the i-th query node of T_r to obtain a variable that did not occur before. The stopping time is the number of queries performed by T_r. □

Combining Lemma 4.17 and Lemma 4.19, we obtain the main result of this section:

Theorem 4.20 (Simulation by uniform sampling with replacement) *For every $\epsilon \geq 0$ and for every randomized decision tree T of worst-case query cost q and expected query cost q_E that (ϵ, δ)-approximates an ϵ-symmetric function $f : \mathcal{X}^n \to \mathcal{Z}$,*

1. *If $\frac{1}{2n} < \delta < \frac{1}{2}$ and $q \leq \sqrt{2\delta n}$, there exists a variable-oblivious U_q-decision tree T' of worst-case query cost q and expected query cost $(1-\delta)q_E + \delta q$ that $(\epsilon, 2\delta)$-approximates f. Furthermore, if $q \leq \sqrt[3]{2n \cdot q_E}$, then T' has expected query cost at most $2q_E$.*

2. *If $\frac{2}{n} < \delta < \frac{1}{2}$ and $q \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$, there exists a variable-aware U_{2q}-decision tree T'' of worst-case query cost $2q$ and expected query cost $2q_E$ that $(\epsilon, 2\delta)$-approximates f.*

4.2.6 Lower bounds for symmetric functions

We are now ready to prove our sampling lower bounds for symmetric functions. We start with three lower bounds on the worst-case query complexity. The first two use the reduction from uniform sampling with replacement to general sampling, and the third one uses only the reduction from uniform sampling without replacement to general sampling. The first two bounds are much easier to apply but yield slightly weaker bounds than the third.

The difference between the first two bounds is that the first one uses the reduction from variable-oblivious uniform sampling with replacement, while the second uses the reduction from variable-aware uniform sampling with replacement. Therefore, the first bound applies only to symmetric functions with worst-case query complexity $O(\sqrt{n})$, and the second bound can be used for almost all symmetric functions. The second bound, however, is in general weaker than the first one.

Theorem 4.21 (Worst-case query complexity of symmetric functions) *For every $\epsilon \geq 0$, every ϵ-symmetric function $f : \mathcal{X}^n \to \mathcal{Z}$, and every ϵ-disjoint inputs x, y, the following lower bounds on $Q^W_{\epsilon,\delta}(f)$ hold under the specified restrictions:*

Bound	*Restrictions*
$\frac{1}{4 \cdot h^2(U_x,U_y)} \ln \frac{1}{8\delta}$	$\frac{1}{2n} < \delta \leq \frac{1}{8}, \quad Q^W_{\epsilon,\delta}(f) \leq \sqrt{2\delta n}, \quad h^2\left(U_x, U_y\right) \leq \frac{1}{2}$
$\frac{1}{8 \cdot V(U_x,U_y)} \ln \frac{1}{8\delta}$	$\frac{2}{n} < \delta \leq \frac{1}{8}, \quad Q^W_{\epsilon,\delta}(f) \leq n/2 - \sqrt{n/(2\delta)}, \quad V\left(U_x, U_y\right) \leq \frac{1}{2}$
$\min\left\{t, \frac{1}{4\alpha} \ln \frac{1}{4\delta}\right\}$	$0 < \delta \leq \frac{1}{4}, \quad \forall q \leq t,\ h^2_M\left(W_{q,x}, W_{q,y}\right) \leq \alpha \leq \frac{1}{2}$

Proof: The first lower bound is an immediate consequence of the first part of Theorem 4.20 and of Theorem 4.11. The third lower bound immediately follows from Lemma 4.17 and from Theorem 4.16. In order to prove the second lower bound, we need to use the following fact:

Claim 4.21.1 $V\left(U_x, U_y\right) = \min\{V\left(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}\right) \mid \pi, \sigma \in S_n\}$

Proof: Since U_x and U_y can be obtained from $\tilde{U}_x$ and $\tilde{U}_y$ by projecting on their second coordinate, then $V\left(U_x, U_y\right) \leq V\left(\tilde{U}_x, \tilde{U}_y\right)$ (see Proposition 2.34). Let $\pi^*, \sigma^* \in S_n$ be the permutations in which the minimum of $V\left(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}\right)$ is obtained. Note that $V\left(U_x, U_y\right)$ is invariant under permutations: for all $\pi, \sigma \in S_n$, $V\left(U_x, U_y\right) = V\left(U_{\pi(x)}, U_{\sigma(y)}\right)$. Thus,

$$\begin{aligned} V\left(U_x, U_y\right) &= V\left(U_{\pi^*(x)}, U_{\sigma^*(y)}\right) \ \leq\ V\left(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}\right) \\ &= \min\{V\left(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}\right) \mid \pi, \sigma \in S_n\}. \end{aligned}$$

For the other direction, we first describe a procedure to permute x and y. We then claim that the obtained permutations π^* and σ^* satisfy $V\left(U_x, U_y\right) \geq V\left(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}\right)$.

The main idea is to move all the shared arguments of x and y to their front, such that they will have a maximal common prefix. Formally, for each $a \in \mathcal{X}$, denote by $f_x(a)$ and $f_y(a)$ the frequencies of a in x and y respectively (i.e., $f_x(a) = |\{i \in [n] \mid x_i = a\}|$ and similarly for $f_y(a)$). Let $m(a) = \min(f_x(a), f_y(a))$ and $M(a) = \max(f_x(a), f_y(a))$. Fix some arbitrary order on $\mathcal{X}$, and permute x and y as follows: enumerate the arguments in $a \in \mathcal{X}$ that have at least one occurrence in either x or y from smallest to largest according to the fixed order; for each such a, move its first $m(a)$ occurrences in x and in y to the front of x and y respectively. Note that after this procedure ends, x and y have a common prefix of length $m \stackrel{\text{def}}{=} \sum_{a \in \mathcal{X}} m(a)$; their two suffixes (of length $n - m$) are different in every position.

Denote by π^* and σ^* the permutations this procedure induces on x and y respectively. By Claim 4.12.1, $V\left(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}\right)$ equals the relative Hamming distance between

$\pi^*(x)$ and $\sigma^*(y)$. Therefore,

$$V\left(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}\right) \;=\; \frac{n-m}{n}.$$

On the other hand, if we define $M \stackrel{\text{def}}{=} \sum_{a\in A} M(a)$, then we have:

$$\begin{aligned} V\left(U_{\pi^*(x)}, U_{\sigma^*(y)}\right) \;&=\; V\left(U_x, U_y\right) \;=\; \frac{1}{2}\sum_{a\in\mathcal{X}} |U_x(a) - U_y(a)| \\ &=\; \frac{1}{2}\sum_{a\in\mathcal{X}} \frac{M(a)-m(a)}{n} \;=\; \frac{M-m}{2n}. \end{aligned}$$

Note that $M + m = 2n$ (since $M + m$ is the sum of frequencies of all $a \in \mathcal{X}$ in x and y), implying that $(M - m)/2n = (n - m)/n$. Therefore,

$$V\left(U_x, U_y\right) \;=\; V\left(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}\right) \;\geq\; \min\{V\left(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}\right) \mid \pi, \sigma \in S_n\}.$$

□

Let $q = Q^W_{\epsilon,\delta}(f)$, and let T be a decision tree of worst-case query cost q that (ϵ, δ)-approximates f. By the second part of Theorem 4.20, T can be simulated by a variable-aware U_{2q}-decision tree T'' of worst-case query cost $2q$ that $(\epsilon, 2\delta)$-approximates f.

Let π^*, σ^* be the permutations in which the minimum of $V\left(\tilde{U}_{\pi(x)}, \tilde{U}_{\sigma(y)}\right)$ over all $\pi, \sigma \in S_n$ is obtained. By Claim 4.21.1, $V\left(U_x, U_y\right) = V\left(\tilde{U}_{\pi^*(x)}, \tilde{U}_{\sigma^*(y)}\right)$. By Claim 4.12.1, the latter equals to $U\left(\{i \mid \pi^*(x)_i \neq \sigma^*(y)_i\}\right)$, and therefore, since $V\left(U_x, U_y\right) \leq 1/2$, also $U\left(\{i \mid \pi^*(x)_i \neq \sigma^*(y)_i\}\right) \leq 1/2$. Note also that since x, y are ϵ-disjoint, and since f is ϵ-symmetric, also $\pi^*(x), \sigma^*(y)$ are ϵ-disjoint. We can now apply Theorem 4.12, using T'', $\pi^*(x)$, and $\sigma^*(y)$,

$$2q \;>\; \frac{1}{4\cdot U\left(\{i \mid \pi^*(x)_i \neq \sigma^*(y)_i\}\right)} \ln\frac{1}{8\delta} \;=\; \frac{1}{4\cdot V\left(U_x, U_y\right)} \ln\frac{1}{8\delta}.$$

□

We can translate the worst-case query complexity lower bounds into expected query complexity lower bounds, using the following general reduction from worst-case query complexity to expected query complexity:

Proposition 4.22 (From expected to worst-case query complexity) *For every function $f : \mathcal{X}^n \to \mathcal{Z}$, every $\epsilon \geq 0$, and every $0 < \delta < 1/3$,*

$$Q^W_{\epsilon,3\delta}(f) \;\leq\; 2Q^E_{\epsilon,\delta}(f) \cdot \lceil \log\frac{1}{\delta} \rceil.$$

Proof: Let T be a decision tree of expected query cost $q_E = Q^E_{\epsilon,\delta}(f)$ that (ϵ,δ)-approximates f. We will construct a decision tree T' of worst-case query cost at most $2q_E\lceil\log\frac{1}{\delta}\rceil$ that $(\epsilon, 3\delta)$-approximates f.

Given input x, T' runs at most $d \stackrel{\text{def}}{=} \lceil\log\frac{1}{\delta}\rceil$ independent simulations of T one after the other. Each simulation is carried out for at most $2q_E$ queries. If at any simulation T outputs a value by the $2q_E$-th query, T' outputs the same value and stops. Otherwise, T' continues to the next simulation. If none of the d simulations outputs a value by the $2q_E$-th query, T' outputs an arbitrary value and stops.

It is clear that T' has a worst-case query complexity of at most $2q_E d = 2q_E\lceil\log\frac{1}{\delta}\rceil$. We are left to show that it $(\epsilon, 3\delta)$-approximates f. Fix some input x, and let $T_1,\dots,T_d$ be the d simulations of T. We call T_i "bad" if $Q(T_i(x)) > 2q_E$ (that is, T_i needs more than $2q_E$ queries). We denote by B_i the event that T_i is bad. Since $\mathrm{E}\left[Q(T_i(x))\right] = Q^E(T_i) = q_E$, then by Markov's inequality, $\Pr(B_i) = \Pr(Q(T_i(x)) > 2q_E) < 1/2$. We denote by C_i the event that $Q(T_i(x)) \le 2q_E$ and $T_i(x) \notin C_{f,\epsilon}(x)$. Since T_i (ϵ,δ)-approximates f, $\Pr(C_i) \le \Pr(T_i(x) \notin C_{f,\epsilon}(x)) < \delta$.

T' fails if and only if one of the following disjoint $i+1$ events occurs: $C_1, B_1 \wedge C_2, B_1 \wedge B_2 \wedge C_3, \dots, \bigwedge_{i=1}^{d-1} B_i \wedge C_d, \bigwedge_{i=1}^{d} B_i$. Using the mutual independence of the simulations we have:

$$\begin{aligned}
\Pr\left(T'(x) \notin C_{f,\epsilon}(x)\right) &= \sum_{i=1}^{d}\left(\prod_{j=1}^{i-1}\Pr(B_j)\cdot\Pr(C_i)\right) + \prod_{j=1}^{d}\Pr(B_j) \\
&< \sum_{i=1}^{d}\frac{1}{2^{i-1}}\cdot\delta + \frac{1}{2^d} \le 3\delta.
\end{aligned}$$

□

Theorem 4.21 and Proposition 4.22 immediately imply the following lower bounds on the expected query complexity. Note that since in the reduction we lose a factor of $O(\log(1/\delta))$ the following lower bounds are not tight in terms of δ:

Theorem 4.23 (Expected query complexity of symmetric functions) *For every $\epsilon \ge 0$, every ϵ-symmetric function $f : \mathcal{X}^n \to \mathcal{Z}$, and every ϵ-disjoint inputs x, y, the following lower bounds on $Q^E_{\epsilon,\delta}(f)$ hold under the specified restrictions:*

Bound	*Restrictions*
$\frac{\ln 2}{8 \cdot h^2(U_x,U_y)}$	$\frac{1}{6n} < \delta < \frac{1}{24}, \quad Q^W_{\epsilon,3\delta}(f) \leq \sqrt{6\delta n}, \quad h^2\left(U_x,U_y\right) \leq \frac{1}{2}$
$\frac{\ln 2}{16 \cdot V(U_x,U_y)}$	$\frac{2}{3n} < \delta < \frac{1}{24}, \quad Q^W_{\epsilon,3\delta}(f) \leq n/2 - \sqrt{n/(6\delta)}, \quad V\left(U_x,U_y\right) \leq \frac{1}{2}$
$\min\left\{\frac{t}{2\log\frac{1}{\delta}}, \frac{\ln 2}{8\alpha}\right\}$	$0 < \delta \leq \frac{1}{12}, \quad \forall q \leq t,\ h^2_M\left(W_{q,x}, W_{q,y}\right) \leq \alpha \leq \frac{1}{2}$

We next prove a direct lower bound on the expected query complexity, which does have the optimal dependence on δ. This bound, however, works only if the expected query complexity is guaranteed to be small enough (at most $O(\sqrt{n})$).

Theorem 4.24 (Query complexity lower bound via KL distance) *For every $\epsilon \geq 0$, every $\sqrt[3]{\frac{1}{2n}} < \delta < \frac{1}{16}$, every ϵ-symmetric function $f : \mathcal{X}^n \to \mathcal{Z}$ with $Q^E_{\epsilon,\delta}(f) \leq \sqrt{2\delta^3 n}$, and every ϵ-disjoint inputs x, y,*

$$Q^E_{\epsilon,\delta}(f) \ > \ \frac{1}{8 \cdot \min\left(D_{KL}\left(U_x \parallel U_y\right), D_{KL}\left(U_y \parallel U_x\right)\right)} \cdot (1 - 16\delta) \cdot \log\left(\frac{1}{16\delta} - 1\right).$$

Proof: Let $q_E = Q^E_{\epsilon,\delta}(f)$, and let T be a decision tree of expected query cost q_E that (ϵ, δ)-approximates f. Since we do not have a bound on the worst-case query cost of T, we transform T into a decision tree T' that $(\epsilon, 2\delta)$-approximates f, has expected query cost q_E, and has worst-case query cost q_E/δ. T' is the same as T, except that all the paths that consist of more than q_E/δ query nodes are pruned to have only q_E/δ query nodes. The $(q_E/\delta + 1)$-st query node in these paths is replaced by an output node that outputs an arbitrary fixed value. Note that by Markov's inequality the probability that T uses more than q_E/δ queries (on any input) is at most δ. Therefore, T' errs with probability at most 2δ.

Let $q = q_E/\delta$. Since $q_E \leq \sqrt{2\delta^3 n}$, then $q \leq \sqrt{4\delta n}$ and also $q \leq \sqrt[3]{2n \cdot q_E}$. Therefore, by the first part of Theorem 4.20, there exists a variable-oblivious U_q-decision tree T'' of worst-case query cost q and expected query cost $2q_E$ that $(\epsilon, 4\delta)$-approximates f.

Using now Theorem 4.13, we have:

$$2q_E \ > \ \frac{1}{4 \cdot \min\left(D_{KL}\left(\nu_0 \parallel \nu_1\right), D_{KL}\left(\nu_1 \parallel \nu_0\right)\right)} \cdot (1 - 16\delta) \cdot \log\left(\frac{1}{16\delta} - 1\right).$$

□

Finally, we prove a lower bound on the deterministic query complexity of symmetric functions:

Theorem 4.25 (Deterministic query complexity of symmetric functions) *For every $\epsilon \geq 0$, for every ϵ-symmetric function $f : \mathcal{X}^n \to \mathcal{Z}$, and for every ϵ-disjoint inputs x, y,*

$$Q_\epsilon^D(f) \;>\; n \cdot (1 - \textsc{Ham}(x, y)) .$$

Proof: Let $t \stackrel{\text{def}}{=} n(1 - \textsc{Ham}(x, y))$, and let $I = (i_1, \ldots, i_t)$ be the sequence of variables on which x and y agree.

Let T be the optimal deterministic decision tree that ϵ-approximates f. Consider the computation path of x in T^I, the I-relabeling of T. By Proposition 4.18, there is a permutation $\pi \in S_n$, such that $\pi(x)$ has the same computation path in T as the computation path of x in T^I. Similarly, there is a permutation $\sigma \in S_n$ such that $\sigma(y)$ has the same computation path in T as the computation path of y in T^I.

Note that since the first t variables queried in T^I are always $i_1, \ldots, i_t$, and since x and y agree on these variables, the first $t + 1$ nodes on the computation path of x in T^I are identical to the first $t + 1$ nodes on the computation path of y in T^I. It follows that also the computation paths of $\pi(x)$ and $\sigma(y)$ in T share the same first $t + 1$ nodes. Now, if the query complexity of T is at most t, then necessarily the computation paths of $\pi(x)$ and $\sigma(y)$ in T share the same leaf (output node). Therefore, $T(\pi(x)) = T(\sigma(y))$. However, since f is ϵ-symmetric and since x and y are ϵ-disjoint, then also $\pi(x)$ and $\sigma(y)$ are ϵ-disjoint. This implies that T outputs a wrong value either on $\pi(x)$ or on $\sigma(y)$. Therefore, $Q^D T > t = n(1 - \textsc{Ham}(x, y))$. □

4.3 An information-theoretic lower bound

The statistical lower bounds presented in the previous section were all based on the fact that any randomized decision tree that approximates a function derives a hypothesis test for Δ_x, Δ_y for every pair of ϵ-disjoint inputs x, y (Lemma 4.6). However, such a decision tree can be used also to form a multiple hypotheses test for $\{\Delta_x\}_{x \in \mathcal{S}}$ for any set $\mathcal{S}$ of pairwise ϵ-disjoint inputs, where $|\mathcal{S}|$ may be larger than 2. Indeed, when the function f has many mutually disjoint inputs (e.g., frequency distribution estimation; see Section 4.4.4) the statistical lower bounds are not tight. In this section we develop a new lower bound technique, based on information theory, which depends on the number of mutually disjoint inputs, and thus produces better (up to optimal) bounds for such functions.

The technique proves that any oblivious sampling algorithm approximating a function f requires $\Omega\left(1/\left(JS_{U(\mathcal{S})}\left(\{\Delta_x\}_{x\in\mathcal{S}}\right)\right)\log|\mathcal{S}|\left(1-\delta\right)\right)$ queries, where $JS_{U(\mathcal{S})}\left(\{\Delta_x\}_{x\in\mathcal{S}}\right)$ is the *Generalized Jensen-Shannon divergence* of $\{\Delta_x\}_{x\in\mathcal{S}}$ w.r.t. the uniform distribution on $\mathcal{S}$ (see Section 2.4.1). This is at least $\Omega\left(1/\left(\mathrm{E}\left[D_{KL}\left(\Delta_X \parallel \Delta_Y\right)\right]\right)\cdot\log|\mathcal{S}|\cdot(1-\delta)\right)$, where $(X,Y)\sim U(S)\times U(S)$. For the case $|\mathcal{S}|=2$, using the relations between the Jensen-Shannon divergence and the Hellinger distance (Proposition 2.51), the bound is $\Omega\left(1/h^2\left(\Delta_x,\Delta_y\right)\cdot(1-\delta)\right)$. Thus, this bound is as good as the best bounds achieved in the previous section in terms of the dependence on ϵ. The dependence on δ is much weaker.

As in the previous section, the bound applies only to oblivious sampling, but can be carried over to general sampling for symmetric functions.

The rest of the section is organized as follows: we start (Section 4.3.1) with an information-theoretic view of hypothesis testing, which allows us to use information theory to obtain a sample complexity lower bound for multiple hypotheses tests. We then use this bound (Section 4.3.2) to obtain query complexity lower bounds for oblivious sampling and for symmetric functions.

4.3.1 Information theory view of hypothesis testing

In this section we introduce a simple information-theoretic view of hypothesis testing, which allows us to use tools and results from information theory (most notably, Fano's inequality) in the analysis of hypothesis tests.

Recall that in the basic framework of information theory (see discussion in Section 2.3.2) a sender wants to send a message $x\in\mathcal{X}$ distributed according to μ_X to a receiver. The communication goes, however, through a noisy channel, and thus what the receiver gets to see is a message $y\in\mathcal{Y}$ distributed according to μ_Y. The receiver then applies a reconstruction function $g:\mathcal{Y}\to\mathcal{X}$, trying to predict what was the sent message x. The prediction error of g, denoted δ_g, is $\Pr\left(g(Y)\neq X\right)$, where $(X,Y)\sim(\mu_X,\mu_Y)$. Fano's inequality (Theorem 2.29) gives a lower bound on δ_g for any g the receiver may use:

$$H_2\left(\delta_g\right)+\delta_g\log\left(|\mathcal{X}|-1\right)\;\geq\;H\left(X\mid Y\right).$$

Multiple hypotheses testing can be casted in this framework as follows (see Figure 4.3). Let T be a δ-error multiple hypotheses test for a family of distributions $\{\nu_b\}_{b\in\mathcal{B}}$ on a domain $\mathcal{X}$, and let $k=S\left(T\right)$ be the sample cost of T. A sender picks a value $B\in\mathcal{B}$ according to the uniform distribution λ on $\mathcal{B}$, and would like to transmit the identity of

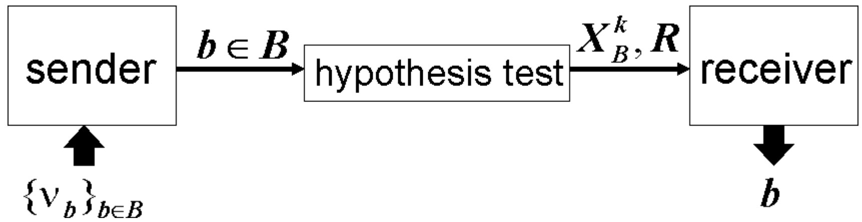

Figure 4.3: Information theory view of multiple hypothesis testing

the chosen distribution ($\log|\mathcal{B}|$ bits of information) to the receiver. However, the only available communication channel is the random samples T draws from ν_B. Thus, the sender's message is an index B distributed uniformly at random, and the received message is (X_B^k, R), the k independent samples T draws from ν_B plus internal random coin tosses R, which is distributed according to a distribution ρ on a domain $\mathcal{R}$. The hypothesis test T can be interpreted as a function $g^T : \mathcal{X}^k \times \mathcal{R} \to \mathcal{B}$, and the error δ is simply $\Pr_{B\sim\lambda, X_B\sim\nu_B, R\sim\rho}\left(g^T\left(X_B^k, R\right) \neq B\right)$. Using Fano's inequality, we have:

Proposition 4.26 $H_2(\delta) + \delta\log(|\mathcal{B}| - 1) \geq H\left(B \mid g^T\left(X_B^k, R\right)\right)$.

We use this connection to prove the following sample cost lower bound for multiple hypotheses tests:

Theorem 4.27 (Sampling lower bound for multiple hypotheses tests) *For every δ-error multiple hypotheses test T for a family of distributions $\{\nu_b\}_{b\in\mathcal{B}}$ on a domain $\mathcal{X}$,*

$$\begin{aligned} S(T) &\geq \frac{1}{JS_\lambda\left(\{\nu_b\}_{b\in\mathcal{B}}\right)}\left(\log|\mathcal{B}|(1-\delta) - H_2(\delta)\right) \\ &\geq \frac{1}{\mathrm{E}_{(B,B')\in\lambda\times\lambda}\left[D_{KL}\left(\nu_B \parallel \nu_{B'}\right)\right]}\left(\log|\mathcal{B}|(1-\delta) - H_2(\delta)\right), \end{aligned}$$

where λ is the uniform distribution on $\mathcal{B}$.

Proof: Let $k = S(T)$, let ρ be a distribution on $\mathcal{R}$ corresponding to the internal random coin tosses used by T, and let $g^T : \mathcal{X}^k \times \mathcal{R} \to \mathcal{B}$ be the function corresponding to the decisions of T. Let $B \sim \lambda$, $X_B \sim \nu_B$, and $R \sim \rho$. By Proposition 4.26,

$$H_2(\delta) + \delta\log(|\mathcal{B}| - 1) \;\geq\; H\left(B \mid g^T\left(X_B^k, R\right)\right). \tag{4.5}$$

Expanding the right-hand-side of Equation 4.5, we have:

$$
\begin{aligned}
& H\left(B \mid g^T\left(X_B^k, R\right)\right) \\
& = H(B) - I\left(B;\ g^T\left(X_B^k, R\right)\right) \quad \text{(Definition of mutual information)} \\
& \geq \log|\mathcal{B}| - I\left(B;\ X_B^k, R\right) \\
& \qquad (B \text{ is uniformly distributed and the data processing inequality (Corollary 2.28))} \\
& = \log|\mathcal{B}| - I(B;\ R) - I\left(B;\ X_B^k \mid R\right) \\
& \qquad \text{(Chain rule for mutual information (Proposition 2.26))} \\
& = \log|\mathcal{B}| - I\left(B;\ X_B^k\right) \quad \left(\left(B, X_B^k\right) \text{ are jointly independent of } R\right).
\end{aligned}
$$

Substituting back in Equation 4.5, we have:

$$
I\left(B;\ X_B^k\right) \ \geq\ \log|\mathcal{B}| - \delta\log\left(|\mathcal{B}| - 1\right) - H_2(\delta) \ \geq\ \log|\mathcal{B}|\left(1 - \delta\right) - H_2(\delta). \tag{4.6}
$$

Consider now the left-hand-side of Equation 4.6:

$$
\begin{aligned}
I\left(B;\ X_B^k\right) &= H\left(X_B^k\right) - H\left(X_B^k \mid B\right) \quad \text{(Definition of mutual information)} \\
&\leq k \cdot H(X_B) - H\left(X_B^k \mid B\right) \quad \text{(Sub-additivity of entropy (Proposition 2.15))} \\
&= k \cdot H(X_B) - k \cdot H(X_B \mid B) \quad \text{(Conditioned on } B,\ X_B^k \text{ is product)} \\
&= k \cdot I(B;\ X_B) \quad \text{(Definition of mutual information).}
\end{aligned}
$$

Combining with Equation 4.6, we obtain:

$$
k \ \geq\ \frac{1}{I(B;\ X_B)} \cdot \left(\log|\mathcal{B}|\left(1 - \delta\right) - H_2(\delta)\right). \tag{4.7}
$$

We are left to analyze the quantity $I(B;\ X_B)$. The distribution of X_B, ν_B, can be written as $\pi\left(\lambda, (\nu_b)_{b\in\mathcal{B}}\right)$, where π is simply the projection of the vector given as the second argument on the coordinate given as the first argument: $\pi\left(z, (x_b)_{b\in\mathcal{B}}\right) = x_z$.

With this view, we can apply the connection between mutual information and the generalized Jensen-Shannon divergence (Proposition 2.53):

$$
I(B;\ X_B) \ =\ I\left(\lambda;\ \pi\left(\lambda, (\nu_b)_{b\in\mathcal{B}}\right)\right) \ =\ JS_\lambda\left(\{\nu_b\}_{b\in\mathcal{B}}\right). \tag{4.8}
$$

Combining Equations 4.7 and 4.8 we obtain:

$$
k \ \geq\ \frac{1}{JS_\lambda\left(\{\nu_b\}_{b\in\mathcal{B}}\right)} \left(\log|\mathcal{B}|\left(1 - \delta\right) - H_2(\delta)\right).
$$

Applying the upper bound on the generalized Jensen-Shannon divergence via the KL distance (Proposition 2.54), we have also:

$$k \;\geq\; \frac{1}{\mathrm{E}_{(B,B')\sim\lambda\times\lambda}\left[D_{KL}\left(\nu_B \parallel \nu_{B'}\right)\right]}\left(\log|\mathcal{B}|\left(1-\delta\right)-H_2\left(\delta\right)\right).$$

□

4.3.2 Lower bounds for oblivious sampling and symmetric functions

We next use the lower bound achieved for multiple hypotheses testing to obtain query complexity lower bounds for oblivious sampling and for symmetric functions.

We first note that Lemma 4.6 easily generalizes to give a multiple hypotheses test for any set of pairwise disjoint inputs:

Lemma 4.28 (Multiple hypotheses tests from oblivious sampling) *Let T be a Δ_q-decision tree of worst-case query cost q and expected query cost q_E that (ϵ,δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$. Then, for any set $\mathcal{S} \subseteq \mathcal{X}^n$ of inputs, each two of which are ϵ-disjoint,*

1. *There is a δ-error randomized multiple hypotheses test for $\{\tilde{\Delta}_{q,x}\}_{x\in\mathcal{S}}$.*

2. *If $\Delta_q = \Delta^q$, then there is a δ-error randomized multiple hypotheses test for $\{\tilde{\Delta}_x\}_{x\in\mathcal{S}}$, whose sample cost is q.*

3. *If $\Delta_q = \Delta^q$, then there is a δ-error sequential randomized multiple hypotheses test for $\{\tilde{\Delta}_x\}_{x\in\mathcal{S}}$, whose expected sample cost on $\tilde{\Delta}_x$, for all $x \in \mathcal{S}$, is at most q_E.*

Furthermore, if T is variable-oblivious, then there are such tests also for $\{\Delta_{q,x}\}_{x\in\mathcal{S}}$ and for $\{\Delta_x\}_{x\in\mathcal{S}}$.

This reduction together with the multiple hypotheses testing lower bound (Theorem 4.27) immediately imply the following:

Theorem 4.29 (Lower bound for oblivious sampling) *For every $\epsilon \geq 0$, every $0 < \delta < 1$, every $\Delta_q = \Delta^q$, every variable-oblivious Δ_q-decision tree T that (ϵ,δ)-approximates a function $f : \mathcal{X}^n \to \mathcal{Z}$, and every set of $\mathcal{S} \subseteq \mathcal{X}^n$ of inputs, each two of which are ϵ-disjoint,*

$$\begin{aligned} Q^W(T) \;&\geq\; \frac{1}{JS_{U(\mathcal{S})}\left(\{\Delta_x\}_{x\in\mathcal{S}}\right)}\left(\log|\mathcal{S}|\left(1-\delta\right)-H_2\left(\delta\right)\right) \\ &\geq\; \frac{1}{\mathrm{E}_{(x,y(\in U(\mathcal{S})\times U(\mathcal{S})}\left[D_{KL}\left(\Delta_x \parallel \Delta_y\right)\right]}\left(\log|\mathcal{S}|\left(1-\delta\right)-H_2\left(\delta\right)\right), \end{aligned}$$

where $U(\mathcal{S})$ is the uniform distribution on S. Furthermore, if T is variable-aware, the same bounds hold, except for replacing Δ_x, Δ_y by $\tilde{\Delta}_x, \tilde{\Delta}_y$.

Using the reduction from uniform sampling with replacement to general sampling for symmetric functions (Theorem 4.20), we can derive the following:

Theorem 4.30 (Lower bounds for symmetric functions) *For every $\epsilon \geq 0$, every ϵ-symmetric function $f : \mathcal{X}^n \rightarrow \mathcal{Z}$, and every set of $\mathcal{S} \subseteq \mathcal{X}^n$ of inputs, each two of which are ϵ-disjoint, the following bounds hold on $Q^W_{\epsilon,\delta}(f)$, under the specified restrictions:*

Bound	*Restrictions*
$\frac{1}{JS_{U(\mathcal{S})}\left(\{U_x\}_{x\in\mathcal{S}}\right)} \left(\log\lvert\mathcal{S}\rvert\,(1-2\delta) - H_2\,(2\delta)\right)$	$\frac{1}{2n} < \delta \leq \frac{1}{2}, \quad Q^W_{\epsilon,\delta}(f) \leq \sqrt{2\delta n}$
$\frac{1}{JS_{U(\mathcal{S})}\left(\left\{\tilde{U}_x\right\}_{x\in\mathcal{S}}\right)} \left(\log\lvert\mathcal{S}\rvert\,(1-2\delta) - H_2\,(2\delta)\right)$	$\frac{2}{n} < \delta \leq \frac{1}{2}, \quad Q^W_{\epsilon,\delta}(f) \leq \frac{n}{2} - \sqrt{\frac{n}{2\delta}}$

Identical bounds hold for the expected query complexity, except that they are divided by a factor of $O(\log(1/\delta))$.

4.4 Applications

In this section we apply the lower bound techniques developed in the previous sections to derive lower bounds on the query complexity of specific functions. For the sake of brevity, we demonstrate mainly the use of the marginal Hellinger distance technique, which produces the best bounds. We demonstrate the use of the easier (but weaker) Hellinger distance technique and the KL distance technique for expected query complexity only in the first application.

We start with an inequality that will be useful in the analysis of several of the applications. We denote by $B(p)$ the Bernoulli distribution with probability of success p. The following is proven by a simple calculation:

Proposition 4.31 *For $0 \leq q < p \leq 1$, $h^2\left(B(p), B(q)\right) \leq \frac{d^2}{a(1-a)}$, where $d = \frac{p-q}{2}$ and $a = \frac{p+q}{2}$.*

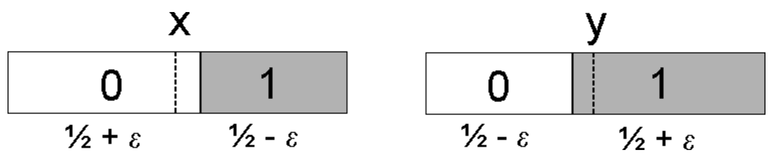

Figure 4.4: ϵ-disjoint inputs for μ_k

4.4.1 Statistical moments

The k-th statistical moment on $[0,1]$ is $\mu_k(x_1,\ldots,x_n) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} x_i^k$. The first moment, for example, is simply the mean.

An additive approximation of μ_k can be easily performed by taking a random sample of size $O\left(1/\epsilon^2 \cdot \log(1/\delta)\right)$, and computing the sample's k-th moment. Chernoff-Hoeffding bound [38, 65] implies that this yields an ϵ-additive approximation with probability $1-\delta$. Canetti, Even, and Goldreich [31] prove that the worst-case query complexity of the mean (and, implicitly, of all the other moments too) is $\Omega\left(1/\epsilon^2 \cdot \log(1/\delta)\right)$. Radhakrishnan and Ta-Shma [104] implicitly extend the bound to hold also for the expected query complexity. We provide simple proofs for both results:

Theorem 4.32 *The following lower bounds hold for μ_k under the specified restrictions:*

Type	*Bound*	ϵ	δ
worst-case	$\min\left\{\frac{\epsilon n}{2},\ \Omega\left(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\right)\right\}$	$0<\epsilon\le\frac{1}{\sqrt{18}}$	$0<\delta\le\frac{1}{4}$
expected	$\Omega\left(\frac{1}{\epsilon^2}\cdot\log\frac{1}{\delta}\right)$	$\Omega\left(\sqrt{\frac{\ln(1/\delta)}{\sqrt{\delta^3 n}}}\right)<\epsilon\le\frac{1}{4}$	$\sqrt[3]{\frac{1}{2n}}<\delta<\frac{1}{16}$

Proof: Consider the two inputs x and y. x has $(\frac{1}{2}+\epsilon)n$ 0's and rest are 1's; y has $(\frac{1}{2}-\epsilon)n$ 0's and the rest are 1's (see Figure 4.4). Since $\mu_k(x)=1/2-\epsilon$ and $\mu_k(y)=1/2+\epsilon$, x and y are ϵ-disjoint.

Note that $U_x = B\left(\frac{1}{2}-\epsilon\right)$ and $U_y = B\left(\frac{1}{2}+\epsilon\right)$, and therefore, by Proposition 4.31, $h^2(U_x,U_y)\le 4\epsilon^2$. Thus, the lower bound on the worst-case query complexity immediately follows from Theorem 4.21, part (1). However, this bound is restricted to large enough δ's. We next use the marginal Hellinger distance technique to get the same bound for all δ. All we need to do is show that $h_M^2(W_{q,x},W_{q,y})\le O(\epsilon^2)$ for $q\le\epsilon n/2$. The lower bound would then follow from Theorem 4.21, part (3).

The first sample from $W_{q,x}$ is distributed according to $U_x = B\left(\frac{1}{2}-\epsilon\right)$. After seeing j samples $b_1,\ldots,b_j \in \{0,1\}$, of which t_1 are ones, the $j+1$-st sample also is distributed Bernoulli, and its success probability is $p_{j,t_1} \stackrel{\text{def}}{=} \left(\left(\frac{1}{2}-\epsilon\right)n - t_1\right)/(n-j) = \left(1+\frac{j}{n-j}\right)\left(\frac{1}{2}-\epsilon-\frac{t_1}{n}\right)$. We want to show that for all j and t_1, $p_{j,t_1} \in \left[\frac{1}{2}-\frac{3\epsilon}{2}, \frac{1}{2}\right]$. Maximizing over j and t_1, $p_{j,t_1} \le \left(1+\frac{q}{n-q}\right)\left(\frac{1}{2}-\epsilon\right) \le \frac{1}{2}-\epsilon+\frac{q}{2(n-q)} \le \frac{1}{2}$, since $q \le \epsilon n/2 \le 2\epsilon(n-q)$. Minimizing over j and t_1, $p_{j,t_1} \ge \frac{1}{2}-\epsilon-\frac{q}{n} \ge \frac{1}{2}-\frac{3\epsilon}{2}$, since $q \le \epsilon n/2$.

A similar analysis shows that the probability of success in the j-th sample of $W_{q,y}$ for any $j \in [q]$ is in the interval $\left[\frac{1}{2}, \frac{1}{2}+\frac{3\epsilon}{2}\right]$. Clearly, the further the probability of success is from $\frac{1}{2}$, the larger is the distance between the two distributions. Therefore, $h_M^2\left(W_{q,x}, W_{q,y}\right) \le h^2\left(B\left(\frac{1}{2}-\frac{3\epsilon}{2}\right), B\left(\frac{1}{2}+\frac{3\epsilon}{2}\right)\right) \le 9\epsilon^2$.

As for the expected query complexity lower bound, all we need to do is calculate the KL distance between U_x and U_y. In the derivation we use the fact $\ln(1+x) \le x$ for $x \ge 0$:

$$\begin{aligned} D_{KL}\left(U_x \parallel U_y\right) &= \left(\frac{1}{2}+\epsilon\right)\log\frac{\frac{1}{2}+\epsilon}{\frac{1}{2}-\epsilon} + \left(\frac{1}{2}-\epsilon\right)\log\frac{\frac{1}{2}-\epsilon}{\frac{1}{2}+\epsilon} = 2\epsilon\log\frac{\frac{1}{2}+\epsilon}{\frac{1}{2}-\epsilon} \\ &= 2\epsilon\log\left(1+\frac{4\epsilon}{1-2\epsilon}\right) \le \frac{2\epsilon}{\ln 2}\cdot\frac{4\epsilon}{1-2\epsilon} \le \frac{16\epsilon^2}{\ln 2}. \end{aligned}$$

The lower bound follows from Theorem 4.24. □

4.4.2 Frequency moments

Given an input sequence $x_1,\ldots,x_n \in [m]$, the frequency of j for $j \in [m]$, is the number of times j occurs in the input sequence: $f_j\left(x_1,\ldots,x_n\right) \stackrel{\text{def}}{=} |\{i \mid x_i = j\}|$. The k-th frequency moment is $\mathrm{F}_k : [m]^n \to \mathbb{R}$, $\mathrm{F}_k\left(x_1,\ldots,x_n\right) \stackrel{\text{def}}{=} \sum_{j=1}^m f_j^k\left(x_1,\ldots,x_n\right)$. Thus, F_1 is the length of the input, n; F_0 (assuming $0^0 = 0$) is the number of distinct elements in the input; F_k, for $k \ge 2$, is a measure of the k-wise collision probability of the sequence: this is the probability to get $x_{i_1} = \ldots = x_{i_k}$ when picking $i_1,\ldots,i_k \in [n]$ uniformly at random with replacement.

We prove upper and lower bounds for relative approximation of F_k, $k \ge 2$, which are almost matching (in terms of n and m).

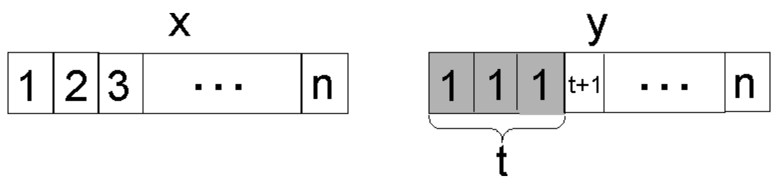

Figure 4.5: ϵ-disjoint inputs for F_k

Theorem 4.33 *Let $r = \min(n, m)$. For all $k \geq 2$, $0 < \epsilon < 1$, and $0 < \delta \leq 1/4$,*

$$\begin{aligned} Q^W_{\epsilon,\delta}(\mathrm{F}_k) &\geq \min\left\{\frac{n}{2},\ \Omega\left(\frac{r^{1-1/k}}{\epsilon^{1/k}}\log\frac{1}{\delta}\right)\right\}. \\ Q^W_{\epsilon,\delta}(\mathrm{F}_k) &\leq O\left(\frac{2^k \cdot k! \cdot r^{1-1/k}}{\epsilon^2}\log\frac{1}{\delta}\right). \end{aligned}$$

An interesting implication of this theorem is that for $k = 2$ there is an exponential gap between the sample complexity of F_2 and the data stream space complexity of F_2 (which is $O(\log m + \log n)$ [4]). However, for $k \geq 3$, the two are polynomially related (the data stream space complexity of F_k, $k \geq 3$, is $\Omega\left(m^{1-2/k}\right)$, as we show in Chapter 6).

Proof: For the lower bound, we will use Theorem 4.21, part (3). For simplicity of exposition, we assume $m \geq n$. The case $m < n$ is almost identical. Consider the following two inputs: the first input is $x = 1, \ldots, n$. The second input is constructed from x as follows. Divide the input of length n into n/t blocks of size $t = (3\epsilon n)^{1/k}$. y is x except in the first block where it is all 1's (see Figure 4.5). Clearly, $\mathrm{F}_k(x) = n$ while $\mathrm{F}_k(y) = t^k + (n - t) = 3\epsilon n + (n - (3\epsilon n)^{1/k}) = n(1 + 3\epsilon) - (3\epsilon n)^{1/k}$, implying x and y are ϵ-disjoint.

We next show that the marginal Hellinger distance between $W_{q,x}$ and $W_{q,y}$ is at most $O\left(\epsilon^{1/k}/n^{1-1/k}\right)$ for $q \leq n/2$. The only sequences of samples that belong both to the support of $W_{q,x}$ and to the support of $W_{q,y}$ are ones in which there is at most a single occurrence of 1 and there are no occurrences of $2, \ldots, t$. Let $a_1, \ldots, a_j$ be such a sequence. We analyze the Hellinger distance between the distribution of the $j+1$-st sample in $W_{q,x}$ and the distribution of the $j+1$-st sample in $W_{q,y}$, conditioned on the first j samples being $a_1, \ldots, a_j$.

There are two possible cases: either 1 occurs in $a_1, \ldots, a_j$ or it does not. In both cases only the values $1, \ldots, t$ have different probabilities to be the $j+1$-st sample under

$W_{q,x}$ and $W_{q,y}$. In the first case 1 has probability $\frac{t-1}{n-j}$ in $W_{q,x}$ and 0 in $W_{q,y}$. In the second case 1 has probability $\frac{t}{n-j}$ in $W_{q,x}$ and $\frac{1}{n-j}$ in $W_{q,y}$. In both cases $2,\ldots,t$ have probability 0 in $W_{q,x}$ and probability $\frac{1}{n-j}$ in $W_{q,y}$. Using the L_2 characterization of the Hellinger distance (Definition 2.35), we conclude that in the first case the Hellinger distance is $\frac{2(t-1)}{n-j}$, and in the second case it is $\frac{(\sqrt{t}-1)^2+t-1}{n-j}$. The maximum of the two is $\frac{2(t-1)}{n-j}$. Maximizing over all j, we have: $h_M^2\left(W_{q,x},W_{q,y}\right) \leq \frac{2(t-1)}{n-q}$. For $q \leq n/2$, the latter is at most $\frac{4t}{n} = \frac{4\cdot 3^{1/k}\cdot\epsilon^{1/k}}{n^{1-1/k}}$. The lower bound now follows from Theorem 4.21, part (3).

Now, we show that this lower bound is almost tight, at least for small values of k. Note that for any input x, $U_x(j) = f_j/n$, where $f_j = f_j(x)$ is the frequency of j in $x = x_1,\ldots,x_n$. Thus, computing $\mathrm{F}_k(x)$ is equivalent to estimating the k-wise collision probability of U_x, since $\mathrm{F}_k(x) = \sum_{j=1}^m f_j^k = n^k \sum_{j=1}^m U_x(j)^k$. So, the sampling algorithm picks ℓ samples $x_1,\ldots,x_\ell$ using U_x and counts the k-wise collisions C_k, our basic estimator. It is clear that $\mathrm{E}\left[C_k\right] = \binom{\ell}{k}||U_x||_k^k$. The following lemma, whose proof generalizes that of Goldreich and Ron [56] for F_2, helps us to bound the variance of this estimator. (A proof appears in Section A.2 of the Appendix).

Lemma 4.34 $\mathrm{V{\scriptsize AR}}\left[C_k\right] \leq 2^k\cdot(k!)^{1-1/k}\cdot\mathrm{E}\left[C_k\right]^{2-1/k}$.

Since $||U_x||_k^k \geq 1/r^{k-1}$, then by Chebyshev's inequality,

$$\Pr\left(\left|C_k - \mathrm{E}\left[C_k\right]\right| \geq \epsilon\mathrm{E}\left[C_k\right]\right) \;\leq\; \frac{2^k\cdot(k!)^{1-1/k}}{\epsilon^2\mathrm{E}\left[C_k\right]^{1/k}} \;\leq\; \frac{2^k\cdot(k!)^{1-1/k}\cdot r^{(k-1)/k}\cdot(k!)^{1/k}}{(\ell-k)\cdot\epsilon^2}.$$

For this to be a constant, say, 3/4, we get $\ell \geq \Omega\left(2^k\cdot k!\cdot r^{1-1/k}/\epsilon^2\right)$. Now, this experiment can be repeated $O\left(\log 1/\delta\right)$ times and the median of these various trials is correct with probability at least $1-\delta$. $\square$

Sampling algorithms for approximating F_0, the number of distinct elements, received much attention both in the database community (e.g., [99, 62, 36]) and in statistics [28]. Charikar, Chaudhuri, Motwani, and Narasayya [36] got the best results theoretically, showing both upper and lower bounds in the order of $\Theta\left(n(1-\epsilon)^2\log(1/\delta)\right)$ on the number of queries required to obtain a ratio approximation of F_0. We next give a simpler exposition of the proof of their lower bound:

Theorem 4.35 *For all* $0 \leq \epsilon \leq 1-\frac{1}{\sqrt{n}}$ *and* $0 < \delta \leq 1/4$,

$$Q_{\epsilon,\delta}^{W}(\mathrm{F}_0) \;\geq\; \min\left\{\frac{n}{2},\; \Omega\left(n\cdot(1-\epsilon)^2\cdot\log\frac{1}{\delta}\right)\right\}.$$

Proof: Let x be the input that is all 1's and let y be the input that has $n - \lfloor 1/(1-\epsilon)^2 \rfloor$ 1's, followed by $2, 3, \ldots, \lfloor 1/(1-\epsilon)^2 \rfloor + 1$. $\mathrm{F}_0(x) = 1, \mathrm{F}_0(y) \geq 1/(1-\epsilon)^2$, implying x and y are ϵ-disjoint. We next analyze the marginal Hellinger distance between $W_{q,x}$ and $W_{q,y}$ for $q \leq n/2$.

The only sequences of samples that belong to both supports are ones that consist only of 1's. After seeing j such samples, the probability of 1 in $W_{q,x}$ is 1 and the probability of 1 in $W_{q,y}$ is $1 - 1/(n-j)(1-\epsilon)^2$. Thus, the Hellinger distance between the distribution of the $j+1$-st sample in $W_{q,x}$ and the distribution of the $j+1$-st sample in $W_{q,y}$ conditioned on seeing j 1's before is $1 - \sqrt{1 - 1/(n-j)(1-\epsilon)^2} \leq 1/(n-j)(1-\epsilon)^2$. Maximizing over all j, this is at most $1/(n-q)(1-\epsilon)^2$, which is at most $2/n(1-\epsilon)^2$ for $q \leq n/2$. The lower bound now follows from Theorem 4.21, part (3). $\square$

4.4.3 Selection functions

The selection function of order p, for $0 \leq p \leq 1$, is defined as follows: $\textsc{Select}_p : \mathbb{R}^n \to \mathbb{R}$, $\textsc{Select}_p(x_1, \ldots, x_n) = x_{i_{\lfloor p(n-1) \rfloor}}$, where $x_{i_0} \leq \cdots \leq x_{i_{n-1}}$. For example, the median is $\textsc{Select}_{\frac{1}{2}}$, minimum is $\textsc{Select}_0$, and maximum is $\textsc{Select}_1$. The approximation for $0 < \epsilon \leq 1$ is $C_{\textsc{Select}_{p,\epsilon}}(x_1, \ldots, x_n) = \{x_{i_j} \mid j \in (\lfloor p'(n-1) \rfloor, \lfloor p''(n-1) \rfloor), p' = \max(p-\epsilon, 0), p'' = \min(p+\epsilon, 1)\}$.

By Chernoff bound, the median of a sample of size $O\left(1/\epsilon^2 \cdot \log(1/\delta)\right)$ yields an ϵ-approximate median. The same result holds for $\textsc{Select}_p$ for any constant $0 < p < 1$. However, for $p = 0$ (minimum) and for $p = 1$ (maximum), a sample of size $O\left(1/\epsilon \cdot \log(1/\delta)\right)$ is sufficient. We next show that these bounds are tight:

Theorem 4.36 *For all* $0 < \epsilon < 1$ *and* $0 < \delta \leq 1/4$,

1. *For* $\epsilon \leq p \leq 1 - \epsilon$, $\quad Q^W_{\epsilon,\delta}(\textsc{Select}_p) \geq \min\left\{\epsilon n,\ \Omega\left(\frac{p(1-p)}{\epsilon^2} \log \frac{1}{\delta}\right)\right\}$.
2. *For* $p < \epsilon, p > 1 - \epsilon$, $\quad Q^W_{\epsilon,\delta}(\textsc{Select}_p) \geq \min\left\{\epsilon n,\ \Omega\left(\frac{1}{\epsilon} \log \frac{1}{\delta}\right)\right\}$.

Proof: Assume, initially, that $\epsilon \leq p \leq 1 - \epsilon$. We pick $x, y \in \{0,1\}^n$ as follows (see Figure 4.6): x has $(p+\epsilon)n$ 0's and the rest are 1's, and y has $(p-\epsilon)n$ 0's and the rest are 1's. Clearly, $C_{\textsc{Select}_{p,\epsilon}}(x) = \{0\}$ and $C_{\textsc{Select}_{p,\epsilon}}(y) = \{1\}$, implying x and y are ϵ-disjoint. We next show that the marginal Hellinger distance between $W_{q,x}$ and $W_{q,y}$ is at most $O\left(\epsilon^2/p(1-p)\right)$ for $q \leq \epsilon n$. The lower bound would follow from Theorem 4.21, part (3).

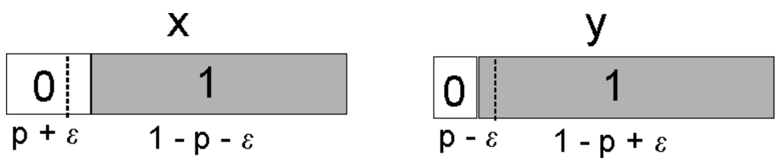

Figure 4.6: ϵ-disjoint inputs for $\textsc{Select}_p$

After seeing j samples, out of which t_1 are 1, the $j+1$-st sample in $W_{q,x}$ is distributed Bernoulli with success probability $\frac{(1-p-\epsilon)n-t_1}{n-j} = \left(1+\frac{j}{n-j}\right)\left(1-p-\epsilon-\frac{t_1}{n}\right)$, and the j+1-st sample in $W_{q,y}$ is distributed Bernoulli with probability of success $\left(1+\frac{j}{n-j}\right)\left(1-p+\epsilon-\frac{t_1}{n}\right)$. It is easy to check that these two probabilities are in the interval $[1-p-2\epsilon, 1-p+2\epsilon]$ for all $j \le q \le \epsilon n$, and for all $t_1 \le j$. Therefore,

$$h_M^2\left(W_{q,x}, W_{q,y}\right) \;\le\; h^2\left(B(1-p-2\epsilon), B(1-p+2\epsilon)\right) \;\le\; \frac{4\epsilon^2}{p(1-p)}.$$

When $p < \epsilon$, we pick x as before and $y = 1^n$, and when $p > 1-\epsilon$, we pick y as before and $x = 0^n$. It is easy to check that in both cases $h_M^2\left(W_{q,x}, W_{q,y}\right) \le O(\epsilon)$ for $q \le \epsilon n$, implying the lower bound. □

4.4.4 Frequency distribution estimation

The frequency distribution function is $\mathrm{FD} : [m]^n \to S_m$, $\mathrm{FD}(x) = U_x$, where S_m is the m-dimensional simplex. We use additive approximation with respect to total variation distance: $C_{\mathrm{FD},\epsilon}(x) = \{\mu \in S_m \mid V(U_x, \mu) < \epsilon\}$.

The best sampling upper bound for this problem is $O\left(1/\epsilon^2 \cdot m \cdot log(m/\delta)\right)$, following from our general upper bound presented in Section 4.5. Using the Hellinger distance technique one can obtain a $\Omega\left(1/\epsilon^2 \cdot \log(1/\delta)\right)$ lower bound, which gives the tight dependence on ϵ and on δ, but not on m. The main problem here is that the function has many pairwise disjoint inputs. We thus need to resort to the information-theoretic technique to obtain a tighter bound. We note that Batu *et al.* [21] showed an $\Omega(m)$ sampling lower bound for this problem.

Theorem 4.37 *For every $0 \le \epsilon \le \frac{1}{4}$ and $\frac{2}{n} < \delta \le \frac{1}{2}$, if $n \ge \Omega\left(m^2 \cdot 1/\epsilon^4 \cdot 1/\delta\right)$, then*

$$Q_{\epsilon,\delta}^W(\mathrm{FD}) \;\ge\; \Omega\left(m \cdot \frac{1}{\epsilon + O(1)} \cdot (1-4\epsilon) \cdot (1-2\delta)\right).$$

Proof: In order to construct a set $\mathcal{S}$ of pairwise disjoint inputs, we use *combinatorial designs*. An (m, d, k)-design is a family of subsets $S_1, \ldots, S_t \subseteq [m]$, of size d each, such that the intersection of any two of them is of size at most k. A probabilistic method argument can show that for any given m and d there exists an (m, d, k)-design of size $t = 2^{\frac{c_1 d^2}{m}}$ with $k = \frac{c_2 d^2}{m}$, where c_1, c_2 are universal constants (cf. [97]).

We will use an $(m, d, d(1-2\epsilon))$ design of size t with $d = m(1-2\epsilon)/c_2$ and $t = 2^{m \cdot c_1 (1-2\epsilon)^2 / c_2^2}$. For simplicity of exposition we assume $d|n$. Each set S_i in the design defines an input $x_i \in [m]^n$ in the natural way: x_i contains n/d occurrences of each element in S_i. U_{x_i} is just the uniform distribution on S_i. $\mathcal{S}$ is the family of all such inputs: $\mathcal{S} = \{x_i\}_{i \in [t]}$.

Clearly, for all $i \neq \ell \in [t]$, $V(U_{x_i}, U_{x_\ell}) \geq U_{x_i}(S_i \setminus S_\ell) \geq 2\epsilon$, implying that x_i, x_ℓ are ϵ-disjoint. We next calculate the Jensen-Shannon divergence of $\{U_{x_i}\}_{i \in [t]}$. We denote by I a random choice of $i \in [t]$ and by U_{x_I} the average distribution of U_{x_i} when i is chosen at random. U_{x_I} is a distribution on $[m]$, having probability $f_j/(td)$ for each $j \in [m]$, where f_j is the number of sets in the design that contain j. Now,

$$
\begin{aligned}
JS_I\left(\{U_{x_i}\}_{i \in [t]}\right) &= \frac{1}{t}\sum_{i=1}^{t} D_{KL}\left(U_{x_i} \parallel U_{x_I}\right) = \frac{1}{t}\sum_{i=1}^{t}\sum_{j \in S_i} \frac{1}{d} \log \frac{1/d}{f_j/(td)} \\
&= \sum_{j=1}^{m} \frac{f_j}{td} \log \frac{td}{f_j} + \log \frac{1}{d} = H\left(U_{x_I}\right) + \log \frac{1}{d} \leq \log \frac{m}{d}.
\end{aligned}
$$

Applying now Theorem 4.30 and noting that $\log(m/d) \leq \log(c_2/(1-2\epsilon)) \leq \log c_2 + 4\epsilon$ derives the lower bound. $\square$

4.4.5 Extractor and disperser lower bounds

Using the lower bounds for the mean and the maximum, we can immediately derive the lower bounds on the seed length of extractors and dispersers originally proved by Radhakrishnan and Ta-Shma [104]. We first recall the definitions of dispersers and extractors. In the definitions the *min-entropy* of a distribution μ on a set $\mathcal{X}$ is $\min_{x \in \mathcal{X}} \log(1/\mu(x))$. U_r denotes the uniform distribution on $\{0,1\}^r$.

Definition 4.38 (Disperser [116]) *$E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ϵ)-disperser, if for all distributions μ on $\{0,1\}^n$ of min-entropy at least k and for every subset $S \subseteq \{0,1\}^m$ of size at least $\epsilon 2^m$, $\Pr_{X \sim \mu, Y \sim U_d}(E(X,Y) \in S) > 0$.*

Definition 4.39 (Extractor [98]) *$E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ϵ)-*extractor*, if for all distributions μ on $\{0,1\}^n$ of min-entropy at least k, $V(E(\mu, U_d), U_m) < \epsilon$.*

Zuckerman [129] showed that any extractor implies a sampling algorithm for the mean:

Lemma 4.40 (Zuckerman) *If there exists a (k, ϵ)-extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, then there exists a sampling algorithm T of worst-case query complexity 2^d that $(\epsilon, 1/2^{n-k-1})$-approximates the mean $\mu_1 : [0,1]^{2^m} \to [0,1]$.*

A similar argument can show that any disperser implies a sampling algorithm for the maximum:

Lemma 4.41 *If there exists a (k, ϵ)-disperser $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, then there exists a sampling algorithm T of worst-case query complexity 2^d that $(\epsilon, 1/2^{n-k})$-approximates the maximum* $\max : \mathbb{R}^{2^m} \to \mathbb{R}$.

Proof: Given the disperser E, our sampling algorithm T for approximating max will work as follows: T picks a random $z \in \{0,1\}^n$, queries the input $x \in \mathbb{R}^{2^m}$ in the locations designated by $\{E(z,r)\}_{r \in \{0,1\}^d}$, and outputs the maximum of the values obtained. Clearly, T has worst-case query complexity 2^d. We next show it indeed $(\epsilon, 1/2^{n-k})$-approximates max.

Suppose not. Then there exists an input $x \in \mathbb{R}^{2^m}$ for which with probability at least $1/2^{n-k}$ over the choice of $z \in U_n$, $\max_{r \in \{0,1\}^d} x_{E(z,r)}$ is not one of the $\epsilon 2^m$ biggest elements of x. Let us denote by $S \subseteq \{0,1\}^m$ the set of indices of these $\epsilon 2^m$ biggest arguments. We thus have:

$$\Pr_{Z \sim U_n}\left(\Pr_{R \sim U_d}(E(Z,R) \in S) = 0\right) \geq \frac{1}{2^{n-k}}.$$

Therefore, if we denote by $A \subseteq \{0,1\}^n$ the set of z's for which $\Pr_{R \sim U_d}(E(z,R) \in S) = 0$, we have $|A| \geq 2^n/2^{n-k} = 2^k$. Therefore, the uniform distribution on A, which we denote by U_A, has min-entropy k and satisfies: $\Pr_{X \sim U_A, R \sim U_d}(E(X,R) \in S) = 0$. This is a contradiction to E being a (k, ϵ)-disperser. □

Using these connections and our lower bounds for the mean and the maximum, we can obtain a simple proof for extractor and disperser seed length lower bounds:

Theorem 4.42 *For all $n, m \leq n$, $k \leq n-3$, and $0 < \epsilon \leq 1/\sqrt{18}$,*

1. *For every (k, ϵ)-disperser $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, $d \geq \log(n-k) + \log \frac{1}{\epsilon} - O(1)$.*

2. For every (k, ϵ)-extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, $d \geq \log(n-k) + 2\log\frac{1}{\epsilon} - O(1)$.

Proof: Using Lemma 4.41, the existence of a (k, ϵ)-disperser E implies the existence of a sampling algorithm T of worst-case query complexity 2^d that $(\epsilon, 1/2^{n-k})$-approximates $\max : \mathbb{R}^{2^m} \to \mathbb{R}$. Our sampling lower bound for the maximum (Theorem 4.36) implies that $2^d \geq \Omega(\frac{1}{\epsilon}\log(2^{n-k})) = \Omega(\frac{1}{\epsilon}(n-k))$, which derives part (1) of the theorem.

Similarly, Lemma 4.40 implies that the existence of a (k, ϵ)-extractor E yields a sampling algorithm T that $(\epsilon, 1/2^{n-k-1})$-approximates $\mu_1 : [0,1]^{2^m} \to [0,1]$ with 2^d queries. Our lower bound for the mean (Theorem 4.32) implies that $2^d \geq \Omega(\frac{1}{\epsilon^2}\log(2^{n-k-1})) = \Omega(\frac{1}{\epsilon^2}(n-k-1))$, which derives part (2) of the theorem. □

4.5 Tightness of the bounds

In this section, we discuss the tightness of our bounds and the relations among them. First, we compare the different statistical lower bounds:

Proposition 4.43 *For all $\epsilon \geq 0$, for every ϵ-symmetric function $f : \mathcal{X}^n \to \mathcal{Z}$, and for every pair of ϵ-disjoint inputs x, y,*

$$\frac{1}{\sqrt{D_{KL}\left(U_x \parallel U_y\right)}} \leq \frac{1}{V\left(U_x, U_y\right)} \leq \frac{1}{h^2\left(U_x, U_y\right)} = \Theta\left(\frac{1}{JS\left(U_x, U_y\right)}\right).$$

Thus, the Hellinger lower bounds are the best, and they are comparable to the Jensen-Shannon lower bound. The KL distance lower bound can be very bad, but can also be as good as the Hellinger bound. The proof of the proposition follows from the connections among the various distance measures described in Section 2.4.1.

We next prove that for symmetric functions the statistical lower bounds are always at least as strong as the lower bound via block sensitivity:

Proposition 4.44 *For all $\epsilon \geq 0$ and for every ϵ-symmetric function $f : \mathcal{X}^n \to \mathcal{Z}$,*

$$\max\left\{\frac{1}{V\left(U_x, U_y\right)} \mid x, y \ \epsilon\text{-disjoint}\right\} \geq bs_\epsilon(f).$$

Proof: Let x be the input that achieves the block sensitivity of f, and let $I \subseteq [n]$ be the largest block to which f is ϵ-sensitive on x. Note that $bs_\epsilon(f) = bs_\epsilon(f, x) \leq n/|I|$.

Let $Q \in \mathcal{X}^{|I|}$ be the assignment to I such that x and $y = x^{(I \leftarrow a)}$ are ϵ-disjoint. Since x and y differ in $|I|$ positions, then $V\left(U_x, U_y\right) \leq |I|/n$. □

We next prove that the statistical lower bounds obtained in Section 4.2 are reasonably tight for functions over small finite domains (e.g., functions over a Boolean domain). The proof, due to Luca Trevisan [119], improves a slightly weaker bound we achieved before. The upper bound works only for functions associated with approximation notions satisfying the "sunflower property":

Definition 4.45 (Sunflower property) *An approximation $C_{f,\epsilon}$ is said to satisfy the* sunflower property, *if for every $\mathcal{S} \subseteq \mathcal{X}^n$, the following condition holds:*

$$\forall x, x' \in \mathcal{S},\ \ C_{f,\epsilon}(x) \cap C_{f,\epsilon}(x') \neq \emptyset \ \Rightarrow \ \bigcap_{x \in \mathcal{S}} C_{f,\epsilon}(x) \neq \emptyset.$$

Note that relative, ratio, rank, and property testing approximations satisfy the sunflower property. Additive approximation also satisfies it if the range of the function is $\mathbb{R}$.

Theorem 4.46 *If $f : \mathcal{X}^n \to \mathcal{Z}$ is an ϵ-symmetric function with approximation $C_{f,\epsilon}$ that satisfies the sunflower property, then there exist ϵ-disjoint inputs x, y such that:*

$$Q^W_{\epsilon,\delta}(f) \leq O\left(|\mathcal{X}| \frac{1}{V^2\left(U_x, U_y\right)} \log \frac{1}{\delta}\right).$$

Proof: Let x, y be the inputs with the minimum $V\left(U_x, U_y\right)$ among all pairs of ϵ-disjoint inputs. Let $t \stackrel{\text{def}}{=} V\left(U_x, U_y\right)$.

Given input z, the algorithm for approximating f will try to estimate U_z, the frequency distribution of z, in total variation distance to within an additive factor of $\beta = t/18$.

The algorithm constructs an estimate distribution ν_z by picking q independent samples $z_1, \ldots, z_q$ uniformly at random from z, and computing for all $a \in \mathcal{X}$: $\nu_z(a) = (1/q)|\{i \mid x_i = a\}|$. We choose $q = |\mathcal{X}|/\beta^2$.

Define $e_a = |\nu_z(a) - U_z(a)|$, the error made on a; we bound $\mathrm{E}\left[e_a\right]$ as follows: $\mathrm{E}\left[e_a\right]^2 \leq \mathrm{E}\left[e_a^2\right] = \mathrm{E}\left[|\nu_z(a) - U_z(a)|^2\right] = \mathrm{Var}\left[\nu_z(a)\right]$. Define for all $i \in [q]$, $Z_i = 1$, if $z_i = a$ and $Z_i = 0$, otherwise. Then, $\nu_z(a) = \frac{1}{q}\sum_{i\in[q]} Z_i$. We obtain $\mathrm{E}\left[e_a\right]^2 \leq \mathrm{Var}\left[\frac{1}{q}\sum_{i\in[q]} Z_i\right] = \frac{1}{q^2}\sum_{i\in[q]} \mathrm{Var}\left[Z_i\right] = \frac{1}{q^2} \cdot q \cdot U_z(a)(1 - U_z(a)) \leq U_z(a)/q$. Now, if $U_z(a) \leq 1/|\mathcal{X}|$, then $\mathrm{E}\left[e_a\right] \leq \sqrt{(1/|\mathcal{X}|)(\beta^2/|\mathcal{X}|)} = \beta/|\mathcal{X}|$. If $U_z(a) > 1/|\mathcal{X}|$, then $\mathrm{E}\left[e_a\right] \leq \sqrt{U_z(a)(\beta^2/|\mathcal{X}|)} \leq \sqrt{U_z(a)^2\beta^2} = \beta U_z(a)$. Thus, $\forall a \in \mathcal{X}$, $\mathrm{E}\left[e_a\right] \leq \max\{\beta/|\mathcal{X}|, \beta U_z(a)\}$.

Note that $V(\nu_z, U_z) = \frac{1}{2}\sum_{a\in\mathcal{X}} e_a$. Thus,

$$\begin{aligned}
\mathrm{E}\left[V(\nu_z, U_z)\right] &= \frac{1}{2}\sum_{a\in\mathcal{X}} \mathrm{E}\left[e_a\right] \;\le\; \frac{1}{2}\left(\sum_{a:U_z(a)\le\frac{1}{|\mathcal{X}|}} \frac{\beta}{|\mathcal{X}|} + \sum_{a:U_z(a)>\frac{1}{|\mathcal{X}|}} \beta U_z(a)\right) \\
&\le \frac{1}{2}(|\mathcal{X}|\cdot(\beta/|\mathcal{X}|) + \beta\cdot 1) \;=\; \beta.
\end{aligned}$$

Therefore, by Markov's inequality, $\Pr\left(V(\nu_z, U_z) \ge 3\beta\right) \le 1/3$.

The algorithm will generate ℓ independent approximations $\nu_z^1, \ldots, \nu_z^\ell$ for U_z, where $\ell = O(\log(1/\delta))$. We call an approximation ν_z^i "good", if it is at distance of less than 3β from U_z. Chernoff bound implies that with probability at least $1-\delta$, more than half of the approximations are good. Note that by the triangle inequality every two good approximations are at distance of less than 6β from each other. The algorithm thus picks an approximation ν_z^j that is of distance less than 6β from more than half of the other approximations, if one exists. If indeed more than half of the approximations are good, then (1) ν_z^j exists (each good approximation can serve as ν_z^j) and (2) any ν_z^j we pick has at least one good approximation at distance of less than 6β; therefore, ν_z^j is at distance of less than 9β from U_z. We conclude that with probability $1-\delta$ our algorithm finds an approximation ν_z that satisfies $V(\nu_z, U_z) < 9\beta = t/2$.

The algorithm now looks for the set $\mathcal{S}$ of all inputs w for which $V(\nu_z, U_w) < t/2$. What we proved above shows that with probability $1-\delta$, $z \in \mathcal{S}$. Moreover, by the triangle inequality for every two inputs $w, w' \in \mathcal{S}$, $V(U_w, U_{w'}) < t$. It follows from the definition of t that $C_{f,\epsilon}(w) \cap C_{f,\epsilon}(w') \ne \emptyset$. Using now the sunflower property of $C_{f,\epsilon}$, we obtain that there exists a value $b \in \bigcap_{w\in\mathcal{S}} C_{f,\epsilon}(w) \subseteq C_{f,\epsilon}(z)$. The algorithm outputs b. □

For small domains this upper bound can be tight up to a constant factor. The tightness depends on the difference between $V(U_x, U_y)$ and $h^2(U_x, U_y)$. Since this difference is at most quadratic, the difference between the upper and lower bounds is at most quadratic.

4.6 Related work

Most of the computer science literature on the query complexity of sampling algorithms concentrated on the variant of the model in which the functions are Boolean (i.e., $\mathcal{X} = \mathcal{Z} = \{0,1\}$) and the computation is required to be exact. This variant is captured by the famous Boolean decision tree model (see [27] for a recent survey about this area).

The query complexity of deterministic, randomized, and non-deterministic computations of Boolean functions is well understood, and is known to be related to various function properties, like sensitivity and block sensitivity, certificate complexity, and degree of representing and approximating polynomials. The situation for the query complexity of approximation of general functions seems to be very different. For example, it is known that the query complexity of (even randomized) computation of any non-constant Boolean symmetric function is $\Omega(n)$ [102, 125]. On the other hand, the query complexity of approximating some non-Boolean symmetric functions (like the mean and the median) is only constant.

We are aware of only a few lower bounds in the theoretical computer science literature on the query complexity of non-Boolean function approximations; all of these are tailored to specific functions. Canetti *et al.* [31] show a lower bound for additive approximation of the mean; Dagum, Karp, Luby, and Ross [41] (and also, implicitly, Schulman and Vazirani [114]) give lower bounds for relative approximation of the mean on any input x. Charikar *et al.* [36] prove a lower bound for ratio approximation of the frequency moment of order 0. Nayak and Wu [91] give a lower bound on the quantum query complexity of the median and some other statistics.

Sampling algorithms can be viewed as a special case of the general framework studied in statistical decision theory [22]. Classical decision theory takes a point of view, which is somewhat orthogonal to our goals, by seeking ways to get the best approximation from a given set of sample points. In particular, in their setting the algorithm has no control over the samples, their number, or their distribution.

Several sampling lower bounds from sub-areas of statistical decision theory are related to our work. The Cramér-Rao inequality (see, e.g., [121], Chapter 2.4) is a classical result in statistical estimation theory, which provides a lower bound on the variance of any unbiased estimator for an unknown parameter (that is, an estimator whose expectation is the true value of the parameter) in terms of the *Fisher information* of the sample distribution. Statistical learning theory [120, 76, 122] includes lower bounds on the number of samples needed to learn an unknown function in terms of the VC dimension of the hypothesis space. It is not clear whether these techniques can be extended to give lower bounds for general sampling algorithms approximating arbitrary functions.

4.7 Open problems

As mentioned above, Boolean query complexity is known to be related to several function properties, one of which is block sensitivity. In this paper we explored only the generalization of block sensitivity to non-Boolean functions. Similar generalizations may be possible also for some of the other properties. However, it is not clear yet whether such generalizations would yield lower or upper bounds for the query complexity of general function approximation.

Our lower bounds for the expected query complexity, which are based on Hellinger distance, are not tight with respect to the confidence δ. The bound via KL distance is not tight with respect to ϵ and applies only to low query complexities. It is left open to find an optimal lower bound on the expected query complexity (like the one we have for the worst-case query complexity via marginal Hellinger distance).

The information-theoretic lower bound is not tight with respect to ϵ and δ. It is left open to settle the combined dependence on ϵ, δ and the number of pairwise disjoint inputs.

Finally, an important open problem is to find tight lower bounds for the query complexity of non-symmetric functions. Currently, we have only the block sensitivity bound, which seems to be too weak. A particularly interesting class of non-symmetric functions for which a lower bound technique is needed is property testing of graph properties in the adjacency matrix model. Goldreich and Trevisan [57] showed for this class of functions a reduction very reminiscent of ours proving that every sampling algorithm for these problems can be simulated by a sampling algorithm that samples nodes from the graph uniformly at random, and then queries all their incident edges. Goldreich and Trevisan did not investigate, however, whether such a reduction can result in a general lower bound technique.

Chapter 5

One-Way and Simultaneous Communication Lower Bounds

This chapter explores lower bound techniques for two restricted models of communication complexity: *one-way communication* and *simultaneous communication* (see Sections 2.2.4 and 2.2.5 for definitions). Our motivation to study these models arises from their tight connection to data stream computations and sketch computations (see Chapter 3). We illustrate the usefulness of the techniques by several applications, including new data stream space lower bounds for counting distinct elements and approximating selection functions and new sketch lower bounds for estimating the frequency moments.

Distributional communication complexity Our main interest is in lower bounds on the randomized communication complexity of functions. The most pervasive technique of proving such bounds is via *distributional communication complexity* (see Section 2.2): we have to choose some distribution μ on the inputs of the function f, for which we wish to prove the lower bound, and then show a lower bound on the μ-distributional deterministic communication complexity of f. By Yao's Lemma (Lemma 2.6), this lower bound is also a lower bound on the randomized communication complexity of f.

The above technique does not specify how one would pick the distribution μ. Indeed, much of the difficulty in proving randomized communication complexity lower bounds is in finding the "right" distribution μ. A certain class of popular input distributions are *product distributions*—distributions in which the input arguments are chosen independently of each other. This independence usually makes the analysis relatively simple, which is the reason for the popularity of product distributions.

One-way communication lower bounds In Section 3.3.3 we showed that lower bounds for the multi-party one-way communication model yield space lower bounds for the data stream model.

Ablayev [2] and Kremer, Nisan, and Ron [79] studied lower bound techniques for the two-party one-way communication model. They showed that every Boolean function $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ has a product distribution μ over its inputs, for which the μ-distributional one-way communication complexity of f is at least the "VC-dimension of f" (see exact definition in Section 5.2). Thus, in order to prove a randomized one-way communication lower bound for f, it is sufficient to show a lower bound on the VC-dimension of f, which is usually easy to do. Kremer *et al.* [79] further proved that the VC-dimension property almost fully characterizes the distributional one-way communication complexity with respect to product distributions: the μ-distributional communication complexity of f for any product μ is at most a constant times the VC-dimension of f.

We give new, information-theoretic, proofs of both results. Our lower bound is stated in terms of *shatter coefficients*—a notion that generalizes VC-dimension. As a result, our lower bound is sometimes strictly stronger than the one of [2, 79]. We also point out an extension of the lower bound to non-Boolean functions and even to approximation of functions.

We then turn to study multi-party one-way communication complexity. Since the multi-party model is weaker than the two-party model, it is conceivable that for some problems in the data stream model strong lower bounds can be achieved only via direct lower bounds for multi-party one-way communication. (Indeed, the frequency moments problem, which will be discussed in the next chapter, is such an example.) We generalize the VC-dimension technique to give lower bounds for multi-party one-way communication, and then show that the VC-dimension property characterizes also the distributional multi-party one-way communication complexity with respect to product distributions. Consequently, we conclude that the two-party and the multi-party models are equivalent for product distributions. Thus, in order to prove stronger lower bounds for multi-party one-way communication, one has to consider non-product distributions.

Relations between simultaneous communication and one-way communication Exploring the relations between the one-way model and the simultaneous model may be beneficial in reducing lower bounds in one model to lower bounds in the other. Kremer *et al.* [79] showed the two-party simultaneous model is not stronger than the two-party one-way model. Furthermore, they claimed the two models are in fact equivalent. We prove that the latter statement is false, by presenting a problem that exhibits an exponential gap between the two models. We prove that the equivalence holds for distributional communication complexity with respect to product distributions. We generalize the equivalence to the multi-party case as well.

Lower bounds for Multi-Party Set-Disjointness in the simultaneous model One of the most fundamental problems studied in communication complexity is Set Disjointness: Alice and Bob are given each a subset of $[n]$ and are to determine whether the sets intersect or are disjoint. The problem is important, since it reduces to many other problems, yielding a multitude of lower bounds. Its communication complexity was intensively studied [127, 7, 73, 109, 4]; Kalyanasundaram and Schnitger [73] proved an optimal $\Omega(n)$ lower bound on the randomized communication complexity of the problem (Razborov [109] later simplified their proof). Alon, Matias, and Szegedy [4] studied the generalization of the problem to t parties: each of the t players gets a subset of $[n]$ and the subsets are guaranteed to be either pairwise disjoint or uniquely intersecting; the players have to determine which of the two occurs. Alon *et al.* [4] generalized Razborov's proof to obtain an $\Omega(n/t^4)$ lower bound for the problem. This bound is not optimal, since the best upper bound is $O(n/t)$. Alon *et al.* [4] used the bound to obtain space lower bounds for data stream computation of the frequency moments.

We show an optimal $\Omega(n/t)$ lower bound for Multi-Party Set-Disjointness in the simultaneous communication model. This bound allows us to obtain optimal sketch lower bounds for the estimation of the frequency moments. In the next chapter we improve the general communication lower bound of Multi-Party Set-Disjointness to $\Omega(n/t^2)$ and the one-way communication lower bound to $\Omega(n/t^{1+\epsilon})$ for arbitrarily small ϵ.

Methods Methodologically, all the results presented in this chapter are obtained via a novel view of the communication complexity model in the framework of information theory. This view enables direct application of powerful information theory tools and results, such as Fano's inequality and the MLE principle, to study communication complexity. The resulting proofs are in many cases simpler and more intuitive than previous proofs that were based on the traditional combinatorial rigor.

The chapter is organized as follows. We start (Section 5.1) with a description of the new information theory view of communication complexity. We then present the lower bounds for one-way communication (Section 5.2), the relationship between simultaneous and one-way communication (Section 5.3), and the lower bound for Multi-Party Set Disjointness in the simultaneous model (Section 5.4). In Sections 5.5 and 5.6 we present our applications: data stream and sketch lower bounds. We conclude with some open problems in Section 5.7.

5.1 Information theory view of communication complexity

In this section we present a simple view of distributional communication complexity in the classical framework of information theory. This new view allows us to apply information theory machinery, such as Fano's inequality and the MLE principle, in the analysis of communication complexity problems.

Recall the basic framework of information theory described in Section 2.3.2: a sender wishes to communicate a message $x \in \mathcal{X}$ distributed according to μ_X to a receiver through a noisy channel. Since the noise in the channel is random, the receiver gets a message $y \in \mathcal{Y}$ distributed according to μ_Y. The receiver tries to reconstruct x, by applying a function $g : \mathcal{Y} \to \mathcal{X}$ on y. The error incurred is $\delta_g \stackrel{\text{def}}{=} \Pr_{(X,Y)\sim(\mu_X,\mu_y)}(g(Y) \neq X)$. Fano's inequality (Theorem 2.29) gives a lower bound on δ_g for any possible g:

$$H_2(\delta_g) + \delta_g \log(|\mathcal{X}| - 1) \;\;\geq\;\; H(X \mid Y).$$

MLE is a reconstruction function that almost achieves the bound stated in Fano's inequality. The MLE of y is the x that maximizes the conditional probability $\mu_X(x|Y = y)$, where ties are broken arbitrarily. The MLE principle (Theorem 2.30) proves that:

$$\delta_{MLE} \;\leq\; 1 - \frac{1}{2^{H(X|Y)}}.$$

We would like to use this framework to study distributional communication complexity. Let $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ be a function, and let μ be some distribution over inputs of f. We fix Π to

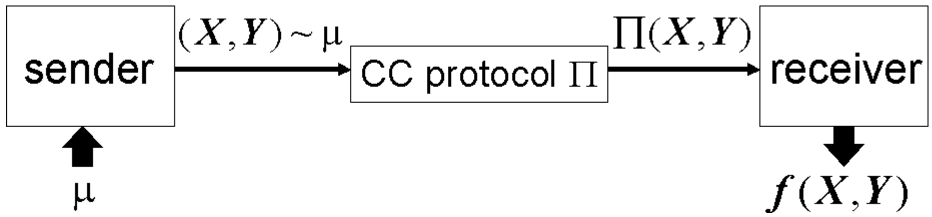

Figure 5.1: Information theory view of distributional communication complexity

be any deterministic protocol that computes f on μ with error probability δ. Our goal is to give lower bounds on the communication cost of Π.

One can view the way Π works as follows (see Figure 5.1). A sender picks an input pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ according to μ and would like to transmit $f(x, y)$ to the receiver. The only communication channel available, though, is the communication complexity protocol Π for f. Thus, what the receiver gets to see is $\Pi(x, y)$—the transcript of messages in Π when running on x, y. The receiver then applies some function g^{Π} to this transcript to deduce $f(x, y)$. Denoting the random inputs chosen according to μ by X and Y, we have $\Pr\left(g^{\Pi}(X,Y) \neq f(X,Y)\right) \leq \delta$. Therefore, by Fano's inequality we obtain the following:

$$H_2(\delta) + \delta \log(|\mathcal{Z}| - 1) \;\; \geq \;\; H\left(f(X,Y) \mid g^{\Pi}(\Pi(X,Y))\right).$$

$H\left(f(X,Y) \mid g^{\Pi}(\Pi(X,Y))\right) \geq H\left(f(X,Y) \mid \Pi(X,Y)\right)$, by the data processing inequality (Corollary 2.28). We therefore have,

Proposition 5.1 $H_2(\delta) + \delta \log(|\mathcal{Z}| - 1) \geq H\left(f(X,Y) \mid \Pi(X,Y)\right).$

Note that in the case f is a Boolean function (i.e., $\mathcal{Z} = \{0,1\}$), we simply have:

$$H_2(\delta) \geq H\left(f(X,Y) \mid \Pi(X,Y)\right).$$

When Π is a one-way protocol then $\Pi(x,y) = A(x) \circ B(A(x), y)$, where $A(x)$ is the message Alice sends on input x and $B(m, y)$ is Bob's announced output on a given message m from Alice and an input y. We can therefore use the data processing inequality to obtain the following simpler form of Fano's inequality in this case:

$$H_2(\delta) \;\; \geq \;\; H\left(f(X,Y) \mid A(X), Y\right).$$

When Π is a simultaneous protocol then $\Pi(x,y) = (A(x), B(y))$, where $A(\cdot)$ and $B(\cdot)$ are the functions Alice and Bob apply, respectively, to their inputs to generate the messages they send to the referee. Thus in this case Fano's inequality takes the following form:

$$H_2(\delta) \;\geq\; H(f(X,Y) \mid A(X), B(Y)).$$

Fano's inequality thus gives a lower bound on the error of δ in terms of the conditional entropy of f given the transcript of messages. The main idea is then to somehow derive a lower bound on the entropy of $\Pi(X,Y)$. Having done that we can use the following simple fact to conclude a lower bound on the communication cost of Π:

Proposition 5.2 $CC(\Pi) \geq H(\Pi(X,Y))$.

The proof is straightforward and is based on our convention that all transcripts are of the same length and the fact that the entropy of a distribution is at most the logarithm of the size of the distribution's domain (Proposition 2.13).

Unfortunately, there is no general recipe for how to move from the lower bound given by Fano's inequality to a lower bound on the entropy of $\Pi(X,Y)$. We were able to do so only in the cases where Π was either a one-way protocol or a simultaneous protocol.

The MLE principle implies that for any protocol Π computing a Boolean function f, the best thing an observer who monitors the communication and would like to deduce the value of the function can do is to use the MLE function. Thus, we can assume that the error δ of Π is bounded as follows:

Proposition 5.3 $\delta \leq H(f(X,Y) \mid \Pi(X,Y))$.

The discussion above easily generalizes to deal with multi-party protocols as well.

5.2 One-way communication complexity lower bounds

In this section we study the randomized communication complexity of two-party and multi-party one-way protocols. The two-party variant was studied by Ablayev [2] and Kremer *et al.* [79], who gave a general lower bound for any Boolean function in terms of the VC-dimension of the function's matrix. They also proved that this bound fully characterizes the distributional one-way communication complexity with respect to product distributions.

We present new, simpler, proofs of the lower bound and of the matching upper bound, based on the information theory view of communication complexity discussed in Section 5.1. Our lower bound is stated in terms of *shatter coefficients*—a notion that generalizes VC-dimension; we show that for some functions the resulting lower bounds are tighter than what could be obtained via VC-dimension.

We then generalize the VC-dimension characterization of distributional one-way communication complexity with respect to product distributions to an arbitrary number of players. As a consequence, we prove that the two-party model and the multi-party model with worst-case partition are equivalent for product distributions. We exhibit non-product distributions for which there exists an unbounded gap between the two models.

5.2.1 Two-party one-way communication lower bounds

We start with the definitions of VC-dimension and shatter coefficients, in terms of which we state the bounds for one-way communication complexity:

Definition 5.4 (VC-dimension [123] and shatter coefficients) *Given a family of Boolean functions $\mathcal{H} = \{h : \mathcal{X} \to \{0,1\}\}$, we view each of the functions as a binary string of length $|\mathcal{X}|$ (the string corresponding to h is the truth table of h). The* shatter coefficient of a subset *$\mathcal{S} \subseteq \mathcal{X}$ is the cardinality of the set $\{h|_{\mathcal{S}}\}_{h\in\mathcal{H}}$. $\mathcal{S}$ is said to be* shattered *by $\mathcal{H}$, if its shatter coefficient is $2^{|\mathcal{S}|}$. The The* Vapnik-Chervonenkis (VC) dimension *of $\mathcal{H}$, $\textsc{vcd}(\mathcal{H})$, is the maximum size of a subset $\mathcal{S}$ that is shattered by $\mathcal{H}$. The* ℓ-th shatter coefficient of *$\mathcal{H}$, $\textsc{sc}(\mathcal{H},\ell)$, is the largest shatter coefficient of $\mathcal{S}$ over all subsets $\mathcal{S}$ of size ℓ.*

Note that $\textsc{vcd}(\mathcal{H})$ is the maximum ℓ for which $\textsc{sc}(\mathcal{H},\ell) = 2^{\ell}$.

Example Figure 5.2 shows an example of a family $\mathcal{H} = \{h : [4] \to \{0,1\}\}$ of Boolean functions, specified as rows of a $|\mathcal{H}| \times 4$ Boolean matrix. The largest set of columns that are shattered by $\mathcal{H}$ is $\{1,3\}$, and therefore the VC dimension of $\mathcal{H}$ is 2. Thus, the second shatter coefficient of $\mathcal{H}$ is 4. The third shatter coefficient, for example, is 6, because the set of columns $\{1,3,4\}$ contain 6 patterns.

Consider a function $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ to be computed by a two-party one-way communication protocol. For every possible input $x \in \mathcal{X}$ for Alice, we define $f_x : \mathcal{Y} \to \{0,1\}$ to be the function induced by this input; that is, $f_x(y) = f(x,y)$. $f_{\mathcal{X}}$ is the family of functions $\{f_x \mid x \in \mathcal{X}\}$. Kremer *et al.* [79] showed that for every $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$, there exists a product distribution μ on $X \times Y$ such that $D^{\rightarrow}_{\mu,\delta}(f) \geq \textsc{vcd}(f_{\mathcal{X}}) \cdot (1 - O(\delta))$. By Yao's Lemma (Lemma 2.6) this gives also a lower bound on the randomized two-party one-way

1	0	**1**	0
1	1	1	1
0	1	**0**	1
0	0	**1**	0
1	0	**0**	0
0	1	0	0

Figure 5.2: VC-dimension and shatter coefficients

communication complexity. We next present a simple argument, based on the information-theoretic view of communication complexity, to prove a more general lower bound in terms of shatter coefficients:

Theorem 5.5 (Lower bound for 2-party 1-way communication complexity) *For every function* $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$*, every* $\ell \geq \textsc{vcd}(f_{\mathcal{X}})$*, and every* $0 < \delta < 1$*, there exists a product distribution* μ *on* $\mathcal{X} \times \mathcal{Y}$*, such that:*

$$D^{\rightarrow}_{\mu,\delta}(f) \;\geq\; \log\left(\textsc{sc}(f_{\mathcal{X}}, \ell)\right) - \ell \cdot H_2(\delta)\,.$$

Note that by taking $\ell = \textsc{vcd}\,(f_{\mathcal{X}})$ we get exactly the lower bound of Kremer *et al.* [79] as a special case.

Proof: Let M_f be the function's matrix such that $M_f(x,y) = f(x,y)$. Let $\mathcal{X}' \subseteq \mathcal{X}$ and $\mathcal{Y}' \subseteq \mathcal{Y}$, where $|\mathcal{Y}'| = \ell$ and $|\mathcal{X}'| = \textsc{sc}\,(f_{\mathcal{X}}, \ell)$, be such that M_f restricted to $\mathcal{X}' \times \mathcal{Y}'$ consists of $\textsc{sc}\,(f_{\mathcal{X}}, \ell)$ distinct rows. In other words, $|\{f_x|_{\mathcal{Y}'} \mid x \in \mathcal{X}'\}| = \textsc{sc}\,(f_{\mathcal{X}}, \ell)$.

Consider the product distribution $\mu = \mu_X \times \mu_Y$, for which μ_X is uniform on $\mathcal{X}'$ and μ_Y is uniform on $\mathcal{Y}'$. Let X and Y denote random elements of $\mathcal{X}'$ and $\mathcal{Y}'$ chosen according to μ_X and μ_Y respectively. Consider a one-way deterministic protocol for f of cost C and error at most δ on μ, in which $A = A(X)$ is the message sent by Alice to Bob. Bob estimates the output of the function given $A(X)$ and Y. Therefore, by Fano's inequality (Theorem 2.29),

$$\begin{aligned} H_2(\delta) \;&\geq\; H\left(f(X,Y) \mid A(X), Y\right) \;=\; \sum_{y \in \mathcal{Y}'} \mu_Y(y) \cdot H\left(f(X,Y) \mid A(X), Y = y\right) \\ &=\; \frac{1}{\ell} \sum_{y \in \mathcal{Y}'} H\left(f(X,y) \mid A(X), Y = y\right). \end{aligned}$$

Note that both $f(X,y)$ and $A(X)$ are random variables that depend only on X since y is a fixed element of $\mathcal{Y}'$. Since X and Y are independent, we can apply Proposition

2.22 to infer that $H\left(f(X,y) \mid A(X), Y=y\right) = H\left(f(X,y) \mid A(X)\right)$. Using this fact in the equation above we obtain:

$$\begin{aligned}\ell \cdot H_2(\delta) &\geq \sum_{y\in\mathcal{Y}'} H\left(f(X,y) \mid A(X)\right) \;\geq\; H\left(\{f(X,y)\}_{y\in\mathcal{Y}'} \mid A(X)\right)\\ &\geq H\left(\{f(X,y)\}_{y\in\mathcal{Y}'}\right) - H\left(A(X)\right),\end{aligned}$$

where we applied the entropy sub-additivity (Proposition 2.15) and the entropy chain rule (Proposition 2.21).

As X ranges over the values in $\mathcal{X}'$ the (ordered) set $\{f(X,y)\}_{y\in\mathcal{Y}'}$ takes on $\textsc{sc}\left(f_{\mathcal{X}}, \ell\right)$ distinct values, each with equal probability. Therefore, $H\left(\{f(X,y)\}_{y\in\mathcal{Y}'}\right) = \log \textsc{sc}\left(f_{\mathcal{X}}, \ell\right)$. Moreover, $H\left(A(X)\right) \leq C$. Thus, $\ell \cdot H_2(\delta) \geq \log \textsc{sc}\left(f_{\mathcal{X}}, \ell\right) - C$ which yields the lower bound. □

Advantage over the VC-dimension technique We next show that for a restricted version of the index function $f : \mathcal{X} \times [n] \to \{0,1\}, f(x,i) = x_i$, $\mathcal{X} = \{w \in \{0,1\}^n : |w| \leq 2\delta n\}$ ($|w|$ is the Hamming weight of w), our bound yields a tighter dependence on δ. $\textsc{vcd}\left(f_{\mathcal{X}}\right) \leq 2\delta n$, because the pattern $1^{2\delta n+1}$ cannot be obtained by $f_x|_{\mathcal{S}}$ for any $x \in \mathcal{X}$ and any $\mathcal{S} \subseteq [n]$. Therefore, the *Kremer et al.* [79] lower bound is at most $2\delta n\left(1 - O(\delta)\right)$. On the other hand, consider the n-th shatter coefficient of $f_{\mathcal{X}}$. Since all the rows of M_f are distinct, $\textsc{sc}\left(f_{\mathcal{X}}, n\right) = \sum_{i=0}^{2\delta n}\binom{n}{i}$, which by Stirling's approximation is at least $2^{nH_2(2\delta)}/O(\sqrt{n})$. Therefore, our lower bound is at least $nH_2(2\delta) - nH_2(\delta) - O(\log n) = n\left(H_2(2\delta) - H_2(\delta)\right) - O(\log n) = \Omega\left(n\delta\log(1/\delta)\right)$ for $\delta < 1/2$.

Extension to partial domains The lower bound was stated for functions over a full domain. It can be easily extended to hold also for functions over a partial domain: $f : \mathcal{T} \to \{0,1\}$, where $\mathcal{T} \subseteq \mathcal{X} \times \mathcal{Y}$. For such functions, the entries of the function's matrix M_f may consist of three possible values: $1, 0$, or $*$, where $*$ is assigned to entries (x,y) for which $(x,y) \notin \mathcal{T}$. In order to calculate the shatter coefficients and the VC-dimension of M_f, we will consider for each given set of columns $\mathcal{S}$ only the rows of M_f, for which the restriction to $\mathcal{S}$ does not consist of any $*$ entry. Thus, $\mathcal{S}$ is shattered, only if all the $2^{|\mathcal{S}|}$ 0-1 patterns are obtained in $\mathcal{S}$, possibly in addition to other patterns involving $*$'s. With these definitions, the same lower bound holds.

Extension to non-Boolean functions and to approximations The technique can be generalized to deal with functions with a non-Boolean range, and even with approximations of functions. At a high level, the idea is as follows: each entry (x, y) of the function's matrix M_f consists of the approximation set $C_{f,\epsilon}(x, y)$ (if we deal with exact computations this set contains only $f(x, y)$). For a given collection $\mathcal{S}$ of ℓ columns of M_f, two restrictions of rows of M_f to $\mathcal{S}$ are called comparable, if for each column $i \in \mathcal{S}$, the approximation sets occurring in the i-th coordinate of these rows are either identical or disjoint. Two comparable rows are different, if there is at least one coordinate, in which their approximation sets are disjoint. The shatter coefficient of $\mathcal{S}$ is the maximum number of rows the restrictions of which to $\mathcal{S}$ are pairwise different. The ℓ-th shatter coefficient of M_f is the maximum shatter coefficient over all collections of ℓ columns. With this definition, and using the general form of Fano's inequality (for non-Boolean messages), one could obtain an almost identical lower bound. This bound generalizes the bound Kremer *et al.* [79] give for non-Boolean functions.

Tightness of the bound Kremer *et al.* [79] showed that in general the bound is not tight: the "greater-than" function has randomized one-way communication complexity of $\Omega(n)$ but the VC-dimension of its matrix is only O(1)! On the other hand, they proved that VC-dimension fully characterizes (up to constant factors and the dependence on δ) the distributional one-way communication complexity with respect to product distributions. It follows that in cases the VC-dimension bound is not tight, one must consider non-product distributions in order to obtain strong lower bounds for the randomized one-way communication complexity.

In order to prove this characterization, Kremer *et al.* [79] used the famous result of Blumer *et al.* [23] connecting PAC learning and VC-dimension. We give a self-contained proof of this bound using the information theory view of communication complexity:

Theorem 5.6 (Characterization of "product 1-way communication" [79]) *For every function $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ with $\mathrm{VCD}\,(f_{\mathcal{X}}) \le |\mathcal{Y}|/2$, every $0 < \delta < 1$, and for every product distribution $\mu = \mu_X \times \mu_Y$ on $\mathcal{X} \times \mathcal{Y}$,*

$$D^{\rightarrow}_{\mu,\delta}(f) \;\le\; \mathrm{VCD}\,(f_{\mathcal{X}}) \cdot O\left((1/\delta)\log(1/\delta)\right).$$

Proof: Let $2 \cdot \mathrm{VCD}\,(f_{\mathcal{X}}) \le m \le |\mathcal{Y}|$ be a parameter to be fixed later. We will first describe a collection of deterministic protocols, each corresponding to a subset $\mathcal{Y}' \subseteq \mathcal{Y}$ of size m. We will then argue that one of these protocols computes f on μ with error at most δ.

For each multiset $\mathcal{Y}'$ consisting of m elements from $\mathcal{Y}$, we define the following protocol. Upon receiving her input x, Alice sends to Bob $A(x, \mathcal{Y}') = \{f(x, y')\}_{y' \in \mathcal{Y}'}$. (If $\mathcal{Y}'$ contain duplicates, this protocol sends redundant information, but we ignore this fact in order to make the analysis simpler.) Bob finds the collection $\mathcal{X}'$ of all $x' \in \mathcal{X}$ that are consistent with Alice's message (that is, $f(x', y') = f(x, y')$ for all $y' \in \mathcal{Y}'$), and outputs the majority value (w.r.t. μ_X) of $\{f(x', y)\}_{x' \in \mathcal{X}'}$. Note that the communication of the protocol is exactly m.

Let X, Y be random inputs chosen according to μ_X, μ_Y respectively. Note that what Bob computes is the MLE of the random variable $f(X, Y)$ given the observations $A(X, \mathcal{Y}')$ and Y. Therefore, by the MLE Principle (Theorem 2.30),

$$\Pr\left(\textsc{mle}\left(A(X, \mathcal{Y}'), Y, \mathcal{Y}'\right) \neq f(X, Y)\right) \;\leq\; H\left(f(X, Y) \mid A(X, \mathcal{Y}'), Y\right).$$

Consider now choosing the set $\mathcal{Y}'$ by picking m elements from $\mathcal{Y}$ independently according to μ_Y. Let S be the corresponding random variable. Since S is independent of X and Y, we obtain,

$$\Pr\left(\textsc{mle}\left(A(X, S), Y, S\right) \neq f(X, Y)\right) \leq H\left(f(X, Y) \mid A(X, S), Y, S\right).$$

Using the chain rule for entropy (Proposition 2.21), we have:

$$H\left(f(X, Y) \mid A(X, S), Y, S\right) = H\left(f(X, Y), A(X, S) \mid Y, S\right) - H\left(A(X, S) \mid Y, S\right). \tag{5.1}$$

By the definition of A and S, $A(X, S) = \left\{f(X, Y^j)\right\}_{j=1}^{m}$, where $Y^1, \ldots, Y^m$ are chosen according to μ_Y. Therefore, the distribution of $f(X, Y)$ is identical to the distribution of $f(X, Y^j)$, $j = 1, \ldots, m$. We can thus rewrite the joint distribution $(f(X, Y), A(X, S))$ as $\left\{f(X, Y^j)\right\}_{j=0}^{m}$, where $Y^0 = Y$. Using the entropy subset inequality (Theorem 2.16), we now obtain:

$$\begin{aligned}
H\left(f(X, Y), A(X, S) \mid Y, S\right) \;&=\; H\left(\left(\left\{f(X, Y^j)\right\}_{j=0}^{m}\right) \mid Y, S\right) \\
\leq \;& (m+1) \cdot \frac{1}{m+1} \cdot \sum_{T \subseteq [m+1], |T| = m} \frac{1}{m} \cdot H\left(\left\{f(X, Y^j)\right\}_{j \in T} \mid Y, S\right) \\
= \;& \frac{1}{m} \cdot (m+1) \cdot H\left(\left\{f(X, Y^j)\right\}_{j=1}^{m} \mid Y, S\right) \;=\; \left(1 + \frac{1}{m}\right) \cdot H\left(A(X, S) \mid Y, S\right).
\end{aligned}$$

Combining with Equation 5.1, we obtain:

$$\Pr\left(\textsc{mle}\left(A(X, S), Y, S\right) \neq f(X, Y)\right) \;\leq\; \frac{1}{m} H\left(A(X, S) \mid Y, S\right).$$

Since conditioning reduces entropy (Proposition 2.19), then:

$$\frac{1}{m}H\left(A(X,S) \mid Y,S\right) \;\le\; \frac{1}{m}H\left(A(X,S) \mid S\right) \;=\; \frac{1}{m}\sum_{\mathcal{Y}'} H\left(A(X,\mathcal{Y}')\right)\Pr\left(S=\mathcal{Y}'\right). \quad (5.2)$$

Consider any choice $\mathcal{Y}'$ for S. Since the entropy is always at most the logarithm of the size of the range (Proposition 2.13), then:

$$\frac{1}{m}H\left(A(X,\mathcal{Y}')\right) \;=\; \frac{1}{m}H\left(\left\{f(X,y')\right\}_{y'\in\mathcal{Y}'}\right) \;=\; \frac{1}{m}H\left(f_X|_{\mathcal{Y}'}\right) \;\le\; \frac{1}{m}\log\left|f_{\mathcal{X}}|_{\mathcal{Y}'}\right|,$$

where $f_{\mathcal{X}}|_{\mathcal{Y}'}$ is the family of functions f_x, $x\in\mathcal{X}$, restricted to the columns in $\mathcal{Y}'$. We now use Sauer's Lemma to bound the size of this family:

Lemma 5.7 (Sauer) *For any family $\mathcal{F}\subseteq\{0,1\}^n$, $|\mathcal{F}|\le\sum_{i=0}^{d}\binom{n}{i}\le(en/d)^d$, where $d=\text{VCD}\left(\mathcal{F}\right)$.*

We thus obtain:

$$\frac{1}{m}\cdot H\left(A(X,\mathcal{Y}')\right) \;\le\; \frac{\text{VCD}\left(f_{\mathcal{X}}|_{\mathcal{Y}'}\right)}{m}\cdot\log\frac{em}{\text{VCD}\left(f_{\mathcal{X}}|_{\mathcal{Y}'}\right)}.$$

Since the function $x\log(1/x)$ is monotone increasing for $x\le 1/2$, $\text{VCD}\left(f_{\mathcal{X}}|_{\mathcal{Y}'}\right)\le\text{VCD}\left(f_{\mathcal{X}}\right)$, and $\text{VCD}\left(f_{\mathcal{X}}\right)/m\le 1/2$, then

$$\frac{1}{m}H\left(A(X,\mathcal{Y}')\right) \;\le\; \frac{\text{VCD}\left(f_{\mathcal{X}}\right)}{m}\cdot\log\frac{em}{\text{VCD}\left(f_{\mathcal{X}}\right)}.$$

By choosing $m=O\left(\text{VCD}\left(f_{\mathcal{X}}\right)/\delta\cdot\log(1/\delta)\right)$, we have $(1/m)H\left(A(X,\mathcal{Y}')\right)\le\delta$. This holds for any choice of $\mathcal{Y}'$, and therefore, substituting in Equation 5.2,

$$\Pr\left(\text{MLE}\left(A(X,S),Y,S\right)\ne f(X,Y)\right) \;\le\; \delta.$$

In particular, there is a choice $\mathcal{Y}'$ for S, for which

$$\Pr\left(\text{MLE}\left(A(X,\mathcal{Y}'),Y,\mathcal{Y}'\right)\ne f(X,Y)\right) \;\le\; \delta.$$

The protocol $(A(\cdot,\mathcal{Y}'),\text{MLE}\left(\cdot,\cdot,\mathcal{Y}'\right))$ is the desired deterministic protocol for f. $\square$

5.2.2 Multi-party one-way communication lower bounds

The VC-dimension lower bound for two parties trivially holds also for multiple parties, because the multi-party model is weaker than the two-party model. The main

1	1	1		1	0	0		1	0	1
0	1	0		0	1	0		0	0	1
1	**0**	**1**		**0**	**0**	**1**		**1**	**0**	**0**

Figure 5.3: Function's hypermatrix and an induced sub-matrix

result of this section is a generalization of the VC-dimension characterization of one-way communication complexity for product distributions to an arbitrary number of players. This implies that in the multi-party model too, if the VC-dimension technique does not produce tight bounds, then one has to consider non-product input distributions. We also obtain as a corollary that the two-party and the multi-party models are equivalent with respect to product distributions.

Let $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \mathcal{Z}$ be any function on t input arguments. We consider all the $t-1$ functions on two arguments obtained by partitioning the input of f into two contiguous parts: for $j = 1, \ldots, t-1$, $f^j : \mathcal{X}^j \times \mathcal{Y}^j \to \mathcal{Z}$, where $\mathcal{X}^j = \mathcal{X}_1 \times \cdots \times \mathcal{X}_j$ and $\mathcal{Y}^j = \mathcal{X}_{j+1} \times \cdots \times \mathcal{X}_t$.

We define for every $x^j \in \mathcal{X}^j$ the induced function $f^j_{x^j} : \mathcal{Y}^j \to \mathcal{Z}$, $f^j_{x^j}(y^j) = f^j(x^j, y^j)$. $f^j_{\mathcal{X}^j}$ is the family of functions $\left\{ f^j_{x^j} \mid x^j \in \mathcal{X}^j \right\}$.

Geometrically, one can view the function f as a t-dimensional $|\mathcal{X}_1| \times \cdots \times |\mathcal{X}_t|$ hypermatrix M_f, whose $(x_1, \ldots, x_t)$ entry consists of $f(x_1, \ldots, x_t)$. The induced function $f^j_{x^j}$ is the sub-matrix of M_f corresponding to $\{x_1\} \times \cdots \times \{x_j\} \times \mathcal{X}_{j+1} \times \cdots \times \mathcal{X}_t$. Figure 5.3 shows an example of a 3-dimensional hypermatrix corresponding to a function $f : [3]^3 \to \{0, 1\}$. The three layers of the cube M_f are laid out side by side. The submatrix of M_f corresponding to f^1_3 is marked in bold face.

We first note that the multi-party one-way model is not stronger than the two-party model:

Proposition 5.8 *For every function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \mathcal{Z}$ and every $0 < \delta < 1$,*

$$R^{\rightarrow}_{\delta}(f) \;\geq\; \max_j R^{\rightarrow}_{\delta}\left(f^j\right).$$

The proof is straightforward; the two players can simulate the t-player protocol as follows: Alice simulates the first j players; it then transmits what the j-th player would have sent to the $j+1$-st player to Bob; Bob completes the computation by simulating the last $t-j$ players.

The result also holds for the μ-distributional complexity for any input distribution μ. It therefore follows from Theorem 5.5 that:

Corollary 5.9 *For every function* $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \{0,1\}$, $\ell \geq \max_j \textsc{vcd}\left(f^j_{\mathcal{X}^j}\right)$, *and* $0 < \delta < 1$, *there exists a distribution* μ *on* $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$, *such that:*

$$D^{\rightarrow}_{\mu,\delta}(f) \;\geq\; \max_j \left(\log\left(\textsc{sc}\left(f_{\mathcal{X}^j}, \ell\right)\right) - \ell \cdot H_2(\delta)\right).$$

Note that as opposed to Theorem 5.5, the distribution μ here is not necessarily product (it is product only with respect to the domain of the function f^j in which the maximum is obtained).

We next prove that this bound is tight for product distributions modulo the dependence on δ and t:

Theorem 5.10 *For every function* $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \{0,1\}$, *every product distribution* μ *on* $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$, *and every* $0 < \delta < 1$,

$$D^{\rightarrow}_{\mu,\delta}(f) \;\leq\; \sum_{j=1}^{t-1} \textsc{vcd}\left(f_{\mathcal{X}^j}\right) \cdot O\left(\frac{t}{\delta} \cdot \log \frac{t}{\delta}\right).$$

Proof: We will construct a protocol Π for t players that uses $t-1$ copies $\Pi_1, \ldots, \Pi_{t-1}$ of the protocol for two players by Kremer *et al.* [79]. Their protocol (which is slightly different from the protocol we described in the proof of Theorem 5.6) uses a fixed set $\mathcal{Y}' \subseteq \mathcal{Y}$. Given her input $x \in \mathcal{X}$, Alice sends to Bob $f_x(y')$ for all $y' \in \mathcal{Y}'$. Bob finds an arbitrary input x' that is consistent with Alice's message (i.e., $f_{x'}(y') = f_x(y')$ for all $y' \in \mathcal{Y}'$) and outputs $f_{x'}(y)$. Using the connection between PAC learning and VC-dimension [23], Kremer *et al.* prove that for any product distribution μ, there exists such a set $\mathcal{Y}'$ of size $\textsc{vcd}(f_{\mathcal{X}}) \cdot O\left((1/\delta) \cdot \log(1/\delta)\right)$, such that the protocol errs with probability at most δ on μ.

For each $j = 1, \ldots, t-1$, Π_j will compute a function $g^j : \mathcal{X}^j \times \mathcal{Y}^j \to \{0,1\}$ to be defined shortly. Note that μ, the given product distribution on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$, can be also viewed as a product distribution on $\mathcal{X}^j \times \mathcal{Y}^j$. Let us denote by $A_j(\cdot)$ the function Alice computes on her input to produce the message to Bob in the protocol Π_j, and by $B_j(\cdot)$ the function Bob applies to the message received from Alice to find a consistent input x'. Π_j will err with probability at most $\delta/(t-1)$ on inputs drawn from μ, meaning that $\Pr_{(X^j,Y^j)\sim\mu}\left(g^j\left(B_j\left(A_j\left(X^j\right)\right), Y^j\right) \neq g^j\left(X^j, Y^j\right)\right) \leq \delta/(t-1)$.

$g^1, \ldots, g^{t-1}$ are defined recursively: $g^1 = f^1$, and for $j = 2, \ldots, t-1$, $g^j\left(x^j, y^j\right) = g^{j-1}\left(B_{j-1}\left(A_{j-1}\left(x^{j-1}\right)\right), y^{j-1}\right)$. That is, g^j is the output of Π_{j-1} when trying to compute g^{j-1} (note that since Π_{j-1} is allowed to err, g^j is not necessarily identical to g^{j-1}). Denote by g^t the output of the last protocol Π_{t-1}.

The protocol Π works as follows: the first player simulates Alice's part in Π_1, and sends $m_1 = A_1(x_1)$ to the second player. The j-th player ($j = 2, \ldots, t-1$), upon receiving a message m_{j-1} from the $j-1$-st player, works in two phases: first he simulates Bob's part in Π_{j-1}, and computes the function $B_{j-1}(m_{j-1})$; he then simulates Alice's part in Π_j, and computes $m_j = A_j\left(B_{j-1}(m_{j-1}) \circ x_j\right)$; he sends m_j to the $j+1$-st player. The last player, upon receiving the message m_{t-1}, simulates Bob's part in Π_{t-1}, and outputs $g^{t-1}\left(B_{t-1}(m_{t-1}), x_t\right)$.

The protocol Π induces a mapping from every input $x \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_t$ to a sequence of $t-1$ input pairs $\left(x^1, y^1\right) \in \mathcal{X}^1 \times \mathcal{Y}^1, \ldots, \left(x^{t-1}, y^{t-1}\right) \in \mathcal{X}^{t-1} \times \mathcal{Y}^{t-1}$. $\left(x^j, y^j\right)$ is the input on which Π runs the simulation of Π_j. $\left(x^1, y^1\right), \ldots, \left(x^{t-1}, y^{t-1}\right)$ are defined recursively: $x^1 = x_1, y^1 = (x_2, \ldots, x_t)$ and for $j = 2, \ldots, t-1$, $x^j = B_{j-1}\left(A_{j-1}\left(x^{j-1}\right)\right) \circ x_j, y^j = (x_{j+1}, \ldots, x_t)$. We also define $x^t = B_{t-1}\left(A_{t-1}\left(x^{t-1}\right)\right) \circ x_t$ and $y^t = \emptyset$. What Π outputs is $g^t\left(x^t, y^t\right)$.

We first prove that Π errs with probability at most δ on inputs X drawn according to μ.

$$\begin{aligned} \Pr\left(\Pi(X) \neq f(X)\right) &= \Pr\left(g^t\left(X^t, Y^t\right) \neq f(X)\right) = \Pr\left(g^t\left(X^t, Y^t\right) \neq g^1\left(X^1, Y^1\right)\right) \\ &\leq \sum_{j=1}^{t-1} \Pr\left(g^{j+1}\left(X^{j+1}, Y^{j+1}\right) \neq g^j\left(X^j, Y^j\right)\right). \end{aligned}$$

For each $j = 1, \ldots, t-1$,

$$\begin{aligned} g^{j+1}\left(x^{j+1}, y^{j+1}\right) &= g^{j+1}\left(B_j\left(A_j\left(x^j\right)\right) \circ x_{j+1}, y^{j+1}\right) \\ &= g^j\left(B_j\left(A_j\left(B_j\left(A_j\left(x^j\right)\right)\right)\right), y^j\right) = g^j\left(B_j\left(A_j\left(x^j\right)\right), y^j\right). \end{aligned}$$

Now, since Π_j computes g^j with error at most $\delta/(t-1)$ on μ, then

$$\Pr\left(g^j\left(B_j\left(A_j\left(X^j\right)\right), Y^j\right) \neq g^j\left(X^j, Y^j\right)\right) \leq \frac{\delta}{t-1}.$$

Thus,

$$\Pr\left(\Pi(X) \neq f(X)\right) \leq \sum_{j=1}^{t-1} \frac{\delta}{t-1} = \delta.$$

In order to bound the communication cost of Π, we prove the following:

Claim 5.10.1 *For every $j = 1, \ldots, t-1$,* $\quad g^j_{\mathcal{X}^j} \subseteq f^j_{\mathcal{X}^j}$.

Proof: We prove the claim by induction on j. The basis, $j = 1$, is trivial, since $g^1 = f^1$. Assume, then, that the claim holds for $j-1$, and we will prove it holds also for j.

Fix any $x^j \in \mathcal{X}^j$, and consider the induced function $g^j_{x^j}$. By definition, for every $y^j \in \mathcal{Y}^j$,

$$g^j_{x^j}\left(y^j\right) \;=\; g^j\left(x^j, y^j\right) = g^{j-1}\left(B_{j-1}\left(A_{j-1}\left(x^{j-1}\right)\right), y^{j-1}\right) \;=\; g^{j-1}_{B_{j-1}(A_{j-1}(x^{j-1}))}\left(y^{j-1}\right).$$

By the definition of A_{j-1}, B_{j-1}, for every $x^{j-1} \in \mathcal{X}^{j-1}$, also $B_{j-1}\left(A_{j-1}\left(x^{j-1}\right)\right) \in \mathcal{X}^{j-1}$. Thus, by the induction hypothesis, there is some $w^{j-1} \in \mathcal{X}^{j-1}$, such that $g^{j-1}_{B_{j-1}(A_{j-1}(x^{j-1}))} = f^{j-1}_{w^{j-1}}$.

Define $w^j = w^{j-1} \circ x_j$. $w^j \in \mathcal{X}^j$, because $\mathcal{X}^j$ is a Cartesian product of $\mathcal{X}^{j-1}$ and $\mathcal{X}_j$. We will show that $g^j_{x^j} = f^j_{w^j}$, which would imply $g^j_{x^j} \in f^j_{\mathcal{X}^j}$.

We showed above that for every $y^j \in \mathcal{Y}^j$, $g^j_{x^j}\left(y^j\right) = f^{j-1}_{w^{j-1}}\left(y^{j-1}\right)$, where $y^{j-1} = x_j \circ y^j$. The claim now follows since $f^{j-1}_{w^{j-1}}\left(y^{j-1}\right) = f\left(w^{j-1} \circ y^{j-1}\right) = f\left(w^j \circ y^j\right) = f^j\left(w^j, y^j\right)$.
$\square$

From Theorem 5.6, the cost of Π_j is at most $\textsc{vcd}\left(g^j_{\mathcal{X}^j}\right) \cdot O\left(t/\delta \cdot \log(t/\delta)\right)$. It follows from Claim 5.10.1 that $\textsc{vcd}\left(g^j_{\mathcal{X}^j}\right) \leq \textsc{vcd}\left(f^j_{\mathcal{X}^j}\right)$. Therefore, the total communication cost of Π is at most $\sum_{j=1}^{t-1} \textsc{vcd}\left(f^j_{\mathcal{X}^j}\right) \cdot O\left(t/\delta \cdot \log(t/\delta)\right)$. $\square$

Equivalence of two-party and multi-party for product distributions The above Theorem combined with the VC-dimension lower bound for the two-party model (Theorem 5.10) imply that the two-party and the multi-party models are equivalent (modulo the dependence on δ and on t) for product distributions:

Corollary 5.11 *For every function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \{0,1\}$, every $0 < \delta < 1$, and every product distribution μ on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$,*

$$D^{\rightarrow}_{\mu,\delta}(f) \;\leq\; \sum_{j=1}^{t-1} D^{\rightarrow}_{\mu,\delta}(f^j) \cdot O\left(\frac{t}{\delta} \cdot \log\frac{t}{\delta} \cdot \frac{1}{1 - H_2\left(\delta\right)}\right).$$

Gap between two-party and multi-party for non-product distributions We now show that for non-product distributions there can be an unbounded gap between the two-party and the multi-party models. The function we consider is the following multi-party version of the disjointness function (cf. [4]): $\textsc{Disj}_{n,t} : (\{0,1\}^n)^t \to \{0,1\}$, where $n = t(2m-1)+1$ for some parameter m. Each player gets a binary string in $\{0,1\}^n$ which we associate with a subset $\mathcal{S} \subseteq [n]$. μ is a distribution on $(\{0,1\}^n)^t$ which is uniform on all collections of sets $\mathcal{S}_1, \ldots, \mathcal{S}_t$ that satisfy the following: (1) $|\mathcal{S}_i| = m$ for all i; (2) either the sets are pairwise disjoint ($\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for all $i \neq j$), or uniquely interesting ($|\bigcap_i \mathcal{S}_i| = 1$ and $|\mathcal{S}_i \cap \mathcal{S}_j| = 1$ for all $i \neq j$). Alon, Matias, and Szegedy [4] proved a lower bound for the general communication complexity of this problem: $D_{\mu,\delta}(f) \geq \Omega\left(n/t^4\right)$ for any small enough constant δ. In particular, the one-way communication complexity of the function is also $\Omega\left(n/t^4\right)$. However, if $t \geq 3$, then for all $j \in [t-1]$, the two-party communication complexity of $\textsc{Disj}^j_{n,t}$ is $O(1)$, because in every partitioning of $[t]$ into two parts one of the players gets at least two sets and can thus immediately determine whether they are disjoint or intersecting.

5.3 One-way communication vs. simultaneous communication

In this section we compare the simultaneous and the one-way communication models. It is well-known [79] that the simultaneous model is not stronger than the one-way model: every simultaneous protocol can be simulated by a one-way protocol with the same complexity. Kremer *et al.* [79] claimed to prove that the two models are in fact equivalent for two players with respect to general distributional complexity . We disprove this statement by showing that a version of the Generalized Addressing Function (GAF) exhibits an exponential separation between the two models. We then prove that the two models are equivalent with respect to product distributions for an arbitrary number of players.

Simulation of simultaneous communication by one-way communication Kremer *et al.* [79] showed that one-way communication complexity is not weaker than simultaneous communication complexity for two players. Specifically, they showed that for every function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ and for every distribution μ on $\mathcal{X} \times \mathcal{Y}$, $D^{||}_{\mu,\delta}(f) \geq D^{A\to B}_{\mu,\delta}(f) + D^{B\to A}_{\mu,\delta}(f)$. We first note the following generalization of this result to an arbitrary number of players. For a function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \mathcal{Z}$, and for every $j \in [t]$, we denote by $f^j : \mathcal{X}^j \times \mathcal{Y}^j \to \mathcal{Z}$ the two argument function in which $\mathcal{X}^j = \mathcal{X}_j$ and $\mathcal{Y}^j = \vec{\mathcal{X}}^{-j}$ (recall that by our standard notation, $\vec{\mathcal{X}}^{-j} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_{j-1} \times \mathcal{X}_{j+1} \times \cdots \times \mathcal{X}_t$). $f^j(x^j, (y^j_1, \ldots, y^j_{t-1})) =$

$f(y_1^j, \ldots, y_{j-1}^j, x^j, y_j^j, \ldots, y_{t-1}^j)$. We thus have:

Proposition 5.12 *For every function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \mathcal{Z}$, every $0 < \delta < 1$, and every distribution μ on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$,*

$$D^{||}_{\mu,\delta}(f) \;\geq\; \sum_{j=1}^{t} D^{\rightarrow}_{\mu,\delta}(f^j).$$

The proof works by a simple simulation: given a simultaneous protocol Π for f, we construct t one-way protocols $\Pi_1, \ldots, \Pi_t$ for $f^1, \ldots, f^t$ whose sum of communication costs is the communication cost of Π. In the protocol Π_j, Alice sends to Bob the message she would have sent in the simultaneous protocol to the referee. Bob is then able to simulate the roles of all the other players in the simultaneous protocol as well as the role of the referee.

Separation of one-way and simultaneous communication Kremer *et al.* claimed to prove that the other direction is also true for two players and for Boolean functions. We now present a counterexample to this statement. Let $G : \{0,1\}^{n+\log n} \times \{0,1\}^{n+\log n} \to \{0,1\}$, $G((f,x),(g,y)) = f(x \oplus y)$ if $f = g$ and 0 otherwise (f and g are Boolean functions on $\log n$ bits and x, y are inputs to these functions). G is a modification of the Generalized Addressing Function. We prove the following separation between one-way protocols and simultaneous protocols for G:

Proposition 5.13 *For every small enough constant $0 < \delta < 1$,*

1. *$R_\delta^{A \to B}(G) = R_\delta^{B \to A}(G) = O(\log n)$.*
2. *There exists a distribution μ on inputs of G such that $D^{||}_{\mu,\delta}(G) \geq \Omega(\sqrt{n})$.*

Proof: The upper bound is easy: Alice and Bob use the Rabin-Yao $O(\log n)$ communication protocol for equality to check whether $f = g$, and Alice also sends x to Bob. Bob outputs 1 iff $f = g$ and $g(x \oplus y) = 1$.

The lower bound relies on the following simultaneous communication complexity lower bound [8, 103] for the Generalized Addressing Function (Alice gets f, x, Bob gets f, y, and the referee outputs $f(x \oplus y)$): $D^{||}_{\mu',\delta}(\textsc{gaf}) \geq \Omega(\sqrt{n})$, where μ' is the uniform distribution on (f,x,y), and δ is a small enough constant. Consider now the (non-product) distribution μ on inputs of G that is uniform on inputs (f,x,g,y) for which $f = g$. Observe that any δ-error simultaneous protocol for G on μ is in fact also a δ-error simultaneous protocol for GAF on μ'. The lower bound follows. $\square$

Equivalence of simultaneous and one-way communication for product distributions We next prove that Proposition 5.12 is tight with respect to product distributions:

Theorem 5.14 *For every Boolean function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_t \to \{0,1\}$, every $\delta, \delta_1, \ldots, \delta_t > 0$ for which $\delta \geq \sum_{j=1}^{t} H_2(\delta_j)$, and every product distribution μ on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$,*

$$D^{||}_{\mu,\delta}(f) \;\leq\; \sum_{j=1}^{t} D^{\rightarrow}_{\mu,\delta_j}(f^j).$$

Proof: Let $\mu = \mu_1 \times \cdots \times \mu_t$. For each $j = 1, \ldots, t$, let Π_j be any δ_j-error one-way protocol for f^j on the distribution $\mu = \mu_j \times \mu^{-j}$. We denote by A_j the function Alice applies to her input in Π_j to compute the message to be sent to Bob. Let X_j be a random variable whose range is $\mathcal{X}_j$ and that is distributed according to μ_j. X_j represents the input to the j-th player in the simultaneous protocol. Since μ is a product distribution, $X_1, \ldots, X_t$ are mutually independent.

The simultaneous protocol works as follows: the j-th player, given her input $x_j \in \mathcal{X}_j$, computes $A_j(x_j)$ and sends it to the referee. The referee computes the MLE of the random variable $f(X_1, \ldots, X_t)$ given the observations $A_j(X_j) = A_j(x_j), j = 1, \ldots, t$. Clearly, the communication cost of the simultaneous protocol is the sum of the communication costs of the one-way protocols. We are left to prove it indeed computes f with error probability at most δ.

By the MLE principle (Theorem 2.30), the error probability of our protocol is:

$$\begin{aligned} &\Pr\left(\textsc{mle}\left(A_1(X_1), \ldots, A_t(X_t)\right) \neq f(X_1, \ldots, X_t)\right) \\ &\leq \;\; H\left(f(X_1, \ldots, X_t) \mid A_1(X_1), \ldots, A_t(X_t)\right). \end{aligned}$$

Define for each j, $X^{-j} \stackrel{\text{def}}{=} (X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_t)$. X^{-j} is the random variable representing Bob's input in the protocol Π_j. In this protocol Bob predicts $f(X_1, \ldots, X_t)$ based on the message $A_j(X_j)$ he receives from Alice and based on his own input X^{-j}. Therefore, by Fano's inequality (Theorem 2.29),

$$H_2(\delta_j) \;\geq\; H\left(f(X_1, \ldots, X_t) \mid A_j(X_j), X^{-j}\right).$$

The theorem now follows from the following entropy inequality and the fact $\delta \geq \sum_{j=1}^{t} H_2(\delta_j)$:

Claim 5.14.1
$H\left(f(X_1, \ldots, X_t) \mid A_1(X_1), \ldots, A_t(X_t)\right) \;\leq\; \sum_{j=1}^{t} H\left(f(X_1, \ldots, X_t) \mid A_j(X_j), X^{-j}\right).$

Proof: For simplicity, we denote $F = f(X_1, \ldots, X_t)$, $A_j = A_j(X_j)$, $A = (A_1, \ldots, A_t)$, and $A^{-j} = A \setminus \{A_j\}$. Our goal is thus to prove:

$$H(F \mid A) \;\leq\; \sum_{j=1}^{t} H\left(F \mid X^{-j}, A_j\right).$$

We first note that

$$H\left(F \mid X^{-j}, A_j\right) \;=\; H\left(F \mid X^{-j}, A^{-j}, A_j\right) \;=\; H\left(F \mid X^{-j}, A\right),$$

because A^{-j} is a function of X^{-j}. Therefore, we can rewrite the right-hand-side of what we need to prove as: $\sum_{j=1}^{t} H\left(F \mid X^{-j}, A\right)$.

For each j, using the definition of mutual information we have:

$$H(F \mid A) \;=\; H\left(F \mid X^{-j}, A\right) + I\left(F;\; X^{-j} \mid A\right).$$

Thus, summing over all j we have:

$$t \cdot H(F \mid A) \;=\; \sum_{j=1}^{t} H\left(F \mid X^{-j}, A\right) + \sum_{j=1}^{t} I\left(F;\; X^{-j} \mid A\right). \tag{5.3}$$

We expand the second sum as follows:

$$\sum_{j=1}^{t} I\left(F;\; X^{-j} \mid A\right) \;=\; \sum_{j=1}^{t} H\left(X^{-j} \mid A\right) - H\left(X^{-j} \mid F, A\right). \tag{5.4}$$

The mutual independence of the X_j's implies that $g(X_j) \stackrel{\text{def}}{=} (X_j, A_j)$ are mutually independent as well. It follows that

$$\begin{aligned}
\sum_{j=1}^{t} H\left(X^{-j} \mid A\right) \;&=\; \sum_{j=1}^{t} \sum_{i \neq j} H(X_i \mid A_i) \\
&=\; (t-1) \cdot \sum_{i=1}^{t} H(X_i \mid A_i) \;=\; (t-1) \cdot H(X \mid A).
\end{aligned} \tag{5.5}$$

We apply the entropy subset inequality (Theorem 2.16) to get:

$$\sum_{j=1}^{t} H\left(X^{-j} \mid F, A\right) \;\geq\; (t-1) \cdot H(X \mid F, A). \tag{5.6}$$

Combining Equations 5.4, 5.5, 5.6, we have:

$$\begin{aligned}
\sum_{j=1}^{t} I\left(F;\; X^{-j} \mid A\right) \;&\leq\; (t-1) \cdot (H(X \mid A) - H(X \mid F, A)) \;=\; (t-1) \cdot I(X;\; F \mid A) \\
&=\; (t-1) \cdot (H(F \mid A) - H(F \mid X, A)) \;=\; (t-1) \cdot H(F \mid A),
\end{aligned}$$

where the last equality follows from the fact $H(F \mid X, A) = 0$, since F is a function of X. Substituting back into Equation 5.3 completes the proof. $\square$

$\square$

5.4 Multi-Party Set-Disjointness in the simultaneous model

In this section we consider the multi-party version of the disjointness function due to Alon *et al.* [4]. We show an (almost) optimal lower bound on the distributional complexity of this problem in the simultaneous communication model.

Recall that in the t-party disjointness problem, $\textsc{Disj}_{n,t}$, t players are given each a set $\mathcal{S}_i$ from a universe $[n]$, and they have to distinguish between two extreme cases: either the sets are pairwise disjoint, or they are uniquely intersecting. We first note an upper bound for this problem in the simultaneous model:

Proposition 5.15 *For every* $0 < \delta < 1$,

$$R_\delta^{||}(\textsc{Disj}_{n,t}) \;=\; O\left(\frac{n}{t} \cdot \log n \cdot \log \frac{t}{\delta}\right).$$

Proof: The protocol works as follows: the i-th player picks a random subset X_i of his input $\mathcal{S}_i$ of size $6(n/t^2)\ln((t+2)/2\delta)$, and sends it to the referee. The referee declares the sets to be disjoint if and only if $X_1, \ldots, X_t$ are pairwise disjoint. Clearly, the total communication of this protocol is $O\left((n/t) \cdot \log n \cdot \log(t/\delta)\right)$.

If $\mathcal{S}_1, \ldots, \mathcal{S}_t$ are pairwise disjoint, this protocol is correct with probability 1. Consider then the case the sets are uniquely intersecting. Let m_i be the size of $\mathcal{S}_i$. Clearly, $\sum_{i=1}^t m_i \le n + (t-1)$, and therefore an averaging argument implies that more than half of the sets are of size at most $2n/t + 2 \le 3n/t$. Call these players "good". The protocol is erroneous if and only if less than two players choose the intersection point. This happens only if at least $t/2$ among the first $t/2+1$ good players do not pick the intersection point. A good player does not choose the intersection point with probability at most $(1 - t/(3n))^{6n/t^2 \ln((t+2)/2\delta)} \le \left(\frac{\delta}{t/2+1}\right)^{2/t}$, implying the error probability of the protocol is at most δ. $\square$

Alon, Matias, and Szegedy [4] showed that for a certain distribution μ', the μ'-distributional communication complexity (for arbitrary, interactive, communication) of this problem is $\Omega\left(n/t^4\right)$. We next improve this bound to $\Omega\left(n/t\right)$ (which is almost optimal, modulo logarithmic factors) for the simultaneous model. Our lower bound holds for a distribution μ which is a slight variation of μ' (albeit, the proof can be tweaked to work for μ' as well). In Chapter 6 we improve the general communication lower bound to $\Omega\left(n/t^2\right)$ and the one-way communication lower bound to $\Omega\left(n/t^{1+\epsilon}\right)$ for arbitrarily small ϵ.

Our choice of the distribution μ and the proof are inspired by Razborov's proof [109] for the two-party case (which was originally proven by Kalyanasundaram and Schnitger [73]). The distribution $\mu = (X_1, \ldots, X_t)$ is defined by the following process. First, let $(P_1, P_2, \ldots, P_t, I)$ be a uniformly chosen partition of the universe $[n], n = (2m-1)t+1$, for which $|P_j| = 2m-1$, $j = 1, \ldots, t$, and $|I| = 1$. Next, choose $X_j \subseteq P_j \cup I$, $j \in [t]$, as follows: With equal probability, either $I \in X_j$, for all j, or $I \notin X_j$, for all j. Then, for each j, choose independently for each $i \in P_j$ to include i in X_j with probability $1/2$.

Theorem 5.16 *For every* $0 < \delta < 1$,

$$D^{||}_{\mu,\delta}\left(\textsc{Disj}_{n,t}\right) \;\geq\; \left(\frac{n-1}{t}+1\right)\cdot\left(1-H_2\left(\delta\right)\right).$$

Proof: In the proof we use the following notation: given a collection of random variables $Z = (Z_1, \ldots, Z_t)$, define for each j, $Z^{<j} \stackrel{\text{def}}{=} (Z_1, \ldots, Z_{j-1})$ and $Z^{-j} \stackrel{\text{def}}{=} Z \setminus \{Z_j\}$.

Consider a δ-error simultaneous protocol for $\textsc{Disj}_{n,t}$ in which $A_j = A_j(X_j)$, for $j \in [t]$, is the message sent by the j-th player to the referee. Note that $H\left(A_j\right)$ is at most the communication of the j-th player in the protocol, and therefore to prove the theorem, we will show that $\sum_j H\left(A_j\right) \geq 2m \cdot \left(1 - H_2\left(\delta\right)\right)$.

Let $F \stackrel{\text{def}}{=} \textsc{Disj}_{n,t}(X_1, \ldots, X_t)$, $P \stackrel{\text{def}}{=} (P_1, \ldots, P_t)$, and $A \stackrel{\text{def}}{=} (A_1, \ldots, A_t)$. Using Fano's inequality (Theorem 2.29) and the chain rule for mutual information (Proposition 2.26),

$$\begin{aligned} H_2\left(\delta\right) &\geq H\left(F \mid A\right) \;\geq\; H\left(F \mid A, P\right) \;=\; H\left(F \mid P\right) - I\left(F;\ A \mid P\right) \\ &= H\left(F \mid P\right) - \sum_{j=1}^{t} I\left(F;\ A_j \mid A^{<j}, P\right). \end{aligned} \tag{5.7}$$

Our goal now is to upper bound each term in the above sum separately. Fix a $j \in [t]$. We have

$$\begin{aligned} I\left(F;\ A_j \mid A^{<j}, P\right) &= H\left(A_j \mid A^{<j}, P\right) - H\left(A_j \mid F, A^{<j}, P\right) \\ &\leq H\left(A_j \mid P\right) - H\left(A_j \mid F, A^{<j}, P\right), \end{aligned} \tag{5.8}$$

because conditioning reduces entropy (Proposition 2.19). For the second term, we use the crucial observation that, given P and F, the X_j's are mutually independent random variables. Therefore, $I\left(X_j;\ X^{<j} \mid F, P\right) = 0$. By the data processing inequality for mutual information (Corollary 2.28), $I\left(A_j;\ A^{<j} \mid F, P\right) = 0$ as well, that is, $H\left(A_j \mid F, A^{<j}, P\right) = H\left(A_j \mid F, P\right)$. Substituting this fact in Equation 5.8 above:

$$\begin{aligned} I\left(F;\ A_j \mid A^{<j}, P\right) &\leq H\left(A_j \mid P\right) - H\left(A_j \mid F, P\right) = I\left(F;\ A_j \mid P\right) \\ &= H\left(F \mid P\right) - H\left(F \mid A_j, P\right). \end{aligned} \tag{5.9}$$

Note that $H\left(F \mid P\right) = 1$ because given P, F is uniform on $\{0, 1\}$. Consider the second term in the equation above. Since P_j is fixed given P^{-j} and I,

$$H\left(F \mid A_j, P\right) = H\left(F \mid A_j, I, P^{-j}\right) = \sum_p \Pr\left(P^{-j} = p\right) \cdot H\left(F \mid A_j, I, P^{-j} = p\right), \tag{5.10}$$

where the sum ranges over all valid assignments p to P^{-j}. Consider a fixed p in the sum above, and for brevity, let E denote the event "$P^{-j} = p$". Let σ denote the complement of the union of the sets in p. This set has $2m$ elements, so let $\sigma = \{1, 2, \ldots 2m\}$, without loss of generality. Given E, I is chosen uniformly from σ, which then fixes P_j as well. Thus, X_j is a subset of σ given E. Let Z_i, for $i \in [2m]$, denote the indicator random variable for whether $i \in X_j$. Given E, the Z_i's completely determine X_j and vice versa. Moreover, given E and "$I = i$", $F = Z_i$. Therefore,

$$\begin{aligned} H\left(F \mid A_j, I, E\right) &= \sum_{i=1}^{2m} \Pr\left(I = i \mid E\right) \cdot H\left(F \mid A_j, I = i, E\right) \\ &= \frac{1}{2m} \cdot \sum_{i=1}^{2m} H\left(Z_i \mid A_j, I = i, E\right). \end{aligned} \tag{5.11}$$

It can be easily verified that given E, the random variables X_j and I are independent. Therefore both Z_i and $A_j = A_j(X_j)$, which depend only on X_j, are also independent of I given E. It follows that $H\left(Z_i \mid A_j, I = i, E\right) = H\left(Z_i \mid A_j, E\right)$. Substituting this into Equation 5.11 above,

$$\begin{aligned} H\left(F \mid A_j, I, E\right) &= \frac{1}{2m} \cdot \sum_{i=1}^{2m} H\left(Z_i \mid A_j, E\right) \geq \frac{1}{2m} \cdot H\left(Z_1, \ldots, Z_{2m} \mid A_j, E\right) \\ &= \frac{1}{2m} \cdot H\left(X_j \mid A_j, E\right), \end{aligned}$$

using the sub-additivity of entropy (Proposition 2.15). Using this bound in Equation 5.10, we obtain

$$\begin{aligned} H(F \mid A_j, P) &\geq \frac{1}{2m} \cdot H\left(X_j \mid A_j, P^{-j}\right) \geq \frac{1}{2m} \cdot H(X_j \mid A_j, P) \\ &\geq \frac{1}{2m} \cdot (H(X_j \mid P) - H(A_j \mid P)) \geq 1 - \frac{H(A_j)}{2m}, \end{aligned}$$

because given P, X_j is a random (multi-)subset of $P_j \cup I$. Combining this bound with Equations 5.9 and 5.7, and again using $H(F \mid P) = 1$, proves the theorem. □

5.5 Data stream lower bounds

In this section we demonstrate the applicability of the lower bound techniques for the one-way model to deriving lower bounds for the data stream model. Specifically, we use the shatter coefficients technique to prove new lower bounds for the data stream space complexity of estimating the number of distinct elements (a.k.a. F_0, the zeroth frequency moment; see definition in Section 4.4.2) and of approximating selection functions, such as the median (see definition in Section 4.4.3).

5.5.1 Counting distinct elements

Approximately counting the number of distinct elements in a data stream, or equivalently estimating the zeroth frequency moment of a data stream, is a well studied problem [47, 4, 52, 49, 18, 16], having applications in database, networking, and information retrieval. The best algorithm to date obtains an ϵ-relative approximation of F_0 with error probability δ in space $O\left(\left(1/\epsilon^2 + \log m\right)\log(1/\delta)\right)$ [16]. Alon *et al.* [4] proved a lower bound of $\Omega(\log m)$ for the problem, with no dependence on the approximation error ϵ. In the following we use the techniques developed in this chapter to prove an $\Omega(1/\epsilon)$ lower bound for the problem:

Theorem 5.17 *For every n, m, $0 < \delta < 1$, and $1/n \leq \epsilon \leq 1/6$,*

$$DS_{\epsilon,\delta}(\mathrm{F}_0) = \Omega\left(\frac{1}{\epsilon} \cdot (1 - H_2(\delta))\right).$$

Proof: The proof works by a reduction from the one-way communication complexity of the following version of the set-disjointness problem to the data stream space complexity of F_0.

The ϵ-set-disjointness problem, for $1/n \le \epsilon \le 1/2$, is defined as follows: $\textsc{Disj}_{n,\epsilon} : \mathcal{T} \to \{0,1\}$, where

$$\mathcal{T} \stackrel{\text{def}}{=} \left\{(x,y) \in \{0,1\}^{2n} \;\middle|\; |x| = n/2,\, |y| = \epsilon n, \text{ and either } x \cap y = \emptyset \text{ or } y \subseteq x\right\}.$$

$\textsc{Disj}_{n,\epsilon}(x,y) = 1$ if and only if $y \subseteq x$. That is, the first input of $\textsc{Disj}_{n,\epsilon}$ is a set of density $1/2$, its second input is a set of density ϵ, and the sets are guaranteed to be either disjoint or one to contain the other.

We use the shatter coefficients technique to give an $\Omega(1/\epsilon)$ lower bound on the one-way communication complexity of this problem. For simplicity we assume ϵn divides $n/2$. We partition $[n/2]$ into $\ell \stackrel{\text{def}}{=} 1/2\epsilon$ consecutive equally sized intervals, and let y_j, for $j \in [\ell]$, be the j-th such interval. We define $\mathcal{S} \stackrel{\text{def}}{=} \{y_1, \ldots, y_\ell\}$. For each 0-1 pattern $\vec{b} = b_1, \ldots, b_\ell$, we construct a set $x_{\vec{b}}$ as follows: for each $j \in [\ell]$, $x_{\vec{b}}$ contains y_j if and only if $b_j = 1$; we then add to $x_{\vec{b}}$ the first $(\ell - |\vec{b}|) \cdot \epsilon n$ elements of the set $\{n/2+1, \ldots, n\}$, in order to make $x_{\vec{b}}$ a set of size $n/2$. The family $\{x_{\vec{b}}\}_{\vec{b}}$ shatters $\mathcal{S}$, and therefore by Theorem 5.5,

$$R_{\delta}^{\rightarrow}(\textsc{Disj}_{n,\epsilon}) \;\ge\; \frac{1}{2\epsilon}(1 - H_2(\delta)).$$

We use the reduction technique outlined in Section 3.3.3 to derive a lower bound for F_0. Let B be an optimal randomized oblivious read-once branching program that (ϵ,δ)-approximates F_0 on inputs from $[m]^n$, and let $S = DS_{\epsilon,\delta}(\mathrm{F}_0)$ be the space used by B. We use B to construct a one-way communication protocol Π for $\textsc{Disj}_{n',\epsilon'}$, where $n' = 2n/(1+6\epsilon)$ and $\epsilon' = 3\epsilon$.

Π works as follows: Alice runs B on the $n'/2$ elements of x, and then sends the reached state to Bob. Bob continues the simulation of B on the $\epsilon' n'$ elements of y. If the output of B is at least $(1-\epsilon)(n'/2 + \epsilon' n')$, Bob declares the sets to be disjoint, and otherwise declares x to contain y.

Note that the communication in the protocol is the number of bits required to represent a state of B, which is exactly S. If x, y are disjoint, then the number of distinct elements given to the simulation of B is $n'/2 + \epsilon' n'$. If B outputs an ϵ-relative approximation of this quantity (which happens with probability at least $1-\delta$), then it has to output a value which is greater than $(1-\epsilon)(n'/2 + \epsilon' n')$, and therefore the protocol indeed declares the sets to be disjoint. If, on the other hand, x contains y, then the number of distinct elements is only $n'/2$, and therefore with probability at least $1-\delta$, B outputs a value which is at most $(1+\epsilon)n'/2$. By our choice of ϵ', this is less than $(1-\epsilon)(n'/2 + \epsilon' n')$, and therefore the protocol will indeed declare x to contain y. We conclude from the lower bound for $\textsc{Disj}_{n,\epsilon}$ that:

$$DS_{\epsilon,\delta}(\mathrm{F}_0) \;=\; S \;\geq\; \frac{1}{2\epsilon'}\left(1 - H_2(\delta)\right) \;=\; \frac{1}{6\epsilon}\left(1 - H_2(\delta)\right).$$

□

5.5.2 Selection functions

Data stream computation of selection functions (a.k.a. quantiles) is an important task in databases. Greenwald and and Khanna [58] (improving on previous work of Manku, Rajagopalan, and Lindsay [86, 87]) gave a deterministic algorithms for ϵ-approximating $\textsc{Select}_p$, for any constant $0 < p < 1$, running in space $O(1/\epsilon \cdot \log(\epsilon n))$. Henzinger, Raghavan, and Rajagopalan [64] showed a lower bound of $\Omega(1/\epsilon)$ on the space required to approximate the median ($\textsc{Select}_{\frac{1}{2}}$) in the data stream model. However, their bound works only for $\epsilon \leq 1/2n$, which basically means their result is just an $\Omega(n)$ lower bound for exact computation of the median. In the following we use the shatter coefficients technique to generalize the proof of Henzinger *et al.* [64] to arbitrary ϵ and to all values of p (not just 1/2):

Theorem 5.18 *For every $0 < p < 1$, every $0 < \epsilon \leq \min\{p/2, (1-p)/2\}$, and every $0 < \delta < 1$,*

$$DS_{\epsilon,\delta}(\textsc{Select}_p) \;\geq\; \Omega\left(\frac{\min\{p, 1-p\}}{\epsilon} \cdot \left(1 - H_2(\delta)\right)\right).$$

Proof: We prove the lower bound by a reduction from the one-way communication complexity of the ϵ-indexing function, which is a restricted variant of the indexing function, defined as follows: $\textsc{Ind}_\epsilon : \mathcal{T} \to \{0,1\}$, where

$$\mathcal{T} \;\stackrel{\text{def}}{=}\; \{(x,i) \in \{0,1\}^n \times [n] \;\mid\; \forall j \in (i - \epsilon n, i + \epsilon n),\; x_j = x_i\},$$

and $\textsc{Ind}_\epsilon(x,i) = x_i$. That is, $\textsc{Ind}_\epsilon$ is the same as $\textsc{Ind}$, except that its inputs (x,i) are guaranteed to satisfy the following requirement: all the entries of x in the vicinity of radius ϵn around x_i contain the same value as x_i. We next use the shatter coefficients technique to prove that the one-way communication complexity of this problem is $\Omega(1/\epsilon)$.

For simplicity, we assume ϵn is an integer and divides n. Let $\mathcal{S}$ be the set of $\ell \stackrel{\text{def}}{=} 1/2\epsilon$ indices $\epsilon n, 3\epsilon n, 5\epsilon n, \ldots$. We construct a set of 2^ℓ strings $x \in \{0,1\}^n$ that shatters $\mathcal{S}$. Let $\vec{b} = b_1, \ldots, b_\ell$ be any 0-1 pattern. We define a string $x_{\vec{b}} \in \{0,1\}^n$ as follows: for each

$j \in [\ell]$, the entries of $x_{\vec{b}}$ in the interval $((2j-2)\epsilon n, 2j\epsilon n]$ are set to b_j. The family $\{x_{\vec{b}}\}_{\vec{b}}$ shatters $\mathcal{S}$, and therefore, by Theorem 5.5,

$$R_{\delta}^{\rightarrow}(\textsc{Ind}_{\epsilon}) \;\geq\; \frac{1}{2\epsilon}(1 - H_2(\delta)).$$

We now reduce this problem to the approximation of $\textsc{Select}_p$ in the data stream model. Assume, initially, that $p \leq 1/2$. Let B be an optimal randomized oblivious read-once branching program that (ϵ, δ)-approximates $\textsc{Select}_p$ on inputs in $\mathbb{R}^n$. Let $S = DS_{\epsilon,\delta}(\textsc{Select}_p)$ the space used by B. We use B to construct a δ-error one-way protocol Π for $\textsc{Ind}_{\epsilon'}$ on inputs in $\mathcal{T}' \subseteq \{0,1\}^{n'} \times [n']$, where $n' = n(p - 2\epsilon)$ and $\epsilon' = \epsilon/(p - 2\epsilon)$.

Alice feeds B with the following sequence of elements: for each $j \in [n']$, she feeds B with $2(j-1) + x_j$. On top of that, she feeds B with ϵn copies of x_1 and ϵn copies of $2(n'-1) + x_{n'}$. She then transmits the reached state of B to Bob. Bob continues the simulation of B with the following sequence of elements: $n(p - \epsilon) - i$ copies of 0 and $n(1 - 2p + \epsilon) + i$ copies of $2n' + 1$. Bob, when getting the output of B, declares x_i to be the parity of this output.

Note that the element of rank p in the sequence of elements Alice and Bob feed into B is exactly $2(i-1) + x_i$. Moreover, by our choice of n' and ϵ', all the values in the vicinity of radius ϵn around the element of rank p also have parity x_i. Therefore, when B outputs an ϵ-approximation of $\textsc{Select}_p$ on the given input sequence (which happens with probability at least $1-\delta$), then Bob announces the correct answer. Since the communication cost of Π is S, we conclude from the lower bound obtained before for $\textsc{Ind}_{\epsilon}$ that,

$$DS_{\epsilon,\delta}(\textsc{Select}_p) \;=\; S \;\geq\; \frac{1}{2\epsilon'}(1 - H_2(\delta)) \;=\; \left(\frac{p}{2\epsilon} - 1\right)(1 - H_2(\delta)).$$

An identical argument works for $p > 1/2$, except that we need to replace all the occurrences of p by $1 - p$. □

5.6 Sketch lower bounds for frequency moments

In this section we use the lower bound for Multi-Party Set-Disjointness in the simultaneous communication model to derive optimal sketch lower bounds for the frequency moments (see definition in Section 4.4.2).

In $\mathrm{F}_{k,t}$, the t-party sketch version of the k-th frequency moment problem, the input $x_1, \ldots, x_n \in [m]^n$ is partitioned into t equally sized blocks. The j-th block is given to the

j-th player. The players compute "sketches" of the input parts and send them to the referee. The referee uses the sketches to compute an ϵ-relative approximation of $\mathrm{F}_k(x_1, \ldots, x_n)$.

Since every oblivious sampling algorithm can be simulated by a sketch algorithm (see Section 3.5.3), then our sampling algorithm for approximating F_k from Section 4.4.2 can be used as a sketch algorithm, whose cost is $O\left(m^{1-1/k} \cdot \log m \cdot (1/\epsilon^2) \cdot \log(1/\delta) \cdot k! \cdot 2^k\right)$. For $k = 2$, Alon *et al.* [4] (implicitly) show an $O(t \log m \cdot (1/\epsilon^2) \cdot \log(1/\delta))$ sketch algorithm for $\textsc{Disj}_{2,t}$. We next prove that for any constant $k \geq 1$ and for any large enough t, the two bounds are almost tight in terms of the dependence on m:

Theorem 5.19 *For every m, $k \geq 1$, $0 < \delta < 1$, $0 < \epsilon \leq \frac{1}{3}$, and $t \geq ((1+3\epsilon)m)^{1/k}$,*

$$SC_{\epsilon,\delta}(\mathrm{F}_{k,t}) \;\geq\; \Omega\left(m^{1-1/k} \cdot (1-3\epsilon)^{1/k} \cdot (1 - H_2(\delta))\right).$$

Proof: Clearly, any t-party sketch algorithm for F_k can be simulated by a t'-party sketch algorithm with the same sketch cost, provided $t' < t$. Therefore, it is enough to prove the lower bound for $t = ((1+3\epsilon)m)^{1/k}$.

We use the connection between sketch computations and simultaneous communication complexity outlined in Section 3.4.2. Let Π be a t-party public-coin simultaneous communication complexity protocol that (ϵ, δ)-approximates $\mathrm{F}_{k,t}$ on inputs in $[m]^n$, where n is to be determined later. We will use Π to construct a randomized t-party simultaneous protocol Π' that computes $\textsc{Disj}_{m,t}$ with the same communication cost. From Theorem 5.16 we know that $\textsc{Disj}_{m,t}$ is hard on inputs of the following form: $\mathcal{S}_1, \ldots, \mathcal{S}_t$ are each of size $\ell \stackrel{\text{def}}{=} (m-1)/2t + 1/2$, and are guaranteed to be either pairwise disjoint or uniquely intersecting. The protocol Π' will thus compute $\textsc{Disj}_{m,t}$ only on inputs of this form. We define n, the size of the input of $\mathrm{F}_{k,t}$, to be $n \stackrel{\text{def}}{=} \ell \cdot t$.

The protocol Π' works as follows. Given a set $\mathcal{S}_j$, the j-th player simulates the j-th player in the protocol Π on the elements of $\mathcal{S}_j = \left\{s_j^1, \ldots, s_j^\ell\right\}$. She then transmits to the referee exactly what the j-th player in Π would have sent to the referee. The referee in Π' simulates the referee in Π, and gets a value b which is with probability at least $1 - \delta$ an ϵ-relative approximation of $\mathrm{F}_k\left(s_1^1, \ldots, s_t^\ell\right)$. If $b \leq (1+\epsilon)m$, the referee declares to the sets to be disjoint, and otherwise declares them to be uniquely intersecting.

If the input sets are disjoint, then each element in $[m]$ has a frequency of at most one in the input sequence $\left\{s_1^1, \ldots, s_t^\ell\right\}$, and therefore $\mathrm{F}_k\left(s_1^1, \ldots, s_t^\ell\right) \leq m$. On the other hand, if the sets are uniquely intersecting, then there is at least one element whose frequency is t, and therefore $\mathrm{F}_k\left(s_1^1, \ldots, s_t^\ell\right) \geq t^k \geq (1+3\epsilon)m$. Since Π produces an answer b, which with probability $1-\delta$ is in the interval $((1-\epsilon)\mathrm{F}_k, (1+\epsilon)\mathrm{F}_k)$, then if the sets are disjoint, then b is with probability $1-\delta$ at most $m(1+\epsilon)$, and if the sets are uniquely intersecting, then b is with probability $1-\delta$ at least $(1-\epsilon)(1+3\epsilon)m > (1+\epsilon)m$. Thus, our protocol is correct on any input with probability $1-\delta$.

Note that Π' uses the same amount of communication as Π. Therefore, by Theorem 5.16, Π's cost is at least $((m-1)/t+1)(1-H_2(\delta)) = \Omega\left(m^{1-1/k}(1-3\epsilon)^{1/k}(1-H_2(\delta))\right)$. $\square$

5.7 Open problems

The most glaring open problem is to extend our method to arbitrary round communication complexity, going beyond the one-way and simultaneous message models. It seems that obtaining the result of [73, 109] would be a great first step.

For the classical $\mathbb{Z}_2$ inner product problem, our methods can be applied, but they yield poor bounds compared to the discrepancy method. Can this be rectified?

It seems that for symmetric functions, the simultaneous model is as powerful as the one-way model, even for non-product distributions, but formalizing this issue seems subtle, and a proof seems considerably difficult.

As for our applications: (1) There is still a quadratic gap left between the upper bound and the lower bound for F_0 in terms of the dependence on ϵ; (2) The sketch lower bound for frequency moments holds only for a large number of players; it is left open to find the exact tradeoff between the number of players and the sketch cost.

Chapter 6

General Communication Lower Bounds

In Chapter 5 we presented lower bound techniques for restricted models of communication. While being sufficient for our primary application domains—data stream and sketch lower bounds—these models are not strong enough to capture more complex scenarios (e.g., the k-pass data stream model). This chapter presents a novel lower bound technique for general, interactive, randomized communication complexity. We illustrate the technique with several applications, including new communication complexity lower bounds for MULTI-PARTY SET-DISJOINTNESS and L_p distance estimation and new data stream lower bounds for the frequency moments and L_p approximation.

The lower bound method Our method applies to *decomposable functions*—ones that can be decomposed into simpler "primitive" functions, together with an appropriate "composer" function. For example, the two-party SET-DISJOINTNESS function can be decomposed into n two-bit AND functions, one per coordinate: $\textsc{Disj}_{n,2}(\vec{x}, \vec{y}) = \bigvee_{i=1}^{n} (x_i \wedge y_i)$. We argue that any correct protocol for a decomposable function must reveal in its transcript enough information to compute each of the constituent primitive functions, a direct-sum property. We formalize this idea through the notion of *information cost* of a communication protocol, which measures the amount of information revealed by the transcript of the protocol about the inputs. The *information complexity* of a function is the minimum information cost incurred by any protocol that computes the function correctly; this measure is a lower bound on the randomized communication complexity of the function. This concept was recently introduced by Chakrabarti, Shi, Wirth, and Yao [32] in the context of simultaneous communication complexity; it is also implicit in the work of Ablayev [2] and Saks and Sun [113] in the context of one-way communication complexity. We give an appropriate generalization of information complexity for general communication models; the highlight of our generalization is that it admits a direct sum theorem. The direct sum property reduces the task of proving information complexity lower bounds for multi-dimensional functions such as Set-Disjointness to that of proving lower bounds for single-dimensional problems, such as And.

Direct sum for information complexity In carrying out an information complexity lower bound, we would like to create an input distribution that is intuitively hard for any communication protocol. It turns out that for many natural examples, these distributions necessarily have a non-product structure. This is one of the main obstacles to extending the direct sum methodology of [32] to general communication protocols. In the proof technique of Saks and Sun [113], the issue of such non-product distributions causes significant complications; they resolve this difficulty for the one-way model by using tools from information theory and Fourier analysis. We approach this problem by expressing the non-product distribution as a convex combination of product distributions; this approach has been previously considered for other problems like the distributional complexity of Set-Disjointness [109] and the parallel repetition theorem [105].

Statistical geometry of communication complexity Lower bounds for "single coordinate problems" have not been considered previously in communication complexity. Indeed, the length of communication is not an appropriate measure for this study, since any non-constant function requires at least one bit of communication (which for one-bit functions is also sufficient). Nevertheless, it is possible to design randomized protocols that compute such functions and reveal less than one bit of information about their inputs. Therefore existing methods for communication complexity seem unsuitable for obtaining information complexity lower bounds for single coordinate functions. Our solution is based on statistical analysis of the transcript distributions corresponding to communication complexity protocols. By considering various measures of statistical distance between these distributions, such as the Hellinger distance, we discover new geometric properties of these distributions, including a "cut-and-paste" property and an inverse-triangle inequality; these are crucial to the proofs of the one-coordinate lower bounds.

Lower bound technique for one-way protocols Our result for MULTI-PARTY SET DISJOINTNESS in the general communication complexity model is not tight. This is due to a limitation in our proof technique and can be attributed to the fact that the square of the Hellinger distance satisfies only a weak form of triangle inequality. This leads us to consider generalizations of the Hellinger distance, which, combined with the Markovian structure of one-way protocols, allows us to derive near-triangle inequalities. To the best of our knowledge, this is the first proof technique for multi-party one-way protocols.

Multi-Party Set-Disjointness In the MULTI-PARTY SET-DISJOINTNESS problem, t players are given each a subset of $[n]$ with the guarantee that the sets are either pairwise disjoint (NO instance) or mutually intersecting and otherwise disjoint (YES instance). The problem can be solved in $\tilde{O}(n/t)$ communication [15], and the best lower bound to date was $\Omega(n/t^4)$, due to Alon, Matias, and Szegedy [4], who generalized Razborov's proof [109] for the two-party case. Using our new methods, we improve the lower bound to $\Omega(n/t^2)$. For the restricted one-way communication model, we obtain a better bound of $\Omega(n/t^{1+\epsilon})$ for arbitrarily small $\epsilon > 0$.

L_p distance estimation In the L_p distance estimation problem, Alice and Bob are given each a vector in $[m]^n$ with the guarantee that the L_p distance between the two vectors is either at most 1 (NO instance) or at least θ (YES instance). The communication complexity of the problem for $p \in [1, 2]$ is logarithmic in n, because there exist efficient data stream algorithms for the problem [4, 46, 48, 68]. Saks and Sun proved that the one-way communication complexity of the problem is $\Omega(n/\theta^2)$ for any $p > 2$. They left the general communication complexity of the problem open. We prove that the same bound holds also for general communication complexity.

Data stream lower bounds for frequency moments Using the new lower bounds for MULTI-PARTY SET-DISJOINTNESS, we are able to finally resolve the space complexity of approximating the frequency moments in the data stream model. Alon *et al.* [4] showed that F_k, the k-th frequency moment, can be approximated in logarithmic space for $k = 0, 1, 2$ and requires $m^{\Omega(1)}$ space for $k > 5$. Their best upper bounds for the interval $k \in (2, 5]$ used

$m^{O(1)}$ space. They conjectured that in fact F_k requires $m^{\Omega(1)}$ space for any real $k > 2$. We resolve their conjecture, and prove that any data stream algorithm for approximating F_k requires $\Omega\left(m^{1-(2+\epsilon)/k}\right)$, for arbitrarily small $\epsilon > 0$.

Data stream lower bounds for $\mathbf{L}_p$ distance approximation Using the communication complexity lower bounds for the L_p distance estimation problem, we obtain corresponding data stream lower bounds for approximating the L_p distance. We show that for any $p \in (2, \infty]$, a data stream algorithm that approximates the L_p distance between two vectors to within an n^ϵ multiplicative factor, requires $\Omega(n^{1-4\epsilon-2/p})$ space. The same bound was shown before by Saks and Sun [113]; our bound has the advantage of being applicable even if the data stream algorithm is allowed a constant number of passes over the data.

The chapter is organized as follows. We start in Section 6.1 with the definition of information complexity and its connection to communication complexity. We then prove in Section 6.2 the direct sum theorem for conditional information complexity. In Section 6.3 we discuss our statistical approach to deriving information complexity lower bounds for primitive functions. In Section 6.4 we develop several statistical-geometric properties of distributions arising from communication complexity protocols. In Sections 6.5 and 6.6 we present our two main applications: lower bounds for Multi-Party Set-Disjointness and L_p estimation. In Section 6.7 we derive the data stream lower bounds for frequency moments and L_p approximation. Section 6.8 reviews extensions of the lower bound technique, and Section 6.9 concludes with subsequent work and some open problems.

6.1 Information complexity

Our lower bound method is built on an information-theoretic measure of communication complexity, called *information complexity*, defined with respect to a given distribution over the inputs to the function; our definitions generalize similar notions that were considered previously [32, 2, 113]. In this section we give definitions of the notion and its variants and review their basic properties. Starting with this section, and continuing in Sections 6.2 and 6.3, we consider the Set-Disjointness problem as an illustration of our new techniques, resulting in a new proof of the $\Omega(n)$ lower bound on the randomized communication complexity of the problem.

Throughout the discussion below we make certain assumptions about the communication model and about the functions for which the lower bound technique works. Most of these assumptions are made for the sole purpose of keeping notation and exposition simple. We make each of the assumptions explicit, and indicate whether it is indeed required. In Section 6.8 we discuss extensions of the technique that eliminate the unnecessary assumptions.

The technique is described in the framework of two-party private-coin randomized communication complexity (see Section 6.8 for extensions to multi-party and to public-coin protocols). Throughout, we fix a function $f : \vec{\mathcal{L}} \to \mathcal{Z}$, where $\vec{\mathcal{L}} \subseteq \vec{\mathcal{X}} \times \vec{\mathcal{Y}}$, and $\vec{\mathcal{X}} = \mathcal{X}^n, \vec{\mathcal{Y}} = \mathcal{Y}^n$ are n-dimensional vectors. We further assume that $\vec{\mathcal{L}} = \mathcal{L}^n$, where $\mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$ (see Section 6.8 for an extension to functions over non-product domains).

In SET-DISJOINTNESS, Alice and Bob hold, respectively, the characteristic vectors $\vec{x}$ and $\vec{y}$ of two subsets of $[n]$. $\textsc{Disj}(\vec{x}, \vec{y}) = 1$ iff $\vec{x} \cap \vec{y} \neq \emptyset$.

Informally, information cost of a protocol is the amount of information one can learn about the inputs from the protocol's transcript of messages. Formally it is defined as follows:

Definition 6.1 (Information cost) *Let $\vec{\mu}$ be a distribution on $\vec{\mathcal{L}} \subseteq \vec{\mathcal{X}} \times \vec{\mathcal{Y}}$, let $(\vec{X}, \vec{Y}) \sim \vec{\mu}$, and let Π be a private-coin randomized protocol Π running on inputs in $\vec{\mathcal{L}}$. The* information cost *of Π with respect to $\vec{\mu}$ is:*

$$IC_{\vec{\mu}}(\Pi) \stackrel{\text{def}}{=} I\left(\vec{X}, \vec{Y};\ \Pi(\vec{X}, \vec{Y}, R_A, R_B)\right).$$

Definition 6.2 (Information complexity) *For $0 < \delta < 1$, the* information complexity *of f with respect to a distribution $\vec{\mu}$ and error parameter δ, denoted $IC_{\vec{\mu},\delta}(f)$, is defined as the minimum information cost of a δ-error protocol for f with respect to $\vec{\mu}$.*

Our technique gives lower bounds on the information complexity of functions; the following proposition shows that this is sufficient for obtaining lower bounds on the randomized communication complexity:

Proposition 6.3 *For any distribution $\vec{\mu}$ and any $0 < \delta < 1$, $\quad R_\delta(f) \geq IC_{\vec{\mu},\delta}(f)$.*

Proof: If Π is the best δ-error protocol for f (in terms of communication), we have: $IC_{\vec{\mu},\delta}(f) \leq IC_{\vec{\mu}}(\Pi) \leq H\left(\Pi(\vec{X}, \vec{Y}, R_A, R_B)\right) \leq \left|\Pi(\vec{X}, \vec{Y}, R_A, R_B)\right| = R_\delta(f)$.

$(H\left(\Pi(\vec{X},\vec{Y},R_A,R_B)\right) \leq \left|\Pi(\vec{X},\vec{Y},R_A,R_B)\right|$, due to our convention that all transcripts are of the same length). $\square$

The reason to deal with information complexity as opposed to a direct treatment of communication complexity is that former possesses an important direct sum property which the latter does not. Suppose $f : \vec{\mathcal{L}} \to \mathcal{Z}$ can be expressed in terms of a simpler "primitive" h applied to each coordinate of the input pair $(\vec{x},\vec{y})$ (the notion will be formalized later). If f depends (say, symmetrically) on the primitive in each coordinate, and if the distribution $\vec{\mu}$ on $\vec{\mathcal{L}}$ is the product of independent copies of a distribution ν on $\mathcal{L}$, then any protocol for f must implicitly solve each instance of the primitive h. Thus, one can hope to show that $IC_{\vec{\mu}}(f) \geq n \cdot IC_\nu(h)$, i.e., a direct sum property for information complexity.

A major technical obstacle in proving this result is that the distribution $\vec{\mu}$ is not necessarily a product distribution on $\mathcal{X}^n \times \mathcal{Y}^n$. This is because ν need not be a product distribution on $\mathcal{X} \times \mathcal{Y}$ (although $\vec{\mu}$ is the product of n copies of ν). In fact, for Set-Disjointness, it becomes essential to consider non-product distributions to obtain an $\Omega(n)$ lower bound [7]. To handle this, we will use the fact that ν may be written as a convex combination $\nu = \sum_{d\in\mathcal{D}} \lambda_d \nu_d$ of product distributions ν_d, where $\mathcal{D}$ is some index set. Such a decomposition always exists, but in general, is not unique; we will choose one where the entropy of the ν_d's is as large as possible.

One way to realize ν is to view the collection of λ_d's as a distribution λ on the index set $\mathcal{D}$ that selects the appropriate ν_d, $d \in \mathcal{D}$. Therefore, conditioned on λ, ν is a product distribution; that is, if $(X,Y) \sim \nu$ and $D \sim \lambda$, then conditioned on $D = d$, for any $d \in \mathcal{D}$, X and Y are independent of each other. We will say that λ partitions ν. Analogously, let $\vec{\lambda}$ denote the distribution of n i.i.d. copies of λ; clearly, $\vec{\lambda}$ partitions $\vec{\mu} = \nu^n$. If $(\vec{X},\vec{Y}) \sim \vec{\mu}$, then conditioned on $\vec{D} = \vec{d}$: (1) $\vec{X}$ and $\vec{Y}$ are independent of each other, and (2) the coordinates $\{(X_j, Y_j)\}_{j\in[n]}$ are mutually independent of each other.

For Set-Disjointness, we will use the non-product distribution ν given by $\nu(0,0) = 1/2$, $\nu(0,1) = \nu(1,0) = 1/4$. We partition ν as follows. Let λ be the uniform distribution on $\{0,1\}$, let $D \sim \lambda$, and let $(U,V) \sim \nu$. If $D = 0$, then let $U = 0$ and let V be a uniform element of $\{0,1\}$; if $D = 1$, then let $V = 0$ and let U be a uniform element of $\{0,1\}$. It is clear that conditioned on D, U and V are independent.

The above discussion motivates the following refinement of the notion of information cost. The *conditional information cost* of a protocol is the amount of extra information one can learn about the inputs from the protocol's transcript, given some prior information about the inputs. Typically, this prior information will be exactly the distribution $\vec{\lambda}$ that partitions the input distribution $\vec{\mu}$.

Definition 6.4 (Conditional information cost) *For a distribution $\vec{\mu}$ on $\vec{\mathcal{L}} \subseteq \vec{\mathcal{X}} \times \vec{\mathcal{Y}}$, for a partitioning $\vec{\lambda}$ of $\vec{\mu}$, and for a private-coin randomized protocol Π running on inputs in $\vec{\mathcal{L}}$, let $(\vec{X}, \vec{Y}) \sim \vec{\mu}$ and $\vec{D} \sim \vec{\lambda}$. The* conditional information cost *of Π with respect to the distribution $\vec{\mu}$ and the partitioning $\vec{\lambda}$ is:*

$$IC_{\vec{\mu}}(\Pi \mid \vec{\lambda}) \stackrel{\text{def}}{=} I\left(\vec{X}, \vec{Y};\ \Pi(\vec{X}, \vec{Y}, R_A, R_B) \mid \vec{D}\right).$$

Definition 6.5 (Conditional information complexity) *For $0 < \delta < 1$, the* conditional information complexity *of f with respect to a distribution $\vec{\mu}$, a partitioning $\vec{\lambda}$ of $\vec{\mu}$, and error parameter δ, denoted $IC_{\vec{\mu},\delta}(f \mid \vec{\lambda})$, is defined as the minimum conditional information cost of a δ-error protocol for f with respect to $\vec{\mu}$ and $\vec{\lambda}$.*

The following lemma shows that conditional information complexity is a lower bound on information complexity:

Lemma 6.6 (Conditioning lemma) *For any protocol Π, $IC_{\vec{\mu}}(\Pi) \geq IC_{\vec{\mu}}(\Pi \mid \vec{\lambda})$.*

Proof: Since $\Pi(\vec{X}, \vec{Y}, R_A, R_B)$ is conditionally independent of $\vec{D}$ given $\vec{X}, \vec{Y}$ (because the private coins of Π are independent of $\vec{D}$), the data processing inequality (Corollary 2.28) implies: $IC_{\vec{\mu}}(\Pi) = I\left(\vec{X}, \vec{Y};\ \Pi(\vec{X}, \vec{Y}, R_A, R_B)\right) \geq I\left(\vec{X}, \vec{Y};\ \Pi(\vec{X}, \vec{Y}, R_A, R_B) \mid \vec{D}\right) = IC_{\vec{\mu}}(\Pi \mid \vec{\lambda})$. □

6.2 A direct sum theorem for conditional information complexity

We now turn to the development of the direct sum theorem for the conditional information complexity of decomposable functions. First, we characterize the notion of decomposability. Later, we will impose a further restriction on the types of distributions $\vec{\mu} = \nu^d$ which will facilitate the direct sum theorem.

Definition 6.7 (Decomposable functions) *$f : \vec{\mathcal{L}} \to \mathcal{Z}$ is g-*decomposable with primitive *h, if it can be written as: $f(\vec{x}, \vec{y}) = g(h(x_1, y_1), \ldots, h(x_n, y_n))$, where $h : \mathcal{L} \to \{0,1\}$ and $g : \{0,1\}^n \to \mathcal{Z}$. Sometimes we simply write f* is decomposable with primitive *h.*

It is easy to see that SET-DISJOINTNESS is OR-decomposable with primitive AND: $\textsc{Disj}(\vec{x}, \vec{y}) = 1$ iff $\bigvee_{i=1}^{n} (x_i \wedge y_i)$. Here $\mathcal{L} = \{0,1\}^2$, $h = \wedge$, $g = \bigvee$.

Other examples of decomposable functions are the following.

1. Inner product: Again $\mathcal{L} = \{0,1\}^2$ and $h = \wedge$; g is the XOR of n bits.
2. L_∞ approximation: Here $\mathcal{L} = [m]^2$, for some m, $h(x,y) = 1$ if $|x - y| \geq \theta_U$ and 0 if $|x - y| \leq \theta_L$, for some $\theta_L < \theta_U$; g is the OR of n bits.
3. The direct sum problem for a function $h : \mathcal{L} \to \{0,1\}$: Here g is the identity.

Our lower bound technique works only for decomposable functions. However, the notion of decomposability can be generalized; for example, we can consider a different primitive h_i for each coordinate. See details in Section 6.8.

We will prove a direct sum theorem for conditional information complexity of functions with respect to any input distribution $\vec{\mu}$ from a certain class of such distributions. $\vec{\mu}$ will be required to be a product with respect to coordinates: $\vec{\mu} = \nu^n$, where ν is a distribution on $\mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$. (This can be extended to a product of different distributions; see details in Section 6.8). The distribution ν, which itself has two coordinates, is not necessarily a product distribution.

The proof of the direct sum theorem has two steps: (1) decompose conditional information complexity to n information quantities, each one depending only on a single coordinate; (2) show that these information quantities are lower bounded by the conditional information complexity of the primitive h.

Lemma 6.8 (Information cost decomposition lemma) *Let $(\vec{X}, \vec{Y}) \sim \vec{\mu}$, let $\vec{\lambda}$ be a distribution that partitions $\vec{\mu}$, and let $\vec{D} \sim \vec{\lambda}$. Then for any protocol Π,*

$$I\left(\vec{X}, \vec{Y};\ \Pi(\vec{X}, \vec{Y}, R_A, R_B) \mid \vec{D}\right) \ \geq\ \sum_j I\left(X_j, Y_j;\ \Pi(\vec{X}, \vec{Y}, R_A, R_B) \mid \vec{D}\right).$$

Proof: In the proof, we will abbreviate the random variable $\Pi(\vec{X}, \vec{Y}, R_A, R_B)$ by Π. By definition, $I\left(\vec{X}, \vec{Y};\ \Pi \mid \vec{D}\right) = H\left(\vec{X}, \vec{Y} \mid \vec{D}\right) - H\left(\vec{X}, \vec{Y} \mid \Pi, \vec{D}\right)$. Since the pairs $\{(X_j, Y_j)\}_{j \in [n]}$ are mutually independent conditioned on $\vec{D}$, $H\left(\vec{X}, \vec{Y} \mid \vec{D}\right) = \sum_j H\left(X_j, Y_j \mid \vec{D}\right)$. By the sub-additivity of entropy, $H\left(\vec{X}, \vec{Y} \mid \Pi, \vec{D}\right) \leq \sum_j H\left(X_j, Y_j \mid \Pi, \vec{D}\right)$. Thus, we have $I\left(\vec{X}, \vec{Y};\ \Pi \mid \vec{D}\right) \geq \sum_j I\left(X_j, Y_j;\ \Pi \mid \vec{D}\right)$. □

The next main step is to lower bound $I\left(X_j, Y_j;\ \Pi \mid \vec{D}\right)$ by $IC_{\nu,\delta}(h \mid \lambda)$. We achieve this by presenting, for each j, a reduction, which constructs a family of protocols for h that use a protocol Π for f as a subroutine, and whose average information cost with respect to ν is exactly $I\left(X_j, Y_j;\ \Pi \mid \vec{D}\right)$. We are able to carry out the reduction only for a restricted class of "collapsing distributions", defined next. In the definition we use the following embedding notation:

Definition 6.9 (Embedding) *For a vector $\vec{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_t$, a subset $\mathcal{S} \subseteq [t]$, $\mathcal{S} = \{j_1, \ldots, j_\ell\}$, and a vector $\vec{w} \in \mathcal{X}_{j_1} \times \cdots \times \mathcal{X}_{j_\ell}$, we define embed$(\vec{x}, \mathcal{S}, \vec{w})$ to be the t-dimensional vector in $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$, whose i-th component, $1 \leq i \leq t$, is defined as follows: embed$(\vec{x}, \mathcal{S}, \vec{w})[i] = x_i$ if $i \notin \mathcal{S}$, and embed$(\vec{x}, \mathcal{S}, \vec{w})[i] = w_k$ if $i = j_k \in \mathcal{S}$.*

Definition 6.10 (Collapsing distribution) *Let $f : \vec{\mathcal{L}} \to \mathcal{Z}$ be a g-decomposable function with primitive $h : \mathcal{L} \to \{0, 1\}$. We call $\vec{\mu}$ a* collapsing distribution for f *if for every $(\vec{x}, \vec{y})$ in the support of $\vec{\mu}$, every $j \in [n]$, and every $(u, v) \in \mathcal{L}$, we have $f(\text{embed}(\vec{x}, j, u), \text{embed}(\vec{y}, j, v)) = h(u, v)$.*

That is, if we have a pair $(u, v) \in \mathcal{L}$, and we pick values for the other coordinates using a collapsing distribution, then the value of f on the resulting input coincides with the value of h on (u, v). We note that not all functions f have a collapsing distribution over their inputs. However, the notion of collapsing distributions can be relaxed to accommodate other functions as well. See discussion in Section 6.8.

> Since our distribution ν for Set-Disjointness never places any mass on the pair $(1, 1)$, for every $(\vec{x}, \vec{y})$ in the support of $\vec{\mu} = \nu^n$, for every $j \in [n]$, $\bigvee_{i \neq j}(x_i \wedge y_i) = 0$, so for every $(u, v) \in \{0, 1\}^2$, Disj$(\textit{embed}(\vec{x}, j, u), \textit{embed}(\vec{y}, j, v)) = u \wedge v$.

A peculiar property of the distribution $\vec{\mu}$ we chose for SET-DISJOINTNESS is that is concentrated only on the zeroes of the function. Note however that the protocols we consider are supposed to work for any input and do not "know" that we provide them only with inputs on which the value of the function is 0. This is the reason why we are still able to obtain strong lower bounds on their information cost.

Lemma 6.11 (Reduction lemma) *Let Π be a δ-error protocol for a decomposable function $f : \vec{\mathcal{L}} \to \mathcal{Z}$ with primitive h. Let $\vec{\mu} = \nu^n$ be a collapsing distribution on $\vec{\mathcal{L}}$, and suppose λ is a distribution that partitions ν and $\vec{\lambda} = \lambda^n$ partitions $\vec{\mu}$. Let $(\vec{X}, \vec{Y}) \sim \vec{\mu}$ and $\vec{D} \sim \vec{\lambda}$. Then for all j,*

$$I\left(X_j, Y_j;\ \Pi(\vec{X}, \vec{Y}, R_A, R_B) \mid \vec{D}\right) \ \geq\ IC_{\nu,\delta}(h \mid \lambda).$$

Proof: As before, we abbreviate $\Pi(\vec{X}, \vec{Y}, R_A, R_B)$ by Π. Noting that $\vec{D} = (D_j, \vec{D}^{-j})$, we have $I\left(X_j, Y_j;\ \Pi \mid \vec{D}\right) = \sum_{\vec{d}} \Pr\left(D^{-j} = \vec{d}\right) \cdot I\left(X_j, Y_j;\ \Pi \mid D_j, \vec{D}^{-j} = \vec{d}\right)$. We wish to lower bound each term by $IC_{\nu,\delta}(h \mid \lambda)$. To this end, we would like to argue that conditioned on $\vec{D}^{-j} = \vec{d}$, Π is essentially a random variable distributed according to the transcript distribution of a communication protocol that computes h. It is in this step that we make crucial use of the condition $\vec{D}^{-j} = \vec{d}$.

We will describe a randomized protocol $\Pi_{j,\vec{d}}$ for the function $h : \mathcal{L} \to \{0, 1\}$. The protocol $\Pi_{j,\vec{d}}$ has the values j and $\vec{d}$ "hardwired" into it; here $\vec{d}$ is indexed by $[n]\backslash\{j\}$. On input pair (u, v), Alice and Bob realize a distribution η on $\vec{\mathcal{L}}$, defined as follows: $((\vec{U}, \vec{V}) \sim \eta)$ Alice sets $U_j = u$, Bob sets $V_j = v$; then they fill in $(U_i, V_i), i \neq j$, by sampling (independently of each other using private coin tosses) from the product distribution $\nu | \{D = d_i\}$. Let $(\vec{u}, \vec{v})$ be the value of $(\vec{U}, \vec{V})$ produced thus. Then Alice and Bob simulate $\Pi(\vec{u}, \vec{v}, R_A, R_B)$ and output whatever $\Pi(\vec{u}, \vec{v}, R_A, R_B)$ outputs. Since ν^n is a collapsing distribution of f, $f(\vec{u}, \vec{v}) = h(u, v)$, so if $\Pi(\vec{u}, \vec{v}, R_A, R_B) = f(\vec{u}, \vec{v})$ then $\Pi_{j,\vec{d}}(u, v, R_A, R_B) = h(u, v)$. Since Π is a δ-error protocol for f, it follows that $\Pi_{j,\vec{d}}$ is a δ-error protocol for h.

We will now show that $IC_\nu(\Pi_{j,\vec{d}}) = I\left(X_j, Y_j;\ \Pi(\vec{X}, \vec{Y}, R_A, R_B) \mid D_j, \vec{D}^{-j} = \vec{d}\right)$. Since $\Pi_{j,\vec{d}}(U, V, R_A, R_B) = \Pi(\vec{U}, \vec{V}, R_A, R_B)$, this amounts to showing

$$I\left(U, V;\ \Pi(\vec{U}, \vec{V}, R_A, R_B) \mid D\right) \ =\ I\left(X_j, Y_j;\ \Pi(\vec{X}, \vec{Y}, R_A, R_B) \mid D_j, \vec{D}^{-j} = \vec{d}\right).$$

This would follow if we show that the joint distribution of $\left(U, V, D, \Pi(\vec{U}, \vec{V}, R_A, R_B)\right)$ is identical to the joint distribution of $\left(X_j, Y_j, D_j, \Pi(\vec{X}, \vec{Y}, R_A, R_B)\right)$, conditioned on the event

$\vec{D}^{-j} = \vec{d}$. Note here that (since all the random variables have been spelled out) Π is a fixed function. Furthermore, R_A and R_B are independent of all the other random variables, $U_j = U$ and $V_j = V$, so it suffices to show that the joint distribution of $(U, V, D, \vec{U}^{-j}, \vec{V}^{-j})$ is identical to the joint distribution of $(X_j, Y_j, D_j, \vec{X}^{-j}, \vec{Y}^{-j})$, conditioned on the event $\vec{D}^{-j} = \vec{d}$. Now, U, V, D are independent of $\vec{U}^{-j}, \vec{V}^{-j}$, and similarly X_j, Y_j, D_j are independent of $\vec{D}^{-j}, \vec{X}^{-j}, \vec{Y}^{-j}$; therefore, it suffices to show that the joint distribution of (U, V, D) is identical to that of (X_j, Y_j, D_j), and that the joint distribution of $(\vec{U}^{-j}, \vec{V}^{-j})$ is identical to that of $(\vec{X}^{-j}, \vec{Y}^{-j})$, conditioned on the event $\vec{D}^{-j} = \vec{d}$. These can be easily seen to be true; we omit the tedious probability statements.

Finally, since $\Pi_{j,\vec{d}}$ is a δ-error protocol for h, we have $IC_\nu(\Pi_{j,\vec{d}} \mid \lambda) \geq IC_{\nu,\delta}(h \mid \lambda)$. Hence,

$$\begin{aligned} I\left(X_j, Y_j;\ \Pi \mid \vec{D}\right) &= \sum_{\vec{d}} \Pr\left(\vec{D}^{-j} = \vec{d}\right) \cdot IC_\nu(\Pi_{j,\vec{d}} \mid \lambda) \\ &\geq \sum_{\vec{d}} \Pr\left(\vec{D}^{-j} = \vec{d}\right) \cdot IC_{\nu,\delta}(h \mid \lambda) \ = \ IC_{\nu,\delta}(h \mid \lambda). \end{aligned}$$

□

Theorem 6.12 (Direct sum theorem) *Let $f : \vec{\mathcal{L}} \to \mathcal{Z}$ be a decomposable function with primitive h. Let $\vec{\mu} = \nu^n$ be a collapsing distribution for f. Let λ be any distribution that partitions ν so that $\vec{\lambda} = \lambda^n$ partitions $\vec{\mu}$. Then,*

$$IC_{\vec{\mu},\delta}(f \mid \vec{\lambda}) \ \geq \ n \cdot IC_{\nu,\delta}(h \mid \lambda).$$

Proof: Let Π be the optimal δ-error protocol for f with respect to information cost. We have $IC_{\vec{\mu},\delta}(f \mid \vec{\lambda}) = IC_{\vec{\mu}}(\Pi \mid \vec{\lambda}) = I\left(\vec{X}, \vec{Y};\ \Pi \mid \vec{D}\right)$. By Lemma 6.8 (Information cost decomposition lemma), this is at least $\sum_j I\left(X_j, Y_j;\ \Pi \mid \vec{D}\right)$. By Lemma 6.11 (Reduction lemma), this is at least $n \cdot IC_{\nu,\delta}(h \mid \lambda)$. □

Corollary 6.13 (of Proposition 6.3, Lemma 6.6, and Theorem 6.12) *With the notation and assumptions of Theorem 6.12,*

$$R_\delta(f) \ \geq \ IC_{\vec{\mu},\delta}(f) \ \geq \ IC_{\vec{\mu},\delta}(f \mid \vec{\lambda}) \ \geq \ n \cdot IC_{\nu,\delta}(h \mid \lambda).$$

For Set-Disjointness, we have $R_\delta(\text{Disj}) \geq IC_{\vec{\mu},\delta}(\text{Disj}) \geq n \cdot IC_\nu(\text{And} \mid \lambda)$. Thus it suffices to show an $\Omega(1)$ lower bound for the information cost of the 1-bit function And!

6.3 Information complexity lower bounds for primitive functions

The direct sum theorem of the previous section reduces the task of proving randomized communication complexity lower bounds for decomposable functions to proving (conditional) information complexity lower bounds for "primitive functions," where the communicating parties are given inputs from a small domain, and wish to check a fairly simple predicate. In this section we present a methodology for obtaining strong information complexity lower bounds for such primitive functions, based on a statistical view of communication complexity.

We continue the illustration of our techniques through the Set-Disjointness problem. As mentioned above, in order to prove an $\Omega(n)$ lower bound for Set Disjointness, we need a lower bound of C, for some constant $C > 0$, on the conditional information cost of the two-bit And problem, with respect to the distribution ν defined in Section 6.1. The contribution of our strong statistical methodology in this case is rather modest, since it affects only the constant in the lower bound. In our other examples, Multi-Party Set-Disjointness and L_∞ approximation, in contrast, the new methodology gives strong dependence on, respectively, the number of players and the approximation error.

6.3.1 An intuitive overview

The basic idea here is the following: we are given a primitive function h, a distribution ν over the inputs of h, and a protocol Π that computes h with small probability of error. We would like to argue that the transcript of Π, when running on inputs drawn from ν, has to reveal a lot of information about these inputs. Intuitively, if Π "behaves" very differently on different inputs of ν, then it leaks a lot of information about these inputs. The "behavior" of Π on a given input (x, y) can be captured by the transcript distribution obtained when running Π on (x, y) (the randomness comes from the private coins of Alice and Bob). Π behaves differently on two input pairs (x, y) and (w, z) if the corresponding transcript distributions are far from each other, under some measure of statistical distance. We conclude that if we prove a lower bound on the statistical distance between transcript distributions corresponding to different inputs in ν, we obtain an information cost lower bound for Π.

The question is then how to prove that two transcript distributions are far from each other. It is easy to see that if $h(x, y) \neq h(w, z)$, then simply from the fact Π computes h with small probability of error on any input, it follows that its transcript distributions on these two inputs have to be far from each other. The problem, however, is that ν is frequently concentrated only on inputs on which the value of h is the same (the distribution ν for AND is an example).

At this point we need to resort to the special structure of the distributions, imposed by the fact they correspond to transcripts of a communication complexity protocol. Recall that the most fundamental property of transcripts of deterministic protocols is the *rectangle property* (a.k.a. "cut & paste property"), which states that if the transcript of Π on two inputs (x, y) and (w, z) is τ, then τ is also the transcript of Π on the inputs (x, z) and (w, y). A simple argument shows a generalization of this property to private-coin randomized protocols: if Π is likely to have τ as a transcript on both (x, y) and (w, z), then it is likely to have τ also on (x, z) and on (w, y). Contrapositively, it follows that if the transcript distributions of Π on (x, y) and on (w, z) are far from each other, so are the transcript distributions of Π on (x, z) and on (w, y).

One could formalize the above argument and use some arbitrary distance measure (e.g., total variable distance) to quantify the difference in distance between the distributions on (x, y) and (w, z) and the distributions on (x, z) and (w, y). However, we show that if one uses the Hellinger distance, then an extremely simple argument proves that the two distances are exactly the same. We call this fact the *Hellinger cut & paste property* of randomized protocols.

Note now, that it is possible that $h(x, y) = h(w, z)$ but $h(x, z) \neq h(w, y)$ (e.g., in the case of AND and the inputs $(0, 1)$ and $(1, 0)$). Thus, from a lower bound on the distance between the distributions corresponding to (x, z) and (w, y), we can derive a lower bound on the distance between (x, y) and (w, z). This paves the way for proving distance lower bounds for distributions corresponding to arbitrary inputs.

The Hellinger cut & paste property is only one of four statistical properties of randomized communication complexity protocols we discover and use in obtaining lower bounds for information complexity of primitive functions. We discuss these properties in detail in Section 6.4. In the rest of this section we formalize the connection between information cost and statistical distance, and use it together with the Hellinger cut & paste property to prove the $\Omega(1)$ lower bound for AND.

6.3.2 From information theory to statistics

Some notation For a protocol Π and inputs x, y, we denote by $\Pi_{x,y}$ the distribution over transcripts corresponding to the execution of Π on x, y. Formally, $\Pi_{x,y} \stackrel{\text{def}}{=} \Pi(x, y, \rho_A, \rho_B)$, where ρ_A and ρ_B are the distributions over the private random strings of Alice and Bob. If the inputs are chosen according to some distribution $\nu = (\nu_X, \nu_Y)$, we denote by Π_{ν_X,ν_Y} the distribution over transcripts obtained by picking (x, y) according to ν and then running Π on (x, y).

We start with a generic lemma showing that the information cost of a protocol with respect to a distribution ν is lower bounded by a convex combination of statistical distances (in this case, Hellinger distances) between transcript distributions corresponding to inputs in the support of ν.

Lemma 6.14 (Information cost lower bound via Jensen-Shannon divergence) *Let $\nu = (\nu_X, \nu_Y)$ be a distribution on $\mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$, let λ be a distribution on $\mathcal{D}$ that partitions ν, let $(X, Y) \sim \nu$, and let $D \sim \nu$. For $d \in \mathcal{D}$, we denote by $\nu_{X,d}$ the conditional distribution $\nu_X|\{D = d\}$. Similarly, $\nu_{Y,d}$ is the conditional distribution $\nu_Y|\{D = d\}$. Then, for any protocol Π running on inputs in $\mathcal{L}$,*

$$IC_\nu(\Pi \mid \lambda) \;\geq\; \sum_{d\in\mathcal{D}} \lambda(d) \cdot \left(JS_{\nu_{X,d}}\left(\left\{\Pi_{x,\nu_{Y,d}}\right\}_{x\in\mathcal{X}}\right) + JS_{\nu_{Y,d}}\left(\left\{\Pi_{\nu_{X,d},y}\right\}_{y\in\mathcal{Y}}\right)\right).$$

Proof: In the following we abbreviate the random variable $\Pi(X, Y, R_A, R_B)$ by Π.

$$\begin{aligned}
IC_\nu(\Pi \mid \lambda) &= \\
&= I(X, Y;\, \Pi \mid D) \;=\; H(X, Y \mid D) - H(X, Y \mid \Pi, D) \\
&\geq H(X \mid D) + H(Y \mid D) - H(X \mid \Pi, D) - H(Y \mid \Pi, D) \\
&\qquad\qquad\qquad\qquad\qquad\qquad (X, Y \text{ are independent given } D) \\
&= I(X;\, \Pi \mid D) + I(Y;\, \Pi \mid D).
\end{aligned}$$

We continue to expand the first of the two terms. The other term is expanded analogously. $I(X;\, \Pi \mid D) = \sum_{d\in\mathcal{D}} \lambda(d) \cdot I(X;\, \Pi \mid D = d)$. Spelling out all the random variables hidden in Π, $\Pi = \Pi(X, Y, R_A, R_B)$, where R_A, R_B are the coins of Alice and Bob. Since R_A, R_B are independent of X, Y, and D, we can rewrite each term in the expectation as: $I(X_d;\ \Pi(X_d, Y_d, R_A, R_B))$, where $X_d \sim \nu_{X,d}$ and $Y_d \sim \nu_{Y,d}$. Note that X_d and Y_d are independent; we can therefore use the connection between mutual information and the generalized Jensen-Shannon divergence (Proposition 2.53):

$$I(X_d;\ \Pi(X_d, Y_d, R_A, R_B)) \;=\; JS_{\nu_{X,d}}\left(\left\{\Pi_{x,\nu_{Y,d}}\right\}_{x\in\mathcal{X}}\right).$$

□

6.3.3 Information complexity lower bound for And

We now formally prove the information complexity lower bound for AND. In the proof we use the Hellinger cut & paste property, which is proved in Section 6.4.5.

Theorem 6.15 *For the distributions ν and D defined in Section 6.1, and for any* $0 < \delta < 1/4$,

$$IC_{\nu,\delta}(\text{AND} \mid \lambda) \;\geq\; \frac{1}{4}\cdot\left(1 - 2\sqrt{\delta}\right).$$

Proof: Let Π be any δ-error protocol for AND. Using Lemma 6.14, we have:

$$IC_{\nu}(\Pi \mid \lambda) \;\geq\; \sum_{d\in\mathcal{D}} \lambda(d)\cdot\left(JS_{\nu_{X,d}}\left(\left\{\Pi_{x,\nu_{Y,d}}\right\}_{x\in\mathcal{X}}\right) + JS_{\nu_{Y,d}}\left(\left\{\Pi_{\nu_{X,d},y}\right\}_{y\in\mathcal{Y}}\right)\right),$$

where $\nu_{X,d} = \nu_X|\{D = d\}$ and $\nu_{Y,d} = \nu_Y|\{D = d\}$. Recall that in our case λ is a uniform distribution on $\{0,1\}$; $\nu_{X,0}$ is concentrated on 0 and $\nu_{Y,0}$ is uniform on $\{0,1\}$; $\nu_{X,1}$ is uniform on $\{0,1\}$ and $\nu_{Y,1}$ is concentrated on 0. Substituting this in the equation above and noting that in this case the generalized Jensen-Shannon divergence coincides with the standard Jensen-Shannon divergence, we have:

$$\begin{aligned}
IC_{\nu}(\Pi \mid \lambda) \;&\geq\; \frac{1}{2}\cdot\left(JS\left(\Pi_{0,0},\Pi_{0,1}\right) + JS\left(\Pi_{0,0},\Pi_{1,0}\right)\right) \\
&\geq\; \frac{1}{2}\cdot\left(h^2\left(\Pi_{0,0},\Pi_{0,1}\right) + h^2\left(\Pi_{0,0},\Pi_{1,0}\right)\right) \quad \text{(Proposition 2.51)} \\
&\geq\; \frac{1}{4}\cdot\left(h\left(\Pi_{0,0},\Pi_{0,1}\right) + h\left(\Pi_{0,0},\Pi_{1,0}\right)\right)^2 \quad \text{(Cauchy-Schwartz)} \\
&\geq\; \frac{1}{4}\cdot h^2\left(\Pi_{0,1},\Pi_{1,0}\right) \quad \text{(Triangle inequality).}
\end{aligned}$$

We thus managed to lower bound the information cost of Π in terms of the distance between the transcript distribution corresponding to the input $(0,1)$ and the transcript distribution corresponding to the input $(1,0)$. Note that the value of AND on both inputs is 0, and thus there is no straightforward lower bound on the distance between the two distributions. Here we resort to the Hellinger cut & paste property, stating that:

$$h^2\left(\Pi_{1,0}, \Pi_{0,1}\right) \;=\; h^2\left(\Pi_{1,1}, \Pi_{0,0}\right).$$

$\textsc{And}(0,0) \neq \textsc{And}(1,1)$, and thus, $h^2\left(\Pi_{0,0}, \Pi_{1,1}\right) \geq 1 - 2\sqrt{\delta}$. The theorem follows. □

As a corollary we obtain the $\Omega(n)$ lower bound for Set-Disjointness:

Theorem 6.16 *For any* $0 < \delta < 1/4$,

$$R_\delta\left(\textsc{Disj}_{n,2}\right) \;\geq\; IC_{\vec{\mu},\delta}(\textsc{Disj}_{n,2}) \;\geq\; n \cdot IC_{\nu,\delta}(\textsc{And} \mid \lambda) \;\geq\; \frac{n}{4} \cdot (1 - 2\sqrt{\delta}).$$

6.4 The statistical geometry of communication complexity

A randomized communication complexity protocol Π computing a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ can be viewed as a family of distributions $\mathcal{F}(\Pi) \stackrel{\text{def}}{=} \{\Pi_{x,y}\}_{(x,y)\in\mathcal{X}\times\mathcal{Y}}$. $\Pi_{x,y}$ is the distribution over transcripts $\tau \in \{0,1\}^*$ of Π obtained by running Π on the input pair (x, y). The source of randomness here is the private and public coins used by Alice and Bob.

The rules of the communication complexity game induce a certain structure on the family $\mathcal{F}(\Pi)$. In this section we explore the geometric structure of $\mathcal{F}(\Pi)$. We associate with each member $\Pi_{x,y}$ of $\mathcal{F}(\Pi)$ a point in a space, and use the statistical distance measures discussed in Section 2.4.1 to impose a distance on this space. Most of the distance measures we consider are not metrics, and thus the resulting space is not a metric space. Our study reveals that some of the distance measures, most notably the Hellinger distance, exhibit several surprising geometric properties of the space. These properties turn out to be very useful in the analysis of randomized protocols.

The geometric properties are based on three fundamental characteristics of randomized protocols: (1) a *rectangle property*, which generalizes the rectangle property of deterministic protocols; (2) the existence of low error hypothesis tests for certain sub-families of $\mathcal{F}(\Pi)$; and (3) a *Markov property* of one-way randomized protocols.

In the rest of the section we prove the three fundamental characteristics of randomized protocols, and then derive four geometric properties of transcript distributions. The last of these properties holds only for one-way communication protocols.

Some of the properties we state hold only for randomized protocols that use solely private coins. We discuss two-party communication protocols; unless stated otherwise, the properties discussed trivially generalize to multi-party protocols as well.

6.4.1 Rectangle property of randomized protocols

Perhaps the most fundamental property of deterministic communication complexity protocols is the *rectangle property* (see Section 2.2), which states that for every possible transcript $\tau \in \{0,1\}^*$, the set of inputs for which $\Pi(x,y) = \tau$ forms a *combinatorial rectangle*—a set $\mathcal{X}' \times \mathcal{Y}'$, where $\mathcal{X}' \subseteq \mathcal{X}$ and $\mathcal{Y}' \subseteq \mathcal{Y}$. This property is at the core of most of the lower bound techniques for deterministic communication complexity (cf. [81]).

In the following, we generalize this property to private-coin randomized protocols. We first define the rectangle property of a family of distributions:

Definition 6.17 (Rectangle property) *A family of distributions $\{\mu_{x,y}\}_{(x,y)\in\mathcal{X}\times\mathcal{Y}}$ on a domain $\mathcal{Z}$ is said to* satisfy the rectangle property*, if there exist two bivariate functions $q_X : \mathcal{X} \times \mathcal{Z} \to [0,1]$ and $q_Y : \mathcal{Y} \times \mathcal{Z} \to [0,1]$, such that for all pairs $(x,y) \in \mathcal{X} \times \mathcal{Y}$ and for all $z \in \mathcal{Z}$,*

$$\mu_{x,y}(z) \;=\; q_X(x,z) \cdot q_Y(y,z).$$

The rectangle property means that each distribution $\mu_{x,y}$ in the family can be "decomposed" into two independent parts: one that depends only on x and another that depends only on y. We prove that the family of distributions associated with a randomized protocol has this property:

Lemma 6.18 (Rectangle property of randomized protocols) *For every private coin randomized protocol Π running on inputs from $\mathcal{X}\times\mathcal{Y}$, the family $\mathcal{F}(\Pi)$ satisfies the rectangle property.*

Proof: First recall that by our convention, Π is well defined for every pair $(x,y) \in \mathcal{X} \times \mathcal{Y}$, even if the function it computes is defined only on a subset of $\mathcal{X} \times \mathcal{Y}$.

The proof directly follows from the rectangle property of deterministic protocols and the observation that any private-coin randomized protocol can be viewed as a deterministic protocol, if we augment its inputs with the private random strings of Alice and Bob. Formally, we view the input of Alice in Π as a pair (x, r_A), where $x \in \mathcal{X}$ and $r_A \in \mathcal{R}_A$ is the random string; similarly, (y, r_B) is the input of Bob. For each possible transcript $\tau \in \{0,1\}^*$

of Π, let $\mathcal{X}(\tau) \times \mathcal{Y}(\tau)$, where $\mathcal{X}(\tau) \subseteq \mathcal{X} \times \mathcal{R}_A$ and $\mathcal{Y}(\tau) \subseteq \mathcal{Y} \times \mathcal{R}_B$, be the combinatorial rectangle consisting of all the inputs $(x, r_A), (y, r_B)$ for which $\Pi(x, y, r_A, r_b) = \tau$.

For each $x \in \mathcal{X}$, let $\mathcal{X}(x, \tau)$ be the projection of $\mathcal{X}(\tau)$ on the pairs of the form $(x, *)$, and let $\mathcal{X}(x)$ be all the pairs of the form $(x, *)$. Similarly, define $\mathcal{Y}(y, \tau)$ and $\mathcal{Y}(y)$ for each $y \in \mathcal{Y}$. We define $q_X(x, \tau) = |\mathcal{X}(x, \tau)| \,/\, |\mathcal{X}(x)|$ and $q_Y(y, \tau) = |\mathcal{Y}(y, \tau)| \,/\, |\mathcal{Y}(y)|$.

Note that on input x, y, Alice chooses a pair (x, R_A) uniformly at random from $\mathcal{X}(x)$ and Bob chooses a pair (y, R_B) uniformly at random from $\mathcal{Y}(y)$. The transcript would be τ if and only if $(x, R_A) \in \mathcal{X}(x, \tau)$ and $(y, R_B) \in \mathcal{Y}(y, \tau)$. Since the choices of R_A and R_B are independent, it follows that $\Pi_{x,y}(\tau) = q_X(x, \tau) \cdot q_Y(y, \tau)$. $\square$

A generalization of the rectangle property holds for multi-party protocols as well. Given a multi-party protocol Π running on inputs in $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$, there exist t bivariate functions $q_j : \mathcal{X}_j \times \{0,1\}^* \to [0,1]$, such that for all $\vec{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_t$ and for all possible transcripts $\tau \in \{0,1\}^*$,

$$\Pi_{\vec{x}}(\tau) \;=\; \prod_{j=1}^{t} q_j(x_j, \tau).$$

6.4.2 Markov property of one-way randomized protocols

We next present a property which is stronger than the rectangle property and is possessed only by one-way randomized protocols.

Definition 6.19 (Markov property) *A family of distributions $\{\mu_{x,y}\}_{(x,y) \in \mathcal{X} \times \mathcal{Y}}$ on $\mathcal{Z}_1 \times \mathcal{Z}_2$ has the* Markov property*, if for every $x \in \mathcal{X}$ there exists a probability distribution q_x on $\mathcal{Z}_1$ and for every $y \in \mathcal{Y}$ there exists a $|\mathcal{Z}_1| \times |\mathcal{Z}_2|$ probability transition matrix M_y, such that:*

$$\forall x \in \mathcal{X}, y \in \mathcal{Y}, z_1 \in \mathcal{Z}_1, z_2 \in \mathcal{Z}_2, \quad \mu_{x,y}(z_1, z_2) \;=\; q_x(z_1) \cdot M_y(z_1, z_2).$$

The Markov property can be intuitively explained as follows: for any distribution $\mu_{x,y}$ in the family, the first part of the chosen object $z = z_1 \circ z_2$ depends only on x; the second part depends only on y and on the first part (that is, Z_2 is independent of X given Z_1).

It can be easily verified that any family of distributions satisfying the Markov property also satisfies the rectangle property. We next show that the transcript distributions of one-way protocols have the Markov property:

Lemma 6.20 (Markov property of one-way randomized protocols) *For any private coin randomized one-way protocol* Π *running on inputs in* $\mathcal{X} \times \mathcal{Y}$*, the family of distributions* $\mathcal{F}(\Pi) = \{\Pi_{x,y}\}_{(x,y)\in\mathcal{X}\times\mathcal{Y}}$ *has the Markov property.*

Proof: Any transcript $\tau \in \{0,1\}^*$ of Π can be partitioned into two contiguous parts $\tau = \tau_A \circ \tau_B$, where τ_A is the single message sent by Alice and τ_B is the single message sent by Bob.

τ_A is solely dependent on Alice's input and on her private coins. Thus, for every $x \in \mathcal{X}$, we define q_x to be the probability distribution over messages of Alice obtained when Alice gets the input x.

τ_B depends only on τ_A, the message from Alice, on Bob's input, and on his private coins. Thus, for each $y \in \mathcal{Y}$ and $\tau_A \in \{0,1\}^*$, we define p_{y,τ_A} to be the probability distribution on messages of Bob obtained when Bob's input is y and when the message from Alice is τ_A. We now define a probability transition matrix M_y, whose τ_A-th row is p_{y,τ_A}.

It is now straightforward to verify that with these definitions, $\mathcal{F}(\Pi)$ satisfies the Markov property. □

The generalization to multi-party protocols works as follows. Let Π be a private-coin randomized multi-party one-way protocol running on inputs in $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$. Every $x_1 \in \mathcal{X}_1$ is associated with a probability distribution q_{1,x_1} on $\{0,1\}^*$, Every $x_j \in \mathcal{X}_j$, $j > 1$, is associated with a probability transition matrix M_{j,x_j}. The Markov property states that for every input $\vec{x}$ and for every transcript $\tau = \tau_1 \circ \cdots \circ \tau_t$,

$$\Pi_{\vec{x}}(\tau) \;=\; q_{1,x_1}(\tau_1) \cdot M_{2,x_2}(\tau_1, \tau_2) \cdot \cdots \cdot M_{t,x_t}(\tau_{t-1}, \tau_t).$$

6.4.3 Hypothesis testing of transcript distributions

The second important property of a protocol's family of transcript distributions is that certain subsets of this family admit low error hypothesis tests (see definition in Section 2.4.2).

Two input pairs $(x,y), (w,z) \in \mathcal{X} \times \mathcal{Y}$ are called f-$\mathsf{different}$, if $f(x,y) \neq f(w,z)$. We show that for any subset $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y}$ of input pairs that are pairwise f-different there exists a multiple hypotheses test for $\{\Pi_{x,y}\}_{(x,y)\in\mathcal{S}}$.

Lemma 6.21 *Let Π be a δ-error randomized protocol for a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Then, for any set $\mathcal{S} \subseteq \mathcal{X} \times \mathcal{Y}$, for which $\forall (x, y) \neq (w, z) \in \mathcal{S}$, $f(x, y) \neq f(w, z)$, there exists a δ-error multiple hypotheses test for $\{\Pi_{x,y}\}_{(x,y)\in\mathcal{S}}$, whose sample cost is 1.*

Proof: Define $\mathcal{Z}' = f(\mathcal{S})$ to be the image of f on inputs from $\mathcal{S}$. Since f is 1-1 on $\mathcal{S}$, there is an inverse mapping $g : \mathcal{Z}' \to \mathcal{S}$.

Recall that one of the components of the formal definition of a communication complexity protocol (see Section 2.2) is a function $O : \{0, 1\}^* \to \mathcal{Z}$, which determines the output of the protocol based on the transcript of messages.

We define a hypothesis test $T : \{0, 1\}^* \to \mathcal{S}$ as follows: if $O(\tau) \in \mathcal{Z}'$, then $T(\tau) = g(O(\tau))$; otherwise $T(\tau) = (x_1, y_1)$ where (x_1, y_1) is some arbitrary fixed pair in $\mathcal{S}$. Now, for each $(x, y) \in \mathcal{S}$ we have,

$$\Pr_{\tau\sim\Pi_{x,y}} (T(\tau) \neq (x, y)) \;\leq\; \Pr_{\tau\sim\Pi_{x,y}} (O(\tau) \neq f(x, y)) \;<\; \delta,$$

where the latter inequality follows from the definition of a randomized communication protocol. □

6.4.4 Diameter property

The existence of good hypothesis tests for transcript distributions corresponding to f-different inputs immediately implies that the distance between any two such distributions is large. We call this the *diameter property* of randomized protocols:

Proposition 6.22 (Diameter property of randomized protocols) *Let $0 < \delta < 1/2$, and let Π be a δ-error randomized protocol for a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Then, for every two input pairs $(x, y), (w, z) \in \mathcal{X} \times \mathcal{Y}$ for which $f(x, y) \neq f(w, z)$,*

$$V\left(\Pi_{x,y}, \Pi_{w,z}\right) \;>\; 1 - 2\delta.$$

The proposition is a direct consequence of the connection between hypothesis testing and total variation distance (Proposition 2.56). Using the connection between total variation distance and Hellinger distance (Corollary 2.39), we can derive a lower bound also on the Hellinger distance between distributions corresponding to f-different inputs:

Corollary 6.23 *For $0 < \delta < 1/4$,* $\quad h^2\left(\Pi_{x,y}, \Pi_{w,z}\right) \;>\; 1 - 2\sqrt{\delta}.$

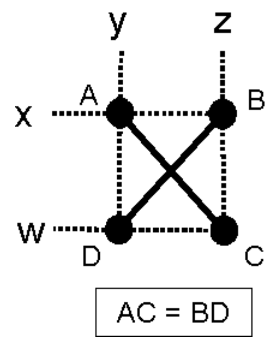

Figure 6.1: Hellinger cut & paste property

6.4.5 Hellinger cut & paste property

An extremely simple but yet surprising property of transcript distributions is the following cut & paste property: for any two input pairs (x, y) and (w, z), the Hellinger distance between $\Pi_{x,y}$ and $\Pi_{w,z}$ is identical to the Hellinger distance between $\Pi_{x,z}$ and $\Pi_{w,y}$. That is, the diagonals of the rectangle spanned by $\Pi_{x,y}, \Pi_{x,z}, \Pi_{w,y}$, and $\Pi_{w,z}$ are of equal length (see Figure 6.1).

Lemma 6.24 (Hellinger cut & paste property of randomized protocols) *For every private-coin randomized protocol* Π *running on inputs in* $\mathcal{X} \times \mathcal{Y}$*, and for every* $x, w \in \mathcal{X}$ *and* $y, z \in \mathcal{Y}$*,*

$$h(\Pi_{x,y}, \Pi_{w,z}) \;=\; h(\Pi_{x,z}, \Pi_{w,y}).$$

Proof: The proof uses the rectangle property of private-coin randomized protocols, and the multiplicativity of the Hellinger distance:

$$\begin{aligned} h^2(\Pi_{x,y}, \Pi_{w,z}) &= 1 - \sum_{\tau} \sqrt{\Pi_{x,y}(\tau) \cdot \Pi_{w,z}(\tau)} \\ &= 1 - \sum_{\tau} \sqrt{q_X(x,\tau) \cdot q_Y(y,\tau) \cdot q_X(w,\tau) \cdot q_Y(z,\tau)} \\ &= 1 - \sum_{\tau} \sqrt{\Pi_{x,z}(\tau) \cdot \Pi_{w,y}(\tau)} \;=\; h^2(\Pi_{x,z}, \Pi_{w,y}). \end{aligned}$$

□

Note that it is possible that $(x, y), (w, z)$ are f-different, but $(x, z), (w, y)$ are not (e.g., when f is the AND of two bits). Thus, the cut & paste property allows us to infer large distances even between inputs that are not f-different.

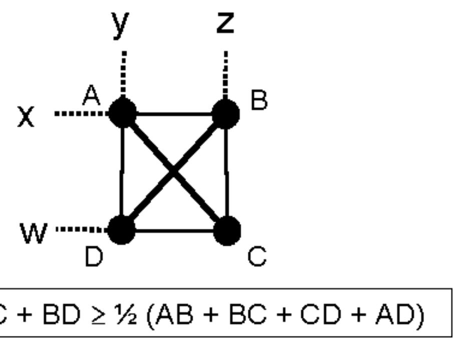

Figure 6.2: Hellinger inverse triangle inequality

The property generalizes to multi-party protocols as well. In order to formally state the property, we use the embedding notation (see Definition 6.9):

Lemma 6.25 *For every private-coin randomized multi-party protocol* Π *running on inputs in* $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$*, every two inputs* $\vec{x}, \vec{y} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_t$*, and every set* $\mathcal{S} \subseteq [t]$*,*

$$h\left(\Pi_{\vec{x}}, \Pi_{\vec{y}}\right) \;=\; h\left(\Pi_{embed(\vec{x},\mathcal{S},\vec{y}|_{\mathcal{S}})}, \Pi_{embed(\vec{y},\mathcal{S},\vec{x}|_{\mathcal{S}})}\right).$$

6.4.6 Hellinger inverse triangle inequality

Consider the rectangle spanned by $\Pi_{x,y}, \Pi_{x,z}, \Pi_{w,y}$, and $\Pi_{w,z}$. In a metric space the triangle inequality implies that the sum of the lengths of the diagonals in a rectangle is at most the sum of the lengths of its sides. We prove that the square Hellinger distance satisfies a (weak) inverse of this property: the sum of the diagonals is at least half of the sum of the sides (see Figure 6.2).

Lemma 6.26 (Hellinger inverse triangle inequality) *For any private coin randomized protocol* Π *running on inputs in* $\mathcal{X} \times \mathcal{Y}$*, and for any* $x, w \in \mathcal{X}$ *and* $y, z \in \mathcal{Y}$*,*

$$h^2\left(\Pi_{x,y}, \Pi_{w,y}\right) + h^2\left(\Pi_{x,z}), \Pi_{w,z}\right) \;\leq\; 2h^2\left(\Pi_{x,y}), \Pi_{w,z}\right).$$

Proof: The proof uses the rectangle property of randomized protocols, the multiplicativity of the Hellinger distance, and the arithmetic-geometric mean inequality.

$$\begin{aligned}
&h^2(\Pi_{x,y},\Pi_{w,y}) + h^2(\Pi_{x,z},\Pi_{w,z}) = \\
&= \; 2 - \sum_\tau \sqrt{\Pi_{x,y}(\tau)\cdot\Pi_{w,y}(\tau)} - \sum_\tau \sqrt{\Pi_{x,z}(\tau)\cdot\Pi_{w,z}(\tau)} \\
&= \; 2 - \sum_\tau \Big(\sqrt{q_X(x,\tau)\cdot q_Y(y,\tau)\cdot q_X(w,\tau)\cdot q_Y(y,\tau)} + \\
&\qquad\qquad \sqrt{q_X(x,\tau)\cdot q_Y(z,\tau)\cdot q_X(w,\tau)\cdot q_Y(z,\tau)}\Big) \\
&= \; 2 - \sum_\tau (q_Y(y,\tau)+q_Y(z,\tau))\sqrt{q_X(x,\tau)\cdot q_X(w,\tau)} \\
&\le \; 2 - 2\sum_\tau \sqrt{q_X(x,\tau)\cdot q_X(w,\tau)\cdot q_Y(y,\tau)\cdot q_Y(z,\tau)} \quad \text{(AM-GM inequality)} \\
&= \; 2h^2(\Pi_{x,y},\Pi_{w,z}).
\end{aligned}$$

□

6.4.7 Rényi semi-triangle inequality for one-way protocols

Previously we proved that the space induced by the transcript distributions and the Hellinger distance squared satisfies some form of an inverse triangle inequality. Here we prove the converse for transcript distributions emanating from one-way randomized protocols. We show that the Hellinger distance squared and its generalizations, the family of Rényi α-divergences, satisfy a semi-triangle inequality. Specifically, we show that in the rectangle spanned by $\Pi_{x,y}, \Pi_{x,z}, \Pi_{w,y}$, and $\Pi_{w,z}$, the sum of the lengths of the diagonals is at most $(1+\epsilon)$ times of the sum of the lengths of the sides (see Figure 6.3). The exact value of ϵ depends on α, and goes to 0 as α tends to 1.

Lemma 6.27 (Rényi semi-triangle inequality for one-way protocols) *Let Π be a private-coin randomized one-way protocol running on inputs in $\mathcal{X}\times\mathcal{Y}$. Let $x, w \in \mathcal{X}$ and $y, z \in \mathcal{Y}$. Then for any $\epsilon > 0$, if $\alpha > 1 - \epsilon^2/(4(1+\epsilon))$,*

$$D_\alpha(\Pi_{x,y} \,\|\, \Pi_{w,z}) \;\le\; (1+\epsilon)\cdot(D_\alpha(\Pi_{x,y} \,\|\, \Pi_{w,y}) + D_\alpha(\Pi_{x,y} \,\|\, \Pi_{x,z})).$$

Proof: The proof uses the Markov property of one-way randomized protocols.

For a probability vector μ on $\mathcal{Z}$ and a probability transition matrix R on $\mathcal{Z}_1\times\mathcal{Z}_2$, let $\mu\circ R$ denote the joint distribution on $\mathcal{Z}_1\times\mathcal{Z}_2$ where $(\mu\circ R)(i,j) = \mu(i)\cdot R(i,j)$. By

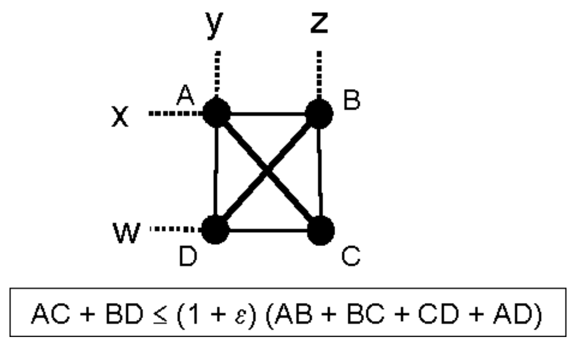

Figure 6.3: Rényi semi-triangle inequality

the Markov property of one-way protocols (Lemma 6.20), for all $x \in \mathcal{X}, y \in \mathcal{Y}$, $\Pi_{x,y}$ can be rewritten as $q_x \circ M_y$. Thus what we need to prove reduces to the following lemma; the proof uses convexity and analytical arguments and is described in Section A.3 of the Appendix.

Lemma 6.28 *Let μ, ν be probability distributions on $\mathcal{Z}_1$, and let R, S be probability transition matrices on $\mathcal{Z}_1 \times \mathcal{Z}_2$. Then for any $\epsilon > 0$, if $\alpha > 1 - \epsilon^2/(4(1+\epsilon))$,*

$$D_\alpha(\mu \circ R \parallel \nu \circ S) \;\; \leq \;\; (1+\epsilon) \cdot (D_\alpha(\mu \circ R \parallel \nu \circ R) + D_\alpha(\mu \circ R \parallel \mu \circ S)).$$

$\square$

In order to describe the generalization to multi-party protocols, we introduce the following notation: for a vector $\vec{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_t$, and an index $i \in [t-1]$, we denote by $\vec{x}_{\leq i}$ the projection of $\vec{x}$ on its first i coordinates and by $\vec{x}_{>i}$ its projection on the last $t-i$ coordinates.

Lemma 6.29 (Rényi semi-triangle inequality for multi-party protocols) *Let Π be a private-coin randomized one-way multi-party protocol running on inputs in $\mathcal{X}_1 \times \cdots \times \mathcal{X}_t$. Let $\vec{x}, \vec{y} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_t$. Then, for any $i \in [t-1]$ and for any $\epsilon > 0$, if $\alpha > 1 - \epsilon^2/(4(1+\epsilon))$,*

$$D_\alpha\left(\Pi_{\vec{x}} \parallel \Pi_{\vec{y}}\right) \;\; \leq \;\; (1+\epsilon) \cdot (D_\alpha\left(\Pi_{\vec{x}} \parallel \Pi_{\vec{y}_{\leq i} \circ \vec{x}_{>i}}\right) + D_\alpha\left(\Pi_{\vec{x}} \parallel \Pi_{\vec{x}_{\leq i} \circ \vec{y}_{>i}}\right)).$$

Proof: By the generalization of the Markov property of one-way randomized protocols to an arbitrary number of players, we can associate with x_1 a probability distributions q_{x_1}, and with $x_2, \ldots, x_t$ probability transition matrices $M_{2,x_2}, \ldots, M_{t,x_t}$, such that for every transcript $\tau = \tau_1 \circ \cdots \circ \tau_t$, $\Pi_{\vec{x}}(\tau) = q_{x_1}(\tau_1) \cdot \prod_{j=2}^{t} M_{j,x_j}(\tau_{j-1}, \tau_j)$.

For two probability transition matrices M_1 on $\mathcal{Z}_1 \times \mathcal{Z}_2$ and M_2 on $\mathcal{Z}_2 \times \mathcal{Z}_3$, let $M_1 \circ M_2$ be a probability transition matrix on $\mathcal{Z}_1 \times (\mathcal{Z}_2 \times \mathcal{Z}_3)$, satisfying $(M_1 \circ M_2)(z_1, (z_2, z_3)) = M_1(z_1, z_2) \cdot M_2(z_2, z_3)$. Define $q_{\vec{x}_{\le i}} \stackrel{\text{def}}{=} q_{x_1} \circ M_{2,x_2} \circ \cdots \circ M_{i,x_i}$ and $M_{\vec{x}_{>i}} \stackrel{\text{def}}{=} M_{i+1,x_{i+1}} \circ \cdots \circ M_{t,x_t}$. We thus obtain: $\Pi_{\vec{x}} = q_{\vec{x}_{\le i}} \circ M_{\vec{x}_{>i}}$. Similarly, $\Pi_{\vec{y}} = q_{\vec{y}_{\le i}} \circ M_{\vec{y}_{>i}}$. The lemma now follows from Lemma 6.28. □

6.5 Lower bounds for Multi-Party Set-Disjointness

We define Multi-Party Set-Disjointness, the generalization of Set Disjointness to any number of players, as $\text{Disj}_{n,t}(\vec{x_1}, \ldots, \vec{x_t}) \stackrel{\text{def}}{=} \bigvee_{j=1}^{n} \bigwedge_{i=1}^{t} x_{i,j}$, where the $\vec{x_i}$'s are length n Boolean vectors. The input domain of $\text{Disj}_{n,t}$ is a subset of $(\{0,1\}^n)^t$: if one views the vectors $\vec{x}_1, \ldots, \vec{x}_t$ as subsets of $[n]$ then these subsets are either pairwise disjoint (i.e., $\vec{x}_i \cap \vec{x}_j = \emptyset$ for $i \neq j$) or mutually intersecting (i.e., $\bigcap_i \vec{x}_i \neq \emptyset$, and each two sets intersect only at $\bigcap_i \vec{x}_i$)[1].

Clearly, $\text{Disj}_{n,t}$ is Or-decomposable, and the induced "primitive" functions are all And_t—the t-bit And. The legal inputs for And_t are the all-zero $\vec{0}$, the all-one $\vec{1}$, and the standard unit vectors $\vec{e}_i$ with 1 in the i-th position.

In Section 5.4 we showed an $\tilde{O}(n/t)$ upper bound (suppressing logarithmic factors) on the simultaneous communication complexity of the problem[2]. This immediately implies an identical upper bound on its general communication complexity. An examination of the proof of the upper bound reveals that it holds also for the one-way communication complexity. We also proved in Section 5.4 a matching $\Omega(n/t)$ lower bound for simultaneous communication complexity. Alon, Matias, and Szegedy [4] showed a $\Omega(n/t^4)$ lower bound on the general communication complexity of the problem. This is also the best known lower bound on its one-way communication complexity. In the following we prove two lower bounds for the problem: (1) we improve the lower bound for general communication complexity to $\Omega(n/t^2)$; and (2) we improve the lower bound for one-way communication complexity to $\Omega(n/t^{1+\epsilon})$ for arbitrarily small $\epsilon > 0$.

[1] This is a slight variation from the definition in Chapter 5, in which we additionally required the mutual intersection to consist of a single point. We chose this definition, for the sake of simplicity only. The bounds we discuss can be made to work for the original version too.

[2] The upper bound was stated for the variant of the problem that requires a unique intersection; the proof, however, carries over to non-unique intersection as well.

6.5.1 General communication lower bound

Theorem 6.30 *For any* $0 < \delta < 1/4$,

$$R_\delta\left(\textsc{Disj}_{n,t}\right) \;\geq\; \frac{n}{t^2} \cdot \left(1 - 2\sqrt{\delta}\right).$$

Proof: We will employ the direct sum technique. We define a distribution ν on inputs of $\textsc{And}_t$ as follows: λ is a uniform distribution on $[t]$, and $D \sim \lambda$. ν is uniform on $\left\{\vec{0}, \vec{e}_D\right\}$. It follows that $\vec{\mu} = \nu^n$ is a collapsing distribution for $\textsc{Disj}_{n,t}$. Thus, all we need to prove is a lower bound on the conditional information complexity of $\textsc{And}_t$ with respect to ν and λ.

Let Π be any δ-error protocol that computes $\textsc{And}_t$. $\vec{X} = (X^1, \ldots, X^t)$ denotes a random input for $\textsc{And}_t$ chosen according to ν. ν_d denotes the conditional distribution of ν given $D = d$ and $\nu_{i,d}$ denotes the conditional distribution of ν_i (the i-th coordinate of ν) given $D = d$. Finally, we denote by $\mathit{embed}(\nu_d, i, x)$, the distribution obtained from ν_d by forcing its i-th coordinate to x and choosing the rest of the coordinates $j \neq i$ according to $\nu_{j,d}$ (since ν_d is product, this makes sense).

A generalization of the connection between information cost and the Jensen-Shannon divergence (Lemma 6.14) to an arbitrary number of players implies that:

$$IC_\nu(\Pi \mid \lambda) \;\geq\; \sum_{d \in \mathcal{D}} \lambda(d) \cdot \left(\sum_{i=1}^{t} JS_{\nu_{i,d}}\left(\left\{\Pi_{\mathit{embed}(\nu_d, i, x)}\right\}_{x \in \mathcal{X}}\right)\right).$$

In our case, $\nu_{i,d}$ is concentrated on 0 if $i \neq d$ and $\nu_{i,i}$ is uniform on $\{0,1\}$. Thus, substituting in the above equation, and noting that also in this case the generalized Jensen-Shannon divergence coincides with the standard one, we have:

$$IC_\nu(\Pi \mid \lambda) \;\geq\; \frac{1}{t} \cdot \sum_{i=1}^{t} JS\left(\Pi_{\vec{e}_i}, \Pi_{\vec{0}}\right) \;\geq\; \frac{1}{t} \sum_{i=1}^{t} h^2\left(\Pi_{\vec{e}_i}, \Pi_{\vec{0}}\right), \tag{6.1}$$

where the last inequality follows from Proposition 2.51. Applying Cauchy-Schwartz gives:

$$IC_\nu(\Pi \mid \lambda) \;\geq\; \frac{1}{t^2} \cdot \left(\sum_{i=1}^{t} h\left(\Pi_{\vec{e}_i}, \Pi_{\vec{0}}\right)\right)^2.$$

Applying the triangle inequality, we can lower bound the first two terms of the sum by $h\left(\Pi_{\vec{e}_1}, \Pi_{\vec{e}_2}\right)$. We now use the generalization of the Hellinger cut & paste property to an arbitrary number of players (see Section 6.4.5):

$$h\left(\Pi_{\vec{e}_1}, \Pi_{\vec{e}_2}\right) \;=\; h\left(\Pi_{\vec{e}_{1,2}}, \Pi_{\vec{0}}\right),$$

where $\vec{e}_{1,2}$ is the vector that has 1 in its first two coordinates and 0's everywhere else. We can now continue and apply a triangle inequality and the cut & paste property to this term and the third term of the sum to obtain a lower bound of $h\left(\Pi_{\vec{e}_{1,2,3}}, \Pi_{\vec{0}}\right)$. After $n-1$ such steps we have:

$$IC_{\nu}(\Pi \mid \lambda) \;\geq\; \frac{1}{t^2} \cdot h^2\left(\Pi_{\vec{1}}, \Pi_{\vec{0}}\right).$$

Since $\textsc{And}_t(\vec{1}) \neq \textsc{And}_t(\vec{0})$, we can use the diameter property (see Section 6.4.4) to derive a $1 - 2\sqrt{\delta}$ lower bound on $h^2\left(\Pi_{\vec{1}}, \Pi_{\vec{0}}\right)$. The theorem follows. $\square$

6.5.2 One-way communication lower bound

The reason we lost a factor of t in the proof of Theorem 6.30 is that the square of the Hellinger distance does not satisfy the triangle inequality, and we thus had to resort to the Cauchy-Schwartz inequality (incurring the loss), in order to move to the Hellinger distance. For one-way protocols we can get around this by moving from the squared Hellinger distance to the Rényi α-divergence, for an appropriately chosen α. We can then apply the semi-triangle inequality for one-way protocols (see Section 6.4.7).

Theorem 6.31 *For any $0 < \delta < 1/4$ and any $0 < \epsilon < 1$,*

$$R_{\delta}^{\rightarrow}\left(\textsc{Disj}_{n,t}\right) \;\geq\; \frac{n}{t^{1+\epsilon}} \cdot \frac{\epsilon^2 \cdot \ln^2 2}{8} \cdot \left(1 - 2\sqrt{\delta}\right).$$

Proof: We use the same distributions ν and $\vec{\mu}$ as in the proof of Theorem 6.30, and employ the same argument. The first point of departure is the derivation after Equation 6.1. Instead of using Cauchy-Schwartz, we bound the Hellinger squared distances by Rényi α-divergence, for an $\alpha = \alpha(\epsilon) > 1/2$ to be chosen later. Note that the square of the Hellinger distance is the Rényi 1/2-divergence, and thus we can use the connection among the different α-divergences (Proposition 2.48) to obtain:

$$IC_{\nu}(\Pi \mid \lambda) \;\geq\; \frac{1}{t} \cdot \frac{1}{2\alpha} \cdot \sum_{i=1}^{t} D_{\alpha}\left(\Pi_{\vec{e}_i} \,\|\, \Pi_{\vec{0}}\right).$$

We next use the Rényi semi-triangle inequality for one-way multi-party protocols. For simplicity of exposition, we assume that t is a power of 2. Consider, for example, the first two terms in the above sum. We partition the input vectors $\vec{0}, \vec{e}_1, \vec{e}_2 \in \{0,1\}^t$ into two parts: one of length 1 and the other of length $t-1$. Thus, if $x = 0, w = 1, y = 0^{t-1}$, and $z = 10^{t-2}$, then $\vec{0} = (x, y), \vec{e}_1 = (w, y), \vec{e}_2 = (x, z)$. We choose $\epsilon' = \epsilon \cdot \ln 2$, and $\alpha \geq 1 - \epsilon'^2/(4(1+\epsilon'))$, so we can apply Lemma 6.29, and obtain:

$$\begin{aligned} D_\alpha\left(\Pi_{\vec{0}} \parallel \Pi_{\vec{e}_1}\right) + D_\alpha\left(\Pi_{\vec{0}} \parallel \Pi_{\vec{e}_2}\right) &= \\ = \ & D_\alpha\left(\Pi_{x,y} \parallel \Pi_{w,y}\right) + D_\alpha\left(\Pi_{x,y} \parallel \Pi_{x,z}\right) \\ \geq \ & \frac{1}{1+\epsilon'} \cdot D_\alpha\left(\Pi_{x,y} \parallel \Pi_{w,z}\right) \ = \ \frac{1}{1+\epsilon'} \cdot D_\alpha\left(\Pi_{\vec{0}} \parallel \Pi_{\vec{e}_{1,2}}\right). \end{aligned}$$

We can apply the same argument to all the $t/2$ pairs of consecutive terms in the above sum, and then recurse for $\log t$ levels. The final outcome is:

$$\begin{aligned} IC_\nu(\Pi \mid \lambda) &\geq \frac{1}{t \cdot 2\alpha} \cdot \left(\frac{1}{1+\epsilon'}\right)^{\log t} \cdot D_\alpha\left(\Pi_{\vec{1}} \parallel \Pi_{\vec{0}}\right) \\ &\geq \frac{1}{t \cdot 2\alpha} \cdot \frac{1}{t^\epsilon} \cdot D_\alpha\left(\Pi_{\vec{1}} \parallel \Pi_{\vec{0}}\right). \end{aligned}$$

Noting that $D_\alpha\left(\Pi_{\vec{1}} \parallel \Pi_{\vec{0}}\right) \geq 2(1-\alpha) \cdot h^2\left(\Pi_{\vec{1}}, \Pi_{\vec{0}}\right) \geq 2(1-\alpha)(1-2\sqrt{\delta})$, due to the diameter property (Corollary 6.23), and that $2(1-\alpha)/(2\alpha) \geq \epsilon^2 \cdot \ln^2 2/8$, by our choice of α and ϵ', completes the proof. □

6.6 Lower bounds for L_∞ and L_p distance estimation

In the $\mathrm{L}_\infty\langle n, m, \theta_L, \theta_U\rangle$ problem, Alice and Bob are, respectively, given two n-dimensional vectors, $\vec{x}$ and $\vec{y}$ from $[m]^n$ with the following promise: either $|x_i - y_i| \leq \theta_L$ for all i, or for some i, $|x_i - y_i| \geq \theta_U$. The function $\mathrm{L}_\infty\langle n, m, \theta_L, \theta_U\rangle = 1$ iff the latter case holds.

Saks and Sun [113] showed the distributional communication complexity (and thus, by Yao's Lemma, also the randomized communication complexity) of $\mathrm{L}_\infty\langle n, m, \theta_L, \theta_U\rangle$ is at least $\Omega(n/(\theta_U/\theta_L)^2)$. However, their bound holds only for the one-way communication model. In the following we strengthen their result, and show the bound holds also for general communication.

Theorem 6.32 *For any* $0 < \delta < 1/4$*,*

$$R_\delta\left(\mathrm{L}_\infty\langle n, m, \theta_L, \theta_U\rangle\right) \geq \frac{n}{(\theta_U/\theta_L)^2} \cdot \frac{1}{4} \cdot \left(1 - 2\sqrt{\delta}\right).$$

Proof: For simplicity of notation, in this proof use $[m]$ to denote the set $\{0, 1, \ldots, m-1\}$, instead of the set $\{1, 2, \ldots, m\}$.

By dividing the input vectors by θ_L, we assume, without loss of generality, that $\theta_L = 1$, and we will write θ for θ_U. Note that $\mathrm{L}_\infty\langle n, m, 1, \theta_U\rangle$ is $\textsc{Or}$-decomposable, since $\mathrm{L}_\infty\langle n, m, 1, \theta_U\rangle(\vec{x}, \vec{y}) = \bigvee_j \textsc{Dist}\langle m, 1, \theta_U\rangle(x_j, y_j)$, where $\textsc{Dist}\langle m, 1, \theta_U\rangle(x, y) = 1$, if $|x - y| \geq \theta$ and $\textsc{Dist}\langle m, 1, \theta_U\rangle(x, y) = 0$ if $|x - y| \leq 1$.

We thus use the direct sum technique with the input distribution $\vec{\mu} = \nu^n$, where $\nu = (\nu_X, \nu_Y)$ is defined as follows. Let λ denote the uniform distribution on $\mathcal{D} \stackrel{\mathrm{def}}{=} [m] \times \{0, 1\} \setminus \{(0, 1), (m - 1, 0)\}$, and let $D \sim \lambda$. If $D = (d, 0)$, then ν_X is concentrated on d and ν_Y is uniform on $\{d, d + 1\}$; if $D = (d, 1)$, then ν_Y is concentrated on d and ν_X is uniform on $\{d - 1, d\}$. It is easy to see that λ partitions ν. Furthermore, since $\textsc{Dist}\langle m, 1, \theta_U\rangle(x, y) = 0$ for all (x, y) in the support of ν, ν^n is a collapsing distribution for $\mathrm{L}_\infty\langle n, m, 1, \theta_U\rangle$. It thus suffices to prove that $IC_{\nu,\delta}(\textsc{Dist}\langle m, 1, \theta_U\rangle \mid \lambda) \;\geq\; 1/\theta^2 \cdot (1 - 2\sqrt{\delta})/4$.

It is enough to prove the lower bound for $m = \theta + 1$. Let Π be any δ-error protocol for $\textsc{Dist}\langle \theta + 1, 1, \theta\rangle$. By Lemma 6.14,

$$IC_\nu(\Pi \mid \lambda) \;\geq\; \sum_{(d,b)\in\mathcal{D}} \lambda(d, b) \cdot \left(JS_{\nu_{X,d,b}} \left(\left\{ \Pi_{x,\nu_{Y,d,b}} \right\}_{x\in\mathcal{X}} \right) + JS_{\nu_{Y,d,b}} \left(\left\{ \Pi_{\nu_{X,d,b},y} \right\}_{y\in\mathcal{Y}} \right) \right).$$

Here, $\nu_{X,d,b}$ is the conditional distribution ν_X given the event $D = (d, b)$, and similarly for $\nu_{Y,d,b}$. Plugging in the distributions in our case, we have:

$$\begin{aligned} IC_\nu(\Pi \mid \lambda) \;&\geq\; \frac{1}{2\theta} \left(\sum_{d=1}^{\theta} JS\left(\Pi_{d,d}, \Pi_{d-1,d}\right) + \sum_{d=0}^{\theta-1} JS\left(\Pi_{d,d}, \Pi_{d,d+1}\right) \right) \\ &\geq\; \frac{1}{2\theta} \left(\sum_{d=1}^{\theta} h^2\left(\Pi_{d,d}, \Pi_{d-1,d}\right) + \sum_{d=0}^{\theta-1} h^2\left(\Pi_{d,d}, \Pi_{d,d+1}\right) \right). \end{aligned}$$

Applying Cauchy-Schwartz on the two sums, and then applying the triangle inequality successively to the resulting terms yields:

$$IC_\nu(\Pi \mid \lambda) \;\geq\; \frac{1}{4\theta^2} \cdot h^2\left(\Pi_{0,0}, \Pi_{\theta,\theta}\right).$$

We cannot use the diameter property to bound the latter distance, since the value of $\textsc{Dist}$ on both inputs is 0. The Hellinger cut & paste property will not help either, since $\textsc{Dist}$ has the same value also on $(0, \theta)$ and $(\theta, 0)$. We thus need to resort to the Hellinger inverse triangle inequality (Lemma 6.26):

$$h^2\left(\Pi_{0,0}, \Pi_{\theta,\theta}\right) \;\geq\; \frac{1}{2}\left(h^2\left(\Pi_{0,0}, \Pi_{\theta,0}\right) + h^2\left(\Pi_{0,\theta}, \Pi_{\theta,\theta}\right)\right) \;\geq\; 1 - 2\sqrt{\delta},$$

where the last inequality follows from the diameter property. The theorem follows. $\square$

The $L_p\langle n, m, \theta_L, \theta_U\rangle$ problem is defined analogously to the $L_\infty\langle n, m, \theta_L, \theta_U\rangle$ problem, except that now Alice and Bob wish to determine whether the L_p distance between their vectors is above θ_U or below θ_L. As a corollary, we obtain a lower bound for this problem too:

Theorem 6.33 *For any $0 < \delta < 1/4$, and for any $p > 2$,*

$$R_\delta\left(L_p\langle n, m, \theta_L, \theta_U\rangle\right) \;\geq\; \frac{n^{1-2/p}}{(\theta_U/\theta_L)^2} \cdot \frac{1}{4} \cdot \left(1 - 2\sqrt{\delta}\right).$$

Proof: The L_∞ and the L_p distances have the following relationship:

$$L_\infty(\vec{x}, \vec{y}) \;\leq\; L_p(\vec{x}, \vec{y}) \;\leq\; n^{1/p} \cdot L_\infty(\vec{x}, \vec{y}).$$

Thus, given a protocol Π that computes $L_p\langle n, m, \theta_L, \theta_U\rangle$ with error probability δ, we can immediately derive a protocol Π' that computes $L_\infty\langle n, m, \theta_U, \theta_L/n^{1/p}\rangle$ with the same error probability and the same communication cost. The lower bound now follows from Theorem 6.32. $\square$

Tightness of the bounds Saks and Sun [113] proved that the lower bound for $L_\infty\langle n, m, \theta_L, \theta_U\rangle$ is tight, by presenting a one-way protocol that matches it. There is no known matching protocol for the $L_p\langle n, m, \theta_L, \theta_U\rangle$ problem. The best protocol, which uses the L_∞ protocol and the connection between L_∞ and L_p, requires $O(n^{1+2/p}/(\theta_U/\theta_L)^2)$ communication.

6.7 Data stream lower bounds

In this section we use the communication complexity lower bounds obtained in the previous sections to derive new data stream space lower bounds for the frequency moments and for L_p estimation.

6.7.1 Frequency moments

Recall the frequency moments problem, defined in Section 4.4.2. For $k = 0, 1, 2$, Alon *et al.* [4] presented data stream algorithms that estimate the k-th frequency moment, F_k, to within a multiplicative error of $1 \pm \epsilon$ using space which is logarithmic in m and n and polynomial in $1/\epsilon$. For $k \geq 3$ their algorithms use space $O(m^{1-1/k})$ (and logarithmic in

n and polynomial in $1/\epsilon$). By reducing the communication complexity of $\textsc{Disj}_{m,m^{1/k}}$ to the space complexity of F_k, they showed that the latter requires space $\Omega(m^{1-5/k})$ in the data stream model. This implies that for $k > 5$, F_k indeed requires $m^{\Omega(1)}$ space. It remained open to settle the space complexity of F_k for $2 < k \leq 5$.

In the following we show that F_k requires space $\Omega(m^{1-(2+\gamma)/k})$ for arbitrarily small $\gamma > 0$. This shows that F_k for any $k > 2$ (even real k's) requires polynomial space, finally settling the space complexity of the frequency moments.

In order to prove the space lower bound we need to adapt the reduction of Alon *et al.* [4] to our case, since their proof worked for a specific hard distribution over instances of $\textsc{Disj}_{m,m^{1/k}}$.

Theorem 6.34 *For all $k \geq 1$, $0 < \delta < 1/4$, and any $0 < \gamma < 1$,*

$$DS_{\epsilon,\delta}(\mathrm{F}_k) \;\geq\; \Omega\left(m^{1-(2+\gamma)/k} \cdot \frac{1}{(1+3\epsilon)^{(2+\gamma)/k}} \cdot \frac{\gamma^2}{1+\gamma}\right).$$

Proof: Let B be any randomized oblivious read-once branching program that (ϵ,δ)-approximates F_k. Let S be the space used by B. We use the reduction from one-way communication complexity to data stream space complexity outlined in Section 3.3.3. We use B to construct a δ-error one-way protocol Π for $\textsc{Disj}_{m,t}$, where $t = ((1+3\epsilon)m)^{1/k}$.

Let $S_1, \ldots, S_t \subseteq [m]$ be the input sets for the t players. The sets translate into an input stream for F_k, where $n = \sum_{i=1}^{t} |S_i|$, in the obvious way: first all the elements of S_1, then all the elements of S_2, and so forth.

The players in Π jointly simulate B as follows: the first player starts the simulation by running B on the elements of S_1. She then transmits the state of B (S bits) to the second player. The second player resumes the execution of B on her part of the stream (the elements of S_2) and sends the reached state to the third player. In the end of the simulation, the t-th player gets b, the output of B, and announces the sets to be disjoint if and only if $b \leq (1+\epsilon)m$.

If the input sets are disjoint, then each element has a frequency of at most one in the stream, and therefore F_k is at most m. On the other hand, if the sets are uniquely intersecting, then there is at least one element whose frequency is t, and therefore F_k is at least $t^k = (1+3\epsilon)m$. Since B produces an answer b, which with probability $1-\delta$ is in the interval $((1-\epsilon)\mathrm{F}_k, (1+\epsilon)\mathrm{F}_k)$, then if the sets are disjoint, then b is with probability $1-\delta$ at most $m(1+\epsilon)$, and if the sets are intersecting, then b is with probability $1-\delta$ at least

$(1-\epsilon)(1+3\epsilon)m > (1+\epsilon)m$. Thus, our protocol is correct on any input with probability $1-\delta$.

We next derive a lower bound on S. Note that the protocol Π uses $S(t-1)+1 \leq St$ bits of communication. By Theorem 6.31, this communication, for $0 < \delta < 1/4$ and for any $0 < \gamma < 1$, is at least $\Omega\left(m/t^{1+\gamma} \cdot \gamma^2/(1+\gamma)\right)$. Therefore, $S \geq \Omega\left(m/t^{2+\gamma} \cdot \gamma^2/(1+\gamma)\right) \geq \Omega\left(m^{1-(2+\gamma)/k}/(1+3\epsilon)^{(2+\gamma)/k} \cdot \gamma^2/(1+\gamma)\right)$. □

6.7.2 L_p distance estimation

Alon *et al.* [4], Feigenbaum *et al.* [46], Fong and Strauss [48], and Indyk [68] showed data stream algorithms that estimate the L_p distance, for any $p \in [1,2]$, to within a multiplicative error of $1 \pm \epsilon$, using space which is logarithmic in m and n and polynomial in $1/\epsilon$. Saks and Sun [113] proved that estimating the L_p distance, for any $p > 2$, in the data stream model, even to within a huge multiplicative error of n^{ϵ}, requires space $n^{\Omega(1)}$. Their proof held only for the one-pass model. We are able to prove that even if the data stream algorithm is allowed a constant number of passes over the data, the same lower bound holds.

Theorem 6.35 *For any $p > 2$, if ϵ and δ satisfy $0 < \epsilon < \frac{1}{4} - \frac{1}{2p}$, $0 < \delta < \frac{1}{4}$, then for any number of passes $\ell \geq 1$,*

$$DS^{\ell}_{n^{\epsilon},\delta}(\mathrm{L}_p) \;\geq\; \Omega\left(\frac{n^{1-4\epsilon-2/p}}{\ell}\right).$$

Proof: Let B be a randomized oblivious read-ℓ-times branching program that (n^{ϵ},δ)-approximates L_p and that accepts the data items as follows: first all the elements of the first vector (in some arbitrary order) and then all the elements of the second vector. Let S the space used by B.

We use B to construct a protocol Π that solves $\mathrm{L}_p\langle n, n^{1+2\epsilon}, n, n^{1+2\epsilon}\rangle$, and whose communication cost is $2\ell S$. In the protocol Π, Alice and Bob jointly simulate B on their inputs. In the end of the simulation, they decide the distance between their vectors is at most n if and only if B's output is at most $n^{1+\epsilon}$. Since B outputs a value in the interval $(\mathrm{L}_p/n^{\epsilon}, \mathrm{L}_p \cdot n^{\epsilon})$ with probability at least $1-\delta$, then the protocol is correct with that much probability.

Since B reads the data ℓ times, Alice and Bob need to exchange roles as the active simulating player at most 2ℓ times. Each such exchange requires sending the current state of B (S bits). Thus, the total communication is at most $2\ell S$. The lower bound follows from Theorem 6.33. □

Unfortunately, there is no matching upper bound for the problem. Saks and Sun [113] showed a data stream algorithm that (n^{ϵ}, δ)-approximates L_∞ in $O(n^{1-4\epsilon})$ space. Using the connection between norms, this derives an $O(n^{1-4\epsilon+2/p})$ space algorithm for L_p.

6.8 Extensions of the technique

In this section we briefly review various possible extensions of the lower bound methodology, which make it usable for more general classes of protocols, functions, and input distributions.

Multi-party protocols The generalization to multi-party protocols is straightforward, and in fact was already illustrated in our lower bound for Multi-Party Set Disjointness.

Public coins In general, lower bounds for private-coin protocols yield almost the same lower bounds for public-coin protocols, because by Newman's Theorem [92] the two models differ by an additive logarithmic factor. This is not the case, however, for simultaneous communication complexity. Our methods can be extended to deal with public-coin protocols as well, and can therefore derive lower bounds for public-coin simultaneous communication complexity.

To deal with public-coin protocols, we need to modify the definition of information cost. Intuitively, we would like the information cost to measure the amount of additional information the transcript reveals to us about the inputs, after observing the public random bits used by the protocol. Formally, the information cost of a protocol Π whose public random string is R and whose private random strings are R_A and R_B is defined as follows:

$$IC_{\vec{\mu}}(\Pi) \stackrel{\text{def}}{=} I\left(\vec{X}, \vec{Y};\ \Pi(\vec{X}, \vec{Y}, R, R_A, R_B) \mid R\right).$$

Note that conditioning on R is essential. Because, consider a protocol Π in which Alice xors her input with the public random bits and sends the result to Bob. Bob obviously can recover Alice's input, and therefore the transcript effectively has full information about Alice's input. If we do not condition on R, however, the transcript has no information at all about Alice's input, because all we get to observe is a random string.

All the definitions and proofs we stated for private-coin protocols carry over to public-coin protocols with obvious modifications. In the proofs we use the characterization of public-coin protocols as distributions over private-coin protocols. We omit the details.

Functions over non-product domains We assumed that functions for which we prove the lower bound are over a domain $\vec{\mathcal{L}} = \mathcal{L}^n$, where $\mathcal{L} \subseteq \mathcal{X} \times \mathcal{Y}$. There are two implicit restrictions here: (1) that $\vec{\mathcal{L}}$ is a product; and (2) that the domain of all the coordinates is the same. The second restriction was put only to be able to denote all the coordinate domains with a single notation. The first restriction is more substantial; the technique applies to some functions over non-product domains (e.g., Multi-Party Set-Disjointness with unique intersection), conditioned on having a product collapsing distribution $\vec{\mu} = \nu^n$ over its inputs. That is, even though the domain is non-product, the input distribution has to be product.

Generalizations of decomposability Decomposability was defined with respect to a single primitive function h. The technique works also if different coordinates have different primitive functions; that is $f(\vec{x}, \vec{y}) = g(h_1(x_1, y_1), \ldots, h_n(x_n, y_n))$.

A more substantial generalization is the following. The definition of collapsing distributions implicitly implies that if a g-decomposable function f has such a distribution, then necessarily g is sensitive to each of its coordinates. As pointed out by Xiaodong Sun [117], the technique can be generalized to work for g-decomposable functions, for which g is not sensitive to all its coordinates but only to some of them. Our direct sum theorem will change then to be $IC_{\vec{\mu},\delta}(f \mid \vec{\lambda}) \geq S(g) \cdot IC_{\nu,\delta}(h \mid \lambda)$, where $S(g)$ is the sensitivity of g.

6.9 Subsequent work and open problems

The most interesting open problem following this work is whether the technique can be generalized to a larger class of functions. One can view the decomposable functions as trees of depth 2: the root contains the gate "g" and the leaves contain the gate "h". In subsequent work, Jayram, Kumar, and Sivakumar [71] were able to extend the technique to AND-OR trees of depth 3. Specifically, they showed a new lower bound for AND of Set-Disjointness problems. Could we make the technique work for trees of higher depth?

The technique gives us direct lower bounds on the randomized communication complexity of problems. In some situations, however, it is beneficial to have the stronger lower bound on the distributional communication complexity with respect to some distribution. In subsequent work, Jayram [70] showed how to obtain such bounds as well using our technique. Interestingly, in his proof in order to prove a μ-distributional complexity lower bound one uses lower bounds on the information complexity of the problem with respect to some distribution μ', which is different from μ.

In most of the problems we studied in this chapter there are some gaps left. First and foremost, the general communication complexity of MULTI-PARTY SET-DISJOINTNESS is not yet resolved, since there is a factor of t difference between the upper and the lower bounds. In subsequent work, Khot [77] improved the lower bound to $\Omega(n/t^{3/2})$. We conjecture that the upper bound is tight, and thus one needs to prove an $\Omega(n/t)$ lower bound. The one-way communication complexity lower bound for the problem is arbitrarily close to the optimum, but is not yet optimal too. Khot [77] showed an optimal $\Omega(n/t)$ lower bound for one-way communication.

There is a gap also between the upper and the lower bounds on the communication complexity of the L_p estimation problem. This gap shows up in the data stream space complexity of the problem as well.

The data stream space complexity of the frequency moments is also not completely resolved, since there is a gap between the lower bound we proved (essentially, $\Omega(m^{1-2/k})$) and the best upper bound ($O(m^{1-1/k})$). Here we conjecture that the upper bound is not tight.

Bibliography

[1] J. Abello, P. M. Pardalos, and M. G.C. Resende, editors. *Handbook of Massive Data Sets*. Kluwer Academic Publishers, 2002.

[2] L. Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theoretical Computer Science*, 157(2):139–159, 1996.

[3] M. Ajtai, T. S. Jayram, R. Kumar, and D. Sivakumar. Approximate counting of inversions in a data stream. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 370–379, 2002.

[4] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[5] A. Amir, G. Benson, and M. Farach. Let sleeping files lie: Pattern matching in Z-compressed files. *Journal of Computer and System Sciences*, 52(2):299–307, 1996.

[6] L. Babai. Communication complexity. Technical Report TR-2001-09, University of Chicago, 2001.

[7] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory. In *Proceedings of the 27th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 337–347, 1986.

[8] L. Babai, P. Kimmel, and S. V. Lokam. Simultaneous messages vs. communication. In *Proceedings of the 12th Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 900 of *Springer Lecture Notes in Computer Science*, pages 361–372, 1995.

[9] L. Babai and P. G. Kimmel. Randomized simultaneous messages. In *Proceedings of the 12th IEEE Conference on Computational Complexity (CCC)*, pages 239–246, 1997.

[10] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols, pseudorandom generators for LOGSPACE, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992.

[11] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proceedings of the 21st ACM Symposium on Principles of Database Systems (PODS)*, pages 1–16, 2002.

[12] Z. Bar-Yossef. Sampling Lower Bounds via Information Theory, 2002. Manuscript.

[13] Z. Bar-Yossef, O. Goldreich, and A. Wigderson. Deterministic amplification of space-bounded probabilistic algorithms. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity (CCC)*, pages 188–198, 1999.

[14] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002. To appear.

[15] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. Information theory methods in communication complexity. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity (CCC)*, pages 93–102, 2002.

[16] Z. Bar-Yossef, T. S. Jayram, R. Kumar, D. Sivakumar, and L. Trevisan. Counting distinct elements in a data stream. In *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, 2002. To appear.

[17] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. Sampling algorithms: Lower bounds and applications. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 266–275, 2001.

[18] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 623–632, 2002.

[19] Z. Bar-Yossef, O. Reingold, R. Shaltiel, and L. Trevisan. Streaming computation of combinatorial objects. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity (CCC)*, pages 165–174, 2002.

[20] D. Barbará, W. DuMouchel, C. Faloutsos, P. J. Haas, J. M. Hellerstein, Y. E. Ioannidis, H. V. Jagadish, T. Johnson, R. T. Ng, V. Poosala, K. A. Ross, and K. C. Sevcik. New Jersey data reduction report. *Data Engineering Bulletin*, 20(4):3–45, 1997.

[21] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *Proceedings of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–269, 2000.

[22] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985.

[23] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[24] A. Z. Broder, M. S. Charikar, A. Frieze, and M. Mitzenmacher. Min-wise independent permutations. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 327–336, 1998.

[25] A. Z. Broder, S. C. Glassman, and M. S. Manasse. Syntactic clustering of the web. In *Proceedings of the 6th International World Wide Web Conference (WWW6)*, pages 1157–1166, 1997.

[26] R. E. Bryant. Graph-based algorithms for boolean function manipulations. *IEEE Transactions on Computers*, 35:677–691, 1986.

[27] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*. To appear.

[28] J. Bunge and M. Fitzpatrick. Estimating the number of species: A review. *Journal of the American Statistical Association*, 88:364–373, 1993.

[29] J. Burbea and C. R. Rao. Entropy differential metric, distance and divergence measures in probability spaces: A unified approach. *Journal of Multivariate Analysis*, 12:575–596, 1982.

[30] J. Burbea and C. R. Rao. On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory*, 28:489–495, 1982.

[31] R. Canetti, G. Even, and O. Goldreich. Lower bounds for sampling algorithms for estimating the average. *Information Processing Letters*, 53:17–25, 1995.

[32] A. Chakrabarti, Y. Shi, A. Wirth, and A. C-C. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proceedings of the 42nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 270–278, 2001.

[33] A. K. Chandra, M. L. Furst, and R. J. Lipton. Multi-party protocols. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC)*, pages 94–99, 1983.

[34] S.-F. Chang. Compressed-domain techniques for image/video indexing and manipulation. In *Invited article in IEEE International Conference on Image Processing (ICIP), Special Session on Digital Image/Video Libraries and Video-on-demand*, pages 314–317, 1995.

[35] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 380–388, 2002.

[36] M. S. Charikar, S. Chaudhuri, R. Motwani, and V. Narasayya. Towards estimation error guarantees for distinct values. In *Proceedings of the 19th Annual ACM Symposium on Principles of Database Systems (PODS)*, pages 268–279, 2000.

[37] M. S. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 693–703, 2002.

[38] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *American Mathematical Society*, 23:493–507, 1952.

[39] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, second edition, 2001.

[40] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., 1991.

[41] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for monte carlo estimation. In *Proceedings of the 36th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 142–149, 1995.

[42] X. Deng, C. H. Papadimitriou, and S. Safra. On the complexity of equilibria. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 67–71, 2002.

[43] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, 1996.

[44] W. DuMouchel, C. Volinsky, T. Johnson, C. Cortes, and D. Pregibon. Squashing flat files flatter. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 6–15, 1999.

[45] R. M. Fano. Class Notes for Transmission of Information, 1952. Course 6.574, MIT, Cambridge, MA.

[46] J. Feigenbaum, S. Kannan, M. Strauss, and M. Viswanathan. An approximate L^1-difference algorithm for massive data streams. In *Proceedings of the 40th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 501–511, 1999.

[47] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

[48] J. H. Fong and M. Strauss. An approximate L^p-difference algorithm for massive data streams. In *Proceedings of the 17th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 193–204, 2000.

[49] P. B. Gibbons. Distinct sampling for highly-accurate answers to distinct values queries and event reports. In *Proceedings of the 27th International Conference on Very Large Data Bases (VLDB)*, pages 541–550, 2001.

[50] P. B. Gibbons and Y. Matias. Synopsis data structures for massive data sets. In *Proceedings of the 10th IEEE Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 909–910, 1999.

[51] P. B. Gibbons, Y. Matias, and V. Poosala. Fast incremental maintenance of approximate histograms. In *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB)*, pages 466–475, 1997.

[52] P. B. Gibbons and S. Tirthapura. Estimating simple functions on the union of data streams. In *Proceedings of the ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 281–291, 2001.

[53] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International Statistical Review*, 2002. to appear.

[54] A. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 389–398, 2002.

[55] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[56] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 2000. TR00-020.

[57] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. In *Proceedings of the 42nd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 460–469, 2001.

[58] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 58–66, 2001.

[59] S. Guha, N. Koudas, and K. Shim. Data streams and histograms. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 471–475, 2001.

[60] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. In *Proceedings of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 359–366, 2000.

[61] S. Guiasu. *Information Theory with Applications*. McGraw-Hill, 1977.

[62] P. J. Haas, J. F. Naughton, S. Seshadri, and L. Stokes. Sampling-based estimation of the number of distinct values of an attribute. In *Proceedings of the 21st International Conference on Very Large Data Bases (VLDB)*, pages 311–322, 1995.

[63] T. S. Han. Non-negative entropy measures of multivariate symmetric correlations. *Information & Control*, 36:133–156, 1978.

[64] M. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. In *DIMACS series in Discrete Mathematics and Theoretical Computer Science*, volume 50, pages 107–118, 1999.

[65] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[66] R. Impagliazzo, N. Nisan, and A. Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 356–364, 1994.

[67] P. Indyk. *High-dimensional Computational Geometry*. PhD thesis, Computer Science Department, Stanford University, 2000.

[68] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Proceedings of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 189–197, 2000.

[69] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[70] T. S. Jayram. Private communication, 2002.

[71] T. S. Jayram, R. Kumar, and D. Sivakumar. A note on randomized versus nondeterministic communication complexity, 2002. Manuscript.

[72] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 712–721, 2001.

[73] B. Kalyanasundaram and G. Schnitger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(5):545–557, 1992.

[74] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. *SIAM Journal on Discrete Mathematics*, 3(2):255–265, 1990.

[75] R. M. Karp, C. H. Papadimitriou, and S. Shenker. A simple algorithm for finding frequent elements in streams and bags. Available at `http://www.cs.berkeley.edu/∼christos`, 2002.

[76] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

[77] S. Khot. Private communication, 2002.

[78] J. M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 599–608, 1997.

[79] I. Kremer, N. Nisan, and D. Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.

[80] S. Kullback and A. Leibler. On information and sufficiency. *IEEE Transactions on Information Theory*, 22:79–86, 1951.

[81] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[82] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 614–623, 1998.

[83] L. Le Cam and G. L. Yang. *Asymptotics in Statistics - Some Basic Concepts*. Springer-Verlag, 1990.

[84] T. Lengauer. VLSI Theory. In *Handbook of Theoretical Computer Science*, volume A, chapter 16, pages 835–868. Elsevier, 1990.

[85] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[86] G. S. Manku, S. Rajagopalan, and B. G. Lindsay. Approximate medians and other quantiles in one pass and with limited memory. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 426–435, 1998.

[87] G. S. Manku, S. Rajagopalan, and B. G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 251–262, 1999.

[88] Y. Matias, J. S. Vitter, and M. Wang. Dynamic maintenance of wavelet-based histograms. In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB)*, pages 101–110, 2000.

[89] P. B. Miltersen. On the cell probe complexity of polynomial evaluation. *Theoretical Computer Science*, 143(1):167–174, 1995.

[90] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

[91] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 384–393, 1999.

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

[93] I. Newman and M. Szegedy. Public vs. private coins flips in one round communication games. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 561–570, 1996.

[94] N. Nisan. CREW PRAMs and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991.

[95] N. Nisan. The communication complexity of threshold gates. In D. Miklós, V. T. SóS, and T. Szönyi, editors, *Combinatorics: Paul Erdös is Eighty*, volume 1, pages 301–315. Janos Bolyai Math. Society, Budapest, Hungary, 1993.

[96] N. Nisan and I. Segal. The communication complexity of efficient allocation problems. In *DIMACS workshop on Computational Issues in Game Theory and Mechanism Design*, 2001.

[97] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

[98] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[99] F. Olken. *Random Sampling from Databases*. PhD thesis, Computer Science Division, U.C. Berkeley, 1993.

[100] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[101] C. H. Papadimitriou and M. Sipser. Communication complexity. *Journal of Computer and System Sciences*, 28(2):260–269, 1984.

[102] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 468–474, 1992.

[103] P. Pudlák, V. Rödl, and J. Sgall. Boolean circuits, tensor ranks and communication complexity. *SIAM Journal on Computing*, 26(3):605–633, 1997.

[104] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[105] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[106] R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM*, 39(3):736–744, 1992.

[107] A. A. Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1):81–93, 1990.

[108] A. A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. In *Proceedings of the 8th International Conference on Fundamentals of Computation Theory (FCT)*, volume 529 of *Springer Lecture Notes in Computer Science (LNCS)*, pages 47–60, 1991.

[109] A. A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.

[110] R.-D. Reiss. *Approximate Distributions of Order Statistics*. Springer-Verlag, 1989.

[111] A. Rényi. On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability*, pages 547–561, 1960.

[112] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[113] M. Saks and X. Sun. Space lower bounds for distance approximation in the data stream model. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 360–369, 2002.

[114] L. Schulman and V. V. Vazirani. Majorizing estimators and the approximation of #P-complete problems. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 288–294, 1999.

[115] D. Siegmund. *Sequential Analysis - Tests and Confidence Intervals*. Springer-Verlag, 1985.

[116] M. Sipser. Expanders, randomness, or time versus space. *Journal of Computer and System Sciences*, 36(3):379–383, 1988.

[117] X. Sun. Private communication, 2002.

[118] F. Topsøe. Some inequalities for information divergence and related measures of discrimination. *RGMIA Research Report Collection*, 2(1), 1999. Article 9.

[119] L. Trevisan. Private communication, 2001.

[120] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[121] H. L. Van Trees. *Detection, Estimation, and Modulation Theory.* Jon Wiley & Sons, Inc., 1968.

[122] V. N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, Inc., 1998.

[123] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, XVI(2):264–280, 1971.

[124] J. S. Vitter. External memory algorithms. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems (PODS)*, pages 119–128, 1998.

[125] J. von zur Gathen and J. R. Roche. Polynomials with two values. *Combinatorica*, 17(3):345–362, 1997.

[126] I. Wegener. *The Complexity of Boolean Functions.* Series in Computer Science. Wiley-Teubner, Stuttgart, 1987.

[127] A. C-C. Yao. Some complexity questions related to distributed computing. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.

[128] A. C-C. Yao. Lower bounds by probabilistic arguments. In *Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 420–428, 1983.

[129] D. Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11(4):345–367, 1997.

[130] D. Zuckerman. Private communication, 2000.

Appendix A

Some Proofs

A.1 Statistical distance measures

A.1.1 Hellinger distance

Proposition 2.37 (restated) *For any distributions μ_1, ν_1 on $\mathcal{X}_1$ and μ_2, ν_2 on $\mathcal{X}_2$,*

$$1 - h^2(\mu_1 \times \mu_2, \nu_1 \times \nu_2) \;=\; (1 - h^2(\mu_1, \nu_1)) \cdot (1 - h^2(\mu_2, \nu_2)).$$

Proof:

$$\begin{aligned}
1 - h^2(\mu_1 \times \mu_2, \nu_1 \times \nu_2) &= \sum_{x_1, x_2 \in \mathcal{X}} \sqrt{(\mu_1 \times \mu_2)(x_1, x_2)(\nu_1 \times \nu_2)(x_1, x_2)} \\
&= \sum_{x_1, x_2 \in \mathcal{X}} \sqrt{\mu_1(x_1)\mu_2(x_2)\nu_1(x_1)\nu_2(x_2)} \\
&= \left(\sum_{x_1 \in \mathcal{X}} \sqrt{\mu_1(x_1)\nu_1(x_1)} \right) \cdot \left(\sum_{x_2 \in \mathcal{X}} \sqrt{\mu_2(x_2)\nu_2(x_2)} \right) \\
&= (1 - h^2(\mu_1, \nu_1)) \cdot (1 - h^2(\mu_2, \nu_2)).
\end{aligned}$$

$\square$

Proposition 2.38 (restated)

$$h^2(\mu, \nu) \;\leq\; V(\mu, \nu) \;\leq\; h(\mu, \nu)\sqrt{2 - h^2(\mu, \nu)}.$$

Proof:

$$\begin{aligned}
h^2(\mu,\nu) &= \frac{1}{2}\sum_{x\in\mathcal{X}}\left(\sqrt{\mu(x)}-\sqrt{\nu(x)}\right)^2 \\
&\le \frac{1}{2}\sum_{x\in\mathcal{X}}\left|\sqrt{\mu(x)}-\sqrt{\nu(x)}\right|\left(\sqrt{\mu(x)}+\sqrt{\nu(x)}\right) \\
&= \frac{1}{2}\sum_{x\in\mathcal{X}}|\mu(x)-\nu(x)| \;=\; V(\mu,\nu)\,.
\end{aligned}$$

Using the Cauchy-Schwartz inequality:

$$\begin{aligned}
V(\mu,\nu) &= \frac{1}{2}\sum_{x\in\mathcal{X}}|\mu(x)-\nu(x)| \\
&= \frac{1}{2}\sum_{x\in\mathcal{X}}\left|\sqrt{\mu(x)}-\sqrt{\nu(x)}\right|\left(\sqrt{\mu(x)}+\sqrt{\nu(x)}\right) \\
&\le \sqrt{\frac{1}{2}\sum_{x\in\mathcal{X}}\left(\sqrt{\mu(x)}-\sqrt{\nu(x)}\right)^2\cdot\frac{1}{2}\sum_{x\in\mathcal{X}}\left(\sqrt{\mu(x)}+\sqrt{\nu(x)}\right)^2} \\
&= h(\mu,\nu)\cdot\sqrt{\frac{1}{2}\left(\sum_{x\in\mathcal{X}}\mu(x)+\sum_{x\in\mathcal{X}}\nu(x)+2\sum_{x\in\mathcal{X}}\sqrt{\mu(x)\nu(x)}\right)} \\
&= h(\mu,\nu)\cdot\sqrt{2-h^2(\mu,\nu)}.
\end{aligned}$$

□

A.1.2 KL distance

Proposition 2.45 (restated)

$$D_{KL}(\mu\,\|\,\nu) \;\ge\; \frac{2}{\ln 2}\,h^2(\mu,\nu) \;\ge\; \frac{1}{\ln 2}\,V^2(\mu,\nu)\,.$$

Proof: The second inequality follows from the connection between the Hellinger and total variation distances (Proposition 2.38). For the first inequality, we use the concavity of log and apply Jensen's inequality: $\log \mathrm{E}[X] \ge \mathrm{E}[\log X]$. We also use the fact that $e^x \ge 1-x$ for all $0 \le x \le 1$:

$$\begin{aligned}
D_{KL}(\mu\,\|\,\nu) &= \sum_x \mu(x)\log\frac{\mu(x)}{\nu(x)} \;=\; -2\sum_x \mu(x)\log\sqrt{\frac{\nu(x)}{\mu(x)}} \\
&\ge -2\log\sum_x\sqrt{\nu(x)\mu(x)} \;=\; -2\log\left(1-h^2(\mu,\nu)\right) \\
&\ge \frac{2}{\ln 2}\,h^2(\mu,\nu)\,.
\end{aligned}$$

□

A.1.3 Rényi α-divergence

Proposition 2.48 (restated) $\forall \alpha < \beta$,

$$\frac{\alpha}{\beta} D_\beta\left(\mu \,\|\, \nu\right) \;\leq\; D_\alpha\left(\mu \,\|\, \nu\right) \;\leq\; \frac{1-\alpha}{1-\beta} D_\beta\left(\mu \,\|\, \nu\right).$$

Proof: We use Hölder's inequality (for vectors v, u and for p, q that satisfy $1/p + 1/q = 1$, $|\langle v, u\rangle| \leq ||v||_p \cdot ||u||_q$) with $p = \beta/\alpha$ and $q = \beta/(\beta - \alpha)$:

$$\begin{aligned}
&1 - D_\alpha\left(\mu \,\|\, \nu\right) \\
&\quad= \sum_x \mu(x)^\alpha \nu(x)^{1-\alpha} \;=\; \sum_x \mu(x)^\alpha \nu(x)^{\alpha/\beta - \alpha} \cdot \nu(x)^{1-\alpha/\beta} \\
&\quad\leq \left(\sum_x \left(\mu(x)^\alpha \nu(x)^{\alpha/\beta-\alpha}\right)^{\beta/\alpha}\right)^{\alpha/\beta} \cdot \left(\sum_x \left(\nu(x)^{1-\alpha/\beta}\right)^{\beta/(\beta-\alpha)}\right)^{(\beta-\alpha)/\beta} \\
&\quad= \left(\sum_x \mu(x)^\beta \nu(x)^{1-\beta}\right)^{\alpha/\beta} \cdot \left(\sum_x \nu(x)\right)^{(\beta-\alpha)/\beta} \\
&\quad= \left(1 - D_\beta\left(\mu \,\|\, \nu\right)\right)^{\alpha/\beta}.
\end{aligned}$$

We now use the following simple analytic claim:

Claim A.0.1 *For any $0 \leq \epsilon, \delta \leq 1$ (excluding the case $\epsilon = 1$ and $\delta = 0$), $(1-\epsilon)^\delta \leq 1 - \delta\epsilon$.*

Proof: The cases $\delta = 0, 1$ are trivial. So assume $\delta \in (0, 1)$ and consider the function $f(\epsilon) = 1 - \delta\epsilon - (1-\epsilon)^\delta$. We need to show f is non-negative in the interval $[0, 1]$. Taking the derivative of f, we have: $f'(\epsilon) = \delta(1/(1-\epsilon)^{1-\delta} - 1) \geq 0$, since $1 - \epsilon \leq 1$ and $1 - \delta > 0$. Therefore, f is non-decreasing in the interval $[0, 1]$, implying its minimum is obtained at $\epsilon = 0$. Since $f(0) = 0$, we have that $f(\epsilon) \geq 0$ for all $\epsilon \in [0, 1]$. □

Since both $D_\beta\left(\mu \,\|\, \nu\right)$ and α/β are in the interval $[0, 1]$ (and $\alpha/\beta > 0$), we obtain the left inequality:

$$1 - D_\alpha\left(\mu \,\|\, \nu\right) \;\leq\; \left(1 - D_\beta\left(\mu \,\|\, \nu\right)\right)^{\alpha/\beta} \;\leq\; 1 - \frac{\alpha}{\beta} \cdot D_\beta\left(\mu \,\|\, \nu\right).$$

For the other direction, we use the first direction and Proposition 2.47:

$$D_\beta\left(\mu \,\|\, \nu\right) \;=\; D_{1-\beta}\left(\nu \,\|\, \mu\right) \;\geq\; \frac{1-\beta}{1-\alpha} D_{1-\alpha}\left(\nu \,\|\, \mu\right) \;=\; \frac{1-\beta}{1-\alpha} D_\alpha\left(\mu \,\|\, \nu\right).$$

□

A.1.4 Jensen-Shannon divergence

Proposition 2.51 (restated)

$$\frac{1}{2\ln 2}h^2\left(\mu,\nu\right) \;\leq\; JS\left(\mu,\nu\right) \;\leq\; \frac{\pi^2}{6\cdot\ln 2}h^2\left(\mu,\nu\right).$$

Proof: For the first inequality, we use the connection between the KL distance and the Hellinger distance (Proposition 2.45):

$$\begin{aligned}
JS\left(\mu,\nu\right) &= \frac{1}{2}\left(D_{KL}\left(\mu \parallel \frac{\mu+\nu}{2}\right)+D_{KL}\left(\nu \parallel \frac{\mu+\nu}{2}\right)\right)\\
&\geq \frac{1}{\ln 2}\left(h^2\left(\mu,\frac{\mu+\nu}{2}\right)+h^2\left(\nu,\frac{\mu+\nu}{2}\right)\right) \quad \text{(From Proposition 2.45)}\\
&\geq \frac{1}{2\ln 2}\left(h\left(\mu,\frac{\mu+\nu}{2}\right)+h\left(\nu,\frac{\mu+\nu}{2}\right)\right)^2 \quad \text{(Cauchy-Schwartz)}\\
&\geq \frac{1}{2\ln 2}h^2\left(\mu,\nu\right) \quad \text{(Triangle inequality).}
\end{aligned}$$

For the second inequality, we use a proof a la Topsøe [118]: Denote by $a_x \stackrel{\text{def}}{=} \left(\mu(x)+\nu(x)\right)/2$, by $d_x \stackrel{\text{def}}{=} \left|\mu(x)-\nu(x)\right|/2$, and by $r_x \stackrel{\text{def}}{=} a_x/d_x$. Note that one of $\mu(x),\nu(x)$ equals to a_x+d_x and the other equals to a_x-d_x. In the following we use two Taylor expansions of ln:

$$\begin{aligned}
\ln(1+x) &= \sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{n}x^n \quad \text{for } |x|\leq 1\\
\ln\left(\frac{x+1}{x-1}\right) &= 2\cdot\sum_{n=1}^{\infty}\frac{1}{x^{2n-1}\cdot(2n-1)} \qquad \text{for } |x|>1.
\end{aligned}$$

$$
\begin{aligned}
2 \cdot JS(\mu,\nu) &= \sum_{x\in\mathcal{X}} \mu(x)\log\frac{\mu(x)}{(\mu(x)+\nu(x))/2} + \nu(x)\log\frac{\nu(x)}{(\mu(x)+\nu(x))/2} \\
&= \sum_{x\in\mathcal{X}} (a_x+d_x)\log\frac{a_x+d_x}{a_x} + (a_x-d_x)\log\frac{a_x-d_x}{a_x} \\
&= \sum_{x\in\mathcal{X}} d_x\left((r_x+1)\log\left(1+\frac{1}{r_x}\right) + (r_x-1)\log\left(1-\frac{1}{r_x}\right)\right) \\
&= \sum_{x\in\mathcal{X}} d_x\left(r_x\log\left(1-\frac{1}{r_x^2}\right) + \log\left(\frac{r_x+1}{r_x-1}\right)\right) \\
&= \frac{1}{\ln 2}\cdot\sum_{x\in\mathcal{X}} d_x\left(\sum_{n=1}^{\infty} r_x\cdot\frac{(-1)^{n+1}}{n}\cdot\left(\frac{-1}{r_x^2}\right)^n + \frac{2}{2n-1}\cdot\frac{1}{r_x^{2n-1}}\right) \\
&= \frac{1}{\ln 2}\cdot\sum_{x\in\mathcal{X}} d_x\left(\sum_{n=1}^{\infty}\frac{1}{n(2n-1)}\cdot\frac{1}{r_x^{2n-1}}\right) \\
&\le \frac{1}{\ln 2}\cdot\sum_{n=1}^{\infty}\frac{1}{n^2}\sum_{x\in\mathcal{X}}\frac{d_x}{r_x} = \frac{\pi^2}{6\cdot\ln 2}\cdot\sum_{x\in\mathcal{X}}\frac{d_x^2}{a_x}.
\end{aligned}
$$

Now, note that by the Arithmetic Mean Geometric Mean inequality,

$$
\begin{aligned}
\frac{d_x^2}{a_x} &= \frac{|\mu(x)-\nu(x)|^2}{2\left(\mu(x)+\nu(x)\right)} = \frac{\left(\sqrt{\mu(x)}-\sqrt{\nu(x)}\right)^2\cdot\left(\sqrt{\mu(x)}+\sqrt{\nu(x)}\right)^2}{2\left(\mu(x)+\nu(x)\right)} \\
&= \left(\sqrt{\mu(x)}-\sqrt{\nu(x)}\right)^2\cdot\frac{\mu(x)+\nu(x)+2\sqrt{\mu(x)\nu(x)}}{2\left(\mu(x)+\nu(x)\right)} \\
&\le \left(\sqrt{\mu(x)}-\sqrt{\nu(x)}\right)^2.
\end{aligned}
$$

Therefore, using the L_2 characterization of the Hellinger distance (Definition 2.35),

$$
JS(\mu,\nu) \le \frac{\pi^2}{12\cdot\ln 2}\cdot\sum_{x\in\mathcal{X}}\left(\sqrt{\mu(x)}-\sqrt{\nu(x)}\right)^2 = \frac{\pi^2}{6\cdot\ln 2}\cdot h^2(\mu,\nu).
$$

□

Proposition 2.53 (restated) *Let μ be a distribution on $\mathcal{X}=\{x_1,\ldots,x_n\}$, let ν be an independent distribution on $\mathcal{Y}$, and let $f:\mathcal{X}\times\mathcal{Y}\to\mathcal{Z}$ be a function. Then,*

$$
I(\mu;\ f(\mu,\nu)) = JS_\mu\left(f(x_1,\nu),\ldots,f(x_n,\nu)\right).
$$

Proof: Let $X \sim \mu$.

$$
\begin{aligned}
I(\mu;\ f(\mu,\nu)) &= \\
&= D_{KL}\left((\mu, f(\mu,\nu)) \parallel \mu \times f(\mu,\nu)\right) \\
&\qquad \text{(KL distance characterization of mutual information)} \\
&= D_{KL}(\mu \parallel \mu) + D_{KL}\left(f(\mu,\nu)|\mu \parallel f(\mu,\nu)\right) \\
&\qquad \text{(Chain rule for KL distance)} \\
&= 0 + \sum_{x\in\mathcal{X}} \mu(x)\cdot D_{KL}\left(f(\mu,\nu)|\{X=x\} \parallel f(\mu,\nu)\right) \\
&\qquad \text{(Definition of conditional KL distance)} \\
&= \sum_{x\in\mathcal{X}} \mu(x)\cdot D_{KL}\left(f(x,\nu) \parallel f(\mu,\nu)\right) \\
&\qquad \text{(Independence of } \mu \text{ and } \nu\text{)} \\
&= JS_\mu\left(f(x_1,\nu),\ldots,f(x_n,\nu)\right) \\
&\qquad \text{(Definition of generalized Jensen-Shannon divergence).}
\end{aligned}
$$

□

A.2 Proof of Lemma 4.34

Lemma 4.34 (restated) $\textsc{Var}[C_k] \le 2^k \cdot (k!)^{1-1/k} \cdot \mathrm{E}[C_k]^{2-1/k}$.

Proof: For every subset $I \subseteq [\ell]$ of size k, we define X_I to be a 0-1 random variable that gets the value 1 iff all the values $\{x_i\}_{i\in I}$ are the same (i.e., I forms a collision). Clearly, $C_k = \sum_{I\subseteq[\ell],|I|=k} X_I$ and $\mathrm{E}[X_I] = ||U_x||_k^k$.

Define $\overline{X}_I \stackrel{\text{def}}{=} X_I - ||U_x||_k^k$. Note that for $I \neq J$, $I \cap J = \emptyset$, X_I and X_J are independent, and therefore $\mathrm{E}\left[\overline{X}_I \cdot \overline{X}_J\right] = \mathrm{E}\left[\overline{X}_I\right] \cdot \mathrm{E}\left[\overline{X}_J\right] = 0$. If $I \cap J \neq \emptyset$, then X_I and X_J do depend on each other, but $X_I \cdot X_J$ is a 0-1 random variable that gets the value 1 iff all the values in $I \cup J$ are the same. Thus, if $|I \cup J| = k + t$ then $\mathrm{E}\left[\overline{X}_I \cdot \overline{X}_J\right] =$

$\mathrm{E}\left[X_I \cdot X_j\right] - ||U_x||_k^{2k} = ||U_x||_{k+t}^{k+t} - ||U_x||_k^{2k}$. We have:

$$\begin{aligned}
\textsc{Var}\left[C_k\right] &= \mathrm{E}\left[\left(\sum_I \overline{X}_I\right)^2\right] = \sum_{I,J} \mathrm{E}\left[\overline{X}_I \cdot \overline{X}_J\right] \\
&= \sum_{t=0}^{k-1} \sum_{I,J,|I\cup J|=k+t} ||U_x||_{k+t}^{k+t} - ||U_x||_k^{2k} \\
&\le \sum_{t=0}^{k-1} \binom{\ell}{k} \cdot \binom{\ell-k}{t} \cdot \binom{k}{t} \cdot ||U_x||_{k+t}^{k+t}.
\end{aligned}$$

Note that $||U_x||_{k+t}^{k+t} \le ||U_x||_\infty^t \cdot ||U_x||_k^k \le ||U_x||_k^t \cdot ||U_x||_k^k = ||U_x||_k^{k+t}$. Therefore,

$$\begin{aligned}
\textsc{Var}\left[C_k\right] &\le \binom{\ell}{k} ||U_x||_k^k \cdot \sum_{t=0}^{k-1} \binom{k}{t} \cdot \binom{\ell-k}{t} \cdot ||U_x||_k^t \\
&\le \binom{\ell}{k} ||U_x||_k^k \cdot \sum_{t=0}^{k-1} \binom{k}{t} \cdot \frac{(\ell-k)^t}{t!} \cdot ||U_x||_k^t.
\end{aligned}$$

Since for all $k \ge 2$ and $0 \le t \le k-1$, $k \le 2t!\,(k-t)$, then $\binom{k}{t}/t! \le 2\binom{k-1}{t}$. Therefore,

$$\begin{aligned}
\textsc{Var}\left[C_k\right] &\le 2\binom{\ell}{k} ||U_x||_k^k \cdot \sum_{t=0}^{k-1} \binom{k-1}{t} \cdot (\ell-k)^t \cdot ||U_x||_k^t \\
&= 2\binom{\ell}{k} ||U_x||_k^k \cdot \left((\ell-k) \cdot ||U_x||_k + 1\right)^{k-1}.
\end{aligned}$$

Note that $||U_x||_k^k \ge 1/r^{k-1}$ (because $\mathrm{F}_k(x) \ge n$ and $(1/m)\mathrm{F}_k(x) \ge (n/m)^k$ for all x). We will choose ℓ to be large enough such that $\ell - k \ge r^{1-1/k}$. Therefore, $(\ell-k) \cdot ||U_x||_k \ge 1$. We thus obtain:

$$\begin{aligned}
\textsc{Var}\left[C_k\right] &\le 2\binom{\ell}{k} ||U_x||_k^k \cdot \left(2\,(\ell-k) \cdot ||U_x||_k\right)^{k-1} \\
&= 2^k\binom{\ell}{k} ||U_x||_k^k \cdot (\ell-k)^{k-1} \cdot ||U_x||_k^{k-1} \\
&\le 2^k\binom{\ell}{k} ||U_x||_k^k \cdot \left(k! \cdot \binom{\ell}{k} \cdot ||U_x||_k^k\right)^{(k-1)/k} \\
&= 2^k \cdot (k!)^{1-1/k} \cdot \mathrm{E}\left[C_k\right]^{2-1/k}.
\end{aligned}$$

$\square$

A.3 Proof of Lemma 6.28

Lemma 6.28 (restated) *Let μ, ν be probability distributions on $\mathcal{Z}_1$, and let R, S be probability transition matrices on $\mathcal{Z}_1 \times \mathcal{Z}_2$. Then for any $\epsilon > 0$, if $\alpha > 1 - \epsilon^2/(4(1+\epsilon))$,*

$$D_\alpha\left(\mu \circ R \parallel \nu \circ S\right) \;\leq\; (1+\epsilon) \cdot \left(D_\alpha\left(\mu \circ R \parallel \nu \circ R\right) + D_\alpha\left(\mu \circ R \parallel \mu \circ S\right)\right).$$

Proof: We define β_i to be $D_\alpha\left(R_i \parallel S_i\right)$—the Rényi α-divergence between the i-th row of R and the i-th row of S (note that since R and S are probability transition matrices their rows are distributions). We rewrite $D_\alpha\left(\mu \circ R \parallel \nu \circ S\right), D_\alpha\left(\mu \circ R \parallel \nu \circ R\right)$, and $D_\alpha\left(\mu \circ R \parallel \mu \circ S\right)$ as follows:

$$\begin{aligned}
D_\alpha\left(\mu \circ R \parallel \nu \circ S\right) &= 1 - \sum_{i \in \mathcal{Z}_1, j \in \mathcal{Z}_2} \left(\mu_i R_{ij}\right)^\alpha \left(\nu_i S_{ij}\right)^{1-\alpha} \;=\; 1 - \sum_{i \in \mathcal{Z}_1} \mu_i^\alpha \nu_i^{1-\alpha} \sum_{j \in \mathcal{Z}_2} R_{ij}^\alpha S_{ij}^{1-\alpha} \\
&= 1 - \sum_{i \in \mathcal{Z}_1} \mu_i^\alpha \nu_i^{1-\alpha}(1-\beta_i) \;=\; D_\alpha\left(\mu \parallel \nu\right) + \sum_{i \in \mathcal{Z}_1} \mu_i^\alpha \nu_i^{1-\alpha} \beta_i \\
D_\alpha\left(\mu \circ R \parallel \nu \circ R\right) &= 1 - \sum_{i \in \mathcal{Z}_1, j \in \mathcal{Z}_2} \left(\mu_i R_{ij}\right)^\alpha \left(\nu_i R_{ij}\right)^{1-\alpha} \;=\; 1 - \sum_{i \in \mathcal{Z}_1} \mu_i^\alpha \nu_i^{1-\alpha} \sum_{j \in \mathcal{Z}_2} R_{ij} \\
&= 1 - \sum_{i \in \mathcal{Z}_1} \mu_i^\alpha \nu_i^{1-\alpha} \;=\; D_\alpha\left(\mu \parallel \nu\right) \\
D_\alpha\left(\mu \circ R \parallel \mu \circ S\right) &= 1 - \sum_{i \in \mathcal{Z}_1, j \in \mathcal{Z}_2} \left(\mu_i R_{ij}\right)^\alpha \left(\mu_i S_{ij}\right)^{1-\alpha} \;=\; 1 - \sum_{i \in \mathcal{Z}_1} \mu_i \sum_{j \in \mathcal{Z}_2} R_{ij}^\alpha S_{ij}^{1-\alpha} \\
&= 1 - \sum_{i \in \mathcal{Z}_1} \mu_i (1-\beta_i) \;=\; \sum_{i \in \mathcal{Z}_1} \mu_i \beta_i.
\end{aligned}$$

Thus, what we need to prove is:

$$\begin{aligned}
& D_\alpha\left(\mu \parallel \nu\right) + \sum_{i \in \mathcal{Z}_1} \mu_i^\alpha \nu_i^{1-\alpha} \beta_i \;\leq\; (1+\epsilon) \cdot \left(D_\alpha\left(\mu \parallel \nu\right) + \sum_{i \in \mathcal{Z}_1} \mu_i \beta_i \right) \\
\iff\;& \sum_{i \in \mathcal{Z}_1} \mu_i^\alpha \nu_i^{1-\alpha} \beta_i \;\leq\; \epsilon \cdot D_\alpha\left(\mu \parallel \nu\right) + (1+\epsilon) \left(\sum_{i \in \mathcal{Z}_1} \mu_i \beta_i \right) \\
\iff\;& \sum_{i \in \mathcal{Z}_1} \beta_i \left(\mu_i^\alpha \nu_i^{1-\alpha} - (1+\epsilon)\mu_i \right) \;\leq\; \epsilon \cdot D_\alpha\left(\mu \parallel \nu\right).
\end{aligned}$$

Let us denote by $\mathcal{Z}_1'$ the set of all $i \in \mathcal{Z}_1$, for which $\mu_i^\alpha \nu_i^{1-\alpha} \geq (1+\epsilon)\mu_i$. Let $\mathcal{Z}_2'' = \mathcal{Z}_1 \setminus \mathcal{Z}_1'$. Since $\beta_i \leq 1$, then

$$\sum_{i \in \mathcal{Z}_1} \beta_i \left(\mu_i^\alpha \nu_i^{1-\alpha} - (1+\epsilon)\mu_i \right) \leq \sum_{i \in \mathcal{Z}_1'} \mu_i^\alpha \nu_i^{1-\alpha} - (1+\epsilon)\mu_i.$$

Thus, it suffices to prove:

$$\sum_{i \in \mathcal{Z}_1'} \mu_i^\alpha \nu_i^{1-\alpha} - (1+\epsilon)\mu_i \ \leq \ \epsilon \cdot D_\alpha\left(\mu \parallel \nu\right). \tag{A.1}$$

We note the following convexity property of the function $f(x,y) = x^\alpha y^{1-\alpha}$:

Claim A.0.2 *For any non-negative numbers $x_1, \ldots, x_n, y_1, \ldots, y_n$,*

$$\sum_{i=1}^n x_i^\alpha y_i^{1-\alpha} \ \leq \ \left(\sum_{i=1}^n x_i\right)^\alpha \cdot \left(\sum_{i=1}^n y_i\right)^{1-\alpha}.$$

The proof follows directly from an application of Hölder's inequality with $p = 1/\alpha$ and $q = 1/(1-\alpha)$.

Define $z = \sum_{i \in \mathcal{Z}_1'} \mu_i$ and $w = \sum_{i \in \mathcal{Z}_1'} \nu_i$. Rephrasing (A.1), and using Claim A.0.2, it suffices to prove the following:

$$\begin{aligned} & z^\alpha w^{1-\alpha} - (1+\epsilon)z \ \leq \ \epsilon \cdot \left(1 - z^\alpha w^{1-\alpha} - (1-z)^\alpha (1-w)^{1-\alpha}\right) \\ \Longleftrightarrow \quad & \epsilon \cdot (1-z)^\alpha (1-w)^{1-\alpha} + (1+\epsilon) \cdot z^\alpha w^{1-\alpha} - (1+\epsilon)z - \epsilon \leq 0. \end{aligned} \tag{A.2}$$

Define $f_\alpha(z,w)$ to be the left-hand-side of (A.2). For any given value of z we will maximize $f_\alpha(z,w)$ as a function of w and show that this maximum is less than 0 for an appropriately chosen α. For simplicity of notations, we denote: $a = (1+\epsilon)z^\alpha$, $b = \epsilon(1-z)^\alpha$ and $\delta = 1-\alpha$. We thus have: $f_{\alpha,z}(w) = aw^\delta + b(1-w)^\delta - (1+\epsilon)z - \epsilon$.

$$\frac{df_{\alpha,z}}{dw} \ = \ a\delta w^{\delta-1} - b\delta(1-w)^{\delta-1}.$$

Thus, the extremal point is at:

$$w^* \ = \ \frac{a^{1/(1-\delta)}}{a^{1/(1-\delta)} + b^{1/(1-\delta)}}.$$

This point is a maximum in the interval $[0,1]$, since

$$\frac{d^2 f_{\alpha,z}}{dw^2} \ = \ a\delta(\delta-1)w^{\delta-2} + b\delta(\delta-1)(1-w)^{\delta-2} < 0.$$

Thus the value of the maximum point is:

$$\begin{aligned} f_{\alpha,z}(w^*) \ &= \ \frac{a^{1/(1-\delta)}}{\left(a^{1/(1-\delta)} + b^{1/(1-\delta)}\right)^\delta} + \frac{b^{1/(1-\delta)}}{\left(a^{1/(1-\delta)} + b^{1/(1-\delta)}\right)^\delta} - (1+\epsilon)z - \epsilon \\ &= \ \left(a^{1/(1-\delta)} + b^{1/(1-\delta)}\right)^{1-\delta} - (1+\epsilon)z - \epsilon \\ &= \ \left((1+\epsilon)^{1/\alpha} z + \epsilon^{1/\alpha}(1-z)\right)^\alpha - (1+\epsilon)z - \epsilon. \end{aligned}$$

We want this maximum to be non-positive for every $z \in [0,1]$. That is,

$$\begin{aligned} \left((1+\epsilon)^{1/\alpha}z + \epsilon^{1/\alpha}(1-z)\right)^{\alpha} &\leq (1+\epsilon)z + \epsilon \\ \iff \quad ((1+\epsilon)z+\epsilon)^{1/\alpha} - (1+\epsilon)^{1/\alpha}z - \epsilon^{1/\alpha}(1-z) &\geq 0. \end{aligned} \tag{A.3}$$

Let $g_\alpha(z)$ be the left-hand-side of (A.3), and for simplicity of notation, let $p = 1/\alpha$. We would like to show that for an appropriate choice of α, $g_\alpha(z) \geq 0$ for all $z \in [0,1]$. Note that $g_\alpha(0) = 0$. Thus, it suffices to show that g is non-decreasing in the interval $[0,1]$.

$$g'(z) = p(1+\epsilon)\left((1+\epsilon)z+\epsilon\right)^{p-1} - (1+\epsilon)^p + \epsilon^p \geq p(1+\epsilon)\epsilon^{p-1} - (1+\epsilon)^p + \epsilon^p,$$

where the last inequality follows from the fact $z \geq 0$. Thus g would be non-decreasing if:

$$p(1+\epsilon)\epsilon^{p-1} - (1+\epsilon)^p + \epsilon^p \geq 0 \iff p\left(\frac{\epsilon}{1+\epsilon}\right)^{p-1} - 1 + \left(\frac{\epsilon}{1+\epsilon}\right)^{p} \geq 0.$$

Write $\gamma = \epsilon/(1+\epsilon)$. Note that $0 < \gamma < 1$. We thus need to prove:

$$\begin{aligned} \gamma^p + p\gamma^{p-1} - 1 \geq 0 \quad &\iff \quad \gamma^{p-1}(\gamma+p) - 1 \geq 0 \\ &\Longleftarrow \quad \gamma^{p-1}(1+\gamma) - 1 \geq 0 \iff \gamma^{p-1} \geq \frac{1}{1+\gamma}. \end{aligned}$$

Since $\gamma < 1$, $1/(1+\gamma) \leq e^{-\gamma/2}$. Thus it suffices that:

$$\gamma^{p-1} \geq e^{-\gamma/2} \iff p - 1 \leq \frac{\gamma}{2\ln(1/\gamma)}.$$

Therefore, we need $\alpha = 1/p$ to satisfy

$$\alpha \geq \frac{1}{1 + \frac{\gamma}{2\ln(1/\gamma)}}.$$

Thus, it suffices that

$$\alpha \geq 1 - \frac{\gamma}{4\ln(1/\gamma)} = 1 - \frac{\epsilon}{4(1+\epsilon)\ln((1+\epsilon)/\epsilon)}.$$

And for the last inequality to hold it suffices that

$$\alpha \geq 1 - \frac{\epsilon^2}{4(1+\epsilon)}.$$

□