Limitations of Local Filters of Lipschitz and Monotone Functions

Pranjal Awasthi, Carnegie Mellon University
Madhav Jha, Pennsylvania State University
Marco Molinaro, Carnegie Mellon University
Sofya Raskhodnikova, Pennsylvania State University

We study local filters for two properties of functions of the form $f : \{0,1\}^d \to \mathbb{R}$: the Lipschitz property and monotonicity. A local filter with additive error a is a randomized algorithm that is given black-box access to a function f and a query point x in the domain of f. It outputs a value $F(x)$ such that (i) the *reconstructed function* $F(x)$ satisfies the property (in our case, is Lipschitz or monotone) and (ii) if the input function f satisfies the property, then for every point x in the domain (with high constant probability) the reconstructed value $F(x)$ differs from $f(x)$ by at most a. Local filters were introduced by Saks and Seshadhri [2010]. The relaxed definition we study is due to Bhattacharyya et al. [2012a], except that we further relax it by allowing additive error. Local filters for Lipschitz and monotone functions have applications to areas such as data privacy.

We show that every local filter for Lipschitz or monotone functions runs in time exponential in the dimension d, even when the filter is allowed significant additive error. Prior lower bounds (for local filters with no additive error, that is, with $a = 0$) applied only to a more restrictive class of filters, e.g., *nonadaptive* filters. To prove our lower bounds, we construct families of hard functions and show that lookups of a local filter on these functions are captured by a combinatorial object that we call a c-connector. Then we present a lower bound on the maximum outdegree of a c-connector and show that it implies the desired bounds on the running time of local filters. Our lower bounds, in particular, imply the same bound on the running time for a class of privacy mechanisms.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms: Theory, Design

Additional Key Words and Phrases: Lipschitz, local filter, monotonicity, privacy

ACM Reference Format:
Pranjal Awasthi, Madhav Jha, Marco Molinaro and Sofya Raskhodnikova, 2014. Limitations of Local Filters of Lipschitz and Monotone Functions. *ACM Trans. Comput. Theory* 7, 1, Article 2 (December 2014), 15 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

In this work we study local reconstruction of properties of functions. Property-preserving data reconstruction [Ailon et al. 2008] is a direction of research in sublinear algorithms that has its roots in property testing [Rubinfeld and Sudan 1996; Goldreich et al. 1998]. Some related notions include locally decodable codes [Katz and Trevisan 2000], program checking [Blum et al. 1993] and, more generally, local computation [Rubinfeld et al. 2011; Alon et al. 2012].

A preliminary version of this paper appeared in RANDOM'12 [Awasthi et al. 2012].

P. Awasthi was supported by NSF grant CCF-1116892, M. Jha and S. Raskhodnikova were supported by NSF grants CAREER CCF-0845701 and CDI-0941553, M. Molinaro was supported by NSF grant CMMI-1024554.

Authors' addresses: P. Awasthi, (Current address) Computer Science Department, Princeton University; M. Jha, (Current address) Sandia National Labs; M. Molinaro, (Current address) School of Industrial and Systems Engineering, Georgia Institute of Technology; S. Raskhodnikova, Department of Computer Science and Engineering, Pennsylvania State University.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2014 ACM 1942-3454/2014/12-ART2 $15.00
DOI: http://dx.doi.org/10.1145/0000000.0000000

To motivate the reconstruction model, consider an algorithm ALG that is computing on a large dataset and whose correctness is contingent upon the dataset satisfying a certain structural property. For example, ALG may require that its input array be sorted or that its input function be Lipschitz. In such situations, ALG could access its input via a *filter* that ensures that data seen by ALG always satisfies the desired property, modifying it at few places on the fly, if required. We can represent the input to ALG as a function f, where $f(x)$ holds the data stored at location x. Instead of accessing $f(x)$ directly, ALG makes a *query* x to the filter. The filter *looks up* the value of f on a small number of points and returns $F(x)$, where F satisfies the desired property and is as close to the original function f as possible. Thus, ALG is computing with reconstructed data F instead of its original input f.

Saks and Seshadhri [2010] introduced a stronger notion of a *local filter*. It has an additional requirement that the reconstruction of $f(x)$ and $f(y)$ on two different queries x and y should be done independently. In particular, the output function F is independent of the order of the queries x made to the filter[1].

Local filters have many desirable features: for example, they can be implemented in a distributed setting, where several processes need to access different parts of the input, and the filter has to ensure that all the parts together are consistent with some function F that satisfies the desired property. This global consistency guarantee enables several applications of local filters described in previous work [Saks and Seshadhri 2010; Bhattacharyya et al. 2012a; Jha and Raskhodnikova 2013], including the application to data privacy that we explain below.

The main goal of this paper is to understand limitations of local filters. This is crucial in order to identify the types of tradeoffs (for instance, output quality vs. lookup complexity) available for a given application. Two natural candidate properties for this evaluation are the Lipschitz property and monotonicity of functions of the form[2] $f : [n]^d \to \mathbb{R}$, studied in previous work [Ailon et al. 2008; Saks and Seshadhri 2010; Bhattacharyya et al. 2012a; Jha and Raskhodnikova 2013]: the first is motivated by the privacy application explained below and the second is a "benchmark" problem in property-preserving reconstruction and property testing. A function $f : [n]^d \to \mathbb{R}$ is *Lipschitz* (with respect to the ℓ_1 metric on $[n]^d$) if $|f(x) - f(y)| \leq \|x - y\|_1$ for all points x, y in the domain $[n]^d$. Intuitively, changing the argument to the Lipschitz function by a small amount does not significantly change the value of the function. A function $f : [n]^d \to \mathbb{R}$ is *monotone* if $f(x) \leq f(y)$ for all points $x \preceq y$ in the domain $[n]^d$, where $\preceq$ denotes the natural partial order on $[n]^d$: for $x = (x_1, \ldots, x_d) \in [n]^d$ and $y = (y_1, \ldots, y_d) \in [n]^d$, we have $x \preceq y$ iff $x_i \leq y_i$ for all coordinates $i \in [d]$. In other words, increasing the coordinates of the argument of a monotone function does not decrease the value of the function.

The original definition of local filters by Saks and Seshadhri [2010] has a requirement that the filter be *distance-respecting*, that is, the reconstructed function F should not differ from the original function f on significantly more points than necessary. Bhattacharyya et al. [2012a] and Jha and Raskhodnikova [2013] removed this requirement and demonstrated that it is not necessary in some applications. Their local filter is simply required to output $F = f$ if the original function has the property; otherwise, F can be an arbitrary function satisfying the property. We relax the notion of local filter further by allowing additive error. Our definition (Definition 2.1) has an additional parameter a, and the function F can differ from f by a small amount on every point, even if f satisfies the property: namely, we require that for every x in the domain, with high constant probability

$$|F(x) - f(x)| \leq a.$$

Local filters considered by Bhattacharyya et al. [2012a] and Jha and Raskhodnikova [2013] are a special case of our local filters with $a = 0$. Our goal is to determine (for small a) if there are local

[1] The lookups made by a local filter on a query point x are not required to be close to x. The word "local" in the name of the filter is inherited from the related notion of locally decodable error-correcting codes.

[2] We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

filters that make only $\mathrm{poly}(n, d)$ lookups in order to output the reconstructed function $F(x)$ at a given point x.

Privacy Application. We observe that local filters with small additive error can still be used in the privacy application described in [Jha and Raskhodnikova 2013]. Consider a server that has a private database with information about individuals, modeled as a point x in $\{0, 1\}^d$, representing whether each of the d possible types of people is present in the database. (More generally, x is modeled as a point in $[n]^d$ representing a histogram that captures how many people of each type are present.) A user who does not have direct access to x can ask the server for some information about this database by specifying a function f for the server to evaluate at the point x. The server's goal is to output a value close to $f(x)$, that reveals almost no information about any single individual. The latter notion has been made precise via the concept of *differential privacy* [Dwork et al. 2006b]. A standard way of obtaining such guarantees is to ask users to *submit only Lipschitz* functions[3], and have the server output $f(x)$ plus some random noise depending on the desired privacy guarantee [Dwork et al. 2006b]. However, if a malicious user submits a function that is not Lipschitz, the differential privacy guarantee is lost. A local filter with the following properties can then be used between the server and the submitted function f to ensure the desired privacy: (i) the reconstructed function F is always Lipschitz; (ii) if f is already Lipschitz, then with high probability $|F(x) - f(x)| \leq a$ for all x, where a is a given parameter. This way, the server always evaluates a Lipschitz function F and thus has the desired privacy guarantees. Furthermore, if the user provides a valid Lipschitz function f, the mechanism outputs a value $F(x)$ in the range $f(x) \pm a$ plus random noise. If a is reasonably small, it is then absorbed in the noise. Thus, bounds on the running time and additive error of the local filter translate directly into bounds on the running time and accuracy of the corresponding privacy mechanism.

1.1. Previous Results on Local Filters

Despite the fact that local filters have been thoroughly studied, lower bounds for general (not necessarily *distance-respecting*) *adaptive* filters remained a big challenge.

Saks and Seshadhri [2010] present a distance-respecting local filter for monotonicity of functions $f : [n]^d \to \mathbb{R}$ with running time $(\log n + 1)^{O(d)}$ per query. For monotonicity of functions $f : \{0, 1\}^d \to \mathbb{R}$, no nontrivial (that is, performing $o(2^d)$ lookups per query) filter is known. Saks and Seshadhri also show that a *distance-respecting* local filter for monotonicity on the domain $\{0, 1\}^d$ must perform $2^{\Omega(d)}$ lookups per query. This lower bound crucially uses the fact that the filter is distance-respecting, and does not apply to general local filters (even when no additive error is allowed).

As we explained, in many applications the extra requirement that the filter be distance-respecting is not necessary. Bhattacharyya et al. [2012a] studied lower bounds for local monotonicity filters that are not necessarily distance-respecting. However, their super-polynomial lower bounds only hold for *nonadaptive* filter. For the domain $\{0, 1\}^d$, Bhattacharyya et al. show that nonadaptive filters must perform $\Omega(\frac{2^{\alpha d}}{d})$ lookups per query in the worst case, where $\alpha \geq 0.1620$. For adaptive filters, their bound quickly degrades with the number of lookups performed to *incomparable* points in the domain ($x, y \in [n]^d$ are *comparable* if $x \preceq y$ or $y \preceq x$ and *incomparable* otherwise). Specifically, their lower bounds for adaptive filters is $\Omega(\frac{2^{\alpha d - \ell}}{d})$, where ℓ is the number of lookups to points incomparable to x made on query x. For arbitrary adaptive filters, this degrades to $\Omega(d)$. (In particular, this lower bound does not rule out local filters that make only $\ell = \alpha d$ lookups per query, all of them to incomparable points.) Prior to our work, no super-polynomial lower bound for adaptive local monotonicity filters was known.

[3] More generally, if a user wants to evaluate a function f with Lipschitz constant at most ℓ, where $\ell > 1$, then the Lipschitz function f/ℓ can be submitted to the server. When the noisy answer returned by the server is multiplied by ℓ, the effect is to add noise proportional to ℓ.

For the Lipschitz property, Jha and Raskhodnikova [2013] obtained a deterministic nonadaptive local filter that runs in time $O((\log n + 1)^d)$ per query. They also show that the lower bound from [Bhattacharyya et al. 2012a] for *nonadaptive* filters, with the same statement, applies to *nonadaptive* local filters of the Lipschitz property.

Previous work left open whether it is possible to obtain (adaptive and not necessarily distance-respecting) local filters for monotonicity and for the Lipschitz property that make only $\mathrm{poly}(n, d)$ lookups per query.

1.2. Our Results and Techniques

We consider local a-filters, that is, local filters with additive error a, described earlier and formally defined in Definition 2.1. These filters do not need to be distance-respecting and can be fully adaptive. Our main results, stated in more detail in Section 2, are that even such relaxed filters need to perform a number of lookups exponential in the dimension d in order to reconstruct a Lipschitz (respectively, monotone) function. This applies even to functions on the domain $\{0,1\}^d$.

THEOREM 1.1 (LIMITATIONS OF LIPSCHITZ FILTERS). *Consider the Lipschitz property of functions $f : \{0,1\}^d \to \mathbb{R}$ and any (randomized) local (not necessarily distance-respecting) $\frac{d}{402}$-filter for this property. Then there is a function f and a query x for which, with constant probability, this filter makes $2^{\Omega(d)}$ lookups.*

The additive error $a = d/402$ in Theorem 1.1 is as large as possible up to a constant factor: the trivial filter that outputs $F(x) = (f(\mathbf{0}) + f(\mathbf{1}))/2$, where $\mathbf{0}$ and $\mathbf{1}$ are all-0 and all-1 vectors, respectively, is a local $\frac{d}{2}$-filter[4]. To see this, note that (i) the reconstructed function $F(x)$ is Lipschitz and (ii) if the input function $f(x)$ is Lipschitz then $|F(x) - f(x)| = \frac{1}{2}|f(\mathbf{0}) + f(\mathbf{1}) - 2f(x)| \leq \frac{1}{2}(|f(\mathbf{0}) - f(x)| + |f(\mathbf{1}) - f(x)|) \leq \frac{1}{2}(\|\mathbf{0} - x\|_1 + \|\mathbf{1} - x\|_1) = \frac{d}{2}$ for every $x \in \{0,1\}^d$.

For monotonicity, we can prove an analogous theorem with no upper bound on a. This is explained by the fact that monotonicity is determined by the order of the values at different points and not their magnitudes. To calibrate the additive error, we state the next theorem for functions with bounded range, namely, $[0, 2a+1]$. The additive error in the theorem is also tight because for functions with that range, the trivial filter above that outputs $F(x) = (f(\mathbf{0}) + f(\mathbf{1}))/2$ is a local $(a + \frac{1}{2})$-filter.

THEOREM 1.2 (LIMITATIONS OF MONOTONICITY FILTERS). *Consider the monotonicity property of functions $f : \{0,1\}^d \to [0, 2a+1]$ and any (randomized) local a-filter for this property. Then there is a function f and query x for which, with constant probability, this filter makes $2^{\Omega(d)}$ lookups.*

To introduce the ideas used in the proofs, we focus for now on deterministic filters. To obtain lower bounds for *nonadaptive* filters in [Bhattacharyya et al. 2012a; Jha and Raskhodnikova 2013], the authors construct two collections of "hard functions" $f^{(x,y)}$ and $f^{(\overline{x,y})}$ (satisfying the Lipschitz property) indexed by $x, y \in \{0,1\}^d$. They show that if a local filter works correctly on $f^{(x,y)}$ and $f^{(\overline{x,y})}$, as well as on a suitably defined function $h^{(x,y)}$ (violating the Lipschitz property on (x, y)), the lookups made on queries x and y need to have a structured interaction. (Note that in this case the lookups are independent of the input function because the filter is nonadaptive.) More precisely, they construct a graph over $\{0,1\}^d$ by (roughly) adding, for every point x, edges from x to all points that are looked up upon query x, and show that this graph is a 2-transitive-closure-spanner (2-TC-spanner) for the hypercube. (TC-spanners were introduced in [Bhattacharyya et al. 2012b]; see Section 3 for definition and comparison with c-connectors that we introduce.) Using the lower bound on the size of a 2-TC-spanner for the hypercube from [Bhattacharyya et al. 2012a], it can be shown that any nonadaptive filter must use exponential lookups on one of the query points.

[4] To simplify the presentation, we did not optimize the constant factor. In particular, the weights $d/3$ and $2d/3$ in Definition 3.1 were not optimized.

In the case of adaptive filters, one cannot assume that the lookups made on a given query point are independent of the input function. One simple idea to try to overcome this obstacle is to consider, for each query x, the *union* of the lookups made on query x over all possible choices of hard functions, and then apply the previous lower bound approach. The problem is that this is overcounting the number of lookups made by the filter on a *single* given function on query x. Due to the large number of "hard functions" considered in [Bhattacharyya et al. 2012a; Jha and Raskhodnikova 2013], this overcounting makes the bound coming from the 2-TC-spanners vacuous for adaptive filters; this is where the factor 2^ℓ lost in [Bhattacharyya et al. 2012a] mentioned above comes from.

In order to remedy this, we build a collection of hard functions that are much "smoother" than those from [Bhattacharyya et al. 2012a; Jha and Raskhodnikova 2013]. This allows us to use fewer functions. However, it comes at a cost: the interactions of the lookups caused by these functions are not as structured as before and do not imply a 2-TC-spanner. We introduce (in Definition 3.2) a type of directed graph called *c-connector* that captures lookup interactions. When arc directions are ignored, a c-connector is a relaxation of 2-TC-spanners (our transformation to c-connectors preserves information on whether x is looked up on query y or vice versa, while this information is lost in the transformation to 2-TC-spanners in [Bhattacharyya et al. 2012a; Jha and Raskhodnikova 2013]). Nevertheless, we can argue that a c-connector has a large maximum outdegree, which relates to the lookup complexity. Indeed, one of the key ingredients for our lower bound is recognizing the limitations of 2-TC-spanners in this context and finding a combinatorial structure with the right amount of flexibility. Given the importance of TC-spanners (see [Raskhodnikova 2010] for a survey), c-connectors might find use outside of this work.

Organization. Section 2 gives basic definitions and a detailed statement of our main results. In Section 3, we define c-connectors, the graph objects on which our lower bounds are based. In Sections 4 and 5, we develop a connection between c-connectors and local filters for the Lipschitz property and monotonicity. In Section 6, we bound the outdegree of c-connectors. The final proof of the theorems stated in Section 1.2 appears in Section 7 and consists of putting these two parts together.

2. DEFINITIONS AND FORMAL STATEMENT OF RESULTS

Given a point $x \in \{0,1\}^d$, we use x_i to denote its ith coordinate and $|x|$ to denote its Hamming weight, that is, $|x| = \sum_i x_i$. We identify each point $x \in \{0,1\}^d$ with the subset of coordinates that are equal to 1, namely, $\{i : x_i = 1\}$. This gives meaning to expressions like $x \subseteq y$, $x \cap y$, $x \cup y$ and $x \setminus y$ for $x, y \in \{0,1\}^d$. For $x \in \{0,1\}^d$, the Hamming weight $|x|$ coincides with the cardinality of the set associated with x.

We now provide a formal definition of local a-filters, i.e., local filters that allow additive error a. It is stated for a general property P of functions with domain D; in our case, P will be either the Lipschitz property or monotonicity.

Definition 2.1 (*Local a-filter*). Let P be a property of functions $f : D \to R$ for some domain D and range $R \subseteq \mathbb{R}$. A local a-filter for P with error probability δ is a randomized algorithm that is given black-box access to a function $f : D \to R$ together with a *query* point $x \in D$. For each random seed σ in the algorithm's probability space $(\Omega, \Pr)$, the filter obtains the value of f on a sequence of points $L(\sigma, f, x) = \{y_1, y_2, \ldots, y_k\}$, called *lookups* (where the choice of y_i depends only on x, σ and $f(y_1), f(y_2), \ldots, f(y_{i-1})$), and outputs a reconstructed value $F(\sigma, f, x)$ for x. The reconstructed function $F_{\sigma,f} : D \to R$ given by $F_{\sigma,f}(x) = F(\sigma, f, x)$ must obey two conditions:

(i) $F_{\sigma,f}$ satisfies property P for all functions f and all random seeds σ;
(ii) if f satisfies property P then for all $x \in D$,

$$\Pr_{\sigma}(F_{\sigma,f}(x) \in [f(x) - a, f(x) + a]) \geq 1 - \delta.$$

To simplify notation, we usually omit the probability space and denote a local a-filter by (L, F).

Notice that one could make requirement (ii) in Definition 2.1 stronger by changing the order of quantifiers and asking that if f satisfies property P then $\Pr_\sigma(\forall x \in D, F_{\sigma,f}(x) \in [f(x)-a, f(x)+a]) \geq 1-\delta$. Any lower bound that applies to filters we defined also applies to filters with this stronger requirement.

The next observation captures the structural rigidity of local filters exploited in our lower bounds. It states that if functions f and g are identical on the lookups performed on query x when the input function is f, then the filter will perform the same lookups on x for both f and g and, consequently, reconstruct the same value.

OBSERVATION 2.2. *Let (L, F) be a local a-filter. Then the following holds for every random seed σ and query point x: if f and g are functions such that $f|_{L(\sigma,f,x)} = g|_{L(\sigma,f,x)}$, then $F(\sigma, f, x) = F(\sigma, g, x)$.*

Now we restate Theorems 1.1 and 1.2, giving more details about parameters we obtain.

THEOREM 2.1. *Consider a sufficiently large integer d and let $a \in [0, d/402]$. Let (L, F) be a local a-filter for the Lipschitz property with error probability at most $1/3$. Then there exists a function $f : \{0,1\}^d \to \mathbb{R}$ and a query $x \in \{0,1\}^d$ such that*

$$\Pr_\sigma(|L(\sigma, f, x)| \geq 2^{0.009d}) \geq 0.15.$$

THEOREM 2.2. *Consider a sufficiently large integer d and let $a \geq 0$. Let (L, F) be a local a-filter for monotonicity with error probability at most $1/3$. Then there exists a function $f : \{0,1\}^d \to [0, 2a+1]$ and a query $x \in \{0,1\}^d$ such that*

$$\Pr_\sigma(|L(\sigma, f, x)| \geq 2^{0.009d}) \geq 0.15.$$

3. C-CONNECTORS

In this section, we introduce c-connectors. A c-connector is a directed graph on the vertex set $\{0,1\}^d$, where certain pairs of nodes share an out-neighbor with some prescribed properties. The motivation for c-connectors will only become clear in Sections 4 and 5, but will describe right away how they are related to 2-TC-spanners.

Definition 3.1. Let X denote the set of points in $\{0,1\}^d$ with Hamming weight $d/3$, and Y denote the set of points in $\{0,1\}^d$ with Hamming weight $2d/3$. Also let $\mathcal{P}$ denote the set of comparable pairs $(x, y) \in X \times Y$, namely, such that $x \prec y$.

Definition 3.2 (*c-connector*). Fix $c \in \mathbb{N}$. Given a subset $\mathcal{P}'$ of $\mathcal{P}$, a digraph G with the node set $\{0,1\}^d$ is a *c-connector for $\mathcal{P}'$* if for every $(x, y) \in \mathcal{P}'$ there exists $z \in \{0,1\}^d$ with the following properties:

— (Connectivity) The arcs (x, z) and (y, z) belong to G;
— (Structure) $|z \setminus y| < c$ and $|z| > \frac{d}{3} - c$.

Observe that for all $\mathcal{P}' \subseteq P$ and all $c < c'$, if G is a c-connector for $\mathcal{P}'$, it is also a c'-connector for $\mathcal{P}'$. A *2-TC-spanner* of the Boolean hypercube (with the usual partial order) is a directed graph H on the node set $\{0,1\}^d$ with the property that for all $x \prec y$ there is a point z satisfying $x \preceq z \preceq y$, such that the arcs (x, z) and (z, y) belong to H [Bhattacharyya et al. 2012b]. If we reorient the arcs in a 2-TC-spanner of the hypercube so that the nodes in Y only have outgoing arcs, we obtain a valid 1-connector for $\mathcal{P}$, because the requirement $x \preceq z \preceq y$ (in the definition of 2-TC-spanner) implies that $|z \setminus y| = 0$ and $|z| \geq |x| = d/3$. Therefore, c-connectors are a relaxation of 2-TC-spanners in two ways:

(1) in a c-connector, only pairs in $\mathcal{P}'$ have a common neighbor with prescribed properties, and
(2) for a c-connector, the requirements on the common neighbor are weaker.

4. LOCAL FILTERS FOR THE LIPSCHITZ PROPERTY IMPLY C-CONNECTORS

In this section, we focus on the Lipschitz property. We construct a family of functions such that a local a-filter that works correctly on functions from the family must perform lookups corresponding to a c-connector. The idea is to start with a Lipschitz function f^0 and then construct other Lipschitz functions f_y^c that agree with f^0 on most points, but where $f_y^c(y)$ is much larger than $f^0(y)$. We argue that if a purported local a-filter makes only "local" lookups on queries x and y, then we can create a function that looks like f_y^c around y (so that the filter is fooled and returns $F(y)$ in the range $f_y^c(y) \pm a$) and looks like f^0 around x (so that the filter is fooled and returns $F(x)$ in the range $f^0(x) \pm a \ll f_y^c(y) \pm a$). Thus, for the returned function, $F(x)$ and $F(y)$ are too far apart, ensuring that it is not Lipschitz.

4.1. Hard Functions for Lipschitz Filters

Recall from Definition 3.1 that Y denotes the set of points in $\{0,1\}^d$ with Hamming weight $2d/3$. To construct hard functions, for a point $y \in Y$, let

$$T_y = \{x \in \{0,1\}^d : x \subseteq y, |x| \geq d/3\}.$$

Define the function f^0 by $f^0(z) = \max\{|z|, d/3\}$ for all $z \in \{0,1\}^d$. Intuitively, for $c \in \mathbb{N}$ and $y \in Y$, we define the function f_y^c as the smallest Lipschitz function that is at least $f^0 + c\chi_{T_y}$, where χ_{T_y} denotes the characteristic function of the set T_y. More specifically, we set $f_y^c(z) = \max\{|z| + c - |z \setminus y|, f^0(z)\}$ for all $z \in \{0,1\}^d$. These functions are depicted in Figure 1. Clearly, function f^0 is Lipschitz. Next we prove that all functions f_y^c are Lipschitz as well.

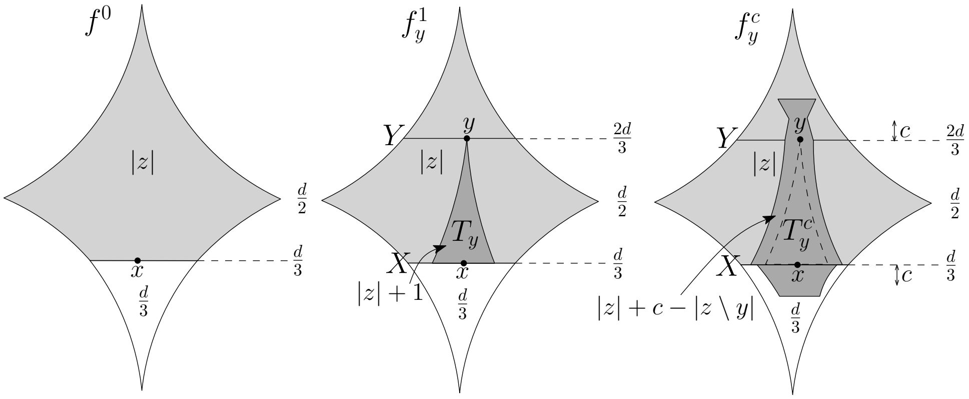

Fig. 1. Functions used in the proof of Lemma 4.3.

LEMMA 4.1. *For all $c \in \mathbb{N}$ and $y \in Y$, the function f_y^c is Lipschitz.*

PROOF. Fix $c \in \mathbb{N}$ and $y \in Y$. Define $g(z) = |z| + c - |z \setminus y|$, so that $f_y^c = \max\{g, f^0\}$. Since f^0 is Lipschitz, and the maximum of two Lipschitz functions is a Lipschitz function, it suffices to show that g is Lipschitz. Take $z, z' \in \{0,1\}^d$ such that $\|z - z'\|_1 = 1$. It remains to show that $|g(z) - g(z')| \leq 1$. Note that either $z \subseteq z'$ or $z' \subseteq z$; without loss of generality assume the former. Since $|z| = |z'| - 1$ and $0 \leq |z' \setminus y| - |z \setminus y| \leq 1$, we obtain that $|g(z) - g(z')| \leq \big||z| - |z'| + |z' \setminus y| - |z \setminus y|\big| \leq 1$. This concludes the proof of the lemma. □

For a point $y \in Y$ and a constant $c \in \mathbb{N}$, let $T_y^c \subseteq \{0,1\}^d$ be the set of points z such that $f_y^c(z) \neq f^0(z)$. Then $T_y^1 = T_y$ and the set T_y^c gets larger as c increases: specifically, $T_y^c \subseteq T_y^{c'}$ for

$c < c'$. The definitions of f_y^c and f^0 directly give the following observation, justifying the structure requirement in the definition of a c-connector.

OBSERVATION 4.2. *All elements z in the set T_y^c satisfy $|z \setminus y| < c$ and $|z| > \frac{d}{3} - c$.*

4.2. Correct Reconstruction of Hard Functions Implies c-Connector

Now we show that if a local a-filter is correct on the hard functions, its lookups correspond to a c-connector for $\mathcal{P}$. (Recall that $\mathcal{P}$ is the set of pairs $(x, y) \in X \times Y$ such that $x \prec y$.) We start by essentially focusing on deterministic filters or, alternatively, by looking at a "good" seed of a randomized filter. The analysis for randomized filters is based on the ability to pick a few of these good seeds and then analyzing the "union" of the behavior of the filter running with these seeds.

Fix a value for c throughout this section. Consider a local a-filter (L, F). Given points $x \in X$ and $y \in Y$, we say that a random seed $\sigma \in \Omega$ is *good* for x and y if $F_{\sigma,f^0}(x) \in [f^0(x) - a, f^0(x) + a]$ and $F_{\sigma,f_y^c}(y) \in [f_y^c(y) - a, f_y^c(y) + a]$. Given a seed σ that is good for x and y, we define the digraph $G_\sigma^{xy} = (\{0,1\}^d, A_\sigma^{xy})$ that captures the lookups made on queries x and y. Specifically, the set A_σ^{xy} consists of all the arcs $\{(x, z) : z \in L(\sigma, f^0, x) \cup \{x\}\}$ and $\{(y, z) : z \in L(\sigma, f_y^c, y) \cup \{y\}\}$. (Notice that while this depends on the value of c, we omitted this parameter in order to keep the notation manageable.)

LEMMA 4.3 (LOCAL FILTER IMPLIES c-CONNECTOR). *Consider a local a-filter (L, F) for the Lipschitz property and an integer $c > 2a$. For all $(x, y) \in \mathcal{P}$, if $\sigma \in \Omega$ is good for x and y then G_σ^{xy} is a c-connector for $\{(x, y)\}$.*

PROOF. For the sake of contradiction, suppose not. Unraveling the definitions and using Observation 4.2, we get that the sets $(L(\sigma, f^0, x) \cup \{x\}) \cap T_y^c$ and $(L(\sigma, f_y^c, y) \cup \{y\}) \cap T_y^c$ do not intersect. Then let A, B be a partition of T_y^c such that A contains $(L(\sigma, f^0, x) \cup \{x\}) \cap T_y^c$ and B contains $(L(\sigma, f_y^c, y) \cup \{y\}) \cap T_y^c$. Define the function f such that $f|_A = f^0|_A$, $f|_B = f_y^c|_B$, and $f|_{\{0,1\}^d \setminus (A \cup B)} = f^0|_{\{0,1\}^d \setminus (A \cup B)} = f_y^c|_{\{0,1\}^d \setminus (A \cup B)}$, where the last equation follows from the definition of T_y^c. (See Figure 1.) To reach a contradiction, we show that the filter does not reconstruct f correctly.

Notice that $f^0|_{L(\sigma,f^0,x)} = f|_{L(\sigma,f^0,x)}$, so Observation 2.2 gives that $F(\sigma, f, x) = F(\sigma, f^0, x)$. Similarly, $f_y^c|_{L(\sigma,f_y^c,y)} = f|_{L(\sigma,f_y^c,y)}$ and hence $F(\sigma, f, y) = F(\sigma, f_y^c, y)$.

Since σ is good for x and y, we have that $F(\sigma, f, x) = F(\sigma, f^0, x) \le f^0(x) + a = \frac{d}{3} + a$ and $F(\sigma, f, y) = F(\sigma, f_y^c, y) \ge f_y^c(y) - a = \frac{2d}{3} + c - a$. Since $c > 2a$ we get $F(\sigma, f, y) - F(\sigma, f, x) > d/3 = \|x - y\|_1$. Hence, the function $F_{\sigma,f}$ is not Lipschitz. This contradicts that (L, F) is a local a-filter and concludes the proof of the lemma. □

Consider subsets $\mathcal{P}_1$ and $\mathcal{P}_2$ of $\mathcal{P}$. Notice that if G_1 is a c-connector for $\mathcal{P}_1$ and G_2 is a c-connector for $\mathcal{P}_2$ then the graph formed by the union of (the arcs of) G_1 and G_2 is a c-connector for $\mathcal{P}_1 \cup \mathcal{P}_2$. We remark that when we take this union we do not add parallel arcs. This directly gives the following result.

COROLLARY 4.4. *Consider a local a-filter (L, F) for the Lipschitz property and an integer $c > 2a$. Suppose that for each $(x, y) \in \mathcal{P}$ there is a random seed $\sigma(x, y) \in \Omega$ that is good for x and y. Then the graph obtained as the union of the graphs $\{G_{\sigma(x,y)}^{xy}\}_{(x,y)\in\mathcal{P}}$ is a c-connector for $\mathcal{P}$. Moreover, this graph has outdegree at most*

$$\max\left\{\max_{x\in X}\left\{\Big|\bigcup_y L(\sigma(x,y), f^0, x)\Big|\right\}, \max_{y\in Y}\left\{\Big|\bigcup_x L(\sigma(x,y), f_y^c, y)\Big|\right\}\right\} + 1. \tag{1}$$

Using this corollary, we show that a local a-filter with small "average" number of lookups implies a c-connector for $\mathcal{P}$ with a small outdegree.

LEMMA 4.5. *Consider a local a-filter (L, F) for the Lipschitz property with error probability δ and an integer $c > 2a$. Consider $\alpha > 0$ and let*

$$M = \max_{f,x} \Pr_{\sigma} \left(|L(\sigma, f, x)| > \alpha\right).$$

If $\delta + M < 1/2$ then there is a c-connector for $\mathcal{P}$ with maximum outdegree at most $2d\alpha/\log\left(\frac{1}{2\delta+2M}\right) + 1$.

PROOF. The idea is to construct, via the probabilistic method, a set $\bar{S} \subseteq \Omega$ of good seeds that attains a small value in (1). Given $(x, y) \in \mathcal{P}$, define the event $E_{x,y} \subseteq \Omega$ as the set of random seeds σ satisfying the following:

(1) σ is good for x and y;
(2) $|L(\sigma, f^0, x)| \leq \alpha$ and $|L(\sigma, f_y^c, y)| \leq \alpha$.

Given the guarantee of the filter and the definition of M, the complement of $E_{x,y}$ holds with probability at most $\gamma \doteq 2\delta + 2M$.

Now let S be a random set obtained by picking independently and with replacement $s \doteq 2d/\log_2(1/\gamma)$ elements from $(\Omega, \Pr)$. For a given $(x, y) \in \mathcal{P}$, it follows from the previous paragraph that the probability (over the construction of S) that S does not intersect $E_{x,y}$ is at most γ^s. Taking a union bound over all such pairs, the probability that there is $(x, y) \in \mathcal{P}$ for which S does not intersect $E_{x,y}$ is strictly less than $2^{2d}\gamma^s = 1$. Therefore, there exists a realization $\bar{S}$ of S that intersects all $E_{x,y}$'s.

Then, for $(x, y) \in \mathcal{P}$, let $\sigma(x, y)$ be a point in $\bar{S} \cap E_{x,y}$. Since each $\sigma(x, y)$ is good for x and y, we can apply Corollary 4.4 using these seeds. By construction, we have that, for all $x \in X$, the set $\bigcup_y L(\sigma(x, y), f^0, x)$ has size at most $|\bar{S}|\alpha$ and, for all $y \in Y$, the set $\bigcup_x L(\sigma(x, y), f_y^c, y)$ has size at most $|\bar{S}|\alpha$. This concludes the proof. □

5. LOCAL FILTERS FOR MONOTONICITY IMPLY 1-CONNECTORS

In this section, we show that the lookups performed by a local a-filter for monotonicity give rise to a c-connector (in this case, with $c = 1$).

5.1. Hard Functions for Monotonicity Filters

Again, we start by defining functions $f^{0,a}$ and f_y^a such that if a local filter is correct on these functions, its lookups correspond to a 1-connector. Recall that for a point $y \in Y$, we define

$$T_y = \{x \in \{0, 1\}^d : x \subseteq y, |x| \geq d/3\}.$$

Define the function $f^{0,a}$ by $f^{0,a}(z) = 2a+1$ if $|z| \geq d/3$ and $f^{0,a}(z) = 0$ if $|z| < d/3$. For a point $y \in Y$, we define the function f_y^a equal to $f^{0,a} - (2a+1)\chi_{T_y}$, namely,

$$f_y^a(z) = \begin{cases} 2a+1 & \text{if } z \geq d/3 \text{ and } z \notin T_y; \\ 0 & \text{otherwise.} \end{cases}$$

These functions are depicted in Figure 2. It can be easily verified that these functions are monotone.

LEMMA 5.1. *For all $y \in Y$ and $a \geq 0$, the functions $f^{0,a}$ and f_y^a are monotone.*

Notice that the functions $f^{0,a}$ and f_y^a differ exactly on points in T_y, and that T_y is the set of points that satisfy the structure property in the definition of a 1-connector.

5.2. Correct Reconstruction of Hard Functions Implies 1-Connector

Recall that $\mathcal{P}$ is the set of comparable pairs $(x, y) \in X \times Y$ or, equivalently, pairs where $x \in T_y$. Consider a local a-filter (L, F) for monotone functions. As before, given $x \in X$ and $y \in Y$, we say that a random seed $\sigma \in \Omega$ is *good* for x and y if $F_{\sigma,f^{0,a}}(x) \in [f^{0,a}(x) - a, f^{0,a}(x) + a]$ and

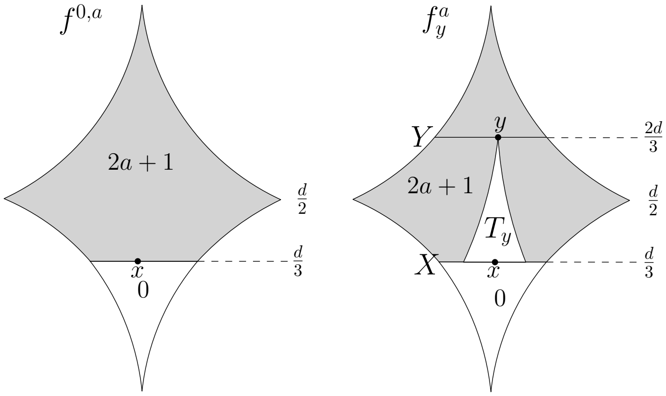

Fig. 2. Functions used in proof of Lemma 5.2. Observe that $f^{0,a}(x) = 2a + 1$ and $f_y^0(y) = 0$.

$F_{\sigma,f_y^a}(y) \in [f_y^a(y) - a, f_y^a(y) + a]$. Given a seed σ that is good for x and y, we define the digraph $G_\sigma^{xy} = (\{0,1\}^d, A_\sigma^{xy})$ in a way similar to what we did in the previous section: A_σ^{xy} contains all the arcs $\{(x,z) : z \in L(\sigma, f^{0,a}, x) \cup \{x\}\}$ and $\{(y,z) : z \in L(\sigma, f_y^a, y) \cup \{y\}\}$. (Again, this graph depends on the value of a, but this parameter is omitted to keep the notation manageable.)

The constructions of our functions and the digraph G_σ^{xy}, together with Observation 2.2 that captures the behavior of local a-filters, give the following lemma.

LEMMA 5.2. *Fix $a \geq 0$ and consider a local a-filter (L, F) for monotonicity. For all $(x, y) \in \mathcal{P}$, if $\sigma \in \Omega$ is good for x and y then G_σ^{xy} is a 1-connector for $\{(x, y)\}$.*

PROOF. For the sake of contradiction suppose not. By definition of 1-connector, this means that the sets $(L(\sigma, f^{0,a}, x) \cup \{x\}) \cap T_y$ and $(L(\sigma, f_y^a, y) \cup \{y\}) \cap T_y$ do not intersect. Then let A, B be a partition of T_y such that A contains $(L(\sigma, f^{0,a}, x) \cup \{x\}) \cap T_y$ and B contains $(L(\sigma, f_y^a, y) \cup \{y\}) \cap T_y$. Define the function f such that $f|_A = f^{0,a}|_A$, $f|_B = f_y^a|_B$, and $f|_{\{0,1\}^d \setminus (A\cup B)} = f_y^a|_{\{0,1\}^d \setminus (A\cup B)} = f^{0,a}|_{\{0,1\}^d \setminus (A\cup B)}$. The last equality holds since f_y^a and $f^{0,a}$ only differ on T_y. To reach the desired contradiction, we show that the filter does not reconstruct f correctly.

Note that $f^{0,a}|_{L(\sigma,f^{0,a},x)} = f|_{L(\sigma,f^{0,a},x)}$, so Observation 2.2 gives that $F(\sigma, f, x) = F(\sigma, f^{0,a}, x)$. Similarly, $f^{0,a}|_{L(\sigma,f_y^a,y)} = f|_{L(\sigma,f_y^a,y)}$ and hence $F(\sigma, f, y) = F(\sigma, f_y^a, y)$.

Since σ is good for x and y, we have that $F(\sigma, f, x) = F(\sigma, f^{0,a}, x) \geq f^{0,a}(x) - a = a + 1$ and $F(\sigma, f, y) = F(\sigma, f_y^a, y) \leq f_y^a(y) + a = a$. That is, $F(\sigma, f, x) > F(\sigma, f, y)$. Hence, the function $F_{\sigma,f}$ is not monotone. This contradicts that (L, F) is a local a-filter and concludes the proof of the lemma. □

Finally, we use the same technique for finding a set of good seeds that achieve small value in (1) as we did in Lemma 4.5. This allows us to obtain the desired connection between local a-filters and 1-connectors for $\mathcal{P}$.

LEMMA 5.3. *Fix $a \geq 0$ and consider a local a-filter (L, F) for monotone functions with error probability δ. Consider $\alpha > 0$ and let*

$$M = \max_{f,x} \Pr_\sigma \left(|L(\sigma, f, x)| > \alpha\right).$$

If $\delta + M < 1/2$ then there is a 1-connector for $\mathcal{P}$ with maximum outdegree at most

$$2d\alpha / \log\left(\frac{1}{2\delta + 2M}\right) + 1.$$

6. LOWER BOUND ON THE MAXIMUM OUTDEGREE OF A C-CONNECTOR

Recall that $\mathcal{P}$ is the set of pairs $(x, y) \in X \times Y$ such that x and y are *comparable*. We show a lower bound on the maximum outdegree of a c-connector for $\mathcal{P}$. The constants in the bound are not optimized.

THEOREM 6.1. *Consider a sufficiently large integer d, and let c be an integer in the range $[d/201, d/200]$. Then the maximum outdegree of any c-connector for $\mathcal{P}$ is at least $2^{0.01d}$.*

To prove this, let G be a c-connector for $\mathcal{P}$. Let

$$\tilde{T}_y^c = \{z : |z \setminus y| < c, |z| > d/3 - c\}$$

be the points that satisfy the structure property in Definition 3.2. Then $T_y \subseteq T_y^c \subseteq \tilde{T}_y^c$ for all $y \in Y$, and for $x \in T_y$ and $z \in \tilde{T}_y^c$, we have $x \cup z \in \tilde{T}_y^c$. We say that a pair $(x, y) \in \mathcal{P}$ is *covered* by a point z if $z \in \tilde{T}_y^c$ and the arcs (x, z) and (y, z) belong to G.

Each pair in $\mathcal{P}$ needs to be covered by a point. For a fixed $x \in X$, the outdegree of x in G is at least the number of distinct points that cover the pairs in $\mathcal{P}$ containing x (and, similarly, for a fixed $y \in Y$). The difficulty in lower-bounding the outdegree of x is that many pairs containing it can be covered by the same point. The heart of the argument is to show that no point can cover too many such pairs. It relies on the fact that the sets $\tilde{T}_y^c$ are "localized". More precisely, consider a point z and let (x, y) be covered by it. Notice that $x \in T_y$ and $z \in \tilde{T}_y^c$, hence $x \cup z \in \tilde{T}_y^c$. If z is not near x, namely, $|z \setminus x|$ is large, then we argue that not too many points y satisfy $x \cup z \in \tilde{T}_y^c$, given the localization of $\tilde{T}_y^c$. On the other hand, if z is near x then there are not too many possibilities for x itself. Our bound is derived by putting these observations together.

To make the above argument work, we divide the pairs in $\mathcal{P}$ into two groups based on the covers they have. Let $\alpha \in [1/15, 1/14]$ be such that αd is an integer (such α exists since d is sufficiently large). For $(x, y) \in \mathcal{P}$ and z that covers (x, y), if $|z \setminus x| \leq \alpha d$, then we say that z is *near* x and that z is a *nearby cover* of (x, y). Let $\mathcal{N}$ denote the set of pairs $(x, y) \in \mathcal{P}$ that have a nearby cover. Let $\mathcal{F} = \mathcal{P} \setminus \mathcal{N}$ be the remaining pairs. For a fixed $y \in Y$, define $\mathcal{N}_y$ as the pairs in $\mathcal{N}$ containing y and for $x \in X$ define $\mathcal{F}_x$ as the pairs in $\mathcal{F}$ containing x. Our goal is to upper-bound $\mathcal{N}$ and $\mathcal{F}$. Towards this goal, define $Z \subseteq \{0, 1\}^d$ to be the set of points that cover at least one pair in $\mathcal{P}$. Furthermore, for a given $x \in X$, let Z_x denote the set of points that cover at least one pair in $\mathcal{P}$ containing x. Define Z_y analogously. Observe that Z is the union of sets Z_x and Z_y over all $x \in X$ and $y \in Y$. The next two lemmas bound the sizes of $\mathcal{N}$ and $\mathcal{F}$, respectively. For each lemma, we give a proof sketch describing the main ideas of the proof. Since the actual proof is somewhat technical, we defer it to Section 6.1.

LEMMA 6.2. *Let $\Theta = d^2 \binom{d/3+\alpha d}{\alpha d} \binom{2d/3+c}{\alpha d+c}$. Then the number of pairs in $\mathcal{N}$ is at most*

$$|Y| \cdot \Theta \cdot \max_{y \in Y}\{|Z_y|\}.$$

PROOF SKETCH. To upper-bound $\mathcal{N}$, we start by arguing that, for a fixed $y \in Y$, a point cannot be a nearby cover for many pairs (x, y) in $\mathcal{N}_y$. To see this, take $z \in Z_y$ and let $(x, y) \in \mathcal{P}$ be such that z is a nearby cover for it. Then notice that x and z are very similar: $|z \setminus x| \leq \alpha d$ and $|x \setminus z| \leq \alpha d + c$; the first bound follows from the definition of a nearby cover and the second uses $|z| \geq |x| - c$ from Observation 4.2. From these constraints, it follows that there are at most $d^2 \binom{d/3+\alpha d}{\alpha d} \binom{2d/3+c}{\alpha d+c}$ possibilities for such x's. Thus, for all $y \in Y$,

$$|\mathcal{N}_y| \leq |Z_y| \cdot d^2 \binom{d/3 + \alpha d}{\alpha d} \binom{2d/3 + c}{\alpha d + c}.$$

By adding over all y, we get the desired bound. □

LEMMA 6.3. *Let $\Phi = d^2\binom{\frac{2d}{3}+c}{c}\binom{\frac{2d}{3}-\alpha d}{\frac{d}{3}-\alpha d+c}$. Then the number of pairs in $\mathcal{F}$ is at most*

$$|X| \cdot \Phi \cdot \max_{x \in X}\{|Z_x|\}.$$

PROOF SKETCH. To upper-bound the size of $\mathcal{F}$, we start by showing that, for a fixed $x \in X$, a point cannot be a (non-nearby) cover for too many pairs in $\mathcal{F}_x$. To see this, take $z \in Z_x$ and suppose $(x, y) \in \mathcal{F}_x$ is covered by z. Notice that $x \cup z$ and y are very similar: $|(x \cup z) \setminus y| \leq c$ and $|y \setminus (x \cup z)| \leq d/3 - \alpha d + c$; the first bound follows from $x \subseteq y$ and Observation 4.2, and the second further uses the fact that $|x \cup z| \geq d/3 + \alpha d$ (since z is not a nearby cover). Then it is easy to see that there are at most $d^2\binom{\frac{2d}{3}+c}{c}\binom{\frac{2d}{3}-\alpha d}{\frac{d}{3}-\alpha d+c}$ such y's. Thus, for each $x \in X$,

$$|\mathcal{F}_x| \leq |Z_x| \cdot d^2\binom{\frac{2d}{3}+c}{c}\binom{\frac{2d}{3}-\alpha d}{\frac{d}{3}-\alpha d+c}.$$

By adding over all x, we get the desired bound on $\mathcal{F}$. □

The maximum outdegree of the c-connector G is bounded from below by

$$M \triangleq \max\{\max_{x \in X}\{|Z_x|\}, \max_{y \in Y}\{|Z_y|\}\}.$$

Since $\mathcal{N}$ and $\mathcal{F}$ partition the set of pairs $\mathcal{P}$, we can add the bounds from Lemmas 6.2 and 6.3 and obtain

$$M \geq \frac{|\mathcal{P}|}{\binom{d}{d/3}(\Theta + \Phi)} = \frac{\binom{2d/3}{d/3}}{\Theta + \Phi}. \tag{2}$$

Standard computations (deferred to Section 6.2) can be used to give a lower bound of $2^{0.01d}$ on the right-hand side of this expression. This concludes the proof of Theorem 6.1.

6.1. Estimates for Lemmas 6.2 and 6.3

Estimates for Lemmas 6.2 and 6.3 come from the following technical claim.

CLAIM 6.4. *Given $t_1, t_2 \in [d]$ and a fixed $u \in \{0,1\}^d$ such that $|u| \geq t_1$, let $S(u, t_1, t_2, \eta)$ be the set of vertices $x \in \{0,1\}^d$ such that $|x| = t_2$ and $|u \setminus x| \leq \eta$. Moreover, assume t_1 (respectively, t_2) is at least $2\eta + 2t_2 - d$ (respectively, η). Then, $|S(u, t_1, t_2)| \leq d^2\binom{t_2+\eta}{\eta}\binom{d-t_1}{\eta+t_2-t_1}$.*

PROOF. First, we show $t_1 \leq |u| \leq \eta + t_2$. The lower bound is part of the premise of the claim. For the upper bound, using $|u \setminus x| \leq \eta$ and $|x| = t_2$ from the premise of the claim, we get

$$|u| = |u \setminus x| + |u \cap x| \leq \eta + |u \cap x| \leq \eta + |x| = \eta + t_2.$$

Next we show $|x \setminus u| \leq \eta + t_2 - t_1$. Observe that $|x \setminus u| = |x \cup u| - |u| = |x| + |u \setminus x| - |u|$. Using $|u \setminus x| \leq \eta$, $|x| = t_2$ and $|u| \geq t_1$ from the premise of the claim, we get

$$|x \setminus u| = |x| + |u \setminus x| - |u| \leq t_2 + \eta - t_1,$$

as required.

Therefore, for every $x \in S(u, t_1, t_2, \eta)$, we have points r and a such that $x = (u \setminus r) \cup a$ satisfying: (i) $r \subseteq u$ and $|r| \leq \eta$; (ii) $a \cap u = \emptyset$ and $|a| \leq \eta + t_2 - t_1$. Since $t_1 \leq |u| \leq \eta + t_2 \leq d$, there are

at most $\sum_{i=0}^{\eta}\binom{t_2+\eta}{i} \leq d\binom{t_2+\eta}{\eta}$ possibilities for r and

at most $\sum_{i=0}^{\eta+t_2-t_1}\binom{d-t_1}{i} \leq d\binom{d-t_1}{\eta+t_2-t_1}$ possibilities for a,

where, in the above bounds, we used the fact that t_1 (respectively, t_2) is at least $2\eta + 2t_2 - d$ (respectively, η). By multiplying these terms, we get the upper bound on $|S(u, t_1, t_2, \eta)|$. □

Estimate for Lemma 6.2. Recall that we have a fixed $z \in \{0,1\}^d$, and we want to upper-bound the number of $x \in X$ satisfying $|z \setminus x| \le \alpha d$ and $|z| \ge |x| - c$ by $\Theta = d^2 \binom{d/3+\alpha d}{\alpha d} \binom{2d/3+c}{\alpha d+c}$. This follows directly by applying Claim 6.4 with parameters $u = z$, $t_1 = d/3 - c$, $t_2 = d/3$, and $\eta = \alpha d$.

Estimate for Lemma 6.3. Recall that we have a fixed $z \in \{0,1\}^d$, and we want to upper-bound the number of $y \in Y$ satisfying $|(x \cup z) \setminus y| \le c$ and $|x \cup z| \ge \frac{d}{3} + \alpha d$ by $\Phi = d^2 \binom{\frac{2d}{3}+c}{c} \binom{\frac{2d}{3}-\alpha d}{\frac{d}{3}-\alpha d+c}$. This follows directly by applying Claim 6.4 with parameters $u = x \cup z$, $t_1 = d/3 + \alpha d$, $t_2 = 2d/3$, and $\eta = c$.

6.2. Bounding $\Theta + \Phi$

In this section we show that

$$\frac{\binom{2d/3}{d/3}}{\Theta + \Phi} \ge 2^{0.01d}.$$

We start with three simple facts about the binomial coefficient $\binom{a}{b}$ for integers $a \ge b$:

(i) $\binom{a+1}{b+1} = \frac{a+1}{b+1}\binom{a}{b} \ge \binom{a}{b}$;
(ii) $\binom{a}{b} \le \left(\frac{ea}{b}\right)^b$;
(iii) if $b = a/2$, then $\binom{a}{b} \ge \frac{2^a}{a}$.

We also observe that $(1/x)^x$ is increasing for x in the range $(0, 1/4]$. Using this, it is easy to see that, by choosing α and c/d small enough, we can get Θ and Φ of the order $\binom{2d/3}{d/3} O(2^{-\epsilon d})$ for a small constant $\epsilon > 0$. We show that the choice of α and c/d in the statement of the lemma works.

From observation (iii) above we have $\binom{2d/3}{d/3} \ge \frac{2^{2d/3}}{d}$. Using observations (i) and (ii) above and the bounds on α, c and d, we obtain the upper bound

$$\begin{aligned}
\frac{\Theta}{\binom{2d/3}{d/3}} &= \frac{d^2 \binom{d/3+\alpha d}{\alpha d} \binom{2d/3+c}{\alpha d+c}}{\binom{2d/3}{d/3}} \\
&\le d^3 2^{-\frac{2d}{3}} \binom{\frac{d}{3}+\alpha d}{\alpha d} \binom{\frac{2d}{3}+c}{\alpha d+c} \\
&\le d^3 2^{-\frac{2d}{3}} \left(\frac{e(d/3+\alpha d)}{\alpha d}\right)^{\alpha d} \binom{\frac{2d}{3}+c}{\alpha d+c}.
\end{aligned}$$

We have that

$$\left(\frac{e(d/3+\alpha d)}{\alpha d}\right)^{\alpha d} = \left(e\left(1 + \frac{1}{3\alpha}\right)\right)^{\alpha d} \le 2^{0.3d}.$$

We also have that

$$\begin{aligned}
\binom{\frac{2d}{3}+c}{\alpha d+c} &= \binom{2d/3}{\alpha d} \frac{(2d/3+c)(2d/3-1+c)\ldots(2d/3+1)}{(\alpha d+c)(\alpha d+c-1)\ldots(\alpha d+1)} \\
&\le \binom{2d/3}{\alpha d} \left(\frac{2d/3+c}{\alpha d}\right)^c \\
&\le \left(\frac{2e}{3\alpha}\right)^{\alpha d} \left(\frac{2/3+c/d}{\alpha}\right)^c \le 2^{0.34d}.
\end{aligned}$$

Hence, we get

$$\frac{\Theta}{\binom{2d/3}{d/3}} \leq d^3 2^{-\frac{2d}{3}} 2^{0.64d} \leq \frac{2^{-0.01d}}{2}. \tag{3}$$

For Φ, using observation (i) and the fact that the central binomial coefficient is the largest one, we have

$$\binom{\frac{2d}{3} - \alpha d}{\frac{d}{3} - \alpha d + c} = \binom{\frac{2d}{3}}{\frac{d}{3} + c} \frac{(\frac{d}{3} + c)(\frac{d}{3} + c - 1) \ldots (\frac{d}{3} + c - \alpha d + 1)}{\frac{2d}{3}(\frac{2d}{3} - 1) \ldots (\frac{2d}{3} - \alpha d + 1)} \leq \binom{\frac{2d}{3}}{\frac{d}{3}} \left(\frac{\frac{1}{3} + \frac{c}{d}}{\frac{2}{3} - \alpha}\right)^{\alpha d}.$$

Again using the bounds on α, c, and d, we obtain

$$\frac{\Phi}{\binom{2d/3}{d/3}} \leq d^2 \binom{\frac{2d}{3} + c}{c} \left(\frac{\frac{1}{3} + \frac{c}{d}}{\frac{2}{3} - \alpha}\right)^{\alpha d} \leq 2^{0.043d} 2^{-0.058d} \leq \frac{2^{-0.01d}}{2}. \tag{4}$$

We conclude the proof by adding the bounds from (3) and (4).

7. CONCLUDING THE PROOF OF THEOREMS 2.1 AND 2.2

PROOF OF THEOREM 2.1. Without loss of generality assume that $a = d/402$. Let $\alpha = 2^{0.009d}$ and $M = \max_{f,x} \Pr_\sigma (|L(\sigma, f, x)| > \alpha)$. We claim that $M \geq 0.15$, which then implies the theorem. For the sake of contradiction, suppose that $M < 0.15$. Since then $1/3 + M < 1/2$, we can take an integer $c \in (d/201, d/200)$ and employ Lemma 4.5 to get that there is a c-connector for $\mathcal{P}$ with maximum outdegree at most $O(d\alpha)$. Since and d is sufficiently large, we obtain that this connector has maximum outdegree less than $2^{0.01d}$. This contradicts Theorem 6.1 and concludes the proof of Theorem 2.1. □

PROOF OF THEOREM 2.2. By definition, a 1-connector is also a c-connector for any $c \geq 1$. We proceed as in the proof of Theorem 2.1, but now with no restriction on a. □

8. CONCLUSION AND FUTURE WORK

We show that local filters for the Lipschitz property and monotonicity require exponentially many (in the dimension) lookups, even when allowed additive error. One can try to further relax the requirements on local filters in order to overcome these lower bounds.

One possibility is to allow the local a-filter to output a reconstructed function that with small probability does not satisfy the desired property P. Such weaker guarantees can still be useful for (ϵ, δ)-differential privacy [Dwork et al. 2006a] instead of "pure" ϵ-differential privacy mentioned in the introduction. Another interesting relaxation of local a-filters, specific to the Lipschitz property, is to allow the reconstructed function F be b-Lipschitz instead of Lipschitz, that is, to require only $|F(x) - F(y)| \leq b \cdot \|x - y\|_1$ for all $x, y \in \{0,1\}^d$. For the privacy application described in the introduction, a and b of the order of $O(\sqrt{d})$ are still acceptable. We remark that the techniques presented here yield similar lower bounds for relaxed notion with b slightly larger than 1, but not for $b \geq 2$.

REFERENCES

Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. 2008. Property-Preserving Data Reconstruction. *Algorithmica* 51, 2 (2008), 160–182.

Noga Alon, Ronitt Rubinfeld, Shai Vardi, and Ning Xie. 2012. Space-efficient local computation algorithms. In *SODA*, Yuval Rabani (Ed.). SIAM, 1132–1139.

Pranjal Awasthi, Madhav Jha, Marco Molinaro, and Sofya Raskhodnikova. 2012. Limitations of Local Filters of Lipschitz and Monotone Functions. In *APPROX-RANDOM (Lecture Notes in Computer Science)*, Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio (Eds.), Vol. 7408. Springer, 374–386.

Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. 2012a. Lower Bounds for Local Monotonicity Reconstruction from Transitive-Closure Spanners. *SIAM J. Discrete Math.* 26, 2 (2012), 618–646.

Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. 2012b. Transitive-Closure Spanners. *SIAM J. Comput.* 41, 6 (2012), 1380–1425.

Manuel Blum, Michael Luby, and Ronitt Rubinfeld. 1993. Self-Testing/Correcting with Applications to Numerical Problems. *J. Comput. Syst. Sci.* 47, 3 (1993), 549–595.

Cynthia Dwork, Krishnaram Kenthapadi, Frank Mcsherry, and Moni Naor. 2006a. Our Data, Ourselves: Privacy via Distributed Noise Generation. In *In EUROCRYPT*. Springer, 486–503.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006b. Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC*. 265–284.

Oded Goldreich, Shafi Goldwasser, and Dana Ron. 1998. Property Testing and its Connection to Learning and Approximation. *J. ACM* 45, 4 (1998), 653–750.

Madhav Jha and Sofya Raskhodnikova. 2013. Testing and Reconstruction of Lipschitz Functions with Applications to Data Privacy. *SIAM J. Comput.* 42, 2 (2013), 700–731.

Jonathan Katz and Luca Trevisan. 2000. On the efficiency of local decoding procedures for error-correcting codes. In *STOC*. 80–86.

Sofya Raskhodnikova. 2010. Transitive-Closure Spanners: A Survey. In *Property Testing (Lecture Notes in Computer Science)*, Oded Goldreich (Ed.), Vol. 6390. Springer, 167–196.

Ronitt Rubinfeld and Madhu Sudan. 1996. Robust Characterization of Polynomials with Applications to Program Testing. *SIAM J. Comput.* 25, 2 (1996), 252–271.

Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. 2011. Fast Local Computation Algorithms. In *ICS*. 223–238.

Michael E. Saks and C. Seshadhri. 2010. Local Monotonicity Reconstruction. *SIAM J. Comput.* 39, 7 (2010), 2897–2926.