Testing and Reconstruction of Lipschitz Functions

Sofya Raskhodnikova
Penn State University

Joint work with Madhav Jha (Penn State)
Lipschitz Continuous Functions

A function $f : D \to R$ has Lipschitz constant $c$ if for all $x, y$ in $D$,

$$\text{distance}_R(f(x), f(y)) \leq c \cdot \text{distance}_D(x, y).$$

A fundamental notion in

- mathematical analysis
- theory of differential equations

Example uses of a Lipschitz constant $c$ of a given function $f$

- probability theory: in tail bounds via McDiarmid’s inequality
- program analysis: as a measure of robustness to noise
- data privacy: to scale noise added to preserve differential privacy
Computing a Lipschitz Constant?

- Infeasible

- Undecidable to even verify if \( f \) computed by a TM has Lipschitz constant \( c \)

- NP-hard to verify if \( f \) computed by a circuit has Lipschitz constant \( c \)
  - even for finite domains
Lipschitz Functions Over Finite Domains

We call a function Lipschitz if it has Lipschitz constant 1.

• can rescale by $1/c$ to get a Lipschitz function from a function with Lipschitz constant $c$

Examples

$f : \{1, \ldots, n\} \rightarrow \mathbb{R}$  

$\begin{array}{cccc}
1 & 2 & 2 & 3 \\
2 & 3 & 2 & 3 \\
3 & 2 & 4 & 3 \\
4 & 3 & 3 & 2 \\
\end{array}$

Nodes = points in the domain; edges = points at distance 1

Node labels = values of the function
Application 1: Program Analysis

Certifying that a program computes a Lipschitz function

[Chaudhuri Gulwani Lublinerman Navidpour 10]

To ensure that a program

- is robust to noise in its inputs
  (e.g., caused by communication/ measurement errors)
- responds well to compiler optimizations that lead to an approximately equivalent program

**Question:** Can we test if a function is Lipschitz?
Application 2: Data Privacy

Typical examples: census, civic archives, medical records,…

[Dwork McSherry Nissim Smith 06]
Lipschitz functions can be released with little noise while satisfying differential privacy.

Question: Can we ensure that the server only answers queries about Lipschitz functions?
Local Property Reconstruction [Saks Seshadhri 10]

Extends [Ailon Chazelle Seshadhri Liu 08]

Oracle

User

Oracle

Filter

User

for each \( f \) and \( r \), function \( g \) satisfies property \( P \)

w.h.p. \( g \) is close to \( f \) (in Hamming distance)

\( g(x) \) can be computed quickly

Local filter: \( g \) does not depend on queries \( x \)
**Local Property Reconstruction** [Saks Seshadhri 10]

Extends [Ailon Chazelle Seshadhri Liu 08]

For each $f$ and $r$, function $g$ satisfies property $P$.

- $g = f$ if $f$ satisfies property $P$.
- $g(x)$ can be computed quickly.
- **Local** filter: $g$ does not depend on queries $x$.
Filter Mechanism for Data Privacy

**Question:**
Can we quickly (locally) reconstruct Lipschitz property?
Question:
Can we quickly locally reconstruct Lipschitz property for functions on the hypergrid domains?
Our Results: Lipschitz Testers

Line $f : \{1, \ldots, n\} \rightarrow R$

- **Upper bound:** $O(\log n / \varepsilon)$ time
  - applies to all discretely metrically convex spaces $R$
    - $(\mathbb{R}^k, \ell_p)$ for all $p \in [1, \infty)$, $(\mathbb{R}^k, \ell_{\infty})$, $(\mathbb{Z}^k, \ell_1)$, $(\mathbb{Z}^k, \ell_{\infty})$
    - the shortest path metric $d_G$ for all graphs $G$
  - generalization of monotonicity tester via transitive-closure-spanners [Dodis Goldreich Lehman R Ron Samorodnitky 99, Bhattacharyya Grigorescu Jung R Woodruff 09]
  - applies to all edge-transitive properties that allow extension

- **Lower bound:** $\Omega(\log n)$ queries for nondaptive 1-sided error tests
  - holds even for range $\mathbb{Z}$
Metric Convexity

- a standard notion in geometric functional analysis

A metric space \((R, d_R)\) is **metrically convex** if for all \(u, v \in R\) and all positive \(\alpha, \beta \in \mathbb{R}\) satisfying \(d_R(u, v) \leq \alpha + \beta\) there exists \(w \in R\) such that \(d_R(u, w) \leq \alpha\) and \(d_R(w, v) \leq \beta\).
Discrete Metric Convexity

- a relaxation of
  a standard notion in geometric functional analysis

A metric space \((R, d_R)\) is **discretely** metrically convex
if for all \(u, v \in R\) and
all positive \(\alpha, \beta \in \mathbb{Z}\) satisfying \(d_R(u, v) \leq \alpha + \beta\)
there exists \(w \in R\) such that \(d_R(u, w) \leq \alpha\) and \(d_R(w, v) \leq \beta\)
A property is **edge-transitive** if

1) it can be expressed in terms conditions on **ordered** pairs of domain points

2) it is **transitive**: whenever \((x, y)\) and \((y, z)\) satisfy (1), so does \((x, z)\)

A property **allows extension** if

3) any function that satisfies (1) on a subset of the domain can be extended to a function with the property
Our Results: Lipschitz Testers

Line $f : \{1, \ldots, n\} \rightarrow R$

- **Upper bound:** $O(\log n / \varepsilon)$ time
  - Applies to all discretely metrically convex spaces $R$
    - $\left(\mathbb{R}^k, \ell_p\right)$ for all $p \in [1, \infty)$, $\left(\mathbb{R}^k, \ell_\infty\right)$, $\left(\mathbb{Z}^k, \ell_1\right)$, $\left(\mathbb{Z}^k, \ell_\infty\right)$
    - The shortest path metric $d_G$ for all graphs $G$
  - Generalization of monotonicity tester via TC-spanners [DGLRRS99, BGJRW09]
  - Applies to all edge-transitive properties that allow extension

- **Lower bound:** $\Omega(\log n)$ queries for nondaptive 1-sided error tests
  - Holds even for range $\mathbb{Z}$
Our Results: Lipschitz Testers

Hypercube $f : \{0,1\}^d \rightarrow R$

- Upper bound: $O(d \cdot \min(d, \text{ImageDiam}(f))/ (\delta \varepsilon))$ time for range $\delta \mathbb{Z}$
  - same time to distinguish Lipschitz and $\varepsilon$-far from $(1+\delta)$-Lipschitz for range $\mathbb{R}$

- Lower bound: $\Omega(d)$ queries
  - tight for range $\{0,1,2\}$
  - reduction from a communication complexity problem
    (new technique due to [Blais Brody Matulef 11])
Our Results: Local Lipschitz Reconstructors

Hypergrid $f : \{1, \ldots, n\}^d \rightarrow \mathbb{R}$

- Upper bound: $O((\log n + 1)^d)$ time
- Lower bound: $\Omega\left(\frac{(\ln n - 1)^{d-1}}{d(4\pi)^d}\right)$ queries

for nonadaptive filters

Hypercube $f : \{0,1\}^d \rightarrow \mathbb{R}$

- Lower bound: $\Omega\left(2^{\alpha d}/d\right)$ queries, where $\alpha \approx 0.1620$

for nonadaptive filters
Hypercube Test: Important Special Case

Testing if \( f : \{0,1\}^d \rightarrow \mathbb{Z} \) is Lipschitz in \( O(d \cdot \min(d, \text{ImageDiam}(f)) / \varepsilon) \) time

- \( f \) is Lipschitz if its values on endpoints of every edge differ by at most 1.

- A an edge \( \{x,y\} \) is violated if \( |f(x) - f(y)| > 1 \)

Goal: Relate the number of violated edges, \( V(f) \), to the distance to the Lipschitz property.
Hypercube Test: Key Lemma

If $f : \{0,1\}^d \to \mathbb{Z}$ is $\varepsilon$-far from Lipschitz then $V(f) \geq \frac{\varepsilon \cdot 2^{d-1}}{\text{ImageDiam}(f)}$.

- **Enough to show:** we can make $f$ Lipschitz by modifying $2 \cdot V(f) \cdot \text{ImageDiam}(f)$ values.

- **Then** $2 \cdot V(f) \cdot \text{ImageDiam}(f) \geq \varepsilon \cdot 2^d$ for $\varepsilon$-far $f$, implying Key Lemma.
Averaging Operator

Plan: Transform $f$ into a Lipschitz function by repairing edges in one dimension at a time.

- As in the analysis of monotonicity tester in [DGLRRS99, GGLRS00]
  - Worked only for Boolean functions
  - General range was handled by induction on the size of the range
  - Function with range $\{0,1\}$ are all Lipschitz,
    with range $\{0,2\}$ are trivially testable
Plan: Repairing edges in one dimension at a time.

### Averaging Operator

For each violated edge \( \{x, y\} \) along dimension \( i \) with \( f(x) < f(y) + 1 \)

\[
\text{Averaging in dimension } i \quad \frac{f(x) + f(y)}{2} \quad \text{Averaging in dimension } i
\]

Issue: might increase the # of violated edges in other dimensions

Intuition: violation is “spread” among the edges in dimension \( j \)
**Potential Function Argument**

**Idea:** Take into account the magnitude of violations.

<table>
<thead>
<tr>
<th>Violation Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Violation score $\text{vs}({x, y}) = \max(0,</td>
</tr>
<tr>
<td>• $\text{VS}^j$ = sum of violation scores of edges along dimension $j$</td>
</tr>
</tbody>
</table>

Want to show: Averaging in dimension $i$ does not increase $\text{VS}^j$ for all dimensions $j \neq i$

**Issue:** averaging operator is complicated
**Basic Step Operator**

Idea: Break up the action of Averaging Operator into basic steps.

<table>
<thead>
<tr>
<th>Basic Step Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>For each violated edge ( {x, y} ) along dimension ( i ) with ( f(x) &lt; f(y) + 1 )</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
  f(x) &\quad \text{Basic Step in dimension } i \\
  f(y) &\quad f(x) + 1 \\
  f(y) - 1 \\
\end{align*}
\]

Averaging in dimension \( i \) = multiple Basic Steps in dimension \( i \)
**Basic Step Operator**

**Idea:** Break up the action of Averaging Operator into basic steps.

<table>
<thead>
<tr>
<th>Basic Step Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>For each violated edge ( {x, y} ) along dimension ( i ) with ( f(x) &lt; f(y) + 1 )</td>
</tr>
</tbody>
</table>

\[
\begin{array}{c}
  f(x) \quad \quad f(y) \\
  \text{Basic Step in dimension } i \\
  f(x) + 1 \quad \quad f(y) - 1
\end{array}
\]

Averaging in dimension \( i = \) multiple Basic Steps in dimension \( i \)
**Basic Step Operator**

**Idea:** Break up the action of Averaging Operator into basic steps.

<table>
<thead>
<tr>
<th>Basic Step Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>For each violated edge ( {x, y} ) along dimension ( i ) with ( f(x) &lt; f(y) + 1 )</td>
</tr>
</tbody>
</table>

\[
f(x) \quad f(y) \quad \text{Basic Step in dimension } i \quad f(x) + 1 \quad f(y) - 1
\]

Averaging in dimension \( i = \) multiple Basic Steps in dimension \( i \)
Basic Step Operator

**Idea:** Break up the action of Averaging Operator into basic steps.

<table>
<thead>
<tr>
<th>Basic Step Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>For each violated edge {x, y} along dimension (i) with (f(x) &lt; f(y) + 1)</td>
</tr>
</tbody>
</table>

\[
f(x) \xrightarrow{\text{Basic Step in dimension } i} f(x) + 1 \xrightarrow{\text{Basic Step in dimension } i} f(y) - 1
\]

Averaging in dimension \(i\) = multiple Basic Steps in dimension \(i\)

**Enough to show:**

Basic Step in dimension \(i\) does not increase \(VS^j\) \(\forall\) dimensions \(j \neq i\).
Basic Step in dimension $i$ does not increase $V_{S^j}$

Enough to prove it for squares

Can be proved by simple case analysis
**Analysis of the Averaging Operator**

*Know*: Averaging dimension $i$

1. repairs all violated edges in dimension $i$ (brings $VS^i$ down to 0)
2. doesn’t increase $VS^j \ \forall \text{dimensions } j \neq i$

- Averaging in dimensions $i = 1, \ldots, d$ repairs all violations because $VS^j = 0$ means “no violated edges in dimension $i$“

![Diagram](image.png)
Analysis of the Averaging Operator

How many function values are changed when averaging dimension \( i \)?

\[
2 \cdot (\text{# of violated edges in dimension } i \text{ after averaging dimensions } 1, \ldots, i - 1)
\]

- Let \( V^i(f) \) be the \# of edges in dimension \( i \) violated by \( f \)

\[
V^i(f) \leq VS^i(f) \leq V^i(f) \cdot \text{ImageDiam}(f)
\]

- Dimension \( i \) starts and ends up with \( VS^i \leq V^i(f) \cdot \text{ImageDiam}(f) \)

- \# of violated edges in dimension \( i \) never exceeds \( V^i(f) \cdot \text{ImageDiam}(f) \)

\# of changes

\[
= 2 \cdot (\text{# of violated edges in dimension } i \text{ after averaging dimensions } 1, \ldots, i - 1)
\leq 2 \cdot V(f) \cdot \text{ImageDiam}(f)
\]
**Lipschitz Test for Functions $f : \{0,1\}^d \rightarrow \mathbb{Z}$**

**Key Lemma**

If $f : \{0,1\}^d \rightarrow \mathbb{Z}$ is $\varepsilon$-far from Lipschitz then $V(f) \geq \frac{\varepsilon \cdot 2^{d-1}}{\text{ImageDiam}(f)}$

- i.e., fraction of violated edges is $\geq \frac{\varepsilon}{d \cdot \text{ImageDiam}(f)}$
- Enough to sample $\Theta(d \cdot \text{ImageDiam}(f)/\varepsilon)$ edges

**Issue:** $\text{ImageDiam}(f)$ can be $> 2^d$

**Observation:** A Lipschitz function on $\{0,1\}^d$ has image diameter at most $d$.

**Algorithm**

1. Sample $\Theta(1/\varepsilon)$ domain points $x$
2. $r = \max_{x} f(x) - \min_{x} f(x)$
3. If $r > d$, reject
4. Sample $\Theta(d \cdot r/\varepsilon)$ edges, and reject if any edge is violated
### Analysis of Lipschitz Hypercube Test

<table>
<thead>
<tr>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Sample $\Theta(1/\varepsilon)$ domain points $x$</td>
</tr>
<tr>
<td>2. $r = \max_{x} f(x) - \min_{x} f(x)$</td>
</tr>
<tr>
<td>3. If $r &gt; d$, reject</td>
</tr>
<tr>
<td>4. Sample $\Theta(d \cdot r/\varepsilon)$ edges, and reject if any edge is violated</td>
</tr>
</tbody>
</table>

If $f$ is Lipschitz, it is always accepted. ✓

Suppose $f$ is $\varepsilon$-far from Lipschitz.
- If $r > d$, the algorithm rejects. ✓
- It remains to consider the case $r \leq d$. 

![Diagram showing function $f$ and distance $\geq \varepsilon$]
Analysis of Lipschitz Hypercube Test

<table>
<thead>
<tr>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Sample $\Theta(1/\varepsilon)$ domain points $x$</td>
</tr>
<tr>
<td>2. $r = \max_x f(x) - \min_x f(x)$</td>
</tr>
<tr>
<td>3. If $r &gt; d$, <strong>reject</strong></td>
</tr>
<tr>
<td>4. Sample $\Theta(d \cdot r/\varepsilon)$ edges, and <strong>reject</strong> if any edge is violated</td>
</tr>
</tbody>
</table>

Suppose $f$ is $\varepsilon$-far from Lipschitz and $r \leq d$.

- W.h.p. $r$ is such that $f$ is $\varepsilon/2$-close to having image diameter $r$
  That is, some function $g$ at distance $< \varepsilon/2$ has image diameter $r$
- Let $a_{\min} = \min_x g(x)$ and $a_{\max} = \max_x g(x)$
  Let $\tilde{f}(x) = \begin{cases} 
    a_{\min} & \text{if } f(x) < a_{\min} \\
    a_{\max} & \text{if } f(x) > a_{\max} \\
    f(x) & \text{otherwise}
  \end{cases}$
- $\tilde{f}$ has image diameter $r$ and is at distance $< \varepsilon/2$ from $f$ $\Rightarrow$ it is $\varepsilon/2$-far from Lipschitz
Analysis of Lipschitz Hypercube Test

<table>
<thead>
<tr>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Sample $\Theta(1/\epsilon)$ domain points $x$</td>
</tr>
<tr>
<td>2. $r = \max_x f(x) - \min_x f(x)$</td>
</tr>
<tr>
<td>3. If $r &gt; d$, <strong>reject</strong></td>
</tr>
<tr>
<td>4. Sample $\Theta(d \cdot r/\epsilon)$ edges, and <strong>reject</strong> if any edge is violated</td>
</tr>
</tbody>
</table>

Suppose $f$ is $\epsilon$-far from Lipschitz and $r \leq d$.

- **We have:** $\tilde{f}$ has image diameter $r$ and is $\epsilon/2$-far from Lipschitz
- By Key Lemma, $V(\tilde{f}) \geq \frac{\epsilon/2}{d \cdot \text{ImageDiam}(\tilde{f})} = \frac{\epsilon}{2d \cdot r}$
- An edge is violated by $\tilde{f}$ only if it is violated by $f$
  \[ V(f) \geq V(\tilde{f}) \geq \frac{\epsilon}{2d \cdot r} \]
- Algorithm rejects w.h.p.
Our Results for the Lipschitz Property

TESTERS

Line \( f : \{1, \ldots, n\} \rightarrow \mathbb{R} \)

Hypercube \( f : \{0,1\}^d \rightarrow \mathbb{R} \)

- Upper bound: \( O(d \cdot \min(d, \text{ImageDiam}(f)) / (\delta \varepsilon)) \) time for range \( \delta \mathbb{Z} \)
  - same time to distinguish Lipschitz and \( \varepsilon \)-far from \((1+\delta)\)-Lipschitz for range \( \mathbb{R} \)

- Lower bound: \( \Omega(d) \) queries
  - tight for range \( \{0,1,2\} \)

LOCAL RECONSTRUCTORS

Hypergrid \( f : \{1,\ldots, n\}^d \rightarrow \mathbb{R} \)

Hypercube \( f : \{0,1\}^d \rightarrow \mathbb{R} \)
Open Questions

Lipschitz Property

• Tight bounds for testers on the hypercube
• Tester on the hypergrid
• Adaptive lower bounds for local filters on the hypercube/hypergrid
• (Nonlocal) reconstruction
• Explore more complicated ranges than $\mathbb{R}$
  – for testers on domains other than the line
  – for reconstructors

Other Properties

• Filters for data privacy mechanisms based on local notions of sensitivity
  – smooth sensitivity [Nissim Raskhodnikova Smith 07]