L_p-Testing

Grigory Yaroslavtsev

Penn
University of Pennsylvania

Appeared in STOC'14.

Joint work with Piotr Berman and Sofya Raskhodnikova

Testing Big Data

- **Q**: How to understand properties of large data looking only at a small sample?
- **Q**: How to ignore noise and outliers?
- **Q**: How to minimize assumptions about the sample generation process?
- **Q**: How to optimize running time?

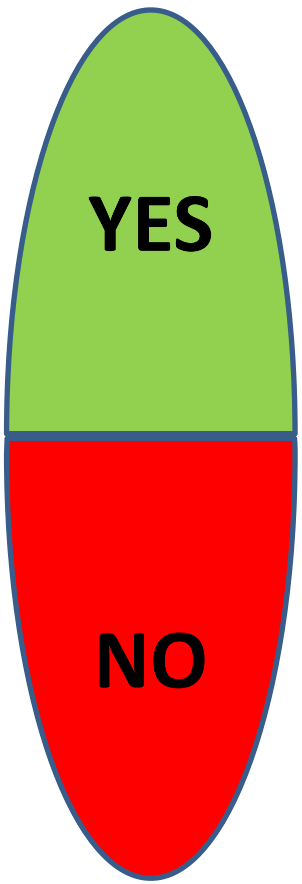
Which stocks were growing steadily?

Data from http://finance.google.com

Property Testing

[Goldreich, Goldwasser, Ron; Rubinfeld, Sudan]

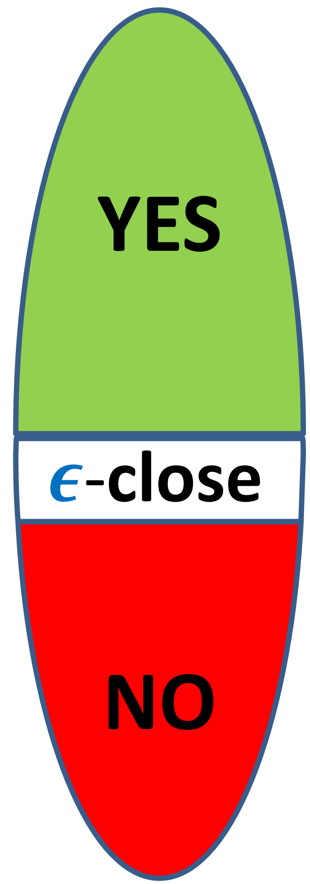
Randomized Algorithm



YES $\Rightarrow$ **Accept with probability** $\geq \frac{2}{3}$

NO $\Rightarrow$ **Reject with probability** $\geq \frac{2}{3}$

Property Tester



YES $\Rightarrow$ **Accept with probability** $\geq \frac{2}{3}$

ϵ-close $\Rightarrow$ **Don't care**

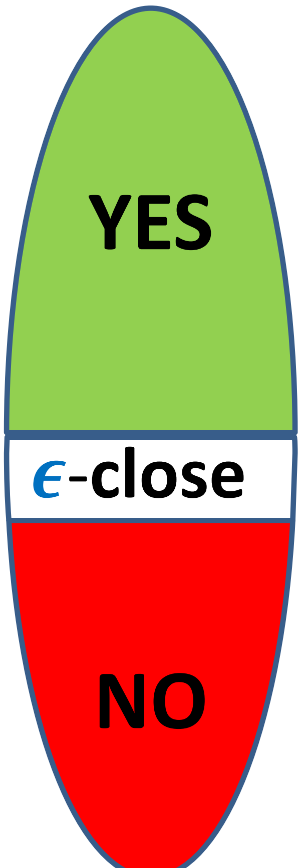
NO $\Rightarrow$ **Reject with probability** $\geq \frac{2}{3}$

ϵ-close : $\leq \epsilon$ fraction has to be changed to become **YES**

Tolerant Property Testing

[Parnas, Ron, Rubinfeld]

Property Tester

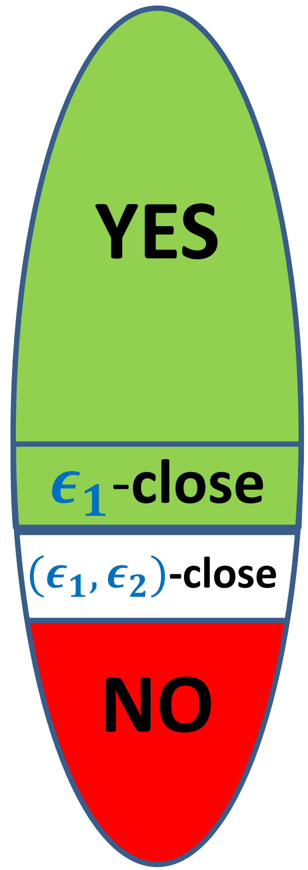


$\Rightarrow$ **Accept with probability $\geq \frac{2}{3}$**

$\Rightarrow$ **Don't care**

$\Rightarrow$ **Reject with probability $\geq \frac{2}{3}$**

Tolerant Property Tester



$\Rightarrow$ **Accept with probability $\geq \frac{2}{3}$**

$\Rightarrow$ **Don't care**

$\Rightarrow$ **Reject with probability $\geq \frac{2}{3}$**

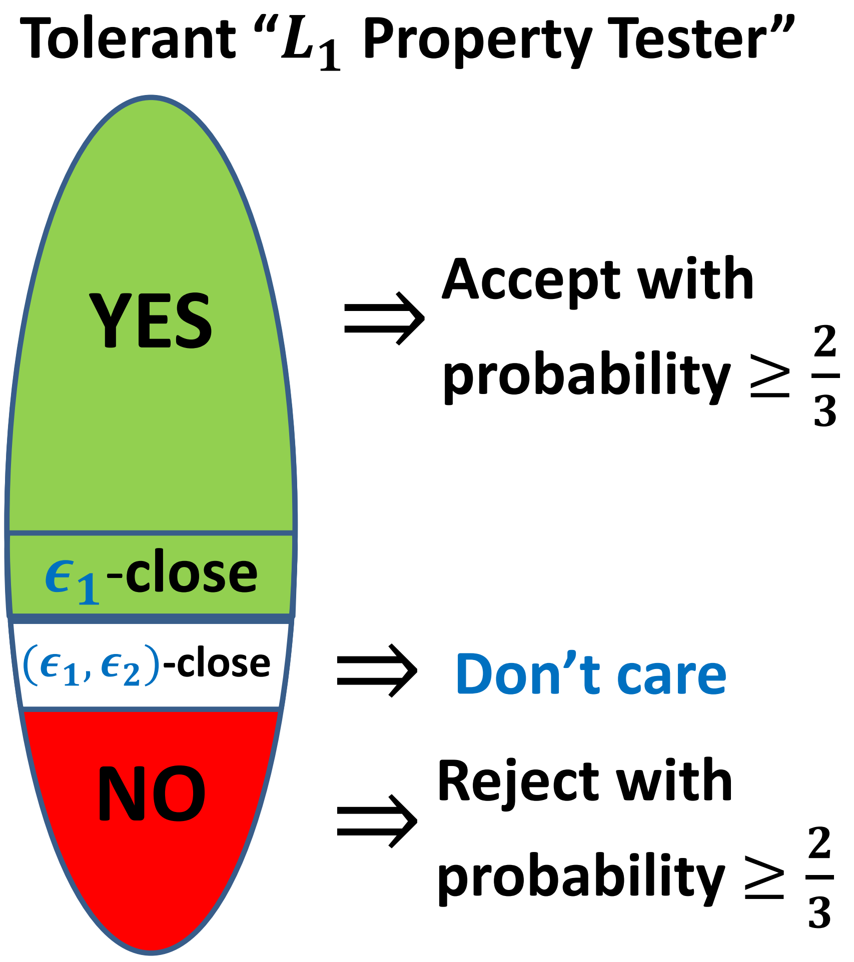
ϵ-**close :** $\leq \epsilon$ fraction has to be changed to become **YES**

Which stocks were growing steadily?

Data from http://finance.google.com

Tolerant "L_1 Property Testing"

- $\boldsymbol{f}: \{1, \ldots, n\} \rightarrow [0,1]$
- $\boldsymbol{P}$ = class of monotone functions
- $dist_1(\boldsymbol{f}, \boldsymbol{P}) = \frac{\min_{\boldsymbol{g} \in \boldsymbol{P}} |\boldsymbol{f} - \boldsymbol{g}|_1}{n}$
- $\boldsymbol{\epsilon}$-close: $dist_1(\boldsymbol{f}, \boldsymbol{P}) \leq \boldsymbol{\epsilon}$



New L_p-Testing Model for Real-Valued Data

- **Generalizes** standard Hamming testing

- For $p > 0$ still have a **probabilistic interpretation**:
$$d_p(f, g) = (\mathbf{E}[|\boldsymbol{f} - \boldsymbol{g}|^{\boldsymbol{p}}])^{1/p}$$

- Compatible with existing **PAC-style learning models** (preprocessing for model selection)

- For Boolean functions, $d_0(f, g) = d_p(f, g)^p$.

Our Contributions

1. Relationships between $L_{\boldsymbol{p}}$-testing models
2. Algorithms
   - $L_{\boldsymbol{p}}$-testers for $\boldsymbol{p} \geq 1$
     - monotonicity, Lipschitz, convexity
   - Tolerant $L_{\boldsymbol{p}}$-tester for $\boldsymbol{p} \geq 1$
     - monotonicity in 1D (sublinear algorithm for isotonic regression)

❖ Our $L_{\boldsymbol{p}}$-testers **beat lower bounds** for Hamming testers

❖ **Simple algorithms** backed up by involved analysis

❖ Uniformly sampled (or **easy to sample**) data suffices

3. Nearly tight lower bounds

Implications for Hamming Testing

Some techniques/results carry over to Hamming testing

- Improvement on **Levin's work investment strategy**
  - Connectivity of bounded-degree graphs [Goldreich, Ron '02]
  - Properties of images [Raskhodnikova '03]
  - Multiple-input problems [Goldreich '13]

- First example of **monotonicity testing** problem where **adaptivity helps**
- Improvements to Hamming testers for Boolean functions

Definitions

- $\boldsymbol{f}: D \rightarrow [0,1]$ (D = finite domain/poset)
- $||\boldsymbol{f}||_{\boldsymbol{p}} = \left(\sum_{x \in D} |\boldsymbol{f}(x)|^{\boldsymbol{p}}\right)^{1/\boldsymbol{p}}$, for $\boldsymbol{p} \geq 1$
- $||\boldsymbol{f}||_{\boldsymbol{0}}$ = Hamming weight (# of non-zero values)
- Property $\boldsymbol{P}$ = class of functions (monotone, convex, linear, Lipschitz, …)
- $dist_{\boldsymbol{p}}(\boldsymbol{f}, \boldsymbol{P}) = \dfrac{\min_{g \in \boldsymbol{P}} ||\boldsymbol{f} - g||_{\boldsymbol{p}}}{||1||_{\boldsymbol{p}}}$

Relationships: L_p-Testing

$Q_{\boldsymbol{p}}(\boldsymbol{P},\boldsymbol{\epsilon})$ = query complexity of $L_{\boldsymbol{p}}$-testing property $\boldsymbol{P}$ at distance $\boldsymbol{\epsilon}$

- $Q_{\mathbf{1}}(\boldsymbol{P},\boldsymbol{\epsilon}) \leq Q_{\mathbf{0}}(\boldsymbol{P},\boldsymbol{\epsilon})$
- $Q_{\mathbf{1}}(\boldsymbol{P},\boldsymbol{\epsilon}) \leq Q_{\mathbf{2}}(\boldsymbol{P},\boldsymbol{\epsilon})$ (Cauchy-Shwarz)
- $Q_{\mathbf{1}}(\boldsymbol{P},\boldsymbol{\epsilon}) \geq Q_{\mathbf{2}}(\boldsymbol{P},\sqrt{\boldsymbol{\epsilon}})$

Boolean functions $\boldsymbol{f}: D \to \{0,1\}$

$Q_{\mathbf{0}}(\boldsymbol{P},\boldsymbol{\epsilon}) = Q_{\mathbf{1}}(\boldsymbol{P},\boldsymbol{\epsilon}) = Q_{\mathbf{2}}(\boldsymbol{P},\sqrt{\boldsymbol{\epsilon}})$

Relationships: Tolerant L_p-Testing

$Q_{\boldsymbol{p}}(\boldsymbol{P}, \boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2})$ = query complexity of tolerant $L_{\boldsymbol{p}}$-testing property $\boldsymbol{P}$ with distance parameters $\boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2}$

- No general relationship between tolerant $L_{\boldsymbol{1}}$-testing and tolerant Hamming testing
- $L_{\boldsymbol{p}}$-testing for $\boldsymbol{p} > 1$ is close in complexity to $L_{\boldsymbol{1}}$-testing

$$Q_{\boldsymbol{1}}(\boldsymbol{P}, \boldsymbol{\varepsilon_1^p}, \boldsymbol{\varepsilon_2}) \leq Q_{\boldsymbol{p}}(\boldsymbol{P}, \boldsymbol{\varepsilon_1}, \boldsymbol{\varepsilon_2}) \leq Q_{\boldsymbol{1}}(\boldsymbol{P}, \boldsymbol{\varepsilon_1}, \boldsymbol{\varepsilon_2^p})$$

For Boolean functions $\boldsymbol{f}: D \to \{0,1\}$

$$Q_{\boldsymbol{0}}(\boldsymbol{P}, \boldsymbol{\varepsilon_1}, \boldsymbol{\varepsilon_2}) = Q_{\boldsymbol{1}}(\boldsymbol{P}, \boldsymbol{\varepsilon_1}, \boldsymbol{\varepsilon_2}) = Q_{\boldsymbol{p}}(\boldsymbol{P}, \boldsymbol{\epsilon_1^{1/p}}, \boldsymbol{\varepsilon_2^{1/p}})$$

Our Results: Testing Monotonicity

- Hypergrid ($D = [n]^d$)

	L_0	L_1
Upper bound	$O\left(\frac{d \log n}{\epsilon}\right)$ [Dodis et al. '99,…, Chakrabarti, Seshadhri '13]	$O\left(\frac{d}{\epsilon} \log \frac{d}{\epsilon}\right)$
Lower bound	$\Omega\left(\frac{d \log n}{\epsilon}\right)$ [Dodis et al.'99…, Chakrabarti, Seshadhri '13]	$\Omega\left(\frac{1}{\epsilon} \log \frac{1}{\epsilon}\right)$ Non-adaptive 1-sided error

- $2^{O(d)}/\epsilon$ **adaptive** tester for Boolean functions

Monotonicity: Key Lemma

- M = class of monotone functions
- Boolean slicing operator $\boldsymbol{f}_{\boldsymbol{y}}: D \to \{0,1\}$

  $\boldsymbol{f}_{\boldsymbol{y}}(x) = 1$, if $\boldsymbol{f}(x) \geq \boldsymbol{y}$,

  $\boldsymbol{f}_{\boldsymbol{y}}(x) = 0$, otherwise.

- **Theorem:**

$$dist_1(\boldsymbol{f}, M) = \int_0^1 dist_0(\boldsymbol{f}_{\boldsymbol{y}}, M) d\boldsymbol{y}$$

Proof sketch: slice and conquer

1) Closest monotone function with **minimal $\boldsymbol{L_1}$-norm** is **unique** (can be denoted as an operator $M_{\boldsymbol{f}}^1$).
2) $\|f - g\|_1 = \int_0^1 \left\| f_{\boldsymbol{y}} - g_{\boldsymbol{y}} \right\| d\boldsymbol{y}$
3) $M_{\boldsymbol{f}}^1$ and $\boldsymbol{f}_{\boldsymbol{y}}$ commute: $\left(M_{\boldsymbol{f}}^1\right)_{\boldsymbol{y}} = M^1{}_{(\boldsymbol{f}_{\boldsymbol{y}})}$

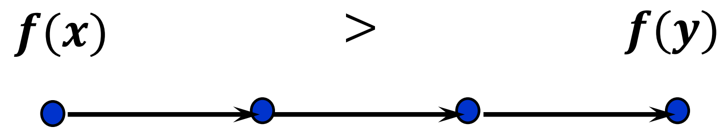
$$dist_1(f, M) \overset{1)}{=} \frac{\left\| f - M_f^1 \right\|_1}{|D|} \overset{2)}{=} \frac{\int_0^1 \left\| f_{\boldsymbol{y}} - (M_f^1)_{\boldsymbol{y}} \right\|_1 d\boldsymbol{y}}{|D|} \overset{3)}{=}$$

$$= \frac{\int_0^1 \left\| f_{\boldsymbol{y}} - M^1_{(f_{\boldsymbol{y}})} \right\|_1 d\boldsymbol{y}}{|D|} = \int_0^1 dist_0(f_{\boldsymbol{y}}, M) d\boldsymbol{y}$$

L_1-Testers from Boolean Testers

Thm: A nonadaptive, 1-sided error L_0-test for monotonicity of $f: D \to \{0,1\}$ is also an L_1-test for monotonicity of $f: D \to [0,1]$.

Proof:



- A violation (x, y):
- A nonadaptive, 1-sided error test queries a random set $Q \subseteq D$ and rejects iff Q contains a violation.
- If $f: D \to [0,1]$ is monotone, Q will not contain a violation.
- If $d_1(f, M) \geq \varepsilon$ then $\exists \boldsymbol{t}^*: d_0\left(\boldsymbol{f}_{(\boldsymbol{t}^*)}, M\right) \geq \boldsymbol{\varepsilon}$
- W.p. $\geq 2/3$, set Q contains a violation (x, y) for $\boldsymbol{f}_{(\boldsymbol{t}^*)}$

$$\boldsymbol{f}_{(\boldsymbol{t}^*)}(x) = 1, \boldsymbol{f}_{(\boldsymbol{t}^*)}(y) = 0$$

$$\Downarrow$$

$$\boldsymbol{f}(x) > \boldsymbol{f}(y)$$

Distance Approximation and Tolerant Testing

Approximating L_1-distance to monotonicity $\pm\delta$ $w.p. \geq 2/3$

f	L_0	L_1
$[n] \to [0,1]$	$\text{polylog}\, n \cdot \left(\frac{1}{\delta}\right)^{O(1/\delta)}$ [Saks Seshadhri 10]	$\Theta\left(\frac{1}{\delta^2}\right)$

- Time complexity of tolerant L_1-testing for monotonicity is

$$O\left(\frac{\varepsilon_2}{(\varepsilon_2 - \varepsilon_1)^2}\right)$$

  - Better dependence than what follows from distance appoximation for $\epsilon_2 \ll 1$
  - Improves $\tilde{O}\left(\frac{1}{\delta^2}\right)$ adaptive distance approximation of [Fattal,Ron'10] for Boolean functions

L_1-Testers for Other Properties

Via combinatorial characterization of L_1-distance to the property

- Lipschitz property $\boldsymbol{f}: [\boldsymbol{n}]^{\boldsymbol{d}} \to [0,1]$:

$$\Theta\left(\frac{\boldsymbol{d}}{\epsilon}\right)$$

Via (implicit) **proper learning**: approximate in L_1 up to error $\boldsymbol{\epsilon}$, test approximation on a random $O(1/\epsilon)$-sample

- Convexity $\boldsymbol{f}: [\boldsymbol{n}]^{\boldsymbol{d}} \to [0,1]$**:**

$$\mathrm{O}\left(\boldsymbol{\epsilon}^{-\frac{\boldsymbol{d}}{2}} + \frac{1}{\boldsymbol{\epsilon}}\right) \text{ (tight for } \boldsymbol{d} \leq 2\text{)}$$

- Submodularity $\boldsymbol{f}: \{0,1\}^{\boldsymbol{d}} \to [0,1]$

$$2^{\tilde{O}\left(\frac{1}{\epsilon}\right)} + poly\left(\frac{1}{\boldsymbol{\epsilon}}\right) \log \boldsymbol{d}$$ [Feldman, Vondrak 13]

Open Problems

- All our algorithms for for $p > 1$ were obtained directly from L_1-testers.

Can one design better algorithms by working directly with L_p-distances?

- Our complexity for L_p-testing convexity grows exponentially with d

Is there an L_p-testing algorithm for convexity with subexponential dependence on the dimension?

- Our L_1-tester for monotonicity is nonadaptive, but we show that adaptivity helps for Boolean range.

Is there a better adaptive tester?

- We designed tolerant tester only for monotonicity (d=1,2).

Tolerant testers for higher dimensions?

Other properties?