Higher-Order Fourier Analysis: Applications to Algebraic Property Testing

Yuichi Yoshida

National Institute of Informatics, and
Preferred Infrastructure, Inc

May 28, 2016

Decision Problems

- Function $f : \mathbb{F}_2^n \to \{0, 1\}$.
- Function property $\mathcal{P}$.
  (such as linearity: $f(x) + f(y) \equiv f(x + y) \bmod 2$ for all x, y.)

Q. How long does it take to decide f satisfies $\mathcal{P}$?

Decision Problems

- Function $f : \mathbb{F}_2^n \to \{0, 1\}$.
- Function property $\mathcal{P}$.
  (such as linearity: $f(x) + f(y) \equiv f(x + y) \bmod 2$ for all x, y.)

Q. How long does it take to decide f satisfies $\mathcal{P}$?
A. Trivially, it takes 2^n time.

Decision Problems

- Function $f : \mathbb{F}_2^n \to \{0, 1\}$.
- Function property $\mathcal{P}$.
  (such as linearity: $f(x) + f(y) \equiv f(x + y) \bmod 2$ for all x, y.)

Q. How long does it take to decide f satisfies $\mathcal{P}$?
A. Trivially, it takes 2^n time.

Q. Can we do something in sublinear or even in constant time?

Property testing

Definition

$f : \{0,1\}^n \to \{0,1\}$ is *ϵ-far* from $\mathcal{P}$ if,

$$d_{\mathcal{P}}(f) := \min_{g \in \mathcal{P}} \frac{\#\{x \in \{0,1\}^n \mid f(x) \neq g(x)\}}{2^n} \geq \epsilon.$$

Property testing

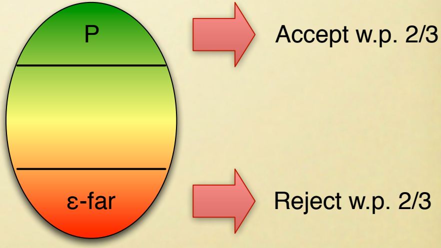
Definition

$f : \{0,1\}^n \to \{0,1\}$ is *ϵ-far* from $\mathcal{P}$ if,

$$d_{\mathcal{P}}(f) := \min_{g \in \mathcal{P}} \frac{\#\{x \in \{0,1\}^n \mid f(x) \neq g(x)\}}{2^n} \geq \epsilon.$$

A *tester* for a property $\mathcal{P}$:
Given

- $f : \{0,1\}^n \to \{0,1\}$ as a query access.
- proximity parameter $\epsilon > 0$.



Testing $f \equiv 1$

Input: a function $f : \{0,1\}^n \to \{0,1\}$ and $\epsilon > 0$.
Goal: $f(x) = 1$ for every $x \in \{0,1\}^n$?

Testing $f \equiv 1$

Input: a function $f : \{0,1\}^n \to \{0,1\}$ and $\epsilon > 0$.
Goal: $f(x) = 1$ for every $x \in \{0,1\}^n$?

1: **for** $i = 1$ to $\Theta(1/\epsilon)$ **do**
2: Sample $x \in \{0,1\}^n$ uniformly at random.
3: **if** $f(x) = 0$ **then** reject.
4: Accept.

Testing $f \equiv 1$

Input: a function $f : \{0,1\}^n \to \{0,1\}$ and $\epsilon > 0$.
Goal: $f(x) = 1$ for every $x \in \{0,1\}^n$?

1: **for** $i = 1$ to $\Theta(1/\epsilon)$ **do**
2: Sample $x \in \{0,1\}^n$ uniformly at random.
3: **if** $f(x) = 0$ **then** reject.
4: Accept.

Theorem

- *If $f \equiv 1$, always accepts. (one-sided error)*

Testing $f \equiv 1$

Input: a function $f : \{0,1\}^n \to \{0,1\}$ and $\epsilon > 0$.
Goal: $f(x) = 1$ for every $x \in \{0,1\}^n$?

for $i = 1$ to $\Theta(1/\epsilon)$ **do**
    Sample $x \in \{0,1\}^n$ uniformly at random.
    if $f(x) = 0$ **then** reject.
Accept.

Theorem

- *If $f \equiv 1$, always accepts. (one-sided error)*
- *If f is ϵ-far, accepts with probability $(1-\epsilon)^{\Theta(1/\epsilon)} \leq 1/3$.*

Testing $f \equiv 1$

Input: a function $f : \{0,1\}^n \to \{0,1\}$ and $\epsilon > 0$.
Goal: $f(x) = 1$ for every $x \in \{0,1\}^n$?

for $i = 1$ to $\Theta(1/\epsilon)$ **do**
  Sample $x \in \{0,1\}^n$ uniformly at random.
  if $f(x) = 0$ **then** reject.
Accept.

Theorem

- *If $f \equiv 1$, always accepts. (one-sided error)*
- *If f is ϵ-far, accepts with probability $(1-\epsilon)^{\Theta(1/\epsilon)} \leq 1/3$.*
- *Query complexity is $O(1/\epsilon) \Rightarrow$ constant!*

Backgrounds

Property testing was introduced by [RS96] for program checking.

Backgrounds

Property testing was introduced by [RS96] for program checking.

Since then, various kinds of objects have been studied.
Ex.: Functions, graphs, distributions, geometric objects, images.

Backgrounds

Property testing was introduced by [RS96] for program checking.

Since then, various kinds of objects have been studied.
Ex.: Functions, graphs, distributions, geometric objects, images.

Q. Why do we study property testing?

Backgrounds

Property testing was introduced by [RS96] for program checking.

Since then, various kinds of objects have been studied.
Ex.: Functions, graphs, distributions, geometric objects, images.

Q. Why do we study property testing?
A. Interested in

- ultra-efficient algorithms.
- connections to inapproximability, locally testable codes, and learning.
- the relation between local view and global property.

Local testability of affine-Invariant properties

Definition

$\mathcal{P}$ is *affine-invariant* if a function $f : \mathbb{F}_2^n \to \{0,1\}$ satisfies $\mathcal{P}$, then $f \circ A$ satisfies $\mathcal{P}$ for any bijective affine transformation $A : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

Local testability of affine-Invariant properties

Definition

$\mathcal{P}$ is *affine-invariant* if a function $f : \mathbb{F}_2^n \to \{0,1\}$ satisfies $\mathcal{P}$, then $f \circ A$ satisfies $\mathcal{P}$ for any bijective affine transformation $A : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

Definition

$\mathcal{P}$ is *(locally) testable* if there is a tester for $\mathcal{P}$ with $q(\epsilon)$ queries.

Local testability of affine-Invariant properties

Some specific locally testable affine-invariant properties:

- Degree-d polynomials [AKK+05, BKS+10]
- Fourier sparsity [GOS+11]
- Odd-cycle-freeness: There exist no $x_1, \ldots, x_{2k+1} \in \mathbb{F}_2^n$ such that $f(x_1) = \cdots = f(x_{2k+1}) = 1$ and $x_1 + \cdots + x_{2k+1} \equiv 0$ [BGRS12].

The goal

Q. Can we characterize locally testable affine-invariant properties? [KS08]

The goal

Q. Can we characterize locally testable affine-invariant properties? [KS08]

A. Yes.

The goal

Q. Can we characterize locally testable affine-invariant properties? [KS08]

A. Yes.

In this talk, we review how we have attacked this question.

- One-sided error testable $\approx$ Affine-subspace hereditary
- Testable $\Leftrightarrow$ Estimable
- Two-sided error testable $\Leftrightarrow$ Regular-reducible

Higher order Fourier analysis has played a crucial role!

Fourier analysis

A function $f : \mathbb{F}_2^n \to \mathbb{R}$ can be uniquely decomposed as

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x),$$

where $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$.
$\widehat{f}(S)$ measures the correlation of f with χ_S. *(Fourier coefficients)*

Fourier analysis

A function $f : \mathbb{F}_2^n \to \mathbb{R}$ can be uniquely decomposed as

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x),$$

where $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$.
$\widehat{f}(S)$ measures the correlation of f with χ_S. *(Fourier coefficients)*

Fourier analysis is

- powerful enough to study specific properties.

Fourier analysis

A function $f : \mathbb{F}_2^n \to \mathbb{R}$ can be uniquely decomposed as

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x),$$

where $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$.
$\widehat{f}(S)$ measures the correlation of f with χ_S. *(Fourier coefficients)*

Fourier analysis is

- powerful enough to study specific properties.
- not powerful enough to obtain general results.

Higher order Fourier analysis

We look at correlations with polynomials instead of linear functions.

Higher order Fourier analysis

We look at correlations with polynomials instead of linear functions.

Main technical tools:

- *Decomposition theorem*
  A function can be decomposed into a structured part + pseudorandom part (with respect to low-degree polynomials)
- *Equidistribution theorem*
  "generic" polynomials look independently distributed.

Higher order Fourier analysis

We look at correlations with polynomials instead of linear functions.

Main technical tools:

- *Decomposition theorem*
  A function can be decomposed into a structured part + pseudorandom part (with respect to low-degree polynomials)
- *Equidistribution theorem*
  "generic" polynomials look independently distributed.

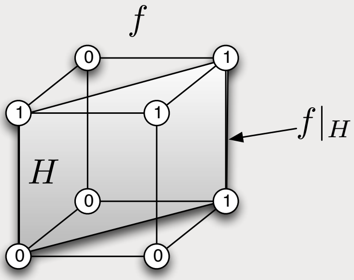
Caveat: In this talk, we do not touch most of technical foundations such as Gowers norm, rank, and bias.

Oblivious tester

Definition

An *oblivious tester* works as follows:

- Take a restriction $f|_H$.
  - H: random affine subspace of dimension $h(\epsilon)$.
- Output based only on $f|_H$.



Motivation: avoid "unnatural" properties such as $f \in \mathcal{P} \Leftrightarrow n$ is even.
For natural properties, $\exists$ a tester $\Rightarrow \exists$ an oblivious tester [BGS10].

Decomposition theorem

$\mu_{f,h}$: the distribution of $f|_H$.

Observation

A tester cannot distinguish f from g if $\mu_{f,h} \approx \mu_{g,h}$.

Decomposition theorem

$\mu_{f,h}$: the distribution of $f|_H$.

Observation

A tester cannot distinguish f from g if $\mu_{f,h} \approx \mu_{g,h}$.

Theorem (Decomposition theorem)

Any function can be decomposed as $f = f_1 + f_2 + f_3$ for $d = d(\epsilon, h)$:

- $f_1 = \Gamma(P_1, \ldots, P_C)$ *for "generic" degree-d polynomials* $\{P_i\}$.
- f_2: *small* L_2 *norm.*
- f_3: *uncorrelated with degree-d polynomials.*

Decomposition theorem

$\mu_{f,h}$: the distribution of $f|_H$.

Observation

A tester cannot distinguish f from g if $\mu_{f,h} \approx \mu_{g,h}$.

Theorem (Decomposition theorem)

Any function can be decomposed as $f = f_1 + f_2 + f_3$ for $d = d(\epsilon, h)$:

- $f_1 = \Gamma(P_1, \ldots, P_C)$ *for "generic" degree-d polynomials* $\{P_i\}$.
- f_2: *small* L_2 *norm.*
- f_3: *uncorrelated with degree-d polynomials.*

The *pseudorandom parts* f_2 and f_3 do not affect $\mu_{f,h}$ much.
$\Rightarrow$ we can focus on the *structured part* f_1.

One-sided error testable $\approx$ Affine-subspace hereditary

Affine-subspace hereditary

Definition

A property $\mathcal{P}$ is *affine-subspace hereditary* if $f \in \mathcal{P} \Rightarrow f|_H \in \mathcal{P}$ for any affine subspace H.

Ex.:

- degree-d polynomials, Fourier sparsity, odd-cycle-freeness
- $f = gh$ for some polynomials g, h of degree $\leq d - 1$.
- $f = g^2$ for some polynomial g of degree $\leq d - 1$.

Characterization of one-sided error testability

Conjecture ([BGS10])

$\mathcal{P}$ is testable with one-sided error by an oblivious tester
$\Leftrightarrow$ $\mathcal{P}$ is (essentially) affine-subspace hereditary

Characterization of one-sided error testability

Conjecture ([BGS10])

$\mathcal{P}$ is testable with one-sided error by an oblivious tester
$\Leftrightarrow$ $\mathcal{P}$ is (essentially) affine-subspace hereditary

$\Rightarrow$ is true [BGS10].

Characterization of one-sided error testability

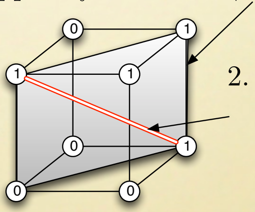
Conjecture ([BGS10])

$\mathcal{P}$ is testable with one-sided error by an oblivious tester
$\Leftrightarrow$ $\mathcal{P}$ is (essentially) affine-subspace hereditary

$\Rightarrow$ is true [BGS10].

Proof sketch:

1. Suppose $f \in \mathcal{P}$ and $f|_H \notin \mathcal{P}$



2. $\exists f|_K$, rejected by the tester

3. f is also rejected w.p.> 0, contradiction.

Alternative formulation via linear forms

Think of *affine-triangle-freeness*:
No $x, y_1, y_2 \in \mathbb{F}_2^n$ s.t. $f(x + y_1) = f(x + y_2) = f(x + y_1 + y_2) = 1$.

Alternative formulation via linear forms

Think of *affine-triangle-freeness*:
No $x, y_1, y_2 \in \mathbb{F}_2^n$ s.t. $f(x + y_1) = f(x + y_2) = f(x + y_1 + y_2) = 1$.

$\Leftrightarrow$ No $x, y_1, y_2 \in \mathbb{F}_2^n$ s.t.

$$f(L_1(x, y_1, y_2)) = \sigma_1 \text{ for } L_1(x, y_1, y_2) = x + y_1 \text{ and } \sigma_1 = 1,$$
$$f(L_2(x, y_1, y_2)) = \sigma_2 \text{ for } L_2(x, y_1, y_2) = x + y_2 \text{ and } \sigma_2 = 1,$$
$$f(L_3(x, y_1, y_2)) = \sigma_3 \text{ for } L_3(x, y_1, y_2) = x + y_1 + y_2 \text{ and } \sigma_3 = 1.$$

Alternative formulation via linear forms

Think of *affine-triangle-freeness*:
No $x, y_1, y_2 \in \mathbb{F}_2^n$ s.t. $f(x+y_1) = f(x+y_2) = f(x+y_1+y_2) = 1$.

$\Leftrightarrow$ No $x, y_1, y_2 \in \mathbb{F}_2^n$ s.t.

$$f(L_1(x,y_1,y_2)) = \sigma_1 \text{ for } L_1(x,y_1,y_2) = x + y_1 \text{ and } \sigma_1 = 1,$$
$$f(L_2(x,y_1,y_2)) = \sigma_2 \text{ for } L_2(x,y_1,y_2) = x + y_2 \text{ and } \sigma_2 = 1,$$
$$f(L_3(x,y_1,y_2)) = \sigma_3 \text{ for } L_3(x,y_1,y_2) = x + y_1 + y_2 \text{ and } \sigma_3 = 1.$$

We call this *$(A = (L_1, L_2, L_3), \sigma = (\sigma_1, \sigma_2, \sigma_3))$-freeness*.

Alternative formulation via linear forms

Think of *affine-triangle-freeness*:
No $x, y_1, y_2 \in \mathbb{F}_2^n$ s.t. $f(x+y_1) = f(x+y_2) = f(x+y_1+y_2) = 1$.

$\Leftrightarrow$ No $x, y_1, y_2 \in \mathbb{F}_2^n$ s.t.

$$f(L_1(x,y_1,y_2)) = \sigma_1 \text{ for } L_1(x,y_1,y_2) = x+y_1 \text{ and } \sigma_1 = 1,$$
$$f(L_2(x,y_1,y_2)) = \sigma_2 \text{ for } L_2(x,y_1,y_2) = x+y_2 \text{ and } \sigma_2 = 1,$$
$$f(L_3(x,y_1,y_2)) = \sigma_3 \text{ for } L_3(x,y_1,y_2) = x+y_1+y_2 \text{ and } \sigma_3 = 1.$$

We call this *$(A = (L_1, L_2, L_3), \sigma = (\sigma_1, \sigma_2, \sigma_3))$-freeness*.

- A is called an *affine* system of *linear forms*.
  $\Rightarrow$ well studied in higher order Fourier analysis.

Testability of subspace hereditary properties

Observation

The following are equivalent:

- $\mathcal{P}$ *is affine-subspace hereditary.*
- *There exists a (possibly infinite) collection* $\{(A^1, \sigma^1), \ldots\}$ *s.t.* $f \in \mathcal{P} \Leftrightarrow f$ *is* (A^i, σ^i)*-free for each* i.

Testability of subspace hereditary properties

Observation

The following are equivalent:

- $\mathcal{P}$ *is affine-subspace hereditary.*
- *There exists a (possibly infinite) collection* $\{(A^1, \sigma^1), \ldots\}$ *s.t.* $f \in \mathcal{P} \Leftrightarrow f$ *is* (A^i, σ^i)*-free for each* i.

Theorem ([BFH+13])

If each (A^i, σ^i) *has bounded complexity, then the property is testable with one-sided error.*

Proof idea

Let's focus on the case $f = \Gamma(P_1, \ldots, P_C)$ and $\mathcal{P} =$ affine $\triangle$-freeness.

Proof idea

Let's focus on the case $f = \Gamma(P_1, \ldots, P_C)$ and $\mathcal{P}$ = affine $\triangle$-freeness.

f is ϵ-far from $\mathcal{P}$
$\Rightarrow$ There are $x^*, y_1^*, y_2^* \in \mathbb{F}_2^n$ spanning an affine triangle.

Proof idea

Let's focus on the case $f = \Gamma(P_1, \ldots, P_C)$ and $\mathcal{P}$ = affine $\triangle$-freeness.

f is ϵ-far from $\mathcal{P}$
$\Rightarrow$ There are $x^*, y_1^*, y_2^* \in \mathbb{F}_2^n$ spanning an affine triangle.

$$\Pr_{x,y_1,y_2}[f(x+y_1) = f(x+y_2) = f(x+y_1+y_2) = 1]$$
$$\geq \Pr_{x,y_1,y_2}[P_i(L_j(x,y_1,y_2)) = P_i(L_j(x^*,y_1^*,y_2^*))\ \forall i \in [C], j \in [3]],$$

which is non-negligibly high from the *equidistribution theorem*.
$\Rightarrow$ Random sampling works.

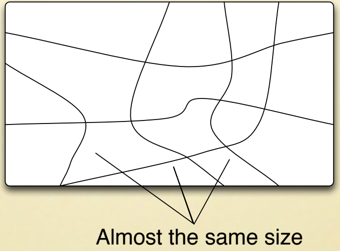
Equidistribution theorem

The space $\mathbb{F}_2^n$ can be divided according to $\{P_i(L_j(x))\}_{i \in [C], j \in [3]}$.

Theorem (Equidistribution theorem)

If P_i's are "generic" enough, then each cell has almost the same size.

$\mathbb{F}_2^n =$



Testability $\Leftrightarrow$ Estimability

Testability $\Leftarrow$ Estimability

Definition

$\mathcal{P}$ is *estimable* if we can estimate $d_{\mathcal{P}}(\cdot)$ to within δ with $q(\delta)$ queries for any $\delta > 0$.

Testability $\Leftarrow$ Estimability

Definition

$\mathcal{P}$ is *estimable* if we can estimate $d_{\mathcal{P}}(\cdot)$ to within δ with $q(\delta)$ queries for any $\delta > 0$.

Trivial direction: $\mathcal{P}$ is estimable $\Rightarrow$ $\mathcal{P}$ is testable.

Testability $\Leftarrow$ Estimability

Definition

$\mathcal{P}$ is *estimable* if we can estimate $d_{\mathcal{P}}(\cdot)$ to within δ with $q(\delta)$ queries for any $\delta > 0$.

Trivial direction: $\mathcal{P}$ is estimable $\Rightarrow$ $\mathcal{P}$ is testable.

Theorem ([HL13])

$\mathcal{P}$ is testable $\Rightarrow$ $\mathcal{P}$ is estimable.

Testability $\Leftarrow$ Estimability

Definition

$\mathcal{P}$ is *estimable* if we can estimate $d_{\mathcal{P}}(\cdot)$ to within δ with $q(\delta)$ queries for any $\delta > 0$.

Trivial direction: $\mathcal{P}$ is estimable $\Rightarrow$ $\mathcal{P}$ is testable.

Theorem ([HL13])

$\mathcal{P}$ is testable $\Rightarrow$ $\mathcal{P}$ is estimable.

Algorithm:

1: $H \leftarrow$ a random affine subspace of a constant dimension.
2: **return** Output $d_{\mathcal{P}}(f|_H)$.

Intuition behind the proof

Why can we hope $d_{\mathcal{P}}(f) \approx d_{\mathcal{P}}(f|_H)$*?*

Intuition behind the proof

Why can we hope $d_{\mathcal{P}}(f) \approx d_{\mathcal{P}}(f|_H)$*?*

(Oversimplified argument)

- Since $\mathcal{P}$ is testable, $d_{\mathcal{P}}(f)$ is determined by the distribution $\mu_{f,h}$.

Intuition behind the proof

Why can we hope $d_{\mathcal{P}}(f) \approx d_{\mathcal{P}}(f|_H)$?

(Oversimplified argument)

- Since $\mathcal{P}$ is testable, $d_{\mathcal{P}}(f)$ is determined by the distribution $\mu_{f,h}$.
- If $f = \Gamma(P_1, \ldots, P_C)$, then $\mu_{f,h}$ is determined by Γ and degrees of $P_1, \ldots, P_C$ (rather than *P_i's themselves*).

Intuition behind the proof

Why can we hope $d_{\mathcal{P}}(f) \approx d_{\mathcal{P}}(f|_H)$?

(Oversimplified argument)

- Since $\mathcal{P}$ is testable, $d_{\mathcal{P}}(f)$ is determined by the distribution $\mu_{f,h}$.
- If $f = \Gamma(P_1, \ldots, P_C)$, then $\mu_{f,h}$ is determined by Γ and degrees of $P_1, \ldots, P_C$ (rather than *P_i's themselves*).
- $f = \Gamma(P_1, \ldots, P_C)$ and $f_H = \Gamma(P_1|_H, \ldots, P_C|_H)$ share the same Γ and degrees.

  $\Rightarrow \mu_{f,h} \approx \mu_{f|_H,h}$.

  $\Rightarrow d_{\mathcal{P}}(f) \approx d_{\mathcal{P}}(f|_H)$.

Two-sided error testability $\Leftrightarrow$ Regular-reducibility

Structured part

Recall that, for $f = \Gamma(P_1, \ldots, P_C) + f_2 + f_3$,

$\mu_{f,h}$ is determined by Γ and degrees of P_i's.

Let's use them as a (constant-size) sketch of f.

Regularity-instance (simplified)

Definition

A *regularity-instance* I is a tuple of

- a complexity parameter $C \in \mathbb{N}$,
- a structure function $\Gamma : \mathbb{F}_2^C \to [0, 1]$,
- a degree-bound parameter $d \in \mathbb{N}$,
- a degree parameter $\mathbf{d} = (d_1, \ldots, d_C) \in \mathbb{N}^C$ with $d_i < d$,

Satisfying a regularity-instance

Definition

Let $I = (C, \Gamma, d, \mathbf{d})$ be a regularity-instance.
f *satisfies I* if it is of the form

$$f(x) = \Gamma(P_1(x), \ldots, P_C(x)) + \Upsilon(x),$$

where

- P_i is a polynomial of degree d_i,
- $P_1, \ldots, P_C$ are "generic" enough.
- Υ is uncorrelated with degree-$(d-1)$ polynomials.

Testing regularity-instances

Theorem ([Yos14a])

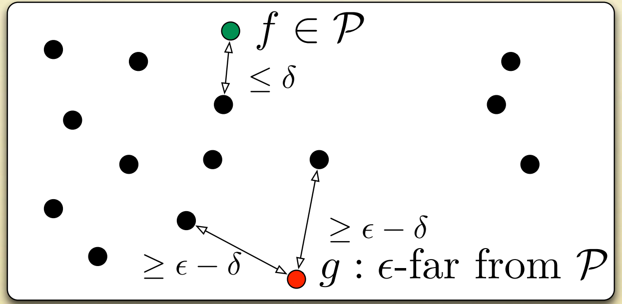
For any regularity-instance I, there is a tester for the property of satisfying I.

Algorithm:

1: $H \leftarrow$ a random affine subspace of a constant dimension.
2: **if** $f|_H$ is close to satisfying I **then** accept.
3: **else** reject.

Regular-reducibility

A property $\mathcal{P}$ is *regular-reducible* if for any $\delta > 0$, there exists a set $\mathcal{R}$ of constant number of regularity-instances such that:



Characterization of two-sided error testability

Theorem

An affine-invariant property $\mathcal{P}$ is testable

$$\Updownarrow$$

$\mathcal{P}$ is regular-reducible.

Characterization of two-sided error testability

Theorem

An affine-invariant property $\mathcal{P}$ is testable

$$\Updownarrow$$

$\mathcal{P}$ is regular-reducible.

Proof sketch:

- Regular-reducible $\Rightarrow$ testable
  Regularity-instances are testable, and testability implies estimability [HL13]. Hence, we can estimate the distance to $\mathcal{R}$.
- Testable $\Rightarrow$ regular-reducible
  The behavior of a tester depends only on $\mu_{f,h}$. Since Γ and $\mathbf{d}$ determines the distribution, we can find $\mathcal{R}$ using the tester.

Notes

- We need to deal with "non-classical" polynomials instead of polynomials.
- Another characterization of testability was shown by introducing "functions limits" [Yos14b].
- Applications of the characterizations:
  - Low-degree polynomials.
  - Having a low spectral norm $\sum_S |\widehat{f}(S)|$.

Summary

Higher order Fourier analysis is useful for studying property testing as

- we care about the distribution $\mu_{f,h}$ for $h = O(1)$,
- which is determined by the structured part given by the decomposition theorem.

Summary

Higher order Fourier analysis is useful for studying property testing as

- we care about the distribution $\mu_{f,h}$ for $h = O(1)$,
- which is determined by the structured part given by the decomposition theorem.

We are almost done, *qualitatively*.

- one-sided error testability $\approx$ affine-subspace hereditary (of bounded complexity)
- two-sided error testability $\Leftrightarrow$ regular-reducibility.

Future direction

Property Testing

- Other groups:
  - Abelian $\Rightarrow$ higher order Fourier analysis exists [Sze12].
  - Non-Abelian $\Rightarrow$ representation theory? [OY16]
- Why is affine invariance easier to deal with than permutation invariance?

Other applications of higher order Fourier analysis.

- Coding theory [BG16, BL15a].
- Learning theory [BHT15].
- Complexity theory [BL15b].
- Algorithms for polynomials [Bha14].