Tolerant Junta Testing and the Connection to Submodular Optimization and Function Isomorphism

Eric Blais* Clément Canonne† Talya Eden‡ **Amit Levi***
Dana Ron‡

* University of Waterloo

† Stanford University

‡ Tel Aviv University

6.5.18

Juntas

Juntas

A function $f\colon \{-1,1\}^n \to \{-1,1\}$ is a **k-junta** if it depends on at most k of its variables.

Juntas

A function $f: \{-1,1\}^n \to \{-1,1\}$ is a **k-junta** if it depends on at most k of its variables.

Why should we care?

Juntas

A function $f\colon \{-1,1\}^n \to \{-1,1\}$ is a **k-junta** if it depends on at most k of its variables.

Why should we care?

- Juntas model learning in the presence of irrelevant attributes.

Juntas

A function $f\colon \{-1,1\}^n \to \{-1,1\}$ is a **k-junta** if it depends on at most k of its variables.

Why should we care?

- Juntas model learning in the presence of irrelevant attributes.
- Central object of study in the analysis of Boolean functions.

Juntas

A function $f\colon \{-1,1\}^n \to \{-1,1\}$ is a **k-junta** if it depends on at most k of its variables.

Why should we care?

- Juntas model learning in the presence of irrelevant attributes.
- Central object of study in the analysis of Boolean functions.
  - Approximates well other (complex) classes of functions.

Juntas

A function $f\colon \{-1,1\}^n \to \{-1,1\}$ is a **k-junta** if it depends on at most k of its variables.

Why should we care?

- Juntas model learning in the presence of irrelevant attributes.
- Central object of study in the analysis of Boolean functions.
  - Approximates well other (complex) classes of functions.
  - Connections to hardness of approximation.

(Tolerant) Testing Juntas

Given query access to unknown $f : \{-1,1\}^n \to \{-1,1\}$ and $0 < \varepsilon_1 < \varepsilon_2 < 1$:

(Tolerant) Testing Juntas

Given query access to unknown $f : \{-1, 1\}^n \to \{-1, 1\}$ and $0 < \varepsilon_1 < \varepsilon_2 < 1$:

- If f is $\boldsymbol{\varepsilon_1}$-close to a **$k$-junta**, accept w.p $2/3$.

(Tolerant) Testing Juntas

Given query access to unknown $f : \{-1,1\}^n \to \{-1,1\}$ and $0 < \varepsilon_1 < \varepsilon_2 < 1$:

- If f is $\boldsymbol{\varepsilon_1}$-close to a **$k$-junta**, accept w.p $2/3$.
- If f is $\boldsymbol{\varepsilon_2}$-far from any **$k$-junta**, reject w.p $2/3$.

(Tolerant) Testing Juntas

Given query access to unknown $f : \{-1,1\}^n \to \{-1,1\}$ and $0 < \varepsilon_1 < \varepsilon_2 < 1$:

- If f is $\boldsymbol{\varepsilon_1}$-close to a **$k$-junta**, accept w.p $2/3$.
- If f is $\boldsymbol{\varepsilon_2}$-far from any **$k$-junta**, reject w.p $2/3$.

Where

$$\mathrm{dist}(f,g) = \Pr_{x\sim U}[f(x) \neq g(x)] .$$

(Tolerant) Testing Juntas

Given query access to unknown $f : \{-1,1\}^n \to \{-1,1\}$ and $0 < \varepsilon_1 < \varepsilon_2 < 1$:

- If f is $\boldsymbol{\varepsilon_1}$-close to a **$k$-junta**, accept w.p $2/3$.
- If f is $\boldsymbol{\varepsilon_2}$-far from any **$k$-junta**, reject w.p $2/3$.

Where

$$\mathrm{dist}(f,g) = \Pr_{x\sim U}[f(x) \neq g(x)]\ .$$

Theorem ([FKRSS04, CG04, Bla09]).
For $\varepsilon_1 = 0$, the query complexity for testing k-juntas is $\tilde{\Theta}(k/\varepsilon_2)$.

(Tolerant) Testing Juntas

Given query access to unknown $f : \{-1,1\}^n \to \{-1,1\}$ and $0 < \varepsilon_1 < \varepsilon_2 < 1$:

- If f is $\boldsymbol{\varepsilon_1}$-close to a **$k$-junta**, accept w.p $2/3$.
- If f is $\boldsymbol{\varepsilon_2}$-far from any **$k$-junta**, reject w.p $2/3$.

Where

$$\mathrm{dist}(f,g) = \Pr_{x\sim U}[f(x) \neq g(x)] \,.$$

Theorem ([FKRSS04, CG04, Bla09]).
For $\varepsilon_1 = 0$, the query complexity for testing k-juntas is $\boldsymbol{\tilde{\Theta}(k/\epsilon_2)}$.

However, in many practical scenarios the function is not exactly a **k-junta** but **close** to such.

Our Results

Theorem 1.
There is an algorithm that for any $\varepsilon > 0$ satisfies the following:

Our Results

Theorem 1.
There is an algorithm that for any $\varepsilon > 0$ satisfies the following:

- *If f is* $\boldsymbol{\varepsilon/10}$ *close to* ***k-junta****, then the algorithm accepts with high probability.*

Our Results

Theorem 1.
There is an algorithm that for any $\varepsilon > 0$ satisfies the following:

- *If f is $\boldsymbol{\varepsilon/10}$ close to **k-junta**, then the algorithm accepts with high probability.*
- *If f is $\boldsymbol{\varepsilon}$ far from any **$2k$-junta**, then the algorithm rejects with high probability.*

Our Results

Theorem 1.
There is an algorithm that for any $\varepsilon > 0$ satisfies the following:

- *If f is $\boldsymbol{\varepsilon/10}$ close to **$\boldsymbol{k}$-junta**, then the algorithm accepts with high probability.*
- *If f is $\boldsymbol{\varepsilon}$ far from any **$\boldsymbol{2k}$-junta**, then the algorithm rejects with high probability.*

The query complexity of the algorithm is $\boldsymbol{poly(k, 1/\varepsilon)}$.

Our Results

Theorem 1.
There is an algorithm that for any $\varepsilon > 0$ satisfies the following:

- *If f is $\boldsymbol{\varepsilon/10}$ close to **k-junta**, then the algorithm accepts with high probability.*
- *If f is $\boldsymbol{\varepsilon}$ far from any **$2k$-junta**, then the algorithm rejects with high probability.*

The query complexity of the algorithm is $\boldsymbol{poly(k, 1/\varepsilon)}$.

The proof uses a techniques from **submodular optimization**

Our Results

Theorem 2 ("Smooth Tradeoff").
There is an algorithm that for any $\varepsilon > 0$ and $\rho \in (0, 1)$ satisfies the following:

Our Results

Theorem 2 ("Smooth Tradeoff").

There is an algorithm that for any $\varepsilon > 0$ and $\rho \in (0,1)$ satisfies the following:

- *If f is* $\boldsymbol{\rho\varepsilon/16}$ *close to* ***k-junta****, then the algorithm accepts with high probability.*

Our Results

Theorem 2 ("Smooth Tradeoff").
There is an algorithm that for any $\varepsilon > 0$ and $\rho \in (0, 1)$ satisfies the following:

- *If f is $\boldsymbol{\rho\varepsilon/16}$ close to **k-junta**, then the algorithm accepts with high probability.*
- *If f is $\boldsymbol{\varepsilon}$ far from any **k-junta**, then the algorithm rejects with high probability.*

Our Results

Theorem 2 (“Smooth Tradeoff”).
There is an algorithm that for any $\varepsilon > 0$ and $\rho \in (0,1)$ satisfies the following:

- *If f is $\rho\varepsilon/16$ close to **k-junta**, then the algorithm accepts with high probability.*
- *If f is ε far from any **k-junta**, then the algorithm rejects with high probability.*

The query complexity of the algorithm is $O\left(\frac{k \log k}{\varepsilon} \cdot \frac{1}{\rho(1-\rho)^k}\right)$.

Our Results

Theorem 3.
There is an algorithm that given query access to f and g and any $\varepsilon > 0$ satisfies the following:

- *If f and g are ε/c-close to **isomorphic**, then the algorithm accepts with high probability.*
- *If f and g are ε-far from **isomorphic**, then the algorithm rejects with high probability.*

The query complexity of the algorithm is $O(2^{k^*/2}/\varepsilon)$ where k^* is the smallest k such that either f or g are ε/c-close to a junta.

For more details, come talk to me during the poster session

Thanks