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

Eric Blais, Clément Canonne, Talya Eden, Amit Levi, and Dana Ron

What is a k-junta, and why should we care?

University of Waterloo

Object of interest: Boolean function $f\colon \{0,1\}^n \to \{0,1\}$. Might want to *learn*, *approximate*, *manipulate* f – but n is **huge**. This will take time, and resources.

Hope: many **irrelevant features**. What if f actually only depended on $k \ll n$ variables? We then could try to "pay" k instead of n everywhere!

Goal: given blackbox access to f and a parameter k, **find out** if the function is a k-variate function "in disguise."

Now, even that may not be enough: we want to be **robust**. If our function only *mostly* depends on k variables, that should be good enough! I.e., we want to be able to **tolerate a little bit of noise.**

Juntas, testing, and tolerance

Definition. *A Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is said to be a k-*junta *if there exists a set $T \subseteq [n]$ of size at most k, such that $f(x) = f(y)$ for every two assignments $x, y \in \{0,1\}^n$ that satisfy $x_i = y_i$ for every $i \in T$.*

We want to detect juntas efficiently, to avoid insane running times depending on n whenever possible. And this can be done:

Theorem ([3, 4, 5, 6])**.** *Testing whether a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is a k-junta has query complexity $\tilde{\Theta}(k/\epsilon)$,* **independent of n.**

But what about this robustness we would like to obtain? Can we test efficiently whether a function is *close* to a junta?

Definition. *A* tolerant testing algorithm for a property $\mathcal{P}$ *is a probabilistic algorithm $\mathcal{T}$ that gets two input parameters $\epsilon_1, \epsilon_2 \in [0,1]$ with $\epsilon_1 < \epsilon_2$, and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$; and outputs a binary verdict that satisfies the following two conditions.*

- *If* $\operatorname{dist}(f, \mathcal{P}) \le \epsilon_1$*, then $\mathcal{T}$ accepts with probability at least $2/3$.*
- *If* $\operatorname{dist}(f, \mathcal{P}) > \epsilon_2$*, then $\mathcal{T}$ rejects with probability at least $2/3$.*

Case $\epsilon_1 = 0$: "usual" testing. But being tolerant is harder – and sometimes **much** harder [2]. *Is it the case here?*

Summary of results

We give two (incomparable) results for **tolerant testing of k-juntas**, each with query complexity *independent of n.*

Theorem. *There exists an algorithm that, given query access to $f\colon \{0,1\}^n \to \{0,1\}$ and parameters $k \ge 1$ and $\epsilon \in (0,1)$, satisfies the following.*

- *If f is $\epsilon/10$-close to some k-junta, then the algorithm accepts with probability at least $2/3$.*
- *If f is ϵ-far from every $2k$-junta, then the algorithm rejects with probability at least $2/3$.*

The query complexity of the algorithm is $\operatorname{poly}(k, \frac{1}{\epsilon})$.

Exploits a connection to **submodular minimization**: approximate minimization of a (noisy) submodular function under a cardinality constraint. Yields an *efficient* algorithm for our testing problem – with a small catch.

Our second algorithm does not include that relaxation of the soundness condition, but features a **tradeoff** between tolerance and query complexity:

Theorem. *There exists an algorithm that, given query access to $f\colon \{0,1\}^n \to \{0,1\}$ and parameters $k \ge 1$, $\epsilon \in (0,1)$ and $\rho \in (0,1)$, satisfies the following.*

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

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

Retrieves weakly tolerant results of Fischer et al. [7] for $\rho = \Theta(1/k)$, and tolerant tester with query complexity $O(2^k/\epsilon)$ for $\rho = \Omega(1)$. Setting ρ, this can also be leveraged to obtain the following:

Application: "instance-by-instance" (tolerant) isomorphism testing of $f, g\colon \{0,1\}^n \to \{0,1\}$. **"Why pay n if there is a better parameter $k = k(f,g)$?"**

References

[1] Eric Blais, Clément Canonne, Talya Eden, Amit Levi, and Dana Ron Tolerant Junta Testing and the Connection to Submodular Optimization and Function Isomorphism. *Manuscript*, 2016. arXiv:1607.03938

[2] Eldar Fischer and Lance Fortnow. Tolerant Versus Intolerant Testing for Boolean Properties. *Theory of Computing*, 2(9):173–183, 2006.

[3] Eric Blais. Improved bounds for testing juntas. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 317–330. Springer, 2008.

[4] Eric Blais. Testing juntas nearly optimally. In *Proceedings of STOC*, pages 151–158. ACM, 2009.

[5] Hana Chockler and Dan Gutfreund. A Lower Bound for Testing Juntas. *Information Processing Letters*, 90(6):301–305, 2004.

[6] Rocco A. Servedio, Li-Yang Tan, and John Wright. Adaptivity helps for testing juntas. In *Conference on Computational Complexity*, volume 33 of *LIPIcs*, pages 264–279. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[7] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753–787, 2004.

[8] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Proceedings of FOCS*, pages 1049–1065. IEEE, 2015.