# Complexity Measures on the Symmetric Group and Beyond


**Neta Dafni**
Department of Computer Science, Technion, Haifa, Israel
netad@cs.technion.ac.il

**Yuval Filmus**
Department of Computer Science, Technion, Haifa, Israel
https://yuvalfilmus.cs.technion.ac.il/
yuvalfi@cs.technion.ac.il

**Noam Lifshitz**
Einstein Institute of Mathematics, Hebrew University of Jerusalem, Israel
noamlifshitz@gmail.com

**Nathan Lindzey**
Department of Computer Science, University of Colorado, Boulder, CO, USA
http://nathanlindzey.com/
Nathan.Lindzey@colorado.edu

**Marc Vinyals**
Department of Computer Science, Technion, Haifa, Israel
http://www.csc.kth.se/~vinyals/
marcviny@cs.technion.ac.il



## Abstract

We extend the definitions of complexity measures of functions to domains such as the symmetric group. The complexity measures we consider include degree, approximate degree, decision tree complexity, sensitivity, block sensitivity, and a few others. We show that these complexity measures are polynomially related for the symmetric group and for many other domains.

To show that all measures but sensitivity are polynomially related, we generalize classical arguments of Nisan and others. To add sensitivity to the mix, we reduce to Huang's sensitivity theorem using "pseudo-characters", which witness the degree of a function.

Using similar ideas, we extend the characterization of Boolean degree 1 functions on the symmetric group due to Ellis, Friedgut and Pilpel to the perfect matching scheme. As another application of our ideas, we simplify the characterization of maximum-size $t$-intersecting families in the symmetric group and the perfect matching scheme.



**2012 ACM Subject Classification** Theory of computation → Computational complexity and cryptography; Mathematics of computing → Discrete mathematics

**Keywords and phrases** Computational Complexity Theory, Analysis of Boolean Functions, Complexity Measures, Extremal Combinatorics

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2021.87

**Category** Extended Abstract

**Related Version** A full version of the paper is available at https://arxiv.org/abs/2010.07405.

**Funding** This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 802020-ERC-HARMONIC.

**Acknowledgements** We thank Nitin Saurabh for many helpful discussions.

# 1 Introduction

A classical result in complexity theory states that a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ of degree $d$ can be computed using a decision tree of depth $\mathrm{poly}(d)$. Conversely, a Boolean function computed by a decision tree of depth $d$ has degree at most $d$. Thus degree and decision tree complexity are polynomially related. Other complexity measures which are polynomially related to the degree include approximate degree, certificate complexity, and block sensitivity. Recently, Huang [9] added sensitivity to the list.

Can we prove similar results for Boolean functions on other domains? Such domains have been introduced to complexity theory in recent years: for example, O'Donnell and Wimmer [13] used Boolean functions on the so-called "slice" to construct optimal nets for monotone functions; Barak et al. [1] used Boolean functions on the Reed–Muller code to construct and analyze the influential "short code"; and recently, Khot, Minzer and Safra [10] proved the 2-to-2 conjecture using Boolean functions on the Grassmann scheme.

Although yet to see applications to complexity theory, perhaps the most appealing domain is the symmetric group. We say that a function $f\colon S_n \to \mathbb{R}$ has degree at most $d$ if any of the following equivalent conditions hold:

1. $f(\pi)$ can be written as a linear combination of *$d$-juntas*, which are functions depending on $\pi(i_1), \dots, \pi(i_d)$ for some $i_1, \dots, i_d \in [n]$.
2. Representing the input as a permutation matrix, $f$ can be written as a degree $d$ polynomial in the entries of the matrix.
3. $f$ has Fourier-degree $d$, that is, it is supported on isotypic components corresponding to partitions $\lambda$ with $\lambda_1 \geqslant n - d$.

(The reader who is not familiar with representation theory can ignore the last definition.)

What is the correct generalization of decision tree? We take our inspiration from the work of Ellis, Friedgut and Pilpel [4], which characterized the Boolean degree 1 functions on $S_n$. These are functions that depend on some $\pi(i)$ or on some $\pi^{-1}(j)$. This suggests the following definition: a decision tree for functions on $S_n$ is a decision tree with queries of the form "$\pi(i) = ?$" and "$\pi^{-1}(j) = ?$". Essentially the same definition ("matching decision trees") is used in lower bounds on the pigeonhole principle [14].

We show that this is a good definition by proving that degree and decision tree complexity are polynomially related for the symmetric group. In fact, we are able to generalize many other complexity measures to the symmetric group, and show that all of them are polynomially related:

▶ **Theorem 1.** *The following complexity measures (appropriately defined) are all polynomially related for Boolean functions over the symmetric group: degree, approximate degree, decision tree complexity, certificate complexity, unambiguous certificate complexity, sensitivity, block sensitivity, fractional block sensitivity, quantum query complexity.*

Our results hold for many other domains, such as the perfect matching scheme (the set of all perfect matchings in $K_{2n}$) and balanced slices (the balanced slice consists of all vectors in $\{0,1\}^{2n}$ with equally many 0s and 1s, and is also known as the Johnson scheme $J(2n, n)$).

We prove Theorem 1 and its generalizations in an abstract framework based on simplicial complexes. In this framework, every point in the domain is a set. For example:

1. Boolean cube: We identify each vector $x \in \{0,1\}^n$ with the set $\{(i, x_i) : i \in [n]\}$.
2. Symmetric group: We identify each permutation $\pi \in S_n$ with the set $\{(i, \pi(i)) : i \in [n]\}$.

A function has degree $d$ if it can be written as a linear combination of functions of the form "the input set contains $S$", where $|S| \leqslant d$; this generalizes the usual notion of degree in these two domains. Our decision trees allow any queries of the form "which element of the set $Q$ does the input set contain?", as long as there is a unique answer for every input.

With this setup in place, we are able to polynomially relate all complexity measures other than sensitivity by generalizing classical arguments, as presented by Buhrman and de Wolf [2], for example. To add sensitivity to the mix, we reduce to Huang's sensitivity theorem [9] using basic representation theory.

Generalizing ideas of Gopalan et al. [8], we also prove the following simple result:

▶ **Theorem 2.** *If a function on the symmetric group has sensitivity $s$, then it can be recovered from its evaluation on a ball of radius $O(s)$ around an arbitrary permutation.*

Using this, we show that low sensitivity functions can be computed efficiently:

▶ **Theorem 3.** *If a function on the symmetric group has sensitivity $s$, then it can be computed using a circuit of size $n^{O(s)}$.*

This should be compared to a decision tree for the function, which corresponds to a balanced formula of size $n^{O(D)}$, where $D \geqslant s$ is the decision tree complexity.

## 1.1 Degree 1 functions

Our results show that in a wide variety of domains, Boolean degree 1 functions can be computed by constant depth decision trees. Can we say more?

Boolean degree 1 functions on the Boolean cube are *dictators*, that is, depend on a single coordinate, and the same holds for functions on the balanced slice. Ellis, Friedgut and Pilpel [4] showed that the same holds for the symmetric group, with the correct interpretation of "dictator": a function depending only on some $\pi(i)$, or only on some $\pi^{-1}(j)$. In contrast, Filmus and Ihringer [6] showed that Boolean degree 1 functions on the Grassmann scheme ($k$-dimensional subspaces of an $n$-dimensional vector space over a finite field) could depend on two different "data points".

Among the domains we consider, in many cases Boolean degree 1 functions are trivially dictators. In some other cases, describing all Boolean degree 1 functions seems difficult. We identify one case in which the problem is feasible:

▶ **Theorem 4.** *A Boolean function on the perfect matching scheme has degree at most $1$ if and only if it is one of the following: a constant function; a function depending on the match of some vertex $i$; or a function depending on whether the perfect matching intersects some triangle.*

Incidentally, this is another example in which there are non-dictatorial degree 1 functions, namely those depending on intersections with a triangle.

We prove Theorem 4 using polyhedral techniques. As in the proof of the corresponding result for the symmetric group by Ellis, Friedgut and Pilpel (which we paraphrase using our methods), we first characterize all *nonnegative* degree 1 functions, using the classical characterization of supporting hyperplanes of the perfect matching polytope. To deduce the result for Boolean functions, we use a simple result from the theory of complexity measures: a degree 1 function has sensitivity at most 1.

The reader is perhaps wondering about nonnegative functions of higher degree. Can we say anything intelligent about them? It turns out that the answer is negative already for the Boolean cube: classical results on the Sherali–Adams hierarchy [7] show that there exist nonnegative degree 2 functions which, if written as nonnegative linear combinations of monomials over literals (that is, products of factors of the form $x_i$ and $1 - x_j$), require degree $\Omega(n)$.

## 1.2 Application to Erdős–Ko–Rado theory

The work of Ellis, Friedgut and Pilpel, which has already been mentioned several times, is about intersecting families of permutations. A subset $\mathcal{F} \subseteq S_n$ is *$t$-intersecting* if any two $\pi_1, \pi_2 \in \mathcal{F}$ agree on the image of at least $t$ points. In other words, if we think of $\pi_1, \pi_2$ as sets (as in our setup), then $|\pi_1 \cap \pi_2| \geqslant t$. How large can a $t$-intersecting family be? One construction is a *$t$-star*:

$$\mathcal{F} = \{\pi \in S_n : \pi(i_1) = j_1, \dots, \pi(i_t) = j_t\}.$$

Ellis et al. show that for large enough $n$ (depending on $t$), these families have the maximum possible size, and moreover uniquely so: every $t$-intersecting family of the maximum size $(n-t)!$ is a $t$-star. Unfortunately, their argument for the uniqueness claim is wrong, see [5]. Uniqueness can be recovered from the work of Ellis [3], which proves a much stronger result, and is quite complicated.

We give a much simpler proof of uniqueness, using the connection between degree and certificate complexity:

▶ **Theorem 5.** *For every $t, d$, the following holds for large enough $n$. If $f$ is the characteristic vector of a $t$-intersecting family and* $\deg f \leqslant d$*, then either $f$ is contained in a $t$-star, or the corresponding family contains $O((n-t-1)!)$ permutations.*

Ellis et al. show that a $t$-intersecting family of size $(n-t)!$ must have degree $t$ (for large enough $n$), and so Theorem 5 shows that for large enough $n$, such a family must be a $t$-star.

Theorem 5 generalizes to other domains for which similar intersection theorems are known, such as the perfect matching scheme [11, 12], and to cross-$t$-intersecting families.

We see Theorem 5 as a contribution of theoretical computer science to extremal combinatorics. It illustrates the usefulness of the theory developed in this work.

## References


1 Boaz Barak, Parikshit Gopalan, Johan Håstad, Raghu Meka, Prasad Raghavendra, and David Steurer. Making the long code shorter. *SIAM J. Comput.*, 44(5):1287–1324, 2015. doi:10.1137/130929394.

2 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoret. Comput. Sci.*, 288(1):21–43, 2002. Complexity and logic (Vienna, 1998). doi:10.1016/S0304-3975(01)00144-X.

3 David Ellis. Stability for $t$-intersecting families of permutations. *J. Combin. Theory Ser. A*, 118(1):208–227, 2011. doi:10.1016/j.jcta.2010.04.005.

4 David Ellis, Ehud Friedgut, and Haran Pilpel. Intersecting families of permutations. *J. Amer. Math. Soc.*, 24(3):649–682, 2011. doi:10.1090/S0894-0347-2011-00690-5.

5 Yuval Filmus. A comment on intersecting families of permutations. *arXiv*, abs/1706.10146, 2017. arXiv:1706.10146.

6 Yuval Filmus and Ferdinand Ihringer. Boolean degree 1 functions on some classical association schemes. *J. Combin. Theory Ser. A*, 162:241–270, 2019. doi:10.1016/j.jcta.2018.11.006.

7 Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. Optimal Sherali-Adams gaps from pairwise independence. In *Approximation, randomization, and combinatorial optimization*, volume 5687 of *Lecture Notes in Comput. Sci.*, pages 125–139. Springer, Berlin, 2009. doi:10.1007/978-3-642-03685-9_10.

8 Parikshit Gopalan, Noam Nisan, Rocco A. Servedio, Kunal Talwar, and Avi Wigderson. Smooth Boolean functions are easy: efficient algorithms for low-sensitivity functions. In *ITCS'16—Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 59–70. ACM, New York, 2016. doi:10.1145/2840728.2840738.

**9** Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Ann. of Math. (2)*, 190(3):949–955, 2019. `doi:10.4007/annals.2019.190.3.6`.

**10** Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in Grassmann graph have near-perfect expansion. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 592–601, 2018.

**11** Nathan Lindzey. Intersecting families of perfect matchings. *ArXiv*, abs/1811.06160, 2018.

**12** Nathan Lindzey. *Matchings and representation theory.* PhD thesis, University of Waterloo, 2018.

**13** Ryan O'Donnell and Karl Wimmer. KKL, Kruskal-Katona, and monotone nets. *SIAM J. Comput.*, 42(6):2375–2399, 2013. `doi:10.1137/100787325`.

**14** Alasdair Urquhart and Xudong Fu. Simplified lower bounds for propositional proofs. *Notre Dame J. Formal Logic*, 37(4):523–544, 1996. `doi:10.1305/ndjfl/1040046140`.