Research Statement

Sourav Chakraborty

My current research interests are in various subjects in Theoretical Computer Science. In the past few years I have mostly worked on the area of property testing.

Other than property testing I have also worked on other areas such as coding theory[CRRS06, CFM11, BC], graph algorithms[CFMY11, CFLY10], algorithmic game theory[CL12, CDK10, CD09], combinatorial measures of Boolean functions[Cha11] and combinatorics[CRR05, CGSM11a].

1 Summary of My Work in Property Testing

Many data sets that arise in the fields of biology, geology, astronomy, climatology, artificial intelligence, etc are massive. In fact they are so huge that even reading the whole data requires impossibly large resources. At the same time we need to analyze and have a good estimate of whether the data has certain property. We can use statistical methods to analyze the data. But statistical methods has it limitations. One alternative is “property testing”: looking at a very small (ideally constant) part of the data and use combinatorial techniques to decide whether the data has a predetermined property or is “far” from satisfying the property.

This way of analyzing the data started in the late eighties and early nineties [BLR90, BFL91, RS96]. Subsequently it had important application in the fields of probabilistically checkable proofs (PCP) [AS98, ALM$^+$98] and learning and approximation [GGR98]. In recent years the field of combinatorial property testing has enjoyed a rapid growth (see, e. g., [AFNS09, AS], cf. [Ron08, Fis01])

The query complexity for a given property is a measure of the fraction of data that has to be accessed/read/queried to determine if the data has the property or is “far” from satisfying the property. I have worked on both classical and quantum query complexity (when the queries are made by a quantum computer) and have designed effective algorithms as well as proved lower bounds on the query complexity for various important properties.

The better understanding of the query complexity for different problems also has applications to other areas in theoretical computer science like learning theory, online algorithms and coding theory.

1.1 Testing of Graph Isomorphism and its Generalizations

Graphs are combinatorial objects that are used to represent relations over a data set. Graph isomorphism is one of the most important and well studied problems in mathematics and theoretical computer science. The problem is to check whether two given graphs are isomorphic to each other. There is no known fast algorithm for testing graph isomorphism.

In the case of property testing the problem is to distinguish between the case where the graphs are isomorphic and the case where we need to change at least a constant fraction of the adjacency matrix to make the graphs isomorphic.

In [BC10] we study a natural and important generalization of the graph isomorphism: testing isomorphisms under group actions. Given two n bit strings x and y and a permutation group G we want to test whether there is a permutation π in G such that by permuting the indices of x by the π we obtain the string y. This abstract generalization has applications in many fields. For example, in image recognition if we want to check whether one images is a shifts of another we consider the group of all shifts of the pixels of the images and test for isomorphism under this group action. Also graph and hyper-graph isomorphism are special cases.

With Babai [BC10] we study this problem when G is a primitive group. Primitive groups acts as building blocks for other permutation groups. Hence understanding query complexity under primitive group action is the first but significant step toward the main problem. In fact various finite geometric transformations (like projective, orthogonal, symplectic, etc) corresponds to primitive groups. Graph and hyper-graph isomorphism also corresponds to primitive groups actions. Eldar and Matsliah [FM08] obtained the exact bounds on the query complexity for testing graph isomorphism. We generalize their results to obtain tight bounds on the query complexity for testing isomorphism under the action of a primitive permutation group G.

In [CFMdW10] with Fischer, Matsliah and de'Wolf we study the problem of testing of graph isomorphism when the graph is accessed by a quantum computer (instead of a classical computer). Quantum computers is known is be much more powerful than classical computers. Property testing using quantum computer has been studied earlier. We show that in case of graph isomorphism also the quantum computer does significantly better that the classical computers - that is the quantum computer can make much less number of queries to the graphs to determine if they are isomorphic or far from isomorphic.

1.2 Testing of Function Isomorphism

Two n-variate functions are said to be isomorphic if both the functions are isomorphic up to a permutation of the input variables. This is a very important problem and has connections to learning theory, artificial intelligence and many other areas in computer science. In property testing we are interested in distinguishing the case when the two functions are isomorphic from the case when the function value has to be changed at least a ϵ fraction of the domain space to make them isomorphic.

The question of testing isomorphism was first formulated explicitly by Fischer, Kindler, Ron, Safra, and Samorodnitsky [FKR$^+$04]. They gave a general upper bound on the query complexity by showing that for every function f that depends on k variables (that is, for every k-junta), the problem of testing isomorphism to f is solvable with polynomial in k number of queries. They gave a lower bound of $\log k$ for a special function (namely the function that is a parity of k variable). No other progress was made on the problem of testing isomorphism on boolean functions until recently, when Blais and O'Donnell [BO10] showed that for every function f that "strongly" depends on k variables testing isomorphism to f requires $\log k$ non-adaptive queries.

In [CGSM11c] and [ABC$^+$] we improve both the upper and lower bounds for testing function isomorphism to a k junta. Our bounds are tight up to a logarithmic factor. These recent results have opened up the flood gates in this area and a lot of work is being done currently. Using our results in [CGSM11c] we also obtained improved lower bounds on testing various function properties [CGSM11b].

The big problem in this area is characterize all function properties that can be tested using "small" number of queries. While such characterization has been obtained in case of testing of graph properties [AFNS09], the problem in the case of general function properties is much more challenging. A crucial component in characterizing testable graph properties was the use of Szemeredi's regularity lemma for graphs. In case of general functions no such regularity type theorem is known.

A first step toward the main problem for characterize all testable function properties is to characterize all the functions f such that isomorphism to f can be tested using small number of queries. Our recent works in [CGSM11c] and [ABC$^+$] does give some insight into this problem. In a recent paper [CFGSM12] we gave a possible characterization of functions for which testing isomorphism to can be tested using small number of queries. We proved various important properties of the characterization that seems to suggest that this can be the right characterization.

1.3 Testing of Monotonicity of Functions

Testing monotonicity of functions [DGL$^+$99] is one of the oldest and most studied problems in Property Testing. The problem is defined as follows. Let $\mathcal{D}$ be a partially ordered set (poset) and let $\mathcal{R} \subseteq \mathbb{Z}$. A function $f : \mathcal{D} \to \mathcal{R}$ is monotone if for every (comparable) pair $x, y \in \mathcal{D}$, $x \leq y$ implies $f(x) \leq f(y)$. A function f is ϵ-far from monotone if it has to be changed on at least an ϵ-fraction of the domain $\mathcal{D}$ to become monotone. A (q, ϵ)-monotonicity tester for domain $\mathcal{D}$ and range $\mathcal{R}$ is a probabilistic algorithm that, given oracle access to a function $f : \mathcal{D} \to \mathcal{R}$, satisfies the following: (a) it makes at most q queries to f; (b) it accepts with probability at least $2/3$ if f is monotone; (c) it rejects with probability at least $2/3$ if f is ϵ-far from monotone.

Most of the property tester for monotonicity falls in the category of "edge-testers" where tester tries to find two neighboring points in the poset where the monotonicity fails. In [BCGSM10] Breit, Garcia-Soriano, Matsliah and I proved that this kind of tester has a limitation and gave a lower bound on the query complexity for testing monotonicity with these kind of testers. The main result of this paper is disproving a 25 year old conjecture by Szymanski [Szy89] on multicuts in the hypercube graph.

We also prove almost tight lower bound on the number of queries for one-sided non-adaptive testing of monotonicity over the n-dimensional hypercube, as well as additional bounds for specific classes of functions and testers.

1.4 Testing of Graph Properties in Orientation Model

Orientation model for graph property testing was first introduced in [HLNT07]. In this model we are given an undirected multi-graph and two distinct vertices s and t. We are supposed to get the orientation of each of the edges by querying. Each edge is oriented in exactly one direction. In this model we want to query as less as possible and find out whether there is a path from s to t in the directed graph or we have to change the orientation of a constant fraction of the edges to get a directed path from s to t. This model is very appropriate for testing properties of sparse and directed graphs.

In [CFL$^+$07] we test whether in a given directed graph G there is a directed path from a specified vertex s to a specified vertex t. The problem of st-connectivity is a classical problem is computer science. We solve the property testing version of this problem. We show that testing st-connectivity in this model can be done using only a constant fraction of queries.

In [CKP] we have shown that testing whether a random walk on a regular graph has a uniform stationary distribution in the orientation can be reduced to the problem of testing whether the graph is Eulerian. This indicated that in this orientation model a global property such as "the stationary distribution being uniform" can be tested by testing a very local property.

References

[ABC$^+$] Noga Alon, Eric Blais, Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Nearly tight bounds for testing function isomorphism.

[AFNS09] Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: It's all about regularity. *SIAM J. Comput.*, 39(1):143–167, 2009.

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and hardness of approximation problems. *J. ACM*, 45:501–555, 1998.

[AS] N. Alon and A. Shapira. Testing subgraphs in directed graphs. pages 700–709.

[AS98] Sanjeev Arora and S. Safra. Probabilistic checking of proofs: a new characterization of np. *J. ACM*, 45:70–122, 1998.

[BC] Rishiraj Bhattacharyya and Sourav Chakraborty. Constant query locally decodable codes against a computationally bounded adversary. *http://www.cmi.ac.in/∼sourav/webpage/Publications.html.*

[BC10] Laszlo Babai and Sourav Chakraborty. Property testing of equivalence under a permutation group action. *To appear in the ACM Transactions on Computation Theory (ToCT)*, 2010.

[BCGSM10] Jop Briët, Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Monotonicity testing and shortest-path routing on the cube. In *APPROX-RANDOM*, pages 462–475, 2010.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, STOC '90, pages 73–83, New York, NY, USA, 1990. ACM.

[BO10] Eric Blais and Ryan O'Donnell. Lower bounds for testing function isomorphism. In *IEEE Conference on Computational Complexity*, pages 235–246, 2010.

[CD09] Sourav Chakraborty and Nikhil R. Devanur. An online multi-unit auction with improved competitive ratio. In *WINE*, pages 79–88, 2009.

[CDK10] Sourav Chakraborty, Nikhil R. Devanur, and Chinmay Karande. Market equilibrium with transaction costs. In *WINE*, pages 496–504, 2010.

[CFGSM12] Sourav Chakraborty, Eldar Fischer, David García-Soriano, and Arie Matsliah. Junto-symmetric functions, hypergraph isomorphism, and crunching. In *CCC*, 2012.

[CFL$^+$07] Sourav Chakraborty, Eldar Fischer, Oded Lachish, Arie Matsliah, and Ilan Newman. Testing *st*-connectivity. In *APPROX-RANDOM*, pages 380–394, 2007.

[CFLY10] Sourav Chakraborty, Eldar Fischer, Oded Lachish, and Raphael Yuster. Two-phase algorithms for the parametric shortest path problem. In *STACS*, pages 167–178, 2010.

[CFM11] Sourav Chakraborty, Eldar Fischer, and Arie Matsliah. Query complexity lower bounds for reconstruction of codes. In *ICS*, pages 264–274, 2011.

[CFMdW10] Sourav Chakraborty, Eldar Fischer, Arie Matsliah, and Ronald de Wolf. New results on quantum property testing. In *FSTTCS*, pages 145–156, 2010.

[CFMY11] Sourav Chakraborty, Eldar Fischer, Arie Matsliah, and Raphael Yuster. Hardness and algorithms for rainbow connection. *J. Comb. Optim.*, 21(3):330–347, 2011.

[CGSM11a] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Cycle detection, order finding and discrete log with jumps. In *ICS*, pages 284–297, 2011.

[CGSM11b] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Efficient sample extractors for juntas with applications. In *ICALP (1)*, pages 545–556, 2011.

[CGSM11c] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Nearly tight bounds for testing function isomorphism. In *SODA*, pages 1683–1702, 2011.

[Cha11] Sourav Chakraborty. On the sensitivity of cyclically-invariant boolean functions. *Discrete Mathematics & Theoretical Computer Science*, 13(4):51–60, 2011.

[CKP] Sourav Chakraborty, Akshay Kamath, and Rameshwar Pratap. Testing uniformity of stationary distribution.

[CL12] Sourav Chakraborty and Oded Lachish. Improved competitive ratio for the matroid secretary problem. In *SODA*, pages 1702–1712, 2012.

[CRR05] Sourav Chakraborty, Jaikumar Radhakrishnan, and Nandakumar Raghunathan. Bounds for error reduction with few quantum queries. *9th International Workshop on Randomization and Computation*, 2005.

[CRRS06] Sourav Chakraborty, Jaikumar Radhakrishnan, Nandakumar Raghunathan, and Prasant Sasatte. Zero error list-decoding capacity of the $q/(q-1)$ channel. *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2006.

[DGL+99] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *RANDOM-APPROX*, pages 97–108, 1999.

[Fis01] Eldar Fischer. The art of uninformed decisions. *Bulletin of the EATCS*, 75:97, 2001.

[FKR+04] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *Journal of Computer and System Sciences*, 68(4):753 – 787, 2004. Special Issue on FOCS 2002.

[FM08] Eldar Fischer and Arie Matsliah. Testing graph isomorphism. *SIAM J. Comput.*, 38(1):207–225, 2008.

[GGR98] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45:653–750, July 1998.

[HLNT07] S. Halevy, Oded Lachish, Ilan Newman, and D. Tsur. Testing properties of constraint-graphs. *Proceedings of the 22nd IEEE Annual Conference on Computational Complexity (CCC)*, 2007.

[Ron08] Dana Ron. Property testing: A learning theory perspective. *Found. Trends Mach. Learn.*, 1:307–402, March 2008.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25:252–271, February 1996.

[Szy89] Ted H. Szymanski. On the permutation capability of a circuit-switched hypercube. In *ICPP (1)*, pages 103–110, 1989.