Property Testing: A Learning Theory Perspective

Dana Ron

Department of EE – Systems, Tel-Aviv University Ramat Aviv, Israel
danar@eng.tau.ac.il

Property testing [15,9] is the study of the following class of problems.

> *Given the ability to perform local queries concerning a particular object (e.g., a function, or a graph), the problem is to determine whether the object has a predetermined global property (e.g., linearity or bipartiteness), or differs significantly from any object that has the property. In the latter case we say it is* far *from (having) the property. The algorithm is allowed a probability of failure, and typically it inspects only a small part of the whole object.*

Property testing problems are usually viewed as relaxations of decision problems. Namely, instead of requiring that the algorithm decide whether the object has the property or does not have the property, the algorithm is required to decide whether the object has the property or is far from having the property. As such, we are interested in testing algorithms that are much more efficient than the corresponding decision algorithms, and in particular have complexity that is sublinear in the size of the object.

Another view of property testing is as a relaxation of learning (with queries and under the uniform distribution)[1]. Namely, instead of asking that the algorithm output a good approximation of the function (object) from within a particular family of functions F, we only require that it decide whether the function belongs to F or is far from any function in F. Given this view, a natural motivation for property testing is to serve as a preliminary step before learning (and in particular, agnostic learning (e.g., [12]): We can first run the testing algorithm to decide whether to use a particular family of functions as our hypothesis class. Here too we are interested in testing algorithms that are more efficient than the corresponding learning algorithms. As observed in [9], property testing is no harder than *proper* learning. Namely, if we have a proper learning algorithm for a family of functions F then we can use it as a subroutine to test the property: "does the function belong to F".

The choice of which of the aforementioned views to take is typically determined by the type of objects and properties in question. Much of property testing

[1] Testing under non-uniform distributions (e.g., [10,1]) and testing with random examples (e.g., [11]) have been considered, but most of the work in property testing deals with testing under the uniform distributions and with queries.

N. Bshouty and C. Gentile (Eds.): COLT 2007, LNAI 4539, pp. 1–2, 2007.

deals with combinatorial objects and in particular graphs (e.g., [9,3]). For such objects it is usually more natural to view property testing as a relaxation of exact decision. Indeed, there are many combinatorial properties for which there are testing algorithms that are much more efficient than the corresponding decision problems. On the other hand, when the objects are functions, then it is usually more natural to look at property testing from a learning theory perspective. In some cases, both viewpoints are appropriate.

This talk will focus on several results that hopefully will be of interest from a learning theory perspective. These include: linearity testing [4] and low-degree testing (e.g., [15]), testing basic Boolean formula [13,7], testing monotonicity (e.g., [8,5]), testing of clustering (e.g., [2]), and distribution-free testing (e.g., [10,1]).

For surveys on property testing see [6,14], and for an online bibliography see: `www.cs.princeton.edu/courses/archive/spring04/cos5987/bib.html`.

References

1. Ailon, N., Chazelle, B.: Information theory in property testing and monotonicity testing in higher dimensions. Information and Computation 204, 1704–1717 (2006)
2. Alon, N., Dar, S., Parnas, M., Ron, D.: Testing of clustering. SIAM Journal on Discrete Math. 16(3), 393–417 (2003)
3. Alon, N., Fischer, E., Newman, I., Shapira, A.: A combinatorial characterization of the testable graph properties: It's all about regularity. In: Proceedings of the Thirty-Eighth Annual ACM Symposium on the Theory of Computing (2006)
4. Blum, M., Luby, M., Rubinfeld, R.: Self-testing/correcting with applications to numerical problems. Journal of the ACM 47, 549–595 (1993)
5. Ergun, F., Kannan, S., Kumar, S.R., Rubinfeld, R., Viswanathan, M.: Spot-checkers. Journal of Computer and System Sciences 60(3), 717–751 (2000)
6. Fischer, E.: The art of uninformed decisions: A primer to property testing. Bulletin of the European Association for Theoretical Computer Science 75, 97–126 (2001)
7. Fischer, E., Kindler, G., Ron, D., Safra, S., Samorodnitsky, S.: Testing juntas. Journal of Computer and System Sciences 68(4), 753–787 (2004)
8. Goldreich, O., Goldwasser, S., Lehman, E., Ron, D., Samordinsky, A.: Testing monotonicity. Combinatorica 20(3), 301–337 (2000)
9. Goldreich, O., Goldwasser, S., Ron, D.: Property testing and its connection to learning and approximation. Journal of the ACM 45(4), 653–750 (1998)
10. Halevy, S., Kushilevitz, E.: Distribution-free property testing. In: Arora, S., Jansen, K., Rolim, J.D.P., Sahai, A. (eds.) RANDOM 2003 and APPROX 2003. LNCS, vol. 2764, pp. 341–353. Springer, Heidelberg (2003)
11. Kearns, M., Ron, D.: Testing problems with sub-learning sample complexity. Journal of Computer and System Sciences 61(3), 428–456 (2000)
12. Kearns, M.J., Schapire, R.E., Sellie, L.M.: Toward efficient agnostic learning. Machine Learning 17(2-3), 115–141 (1994)
13. Parnas, M., Ron, D., Samorodnitsky, A.: Testing boolean formulae. SIAM Journal on Discrete Math. 16(1), 20–46 (2002)
14. Ron, D.: Property testing. In: Rajasekaran, S., Pardalos, P. M., Reif, J. H., Rolim, J. D. P.(eds.) Handbook on Randomization, Volume II, pp. 597–649 (2001)
15. Rubinfeld, R., Sudan, M.: Robust characterization of polynomials with applications to program testing. SIAM Journal on Computing 25(2), 252–271 (1996)