DOI:10.1145/3394623

To view the accompanying paper, visit doi.acm.org/10.1145/3394625

Technical Perspective
Algorithm Selection as a Learning Problem

By Avrim Blum

THE FOLLOWING PAPER by Gupta and Roughgarden—"Data-Driven Algorithm Design"—addresses the issue that the best algorithm to use for many problems depends on what the input "looks like." Certain algorithms work better for certain types of inputs, whereas other algorithms work better for others. This is especially the case for NP-hard problems, where we do not expect to ever have algorithms that work well on all inputs: instead, we often have various heuristics that each work better in different settings. Moreover, heuristic strategies often have parameters or hyperparameters that must be set in some way.

Traditional theoretical analysis of such problems would select among algorithms or parameter choices by looking at their worst-case performance, such as via their approximation ratio. Another traditional alternative would be to consider average-case analysis, looking at performance on some clean (typically uniform) distribution over inputs. However, the inputs arising in a given domain are typically neither worst-case nor average-case from a clean distribution: instead, they often have structure and can be thought of as drawn from some complex and difficult-to-characterize probability distribution over inputs associated with the given domain. This fact suggests treating algorithm selection as a learning problem—using previously seen examples of inputs from a given domain to tune parameters of a heuristic or to select among various choices of algorithm. Indeed, this approach has been used in practice for many years, for example, Hutter[5] and Smith-Miles.[6] What the authors present is a theoretical formulation and analysis of this process, using the well-developed framework of PAC-learning from learning theory to analyze fundamental questions. For instance, how many examples of previous inputs does one need to see to have confidence that optimizing parameters over these examples will produce a rule that is near optimal on average for future inputs drawn from the same (complex and difficult-to-characterize) probability distribution? The paper then provides answers to such questions by analyzing formal measures of complexity (in particular, the pseudo-dimension) of various natural algorithm families such as greedy algorithms for knapsack and other object assignment problems, resulting in bounds on the number of previous inputs that must be seen for good generalization.

The authors present a theoretical formulation and analysis of algorithm selection using the well-developed framework of PAC-learning to analyze fundamental learning questions.

Subsequent to the initial publication of this paper there have been a number of exciting results building on the framework developed here, showing how principled algorithm selection and automated algorithm design more generally can be applied to a wide variety of other important problems. For instance, Balcan et al.[3,4] analyze a large number of classic clustering problems, giving algorithms and sample complexity guarantees for learning near-optimal parameters and hyperparameters for many popular clustering methods. Balcan, Dick, Sandholm, and Vitercik[1] show how learning-theoretic methods can be used to optimize branching in integer linear programming solvers. Most recently, Balcan, Dick, and Vitercik[2] show how principled data-driven algorithm design can be extended to the online setting in which there is no distribution over problem instances at all, just an arbitrary stream of examples, and one wishes to achieve low regret with respect to the best parameter setting in hindsight. While the highly discontinuous nature of the cost functions make this impossible in the worst case due to well-known lower bounds, this work identifies a new notion called dispersion that enables positive results. They also show how these same ideas can address problems of preserving privacy, which can be a concern if algorithm parameters used on one input are set based on possibly sensitive information from previously seen inputs.

References

1. Balcan, M.-F., Dick, T., Sandholm, T. and Vitercik, E. Learning to branch. In *Proceedings of the 35th Intern. Conf. Machine Learning*, (Stockholm, Sweden, July 10–15, 2018), 353–362.
2. Balcan, M.-F., Dick, T., and Vitercik, E. Dispersion for data-driven algorithm design, online learning, and private optimization. In *Proceedings of the 59th IEEE Annual Symp. Foundations of Computer Science*, (Paris, France, Oct. 7–9, 2018), 603–614.
3. Balcan, M.-F., Dick, T. and White, C. Data-driven clustering via parameterized Lloyd's families. In *Proceedings of the Annual Conf. on Neural Information Processing Systems* (Montreal, Canada, Dec. 3–9, 2018), 10664–10674.
4. Balcan, M.-F., Nagarajan, V., Vitercik, E. and White, C. Learning theoretic foundations of algorithm configuration for combinatorial partitioning problems. In *Proceedings of the 30th Conf. Learning Theory* (Amsterdam, The Netherlands, July 7–10, 2017), 213–274.
5. Hutter, F., Hoos, H.H., Leyton-Brown, K. and Stützle, T. ParamILS: An automatic algorithm configuration framework. *J. Artificial Intelligence Research 36*, (2009), 267–306.
6. Smith-Miles, K.A. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys 41*, 1 (2009).

Avrim Blum is a professor and Chief Academic Officer at Toyota Technological Institute at Chicago, IL, USA.