Accepted Manuscript

Circuit Lower Bounds from Learning-theoretic Approaches

Akinori Kawachi

PII: S0304-3975(18)30276-7
DOI: https://doi.org/10.1016/j.tcs.2018.04.038
Reference: TCS 11568

To appear in: *Theoretical Computer Science*

Circuit Lower Bounds from Learning-theoretic Approaches

Akinori Kawachi[a]

[a]Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan

Abstract

An important open problem in computational complexity theory is to prove the size of circuits, namely, Boolean circuit lower bounds, necessary to solve explicit problems. We survey learning-theoretic approaches to proofs of Boolean circuit lower bounds in this paper. In particular, we discuss how to prove circuit lower bounds in uniform classes by assuming (or constructing) circuit-learning algorithms in several settings, such as the exact, probably approximately correct (PAC), and statistical query learning models.

1. Introduction

A deep understanding of the power of a non-uniform computation model such as a Boolean circuit, which we refer to simply as a "circuit" below, is one of the central goals in the computational complexity theory.

The circuit as a computational model has some similarities to the standard computational model, the Turing machine. It is known that circuits can efficiently simulate Turing machines, and thus, polynomial-size circuits can simulate polynomial-time Turing machines. Conversely, Turing machines can efficiently simulate circuits if they are given descriptions of the circuits as supplementary information, and thus, a polynomial-time Turing machine can perform the computation of a polynomial-size circuit if the description of the circuit is given. (For simplicity, we sometimes refer to a circuit as a family of circuits, which contains only one circuit for each input length $n \in \mathbb{N}$, in this paper. Precisely, the above simulation of Turing machines is done by families of circuits.) However, circuits are known to possess some weird computational power in some cases. While there exist problems, such as the halting problem, that no (even time-unbounded) Turing machine can compute, every problem can be solved by an exponential-size circuit. (Precisely, each problem can be solved by a family of exponential-size circuits.)

In addition, we can easily prove that there exists a problem that can be solved in exponential time that no polynomial-time Turing machine can solve, i.e., $\mathrm{EXP} \not\subseteq \mathrm{P}$ by the standard diagonalization argument, but it is a big open problem in circuit complexity whether EXP has a hard problem against polynomial-size circuits or not, i.e., whether $\mathrm{EXP} \not\subseteq \mathrm{SIZE}(\mathrm{poly})$ or not.

However, it is difficult to imagine how the computational power of Turing machines can be enhanced by supplementary information such as circuit descriptions in order to efficiently compute hard problems in EXP or even NP-complete problems. It would be natural to conjecture that no polynomial-size circuit can compute NP-complete problems. This conjecture is also connected to the major open problem $\mathrm{NP} \neq \mathrm{P}$ in theoretical computer science, since we can prove $\mathrm{NP} \neq \mathrm{P}$ if some NP-complete problem has superpolynomial circuit lower bounds; in other words, the size of circuits necessary to compute the problem is superpolynomial.

Email address: `kawachi@is.titech.ac.jp` (Akinori Kawachi)