40-year-old algorithm proven the best possible

June 11 2015, by Larry Hardesty

Credit: AI-generated image (disclaimer)

Comparing the genomes of different species—or different members of the same species—is the basis of a great deal of modern biology. DNA sequences that are conserved across species are likely to be functionally important, while variations between members of the same species can indicate different susceptibilities to disease.

The basic algorithm for determining how much two sequences of symbols have in common—the "edit distance" between them—is now more than 40 years old. And for more than 40 years, computer science researchers have been trying to improve upon it, without much success.

At the ACM Symposium on Theory of Computing (STOC) next week, MIT researchers will report that, in all likelihood, that's because the algorithm is as good as it gets. If a widely held assumption about computational complexity is correct, then the problem of measuring the difference between two genomes—or texts, or speech samples, or anything else that can be represented as a string of symbols—can't be solved more efficiently.

In a sense, that's disappointing, since a computer running the existing algorithm would take 1,000 years to exhaustively compare two human genomes. But it also means that computer scientists can stop agonizing about whether they can do better.

"This edit distance is something that I've been trying to get better algorithms for since I was a graduate student, in the mid-'90s," says Piotr Indyk, a professor of computer science and engineering at MIT and a co-author of the STOC paper. "I certainly spent lots of late nights on that—without any progress whatsoever. So at least now there's a feeling of closure. The problem can be put to sleep."

Moreover, Indyk says, even though the paper hasn't officially been presented yet, it's already spawned two follow-up papers, which apply its approach to related problems. "There is a technical aspect of this paper, a certain gadget construction, that turns out to be very useful for other purposes as well," Indyk says.

Squaring off

Edit distance is the minimum number of edits—deletions, insertions, and substitutions—required to turn one string into another. The standard algorithm for determining edit distance, known as the Wagner-Fischer algorithm, assigns each symbol of one string to a column in a giant grid and each symbol of the other string to a row. Then, starting in the upper left-hand corner and flooding diagonally across the grid, it fills in each square with the number of edits required to turn the string ending with the corresponding column into the string ending with the corresponding row.

Computer scientists measure algorithmic efficiency as computation time relative to the number of elements the algorithm manipulates. Since the Wagner-Fischer algorithm has to fill in every square of its grid, its running time is proportional to the product of the lengths of the two strings it's considering. Double the lengths of the strings, and the running time quadruples. In computer parlance, the algorithm runs in quadratic time.

That may not sound terribly efficient, but quadratic time is much better than exponential time, which means that running time is proportional to N^2, where N is the number of elements the algorithm manipulates. If on some machine a quadratic-time algorithm took, say, a hundredth of a second to process 100 elements, an exponential-time algorithm would take about 100 quintillion years.

Theoretical computer science is particularly concerned with a class of problems known as NP-complete. Most researchers believe that NP-complete problems take exponential time to solve, but no one's been able to prove it. In their STOC paper, Indyk and his student Artūrs Bačkurs demonstrate that if it's possible to solve the edit-distance problem in less-than-quadratic time, then it's possible to solve an NP-complete problem in less-than-exponential time. Most researchers in the computational-complexity community will take that as strong evidence that no subquadratic solution to the edit-distance problem exists.

Can't get no satisfaction

The core NP-complete problem is known as the "satisfiability problem": Given a host of logical constraints, is it possible to satisfy them all? For instance, say you're throwing a dinner party, and you're trying to decide whom to invite. You may face a number of constraints: Either Alice or Bob will have to stay home with the kids, so they can't both come; if you invite Cindy and Dave, you'll have to invite the rest of the book club, or they'll know they were excluded; Ellen will bring either her husband, Fred, or her lover, George, but not both; and so on. Is there an invitation list that meets all those constraints?

In Indyk and Bačkurs' proof, they propose that, faced with a satisfiability problem, you split the variables into two groups of roughly equivalent size: Alice, Bob, and Cindy go into one, but Walt, Yvonne, and Zack go into the other. Then, for each group, you solve for all the pertinent constraints. This could be a massively complex calculation, but not nearly as complex as solving for the group as a whole. If, for instance, Alice has a restraining order out on Zack, it doesn't matter, because they fall in separate subgroups: It's a constraint that doesn't have to be met.

At this point, the problem of reconciling the solutions for the two subgroups—factoring in constraints like Alice's restraining order—becomes a version of the edit-distance problem. And if it were possible to solve the edit-distance problem in subquadratic time, it would be possible to solve the satisfiability problem in subexponential time.

"This is really nice work," says Barna Saha, an assistant professor of computer science at the University of Massachusetts atAmherst. "There are lots of people who have been working on this problem, because it has a big practical impact. But they won't keep trying to develop a subquadratic algorithm, because that seems very unlikely to happen, given the result of this paper."

As for the conjecture that the MIT researchers' proof depends on—that NP-complete problems can't be solved in subexponential time—"It's a very widely believed conjecture," Saha says. "And there are many other results in this low-polynomial-time complexity domain that rely on this conjecture.

More information: "Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false)" arxiv.org/abs/1412.0348

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

Citation: 40-year-old algorithm proven the best possible (2015, June 11) retrieved 26 June 2024 from https://phys.org/news/2015-06-year-old-algorithm-proven.html