Abstract

The *Ulam distance* between two permutations of length n is the minimum number of insertions and deletions needed to transform one sequence into the other. Equivalently, the Ulam distance d is n minus the length of the longest common subsequence (LCS) between the permutations. Our main result is an algorithm, that for any fixed $\varepsilon > 0$, provides a $(1+\varepsilon)$-multiplicative approximation for d in $\tilde{O}_\varepsilon(n/d + \sqrt{n})$ time, which has been shown to be optimal up to polylogarithmic factors. This is the first sublinear time algorithm (provided that $d = (\log n)^{\omega(1)}$) that obtains arbitrarily good multiplicative approximations to the Ulam distance. The previous best bound is an $O(1)$-approximation (with a large constant) by (A. Andoni and H. L. Nguyen, *Near-Optimal Sublinear Time Algorithms for Ulam Distance*, Proceedings of the 21st Symposium on Discrete Algorithms (SODA), 76–86, 2010) with the same running time bound (ignoring polylogarithmic factors).

The improvement in the approximation factor from $O(1)$ to $(1 + \varepsilon)$ allows for significantly more powerful sublinear algorithms. For example, for any fixed $\delta > 0$, we can get additive δn approximations for the LCS between permutations in $\tilde{O}_\delta(\sqrt{n})$ time. Previous sublinear algorithms require δ to be at least $1 - 1/C$, where C is the approximation factor, which is close to 1 when C is large. Our algorithm is obtained by abstracting the basic algorithmic framework of (A. Andoni and H.L. Nguyen, ibid.) and combining it with the sublinear approximations for the longest increasing subsequence by (M. Saks and C. Seshadhri, *Estimating the longest increasing sequence in polylogarithmic time*, Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), 458–467, 2010).