Abstract:

The edit distance, also known as Levenshtein distance, is the minimum number of character insertions, deletions, and substitutions needed to transform one string into another. The textbook dynamic programming algorithm computes the edit distance of two length-n strings in \(O(n^2)\) time, which is optimal up to sub-polynomial factors and conditioned on the Strong Exponential Time Hypothesis (SETH). In a bounded setting, where the running time is parameterized by the edit distance \(k\), the celebrated algorithm by Landau and Vishkin (JCSS'88) achieves a running time of \(O(n+k^2)\), which is optimal as a function of \(n\) and \(k\) (again, up to sub-polynomial factors and conditioned on SETH). While the theory community thoroughly studied the Levenshtein distance, most practical applications rely on a more general weighted edit distance, where each edit has a weight depending on its type and the involved characters from the alphabet \(\Sigma\). This is formalized through a weight function \(w: \Sigma \cup \{\epsilon\} \times \Sigma \cup \{\epsilon\} \rightarrow \mathbb{R}\) normalized so that \(w(a,a) = 0\) for \(a \in \Sigma \cup \{\epsilon\}\) and \(w(a,b) \geq 1\) for \(a,b \in \Sigma \cup \{\epsilon\}\) with \(a \neq b\). The classic \(O(n^2)\)-time algorithm supports this setting seamlessly, but for many decades only a straightforward \(O(nk)\)-time solution was known for the bounded version of the weighted edit distance problem. In this talk, I will present an \(O(n^2 k)\)-time algorithm (joint work with Das, Gilbert, Hajiaghayi, and Saha; STOC'23; arXiv:2302.04229) and a very recent \(\tilde{O}(n+\sqrt{nk^3})\)-time algorithm (joint work with Cassis and Wellnitz). I will also sketch a lower bound that proves the optimality of the latter algorithm for \(\sqrt{n} \leq k \leq n\) (up to sub-polynomial factors and conditioned on the All-Pairs Shortest Paths Hypothesis).