A Novel Approach to the Longest Common Subsequence Problem with Distance Constraints

Vesta Torlow Ballah

March 3, 2025

Abstract

This paper presents a novel approach to the Longest Common Subsequence (LCS) problem by introducing a distance constraint. Traditional LCS algorithms consider all global alignments, often leading to misplaced matches. Our method enforces a strict positional constraint based on the absolute difference in sequence lengths, ensuring that matches remain within a valid alignment window. We formally prove this constraint and present an efficient algorithm that dynamically updates matches without fully populating the dynamic programming (DP) table. Preliminary results suggest that this method maintains correctness while potentially reducing computational overhead.

1 Introduction

The Longest Common Subsequence (LCS) problem is fundamental in computer science and bioinformatics. Given two sequences A and B of lengths m and n, the objective is to find the longest subsequence that appears in both A and B while preserving the original order. Traditional solutions employ dynamic programming (DP) with a time complexity of $O(mn)$. However, these methods consider global alignments without constraints, which can lead to suboptimal matching in practical applications.

1.1 Distance Constraint

Our method introduces a **distance constraint** to regulate valid matches, preventing excessive misalignment. This constraint is defined as:

$$|i - j| \leq |m - n| \tag{1}$$

where (i, j) represents a matched pair of indices in sequences A and B. This ensures that matches remain within a bounded alignment range, preventing large gaps.

Special Cases:

- **Perfect Alignment Case:** If $|m - n| = 0$, then all valid matches occur at identical indices $(i = j)$, ensuring a one-to-one correspondence.

- **Near-Perfect Alignment Case:** If $|m - n| = 1$, then matches may shift by at most one index while maintaining a strong alignment structure.

- **General Case:** The maximum allowable shift is dictated by $|m - n|$, ensuring that even for sequences of differing lengths, valid matches remain constrained within a reasonable range.

2 Problem Definition

Given two sequences:

$$A = \{a_1, a_2, ..., a_m\}$$
$$B = \{b_1, b_2, ..., b_n\}$$

we define a **valid match** as a pair (i, j) satisfying:

$$A[i] = B[j] \quad \text{and} \quad |i - j| \leq |m - n|. \tag{2}$$

The objective is to compute the longest subsequence satisfying these constraints.

3 Proof of Distance Constraint

We prove that any optimal LCS solution cannot contain matches where $|i - j| > |m - n|$.

If (i, j) is a valid match in the LCS of A and B, then:

$$|i - j| \leq |m - n|. \tag{3}$$

Proof. We prove this by contradiction.

Assume there exists an optimal LCS solution containing a pair (i, j) such that:

$$|i - j| > |m - n|. \tag{4}$$

This implies that $A[i]$ and $B[j]$ are matched despite being excessively misaligned. Since LCS preserves order, such a match forces additional gaps, distorting future alignments. However, we can always find an alternative LCS solution where all matches satisfy $|i - j| \leq |m - n|$ without reducing the sequence length. This contradicts the assumption of optimality, proving that no valid LCS contains matches violating this bound. $\square$

4 Algorithm Description

We propose an efficient approach that dynamically tracks and updates optimal matches, avoiding full DP table computation.

Algorithm 1 Optimized LCS with Distance Constraint

 1: **procedure** LCSEQ(A, B)
 2:     Initialize an empty list L for valid matches
 3:     Initialize $best\_match\_for\_B$ as an empty dictionary
 4:     **for** $i = 1$ to m **do**
 5:         **for** $j = 1$ to n **do**
 6:             **if** $A[i] = B[j]$ and $|i - j| \leq |m - n|$ **then**
 7:                 Update match tracking dynamically
 8:             **end if**
 9:         **end for**
10:     **end for**
11:     **return** L, $|L|$
12: **end procedure**

---

5   Conclusion and Future Work

This paper introduced a novel LCS approach incorporating a distance constraint to regulate match alignment. Our theoretical proof ensures correctness, and preliminary findings suggest computational benefits. Future research includes empirical validation, extending this method to approximate LCS variations, and applying it to bioinformatics and text similarity analysis.

References