Approximate Pattern Matching in Ordered Sequences

Julien David and Thierry Lecroq
Contact:julien.david@unicaen.fr

12 avril 2024

1 Abstract

We are interested in pattern matching in time series (for instance : electrocardiograms, stock market fluctuations, music scores, sysmogram, ...) and in general in sequences of numerical values. In particular, we are interested in the notion of Cartesian Trees, that allows to establish the similarity between two series/curves if their relatives ups and downs occur at the same moments, independently of their values. Efficient algorithmic solutions have already been proposed to solve the exact pattern matching problem based on Cartesian Trees, that is a version in which the definition of Cartesian Trees is strictly respected. The main goal of this thesis is to produce notions of approximate pattern matching, where even more differences are tolerated between the two sequences, still regarding them as similar. Indeed, depending on the context, one might consider that

— data might contain input errors, whether it is in the values or the order in which they appear,
— data might contain some noise,
— the notion of pattern given by Cartesian Trees might be too strict.

It might therefore be useful to relax some constraints, in order to automatically detect new similarities between two sequences One we will have proposed those notions, we will also provide efficient algorithmic solutions.

2 Context and objectives

Cartesian Trees were introduces by Vuillemin [1] in the early 80's. Since then, they have been shown to be related to Lyndon Words [2], Range Minimum Queries [3], or the parallel construction of suffix trees [4]. Recently, Park et al. [5] have introduced a new type of generalized pattern on sequences, called *Cartesian Tree Matching (CTM)*.

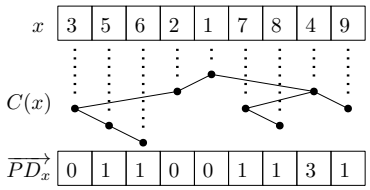

FIGURE 1 – A sequence $x = (3, 5, 6, 2, 1, 7, 8, 4, 9)$, its Cartesian Tree $C(x)$ and its parent-distance table

Let x be a sequence of length n, its Cartesian Tree $C(x)$ is recursively defined as follows (see example Figure 2) :

— if x is empty, then $C(x)$ is the empty tree ;

— if $x[1 \dots n]$ is not empty and $x[i]$ is the smallest value of x, $C(x)$ is a Cartesian Tree enrooted in i, the Cartesian Tree of $x[1 \dots i-1]$ is its left subtree and the Cartesian Tree of $x[i+1 \dots n]$ is its right subtree.

It is possible to reconstruct a permutation from a Cartesian Tree by doing a simple in-order depth first traversal of the tree.

We note $x \approx_{CT} y$ if two sequences x and y share the same Cartesian Tree. For instance, $x = (3, 5, 6, 2, 1, 7, 8, 4, 9) \approx_{CT} (3, 4, 8, 2, 1, 7, 9, 5, 6)$. The *Cartesian tree matching (CTM)* problem consist in finding all the factors of a text that share the same Cartesian Tree as a given pattern. Formally, Park *et al.* [6] define it as :

Definition 1 (Cartesian tree matching)**.** *Let $p[1 \dots m]$ and $t[1 \dots n]$ be two sequences. Find all positions $1 \leq i \leq n - m + 1$ such that $t[i \dots i + m - 1] \approx_{CT} p[1 \dots m]$.*

In order to solve the problem without building every possible Cartesian Tree, an efficient representation of those trees has been introduced by Park *et al.* [6], called the *parent-distance* table (see example Figure 2) :

Definition 2 (Parent-distance Table)**.** *Let $x[1 \dots n]$ be a sequence, the* parent-distance *table of x is a sequence of integers $\overrightarrow{PD}_x[1 \dots n]$, defined as follows :*

$$\overrightarrow{PD}_x[i] = \begin{cases} i - max_{1 \leq j < i}\{j \mid x[j] < x[i]\} & \text{if } j \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

This table allows them to obtain equivalents of famous linear algorithms such as Morris-Pratt (for one pattern) and Aho-Corasick (multiple patterns) for the *CTM* problem. Solutions that are more efficient in practice have been given in [7, 8]. Also, new results on the CTM problem have been published [5, 9, 4] during the past years.

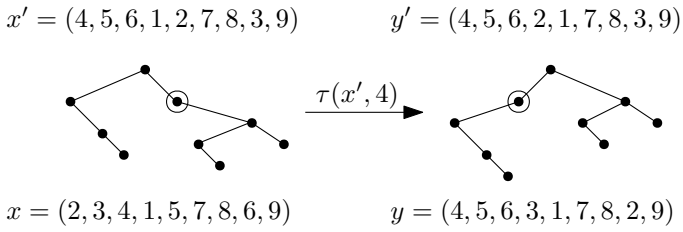

FIGURE 2 – For the sequences x and y we have $x \overset{\tau}{\approx}_{CT} y$. A swap at position 4 moves the circled node of the root's right subtree to its left subtree. In general, a swap at position i consist in either moving the leftmost descendant of the right subtree to **a** rightmost position on the left subtree (that is when $x[i] < x[i+1]$), or to move the rightmost descendant of the left subtree to **a** leftmost position on the right subtree. Let's note that we also have $x \overset{\tau}{\approx}_{CT} y'$, $x' \overset{\tau}{\approx}_{CT} y$ and $x' \overset{\tau}{\approx}_{CT} y'$.

3 Recent results and objectives : approximate pattern matching

The objective of this PHD is to define approximated versions of the CTM problem and to propose algorithmic solutions for those new versions. Some work has already been done in that direction, using the notion of swap on sequences.

Definition 3 (Swap)**.** *Let x and y be two sequences of length n, and $i \in \{1, \ldots, n-1\}$. We note $y = \tau(x, i)$ to describe a* swap*, that is :*

$$y = \tau(x, i) \text{ if } \begin{cases} x[j] = y[j], \forall j \notin \{i, i+1\} \\ x[i] = y[i+1] \\ x[i+1] = y[i] \end{cases}$$

This kind of transposition is, for instance, the one used in the Bubble Sort. It is therefore a "natural" operation on permutation and sequences. Figure 2 shows the effect of a transposition on the Cartesian Tree. We use the swap notion to define an approximate version of the CTM problem.

Let x and y be two sequences of length n. We have $x \overset{\tau}{\approx}_{CT} y$ if :

$$\begin{cases} x \approx_{CT} y, \text{ or} \\ \exists\ x',\ y', \exists\ i \in \{1, \ldots, n-1\}, x' \approx_{CT} x, y' \approx_{CT} y, x' = \tau(y', i) \text{ and } y' = \tau(x', i) \end{cases}$$

Definition 4 (CT_τ Matching)**.** *Let x and y be two sequences of length n and m. Find all positions $1 \leq i \leq n - m + 1$ such that $x[i \ldots i + m - 1] \overset{\tau}{\approx}_{CT} y[1 \ldots m]$.*

In [10], we have studied the effect of a swap on the parent-distance table and obtained two algorithms that can solve the CT_τ Matching problem.

4 Detailed Project

The objective of the PHD will be to continue and extend the word that has been done on approximated pattern matching for Cartesian Trees.

In a first period, the candidate will be asked the following questions :

— what is the average complexity of the algorithms in [10] ? We in fact conjecture that one of the two algorithms is actually more efficient in practice than what the worst-case complexity indicates.
— is there an efficient algorithm if several swaps are allowed ?

We will then focus on the following question : does there exists other interesting notions of approximate pattern ? Can we propose a satisfying algorithmic solutions for those ? The Hamming distance and Levenshtein distance might be considered.

5 Location

The PHD student will work at the university of Caen, in Normandy, France. A co-supervisor, Thierry Lecroq, works at the university of Rouen, also in Normandy.

French speaking student might also teach at the university if they wish to.

Références

[1] J. Vuillemin, "A unifying look at data structures," *Commun. ACM*, vol. 23, p. 229–239, apr 1980.

[2] M. Crochemore and L. M. Russo, "Cartesian and Lyndon trees," *Theoretical Computer Science*, vol. 806, pp. 1–9, 2020.

[3] E. Demaine, G. Landau, and O. Weimann, "On cartesian trees and range minimum queries," vol. 68, pp. 341–353, 01 2009.

[4] J. Shun and G. E. Blelloch, "A simple parallel cartesian tree algorithm and its application to parallel suffix tree construction," *ACM Trans. Parallel Comput.*, vol. 1, oct 2014.

[5] S. Park, A. Amir, G. Landau, and K. Park, "Cartesian tree matching and indexing," in *30th Annual Symposium on Combinatorial Pattern Matching, CPM 2019* (N. Pisanti and S. Pissis, eds.), Leibniz International Proceedings in Informatics, LIPIcs, (Germany), pp. 16 :1–16 :14, Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, June 2019.

[6] S. Park, A. Amir, G. Landau, and K. Park, "Cartesian tree matching and indexing," in *CPM*, vol. 16, (Pisa, Italy), pp. 1–14, 2019.

[7] S. Song, G. Gu, C. Ryu, S. Faro, T. Lecroq, and K. Park, "Fast cartesian tree matching," (Segovia, Spain), pp. 124–137, 2019.

[8] S. Song, G. Gu, C. Ryu, S. Faro, T. Lecroq, and K. Park, "Fast algorithms for single and multiple pattern cartesian tree matching," *Theor. Comput. Sci.*, vol. 849, pp. 47–63, 2021.

[9] S. G. Park, M. Bataa, A. Amir, G. M. Landau, and K. Park, "Finding patterns and periods in cartesian tree matching," *Theoretical Computer Science*, vol. 845, pp. 181–197, 2020.

[10] B. Auvray, J. David, R. Groult, and T. Lecroq, "Approximate cartesian tree matching : An approach using swaps," in *String Processing and Information Retrieval - 30th International Symposium, SPIRE 2023, Pisa, Italy,*

September 26-28, 2023, Proceedings (F. M. Nardini, N. Pisanti, and R. Venturini, eds.), vol. 14240 of *Lecture Notes in Computer Science*, pp. 49–61, Springer, 2023.