On the Complexity of the Discrete Fréchet Distance under L_1 and L_∞

Omer Gold[*] Micha Sharir[*]

Abstract

We study the decision tree complexity of the discrete Fréchet distance (decision version) under the L_1 and L_∞ metrics over $\mathbb{R}^d$. While algorithms for the Euclidean (L_2) discrete Fréchet distance were studied extensively, the problem in other metrics such as L_1 and L_∞ seems to be much less investigated.

For the L_1 discrete Fréchet distance in $\mathbb{R}^d$ we present a $2d$-linear decision tree with depth $O(n \log n)$, for any constant d. For the L_∞ discrete Fréchet distance in $\mathbb{R}^d$ we present a 2-linear decision tree with depth $O(n \log n)$, for any constant d. We hope that these near-linear depth decision trees will motivate the study of the problem in these metrics and, in particular, will lead to the development of improved algorithms.

1 Introduction

The Fréchet distance is a measure of similarity between curves that takes into account the location and ordering of the points along the curves. Therefore it is often better than the well-known Hausdorff distance as a metric for comparing parameterized shapes. This measure was introduced by Fréchet in 1906 [6].

Eiter and Mannila [5] introduced the *discrete Fréchet distance*, a variant also known as the *coupling distance*. They showed that this distance provides a good approximation for the Fréchet distance between curves, and provided a quadratic dynamic programming algorithm to compute it.

Since then many studies have been made about the *discrete* problem in the *Euclidean plane*: Agarwal et al. [1] showed a subquadratic algorithm with $O(n^2 \log\log n / \log n)$ runtime[1], Buchin et al. [3] showed an *algebraic computation tree* lower bound of $\Omega(n \log n)$, and Bringmann [2] recently showed that there is no algorithm with runtime $O(n^{2-\Omega(1)})$ (also known as "truly subquadratic time"), assuming the *Strong Exponential Time Hypothesis.* These bounds hold for computing the exact distance and for the decision version of the problem.

While much work has been made on the Euclidean discrete Fréchet distance, the problem in other metrics, such as L_1 and L_∞ has been much less investigated.

Buchin et al. [4] recently showed that the decision tree complexity of the *Euclidean* discrete Fréchet distance in the *plane* is[2] $\widetilde{O}(n^{4/3})$. This result is obtained by using a range searching technique of Katz and Sharir [9]. We will briefly review this result, and argue that, for the problem under the L_1 and L_∞ metrics in $\mathbb{R}^d$, the standard range searching approach does not seem capable of giving us the results we aim for, which we will establish using a different approach.

From now on, the term L_p discrete Fréchet distance refers to the *decision problem* of determining whether the discrete Fréchet distance with underlying norm L_p is at most some parameter $\varepsilon \geq 0$.

The contribution of this paper is given in the following theorems:

Theorem 1 *Given two polygonal curves P, Q in $\mathbb{R}^d$ with total complexity n (i.e., number of vertices), there is a $2d$-linear decision tree[3] with depth $O(n \log n)$ for the L_1 discrete Fréchet distance between P and Q, for any constant d.*

Theorem 2 *Given two polygonal curves P, Q in $\mathbb{R}^d$ with total complexity n, there is a 2-linear decision tree with depth $O(n \log n)$ for the L_∞ discrete Fréchet distance between P and Q, for any constant d.*

For Theorem 1 and Theorem 2, we generalize an observation originated in Fredman's 1976 work on the decision tree complexity of $(\min, +)$-matrix multiplication [7], a fundamental problem in **P**, known for being computationally equivalent to the APSP (all pairs shortest paths) problem in directed graphs with arbitrary real edge weights.

At the basis of Fredman's technique is the trivial (albeit ingenious) observation that $a + b < c + d$ iff $a - c < d - b$. This observation is often referred to as *Fredman's trick.* Fredman's trick was also liberally used by Grønlund and Pettie in their recent 3SUM breakthrough [8].

2 Fréchet Distance

The Fréchet distance is often illustrated by a man and a dog, each walking along a path (curve). The man

[*]School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel; {omergold, michas}@post.tau.ac.il

[1]For the decision version, they showed a bound of $O(n^2 \log\log n / \log^2 n)$. Both are in the word RAM model.

[2]The notation $\widetilde{O}(\cdot)$ hides poly-logarithmic factors.

[3]A k-linear decision tree is one in which each branching is based on a sign test of a linear expression with at most k terms.

has the dog on a leash. Each of them may choose their own speed and may stop but cannot walk backwards. Then the Fréchet distance is the length of the shortest leash that allows them to walk on their respective curves from beginning to end.

More formally, following [5] we define a curve as a continuous mapping $f : [0,1] \to V$, where (V,ρ) is a metric space. Given two curves $f : [0,1] \to V$ and $g : [0,1] \to V$, their Fréchet distance is defined as

$$\delta_F(f,g) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \rho(f(\alpha(t)), g(\beta(t))),$$

where α and β are arbitrary continuous nondecreasing functions from $[0,1]$ onto $[0,1]$.

When computing the Fréchet distance between arbitrary curves, one typically approximates the curves by polygonal curves. Eiter and Mannila [5] defined the *discrete Fréchet distance* between polygonal curves and showed it gives a good approximation to the Fréchet distance between them.

A polygonal curve with n edges is a curve $P : [0,1] \to V$, such that for each $i \in \{0,1,\ldots,n-1\}$, the restriction of P to the interval $\left[\frac{i}{n}, \frac{i+1}{n}\right]$ is affine. Since the Fréchet distance is invariant under reparametrization, we can assume a polygonal curve P to be given by the ordered list of its vertices, i.e., a sequence $P = (p_0,\ldots,p_n)$.

Let $P = (p_0,\ldots,p_n)$ and $Q = (q_0,\ldots,q_m)$ be two polygonal curves given by their ordered lists of vertices. A *coupling* $C = (c_0,\ldots,c_k)$ between P and Q is an ordered sequence of distinct pairs of vertices in P, Q, such that $c_0 = (p_0,q_0)$, $c_k = (p_n,q_m)$ and $c_r = (p_i,q_j) \Rightarrow c_{r+1} \in \{(p_{i+1},q_j),(p_i,q_{j+1}),(p_{i+1},q_{j+1})\}$. The *discrete Fréchet distance* between P and Q is

$$\delta_{dF}(P,Q) = \min_{C\,coupling} \max_{(p_i,q_j)\in C} \rho(p_i,q_j).$$

Eiter and Mannila [5] showed that

$$\delta_F(P,Q) \le \delta_{dF}(P,Q) \le \delta_F(P,Q) + \max\{D(P),D(Q)\},$$

where $D(P)$ (resp., $D(Q)$) is the length of the longest edge in P (resp., Q). Thus, if we add vertices to the curves P, Q so that their edge lengths tend to zero, their discrete Fréchet distance will tend to their Fréchet distance.

Dynamic Programming Algorithm. Following [5], we quickly review the standard quadratic dynamic programming algorithm for the decision version of the discrete Fréchet distance, in a metric space (V,ρ).

Given two point sequences $A = (a_1,\ldots,a_n)$, $B = (b_1,\ldots,b_n)$, and a parameter $\varepsilon \ge 0$, the algorithm creates an $n \times n$ Boolean matrix M, whose rows and columns correspond to the points of A and B, respectively. The algorithm fills the matrix with values 0/1 row by row. Every cell $M_{i,j}$ in the matrix is filled by 1 iff both conditions hold:

1. At least one of the cells $M_{i-1,j}$, $M_{i,j-1}$, $M_{i-1,j-1}$ is filled with 1.

2. The distance $\rho(a_i,b_j)$ is at most ε.

Otherwise, $M_{i,j}$ is filled by 0. Intuitively, an entry $M_{i,j}$ is equal to 1 iff the pair (a_i,b_j) is reachable from the starting placement (a_1,b_1) of the trip with a “leash” of length ε. Otherwise, $M_{i,j}$ is equal to 0.

The runtime of the algorithm is quadratic and the number of input comparisons it does is also quadratic, as there are potentially n^2 distinct pairs of points (a_i,b_j) to check whether $\rho(a_i,b_j) \le \varepsilon$.

3 Decision Tree for the Euclidean Plane

Buchin et al. [4] showed a quadratic algebraic decision tree[4] with depth $O(n^{4/3}\log n)$ for the Euclidean discrete Fréchet distance in the plane.

The decision tree is based on invoking the quadratic dynamic programming algorithm following a preprocessing stage. All the input comparisons in the dynamic programming algorithm are made by checking if the distance of a point $a_i \in A$ from a point $b_j \in B$ is less than the fixed given parameter ε. The preprocessing stage will compute and store the answers for these pairwise distance queries in a Boolean matrix $T \overset{\text{def}}{=} (t_{ij})$, where $t_{ij} = 1$ if $\|a_i - b_j\|_2 \le \varepsilon$, otherwise $t_{ij} = 0$.

Given two point sequences A, B, with $|A| = n$, $|B| = m$, and a parameter $\varepsilon > 0$, denote, for each point $a \in A$, the circle of radius ε centered at a as c_a. A point $b \in B$ lies inside a circle c_a iff $\|a - b\|_2 \le \varepsilon$. We obtain a set C of n congruent circles (all of radius ε) and a set $P(= B)$ of m points.

Katz and Sharir [9] showed that one can compute a compact representation of the set of pairs of the form (c,p), where $p \in P$, $c \in C$, and p lies inside c, in $O((m^{2/3}n^{2/3} + m + n)\log n)$ time and space. This information suffices to construct T and invoke the dynamic programming algorithm without using further input comparisons.

Thus in total, when $|A| = |B| = n$, the number of input comparisons is $O(n^{4/3}\log n)$.

4 Decision Trees for L_1 and L_∞ in $\mathbb{R}^d$

Similar to the Euclidean case, range searching techniques can also be used for the problem under other metrics, for computing the pairwise distance queries in the decision tree. However, as we now show, these techniques, when routinely implemented, will give much weaker results than those stated in Theorem 1 and Theorem 2.

[4]Namely, each branching is a sign test of a quadratic expression.

The simpler case is for the L_∞ metric, for which the unit ball in $\mathbb{R}^d$ has the form of a d-dimensional hypercube. One can compute a d-dimensional range tree data structure for the points of A, in time $O(n \log^{d-1} n)$. For each point $b = (b_1, \ldots, b_d) \in B$, denote by c_b its corresponding d-sphere (under L_∞) of radius ε, centered at b. Clearly, $c_b = [x_1, y_1] \times [x_2, y_2] \times \cdots \times [x_d, y_d]$ is a d-dimensional hypercube.

For each $b \in B$, we query the range tree with its corresponding hypercube c_b. This will give us all the points of A that lie in c_b. Since for each interval of c_b, the query takes $O(\log n)$ time, the query for c_b takes $O(\log^d n)$ time. Using *fractional cascading*, this can be improved to $O(\log^{d-1} n)$ time. In total, this approach leads to a 2-linear decision tree of depth $O(n \log^{d-1} n)$.

For the L_1 metric, a similar approach will lead to a much weaker result. The unit ball under the L_1 metric forms a d-dimensional cross-polytope with 2^d facets. Thus, querying such a ball will require 2^d queries, each performing $O(\log n)$ $2d$-linear comparisons, resulting in a $2d$-linear decision tree of depth $O(n \log^{2^d} n)$.

The range searching data structure is appropriate also when the queries are not known in advance. Using Fredman's trick, we leverage the fact that in our case all the queries are known in advance, to obtain better decision trees.

The L_1 discrete Fréchet distance. We start by presenting a 4-linear decision tree with depth $O(n \log n)$ for the L_1 discrete Fréchet distance in $\mathbb{R}^2$, and then we explain how to modify it to obtain a $2d$-linear decision tree with depth $O(n \log n)$ for the problem in $\mathbb{R}^d$. This will prove Theorem 1.

The following property will allow us to apply *Fredman's trick* on pairwise distance queries under the L_1 norm.

For any real numbers $x, y, z \in \mathbb{R}$, $|x| + |y| \le z$ *if and only if* all the following inequalities hold.

$$\begin{aligned} x + y &\le z, & x - y &\le z, \\ -x + y &\le z, & -x - y &\le z. \end{aligned}$$

Since the L_1 distance between a point $a_i = (x_i, y_i)$ and a point $b_j = (x_j, y_j)$ is defined by

$$\|a_i - b_j\|_1 = |x_i - x_j| + |y_i - y_j|,$$

the property above leads to the following observation.

Observation 1 *For $a_i = (x_i, y_i)$, $b_j = (x_j, y_j) \in \mathbb{R}^2$, $\|a_i - b_j\|_1 \le \varepsilon$ if and only if all the following inequalities hold.*

$$\begin{aligned} x_i + y_i &\le x_j + y_j + \varepsilon, \\ x_i - y_i &\le x_j - y_j + \varepsilon, \\ y_i - x_i &\le y_j - x_j + \varepsilon, \\ -x_i - y_i &\le -x_j - y_j + \varepsilon. \end{aligned}$$

This observation is a sort of generalization of *Fredman's trick* for the L_1 distance between two points in the plane.

Recall that we are given two point sequences in the plane $A = (a_1, \ldots, a_n)$, $B = (b_1, \ldots, b_n)$, and a distance parameter ε. The following algorithm determines whether $\delta_{dF}(A, B) \le \varepsilon$.

1. Sort $D_1 \overset{\text{def}}{=} \{x_i + y_i,\ x'_j + y'_j + \varepsilon \mid a_i = (x_i, y_i) \in A,\ b_j = (x'_j, y'_j) \in B\}$.
2. Sort $D_2 \overset{\text{def}}{=} \{x_i - y_i,\ x'_j - y'_j + \varepsilon \mid a_i = (x_i, y_i) \in A,\ b_j = (x'_j, y'_j) \in B\}$.
3. Sort $D_3 \overset{\text{def}}{=} \{y_i - x_i,\ y'_j - x'_j + \varepsilon \mid a_i = (x_i, y_i) \in A,\ b_j = (x'_j, y'_j) \in B\}$.
4. Sort $D_4 \overset{\text{def}}{=} \{-x_i - y_i,\ -x'_j - y'_j + \varepsilon \mid a_i = (x_i, y_i) \in A,\ b_j = (x'_j, y'_j) \in B\}$.
5. Using Observation 1, given the sorted orders on $D_1, \ldots, D_4$, construct the $n \times n$ Boolean matrix
$$T \overset{\text{def}}{=} (t_{ij}), \text{ where } t_{ij} = \begin{cases} 1 & \text{if } \|a_i - b_j\|_1 \le \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$
6. Invoke the dynamic programming algorithm using T for the distance queries.

Steps 1–4 require $O(n \log n)$ comparisons. Using Observation 1, Step 5 requires no comparisons (on the raw data) at all, given the sorted orders on $D_1, \ldots, D_4$. Specifically, to test whether $\|a_i - b_j\|_1 \le \varepsilon$, we test the four corresponding inequalities from Observation 1. Each inequality test is resolved by the sorted orders on $D_1, \ldots, D_4$. Step 6 requires no comparisons, given the matrix T from Step 5. All comparisons are sign tests of 4-linear expressions. In total, the number of comparisons is $O(n \log n)$. The algorithm can be *implemented* to run in $O(n^2)$ time, using only $O(n \log n)$ input comparisons.

The algorithm can easily be extended to $\mathbb{R}^d$, by using additional sorting steps (similar to steps 1–4), and lead to a $2d$-linear decision tree with depth $O(n \log n)$. A generalization of Observation 1 to points $a_i = (x_{i_1}, \ldots, x_{i_d})$, $b_j = (x_{j_1}, \ldots, x_{j_d})$ in $\mathbb{R}^d$ leads to 2^d inequalities, each defined by a vector $\delta \in \{-1, 1\}^d$, and has the form

$$\sum_{k=1}^{d} \delta_k x_{i_k} \le \sum_{k=1}^{d} \delta_k x_{j_k} + \varepsilon.$$

Each such inequality is a $2d$-linear expression. Thus, for the same problem in $\mathbb{R}^d$, the algorithm has 2^d sorting steps, and all comparisons are sign tests of $2d$-linear expressions. This proves Theorem 1. $\square$

The L_∞ discrete Fréchet distance. The previous algorithm can easily be modified (and simplified) for the L_∞ norm. As before, we first consider the problem in $\mathbb{R}^2$, and later extend it to $\mathbb{R}^d$. The L_∞ distance between a point $a_i = (x_i, y_i) \in \mathbb{R}^2$ and a point $b_j = (x_j, y_j) \in \mathbb{R}^2$ is defined by $\|a_i - b_j\|_\infty = \max\{|x_i - x_j|, |y_i - y_j|\}$. Hence,

$$\|a_i - b_j\|_\infty \le \varepsilon \Leftrightarrow (|x_i - x_j| \le \varepsilon) \wedge (|y_i - y_j| \le \varepsilon).$$

Thus we obtain the following observation.

Observation 2 *For $a_i = (x_i, y_i)$, $b_j = (x_j, y_j) \in \mathbb{R}^2$, $\|a_i - b_j\|_\infty \le \varepsilon$ if and only if all the following inequalities hold.*

$$\begin{aligned} x_i &\le x_j + \varepsilon, & x_j &\le x_i + \varepsilon, \\ y_i &\le y_j + \varepsilon, & y_j &\le y_i + \varepsilon. \end{aligned}$$

This leads to the following variant of the previous algorithm, where the sets to be sorted are:

$$\begin{aligned} D_1 &\overset{\text{def}}{=} \{x_i,\, x'_j + \varepsilon \mid a_i = (x_i, y_i) \in A,\, b_j = (x'_j, y'_j) \in B\}, \\ D_2 &\overset{\text{def}}{=} \{x'_j,\, x_i + \varepsilon \mid a_i = (x_i, y_i) \in A,\, b_j = (x'_j, y'_j) \in B\}, \\ D_3 &\overset{\text{def}}{=} \{y_i,\, y'_j + \varepsilon \mid a_i = (x_i, y_i) \in A,\, b_j = (x'_j, y'_j) \in B\}, \\ D_4 &\overset{\text{def}}{=} \{y'_j,\, y'_i + \varepsilon \mid a_i = (x_i, y_i) \in A,\, b_j = (x'_j, y'_j) \in B\}. \end{aligned}$$

Using Observation 2, given the sorted orders on $D_1, \ldots, D_4$, one can construct the Boolean matrix

$$T \overset{\text{def}}{=} (t_{ij}), \text{ where } t_{ij} = \begin{cases} 1 & \text{if } \|a_i - b_j\|_\infty \le \varepsilon \\ 0 & \text{otherwise,} \end{cases}$$

with no further comparisons. Now, one can invoke the dynamic programming algorithm and use T for the distance queries.

Similarly to the L_1 norm, the above algorithm uses $O(n \log n)$ input comparisons and can be implemented to run in $O(n^2)$ time. Each comparison is a sign test of a 2-linear expression.

Following a generalization of Observation 2 to points in $\mathbb{R}^d$, the algorithm can be extended to $\mathbb{R}^d$ by adding additional sorting steps. We have $2d$ sorting steps for the problem over $\mathbb{R}^d$, two for each coordinate. Each comparison will still be a 2-linear expression, independent of d. Thus in total we obtain a 2-linear decision tree with depth $O(n \log n)$ for the problem in $\mathbb{R}^d$, for any constant d. This proves Theorem 2. $\square$

5 Discussion

An intriguing aspect of the presented results is the "large" gap we obtain between the nonuniform and the known uniform complexity of the problems.

For some archetypal problems in $\mathbf{P}$, a gap of $\sqrt{n}$ was shown[5], starting with the $(\min, +)$-matrix multiplication [7], to the recent 3SUM and Zero Triangle results [8]. For the *Euclidean* discrete Fréchet distance in the plane, a gap of $n^{2/3}$ was noted above.

The quadratic time algorithm of Eiter and Mannila [5] can compute the discrete Fréchet distance in any metric space. For the L_1 and L_∞ versions, our $O(n \log n)$ decision trees give a gap of n. We hope that this "large" gap will motivate the study of the problem in these metrics. In particular, can one obtain a *truly subquadratic* algorithm for these problems? or on the other hand, does a similar result to the conditional lower bound of Bringmann [2] (for the *Euclidean* discrete Fréchet distance) can be obtained for the problem under metrics L_1 and L_∞?

Another open question is whether our decision trees are optimal. The $\Omega(n \log n)$ lower bound proof in [3] cannot be applied to the L_1 or L_∞ versions, as it exploits the *strict convexity* of the *Euclidean* plane.

References

[1] P. K. Agarwal, R. Ben Avraham, H. Kaplan, and M. Sharir. Computing the discrete Fréchet distance in subquadratic time. *SIAM J. Comput.*, 43(2):429–449, 2014.

[2] K. Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. *Proc. 55th IEEE Annu. Sympos. Foundations of Computer Science*, pages 661–670, 2014.

[3] K. Buchin, M. Buchin, C. Knauer, G. Rote, and C. Wenk. How difficult is it to walk the dog? *Proc. 23rd Euro. Workshop Comput. Geom.*, pages 170–173, 2007.

[4] K. Buchin, M. Buchin, W. Meulemans, and W. Mulzer. Four soviets walk the dog - with an application to Alt's conjecture. *Proc. 25th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 1399–1413, 2014.

[5] T. Eiter and H. Mannila. Computing discrete Fréchet distance. Technical report, TU Vienna, Austria, 1994.

[6] M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Mathematico di Palermo*, 22:174, 1906.

[7] M. L. Fredman. New bounds on the complexity of the shortest path problem. *SIAM J. Comput.*, 5(1):83–89, 1976.

[8] A. Grønlund and S. Pettie. Threesomes, degenerates, and love triangles. *Proc. 55th IEEE Annu. Sympos. Foundations of Computer Science*, pages 621–630, 2014.

[9] M. J. Katz and M. Sharir. An expander-based approach to geometric optimization. *SIAM J. Comput.*, 26(5):1384–1408, 1997.

[5]We refer to such gaps by excluding $o(n^\varepsilon)$ factors, for any $\varepsilon > 0$.