Advanced Computational Geometry Presentation
Locality Sensitive Hashing

Seyed Iman Hosseini Zavaraki*

July 28, 2018

1 GitHub

All the resources of the presentation, including the slides, the tex files of this document, and the code related to LSH algorithm which I developed for this presentation is available at my github: https://github.com/ImanHosseini/LSH

2 The Paper

This paper is from Proceedings of 33rd International Symposium on Computational Geometry (SoCG) 2017. [1] Anne Driemel is Assistant Professor in TU Eindhoven and Francesco Silvestri is Assistant Professor in University of Padova.

The paper deals with the problem of designing data structures to accommodate fast query on a set of polygonal curves to retain similar curves via Frechet or Dynamic Time Warping (DTW) distance. The state of the art, prior to this paper, was by Indyk as established by this[2] paper in 2002.

3 Similarity of Curves

There are various notions for similarity between curves. In this paper, two of them are taken in to consideration.

Dynamic time warping (DTW) is one of the means of measuring similarity of curves (usually the so-called curves are actually temporal sequences) which may vary in speed. For

*hosseini.iman@yahoo.com

[1] https://arxiv.org/abs/1703.04040

[2] P. Indyk. Approximate nearest neighbor algorithms for Frchet distance via product metrics. In Proc. 18th Symp. on Computational Geometry (SOCG)

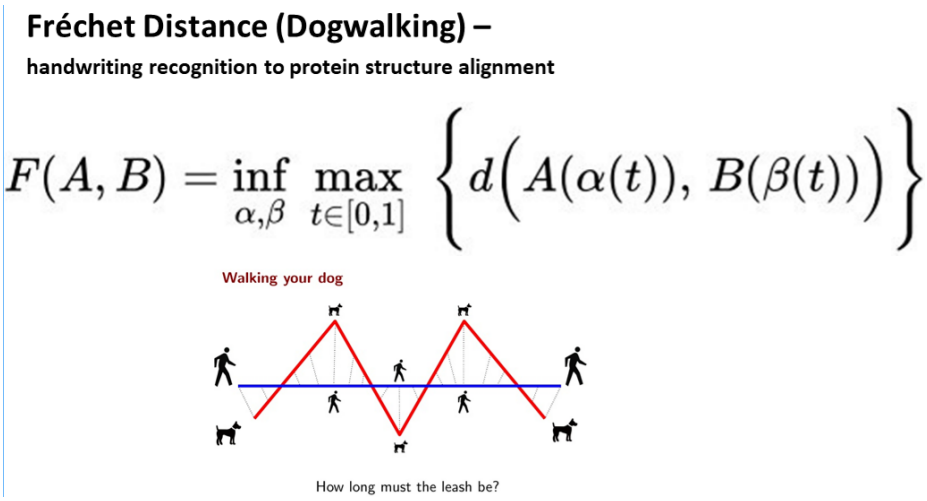


Figure 1: Frechet Distance

instance, similarities in walking could be detected using DTW, even if one person was walking faster than the other. DTW has been applied to temporal sequences of video, audio, and graphics data, and automatic speech recognition, to cope with different speaking speeds. Other applications include speaker recognition and online signature recognition. Also it can be used in partial shape matching application.

In general, DTW induces a metric on given sequences (e.g. time series) by assigning an optimal match with certain restriction and rules:

1) Every index from the first sequence must be matched with one or more indices from the other sequence, and vice versa The first index from the first sequence must be matched with the first index from the other sequence (but it does not have to be its only match)
2) The last index from the first sequence must be matched with the last index from the other sequence (but it does not have to be its only match)
3) The mapping of the indices from the first sequence to indices from the other sequence must be monotonically increasing, and vice versa, i.e. if $j > i$ are indices from the first sequence, then there must not be two indices $l > k$ in the other sequence, such that index i is matched with index l and index j is matched with index k, and vice versa.

Frechet distance, also known as dog-walking distance, is the length of the shortest leash sufficient for a dog and the owner to traverse their respective curvess. Note that the definition is symmetric with respect to the two curves.

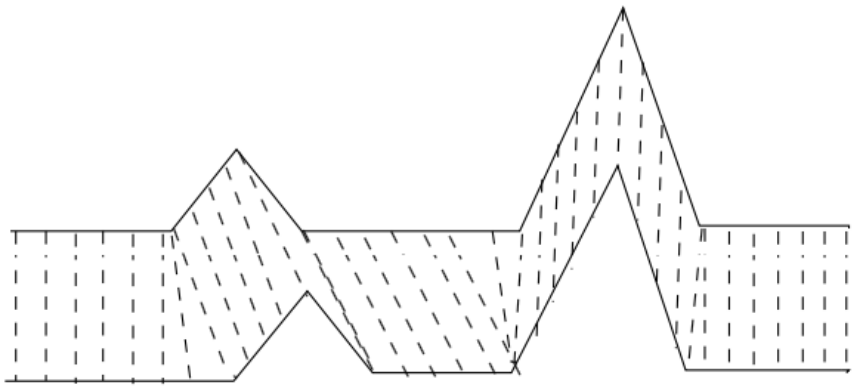


Figure 2: Dynamic Time Warping

▶ **Lemma 3.** *For any two curves $P = p_1, \ldots, p_{m_1}$ and $Q = q_1, \ldots, q_{m_2}$, there always exists an optimal traversal T with the following two properties:*

(i) T consists of at most $m = \min\{m_1, m_2\}$ disconnected components.

(ii) Each component is a star, i.e., all edges of this component share a common vertex.

Figure 3: The Solution

4 Locality Sensitive Hashing

Before introducing the main idea, an observation is made (as seen in the following lemma) which serves as the basis for the main idea:

The first part is trivial (charge each component to a vertex of the shorter curve that is contained in it), and the second part can be proved by contradiction. (iteratively remove reduntant edges)

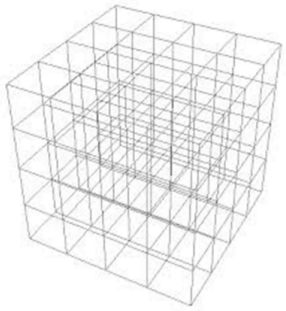
The main idea is that we can snap any curve to discrete grids. We make a discrete grid on the space and then any point on the curve would be replaced by the closest grid-point to it. Naturally, the more fine-grained our grid is, the curve would undergo less change, but what this snapping to grid does, is that now different points can be identified after this algorithm is applied.

Unlike Indyk's work, in this work a non deterministic approach is investigated. This work's novelty is that it is proving bounds on a non-deterministic algorithm, proving bounds on probabilities of collision of hashes. Imagine a family of grids, as described above we can shift the grids by a random vector. Thus we obtain a family of grids, and in a non-deterministic way, corresponding to the exact value of that random vector (which say, comes form a uniform distribution) we get new curves instead of the original curves.

Snapping to grid

• $G_\delta = \{(x1, \ldots, xd) \in \mathbb{R}^d \mid \forall\ 1 \le i \le d\ \exists\ j \in \mathbb{N} : x_i = j \cdot \delta\ \}$

$$\widehat{G}_\delta^t = \{p + t \mid p \in G_\delta\}$$



t picked uniformly from $[0, \delta)^d$

Figure 4: Family of grids

As discussed, after snapping the curves to a grid, they can coincide, and some 'collide' this way. Naturally, the more similar the original curves are, there is more chance of collision. This intuition is quantified in Lemmas 5 and 6 and after some change of variable they result in Theorem 7. Lemma 3 is the cornerstone of this theorem, as it allowed us to prove Lemma 5. Lemma 3 characterizes the optimal matchings, and we later use those characterizations to prove the bounds. It puts an upper-bound on the number of connected components, and states that the connected components are stars, and these, are directly derivable from the optimality of the matching. The rest of the paper, is then dedicated to expanding the algorithm to incorporate constraints on the matching, and to establish -quantitively- the trade-off between approximation factor and the query time.

▶ **Theorem 7.** *Let $P, Q \in \Delta^d$ be two curves with m_1 and m_2 points, respectively, and let $m = \min\{m_1, m_2\}$, $\delta = 4dmr$ and $c = 4d^{\frac{3}{2}}m$. It holds that:*
(i) if $d_F(P,Q) < r$, then $Pr_{\mathcal{H}_\delta^L}(h_\delta^t(P) = h_\delta^t(Q)) > \frac{1}{2}$;
(ii) if $d_F(P,Q) > cr$, then $Pr_{\mathcal{H}_\delta^L}(h_\delta^t(P) = h_\delta^t(Q)) = 0$.

Proof. The first claim follows by plugging in the bounds of Lemma 5:

$$Pr\big(h_\delta^t(P) = h_\delta^t(Q)\big) > 1 - \left(2dm \cdot \frac{d_F(P,Q)}{\delta}\right) > 1 - \frac{d_F(P,Q)}{2r} > \frac{1}{2}.$$

On the other hand, the second claim follows from Lemma 6:

$$d_F(P,Q) > c \cdot r = 4d^{3/2}mr = \sqrt{d} \cdot \delta \quad \Rightarrow \quad h_\delta^t(P) \neq h_\delta^t(Q).$$

Figure 5: Theorem 7

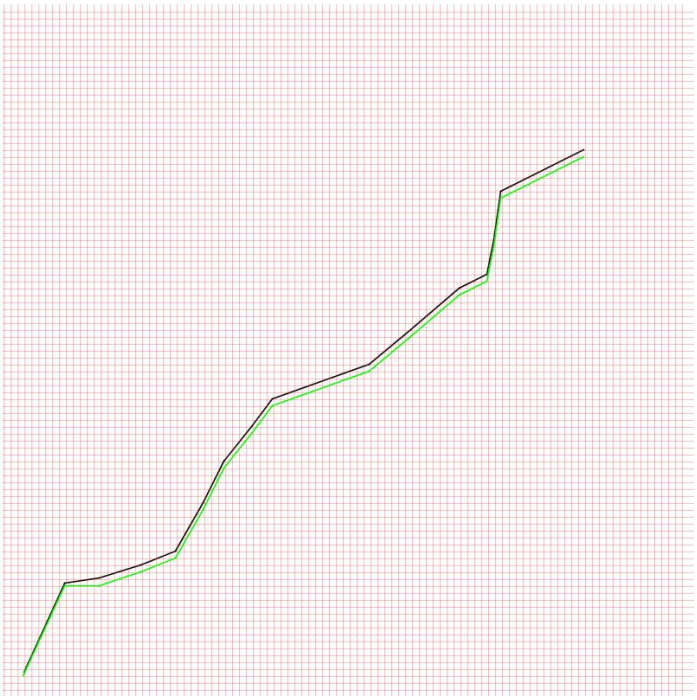


Figure 6: Snapping a curve to grid