The One-Way Communication Complexity of Dynamic Time Warping Distance

Vladimir Braverman
Johns Hopkins University, Baltimore MD, USA
vova@cs.jhu.edu

Moses Charikar
Stanford University, Stanford CA, USA
moses@cs.stanford.edu

William Kuszmaul
Massachusetts Institute of Technology, Cambridge MA, USA
kuszmaul@mit.edu

David P. Woodruff
Carnegie Mellon University, Pittsburgh PA, USA
dwoodruf@cs.cmu.edu

Lin F. Yang
Princeton University, Princeton NJ, USA
lin.yang@princeton.edu

Abstract

We resolve the randomized one-way communication complexity of Dynamic Time Warping (DTW) distance. We show that there is an efficient one-way communication protocol using $\widetilde{O}(n/\alpha)$ bits for the problem of computing an α-approximation for DTW between strings x and y of length n, and we prove a lower bound of $\Omega(n/\alpha)$ bits for the same problem. Our communication protocol works for strings over an arbitrary metric of polynomial size and aspect ratio, and we optimize the logarithmic factors depending on properties of the underlying metric, such as when the points are low-dimensional integer vectors equipped with various metrics or have bounded doubling dimension. We also consider linear sketches of DTW, showing that such sketches must have size $\Omega(n)$.

2012 ACM Subject Classification Theory of computation → Communication complexity

Keywords and phrases dynamic time warping, one-way communication complexity, tree metrics

Digital Object Identifier 10.4230/LIPIcs.SoCG.2019.16

Related Version A full version of this paper is available at `https://arxiv.org/abs/1903.03520`.

Funding *Vladimir Braverman*: Supported by NSF CAREER grant 1652257, ONR Award N00014-18-1-2364, and the Lifelong Learning Machines program from DARPA/MTO.
Moses Charikar: Supported by NSF grant CCF-1617577 and a Simons Investigator Award.
William Kuszmaul: Supported by an MIT Akamai Fellowship and a Fannie & John Hertz Foundation Fellowship. Also supported by NSF grants 1314547 and 1533644.
David P. Woodruff: Partially supported by NSF Big Data grant 1447639.
Lin F. Yang: This work was done while the author was visiting IBM in 2017, hosted by David Woodruff, and supported by NSF CAREER grant 1652257.

Acknowledgements David P. Woodruff would like to thank the Simons Institute for the Theory of Computing where part of this work was done.

1 Introduction

The Dynamic Time Warping (DTW) distance is a widely used distance measure between time series. It is particularly flexible in dealing with temporal sequences that vary in speed. To measure the distance between two sequences, each sequence is “warped" non-linearly in the time dimension (i.e., portions of each sequence are stretched by varying amounts) and the warped sequences are compared by summing up distances between corresponding elements. DTW was popularized in the speech recognition community by Sakoe and Chiba [34]. It was introduced in the data mining community for mining time series by Berndt and Clifford [9]. Its many applications include phone authentication [14], signature verification [29], speech recognition [28], bioinformatics [1], cardiac medicine [11], and song identification [38]. Several techniques and heuristics have been developed to speed up natural dynamic programming algorithms for it [19, 34, 25, 26, 24, 6, 33]. We refer the reader also to Section 2 of [4] for more references.

Distance measures on sequences and time series have been extensively studied in the literature. Given two sequences $x = x_1, x_2, \ldots, x_m$ and $y = y_1, y_2, \ldots, y_n$ of points in $\mathbb{R}^d$ (or a metric space), one seeks to “match the points up” as closely as possible. One way of doing this is to define a “correspondence” $(\bar{x}, \bar{y})$ between x, y by considering expansions of x and y to produce sequences of equal length, i.e., we duplicate each point x_i some number m_i times (to produce $\bar{x}$) and each point y_j some n_j times (to produce $\bar{y}$), so that $\sum_{i=1}^{m} m_i = \sum_{i=1}^{n} n_i$. Now, we define a vector z with $z_i = d(\bar{x}_i, \bar{y}_i)$, for some underlying distance function d and choose the correspondence which minimizes a certain function of z. For example, minimizing $\sum z_i$ leads to the Dynamic Time Warping distance. Minimizing $\max_i z_i$ leads to the discrete Fréchet distance. The edit distance between strings can be similarly cast in this framework. One unusual aspect of DTW (in contrast to its close cousins, edit distance and Fréchet distance) is that it does not satisfy the triangle inequality.

Edit distance and Fréchet distance have received a lot of attention in the theory community. Fundamental questions such as exact and approximation algorithms, nearest neighbor search, sketching, and communication complexity have been intensively studied. However, there are relatively few results about DTW. Similar to edit distance, DTW can be computed by a quadratic-time dynamic program. Recently, it was shown that there is no strongly subquadratic-time algorithm for DTW unless the Strong Exponential Time Hypothesis is false [10, 2]; approximation algorithms for DTW were obtained under under certain assumptions about properties of the input strings [3, 37]; and slightly subquadratic algorithms for DTW have also been obtained [18]. DTW was studied in the context of LSH [15] and nearest neighbor search [35, 16]. To the best of our knowledge, until now, there has been no study of the communication complexity of this basic distance measure on sequences.

In this paper, we study the *one-way communication complexity* of DTW. For a distance measure $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{\geq 0}$ such as DTW, the goal in the one-way communication model is to define a randomized function S and an estimation procedure E so that for any $x, y \in \mathcal{X}$, given $S(x)$ and y, the output $E(S(x), y) \approx d(x, y)$ with large probability. There are various notions of approximation, but a natural one is that $d(x, y) \leq E(S(x), y) < \alpha d(x, y)$ for an approximation factor $\alpha > 1$. The challenge is to understand how large $S(x)$ needs to be (for sequences of length n) in order to obtain approximation factor α. A closely related notion is that of sketching, where the estimation procedure takes $S(x)$ and $S(y)$ and we require that $E(S(x), S(y)) \approx d(x, y)$ with large probability. This one-way communication complexity question has been studied previously for edit distance, in the context of document exchange [7, 8, 20]. This model captures a number of applications, e.g., lower bounds in it apply to data stream algorithms and to sketching protocols. Upper bounds in it are appropriate for nearest neighbor search; indeed, the natural thing to do here is a lookup table, so a (one-way) sketch of size b bits creates a table of size 2^b (see e.g., [5]). One-way communication is one of the simplest and most natural settings in which one can study communication complexity, and it has rich connections to areas such as information theory, coding theory, on-line computing, and learning theory [27].

1.1 Our results

Our main result is a tight $\tilde{\Theta}(n/\alpha)$ bound, up to logarithmic factors, on the one-way communication complexity of computing an α-approximation to DTW. The results are discussed in more detail below.

We present a communication protocol using $\tilde{\Theta}(n/\alpha)$ bits which works for DTW on any underlying metric space of polynomial size and aspect ratio (Theorem 8). We optimize the logarithmic factors in the important case when the points are natural numbers and the distance $d(a, b) = |a - b|$, as well as more generally when the points are low-dimensional integer vectors equipped with various metrics (Theorem 9); we also optimize for the important case where the underlying metric has small doubling dimension (Theorem 9). At the cost of an extra logarithmic factor in complexity, all of our protocols are also time-efficient, in that Alice and Bob each run in polynomial time.

Next, we turn to lower bounds. Our communication protocol is non-linear, and we show that in general linear sketches must have size $\Omega(n)$ (Theorem 12). Moreover, we prove that our upper bounds are within a polylogarithmic factor of tight, establishing a randomized one-way communication lower bound of $\Omega(n/\alpha)$ for any underlying metric space of size at least three, for one-way communication algorithms which succeed with constant success probability (Theorem 11). We optimize this in several ways: (1) when the underlying metric is generalized Hamming space over a point set of polynomial size $n^{1+\Omega(1)}$, we improve the lower bound to $\Omega(n/\alpha \cdot \log n)$ for algorithms which succeed with probability $1 - 1/n$, and show this is optimal (Theorem 10); (2) for the natural numbers, we improve the lower bound to $\Omega(n/\alpha \cdot \log(\min(\alpha, |\Sigma|)))$ for algorithms which succeed with probability at least $1 - 1/\min(\alpha, |\Sigma|)$ (see the extended paper [36]). We note that our lower bound of $\Omega(n/\alpha)$ applies even to approximating DTW in the low distance regime (i.e., distinguishing $\mathrm{DTW}(x, y) \leq 1$ versus $\mathrm{DTW}(x, y) > \alpha$ with constant probability), and that in this regime the edit distance admits a much smaller sketching complexity [7, 21]. To the best of our knowledge, our result provides the first separation between the DTW and the edit distance.

We summarize our results in Table 1. The layout of the paper is as follows: We present preliminaries in Section 2. We give a detailed overview of our techniques and results in Section 3. Then in Section 4 we give a complete treatement of several of the core results. A full presentation of all of the technical results appears in the extended paper [36].

2 Preliminaries

As a convention, we say an event occurs with *high probability* if it happens with probability at least $1 - \frac{1}{\mathrm{poly}(n)}$ for a polynomial of our choice. Throughout the paper, we use (Σ, d) to denote a finite metric space. We denote by Σ^n the set of strings of length n over Σ and by $\Sigma^{\leq n}$ the set of strings of length at most n over Σ. An important property of Σ will be its *aspect ratio*, which is defined as the ratio between the diameter of Σ and the smallest distance between distinct points in Σ.

Table 1 Summary of results on computing α multiplicative approximation of DTW_n over a metric space Σ with aspect ratio $\text{poly}(n)$. *These upper bounds are also time efficient. Inefficient protocols can remove an additional $\log\alpha$ factor in the communication complexity. †These lower bounds hold for protocols that are correct with probability $1-1/n$ or $1-1/\min(\alpha,|\Sigma|)$.

Model	Metric Space	Communication Bounds	Theorem
One-way	Finite	$O(n/\alpha\cdot\log\alpha\cdot\log^3 n)^*$	8
	Natural Numbers	$O(n/\alpha\cdot\log\alpha\cdot\log^2 n\cdot\log\log\log n)^*$	9
	ℓ_p^d	$O_{p,d}(n/\alpha\cdot\log\alpha\cdot\log^2 n\cdot\log\log\log n)^*$	9
	doubling constant λ	$O(\log\lambda\cdot n/\alpha\cdot\log\alpha\cdot\log^2 n\cdot\log\log\log n)^*$	9
	Finite	$\Omega(n/\alpha)$	11
	Generalized Hamming	$\Theta(n/\alpha\cdot\log n)^\dagger$	10
Linear Sketch	Finite	$\Omega(n)$	12

Dynamic Time Warping Distance

We study the *dynamic warping distance* (DTW) of strings $x, y \in \Sigma^{\le n}$. Before we formally define the DTW distance, we first introduce the notion of an *expansion* of a string.

▶ **Definition 1.** *The* runs *of a string $x \in \Sigma^{\le n}$ are the maximal substrings consisting of a single repeated letter. Any string obtained from x by extending x's runs is an* expansion *of x.*

For example, the runs of *aabbbccd* are *aa*, *bbb*, *cc*, and *d*. Given a string x, we can *extend* a run in x by further duplicating the letter which populates the run. For example, the second run in *aabbbccd* can be extended to obtain *aabbbbbccd*, and we say the latter string is an expansion of the first.

Using the notion of an expansion, we can now define dynamic time warping.

▶ **Definition 2.** *Consider two strings $x, y \in \Sigma^{\le n}$. A* correspondence[1] *between x and y is a pair $(\overline{x}, \overline{y})$ of equal-length expansions of x and y. The* edges *in a correspondence are the pairs of letters $(\overline{x}_i, \overline{y}_i)$, and the* cost *of an edge is given by $d(\overline{x}_i, \overline{y}_i)$. The* cost *of a correspondence is the sum $\sum_i d(\overline{x}_i, \overline{y}_i)$ of the costs of the edges between the two expansions. A correspondence between x and y is said to be* optimal *if it has the minimum attainable cost, and the resulting cost is called the* dynamic time warping distance $\text{DTW}(x, y)$.

When discussing a correspondence $(\overline{x}, \overline{y})$, the following terms will be useful.

▶ **Definition 3.** *A run in $\overline{x}$* overlaps *a run in $\overline{y}$ if there is an edge between them. A letter x_i is* matched *to a letter y_j if the extended run containing x_i overlaps the extended run containing y_j.*

Note that any minimum-length optimal correspondence between strings $x, y \in \Sigma^{\le n}$ will be of length at most $2n$. In particular if in an optimal correspondence a run r_1 in x and a run r_2 in y have both been extended and overlap by at least one letter, then there is a shorter optimal correspondence in which the length of each run is reduced by one. Thus any minimum-length optimal correspondence has the property that every edge $(\overline{x}_i, \overline{y}_i)$ contains at least one letter from a run that has not been extended, thereby limiting the length of the correspondence to at most $2n$.

[1] A related concept, *traversal*, is sometimes used in the literature. A traversal can be viewed as the the set of matching edges of a correspondence.

DTW can be defined over an arbitrary metric space (Σ, d), and is also well-defined when d is a distance function not satisfying the triangle inequality.

Throughout our proofs, we will often refer to DTW over generalized Hamming space, denoted by $\mathrm{DTW}_0(x, y)$. As a convention, regardless of what metric space the strings x and y are initially taken over, $\mathrm{DTW}_0(x, y)$ is defined to be the DTW-distance between x and y obtained by redefining the distance function $d(\cdot, \cdot)$ to return 1 on distinct inputs.

One-Way Communication Complexity

In this paper, we focus on the one-way communication model. In this model, Alice is given an input x, Bob is given an input y, and Bob wishes to recover a valid solution to a problem with some solution-set $f(x, y) \subseteq \mathbb{R}$. (For convenience, we will refer to the problem by its solution set $f(x, y)$.) Alice is permitted to send Bob a single message $\mathrm{sk}(x)$, which may be computed in a randomized fashion using arbitrarily many public random bits. Bob must then use Alice's message $\mathrm{sk}(x)$ in order to compute some $F(\mathrm{sk}(x), y)$, which he returns as his proposed solution to $f(x, y)$.

The pair (sk, F) is a *p-accurate one-way communication protocol* for the problem $f(\cdot, \cdot)$ if for all x and y, the probability $\Pr[F(\mathrm{sk}(x), y) \in f(x, y)]$ that Bob returns a correct answer to $f(x, y)$ is at least p. The protocol is said to have *bit complexity* at most m if Alice's message $\mathrm{sk}(x)$ is guaranteed not to exceed m in length. Moreover, the protocol is said to be *efficient* if both sk and F can be evaluated in time polynomial in the length of x and y.

Fix a parameter $p \in (0, 1]$, the randomized *one-way communication complexity* $\mathrm{CC}_p(f)$ of the problem f is the minimum attainable bit complexity of a p-accurate one-way communication protocol for f. The focus of this paper is on the one-way communication complexity of the α-DTW problem, defined as follows:

▶ **Definition 4** (α-DTW). *The α-DTW$(\Sigma^{\leq n})$ problem is parameterized by an approximation parameter $1 \leq \alpha \leq n$. The inputs are a string $x \in \Sigma^{\leq n}$ and a string $y \in \Sigma^{\leq n}$. The goal is recover an α-approximation for* $\mathrm{DTW}(x, y)$. *In particular, the set of valid solutions is*

$$\{t \mid \mathrm{DTW}(x, y) \leq t < \alpha \cdot \mathrm{DTW}(x, y)\}.$$

One can also consider the decision version of this problem, in which one wishes to distinguish between distances at most r and distances at greater than $r\alpha$:

▶ **Definition 5** (DTEP). *The Decision Threshold Estimation Problem* $\mathrm{DTEP}_r^{\alpha}(\Sigma^{\leq n})$*, is paramaterized by a positive threshold $r > 0$ and an approximation parameter $1 \leq \alpha \leq n$. The inputs to the problem are a string $x \in \Sigma^{\leq n}$ and a string $y \in \Sigma^{\leq n}$. An output of 0 is a valid solution if* $\mathrm{DTW}(x, y) \leq r\alpha$*, and an output of 1 is a valid solution of* $\mathrm{DTW}(x, y) > r$.

Notice that any algorithm for α-DTW immediately gives an solution for DTEP_r^{α} for any $r > 0$. Conversely, any lower bound for the communication complexity of DTEP gives a lower bound for the communication complexity of α-DTW. For both of the above two definitions, we may omit the sequence space $\Sigma^{\leq n}$ if it is clear from the context.

3 Technical overview

In this section, we present the statements and proof overviews of our main results.

Complexity Upper Bounds

Our starting point is the following: suppose that $x, y \in \Sigma^n$ for a metric space Σ of polynomial size and aspect ratio, and further that the distances between points are always either 0 or at least 1. Alice and Bob wish to construct a 2/3-accurate one-way protocol for α-DTW.

Collapsing Repeated Points. Consider the strings $c(x)$ and $c(y)$, formed by reducing each run of length greater than one in x and y to the same run of length one. If we define l to be the length of the longest run in x or y, then $\mathrm{DTW}(x, y) \le l \cdot \mathrm{DTW}(c(x), c(y))$. Indeed, any correspondence $(\overline{c(x)}, \overline{c(y)})$ between $c(x)$ and $c(y)$ gives rise to a correspondence $(\bar{x}, \bar{y})$ between x and y obtained by duplicating each coordinate in $\overline{c(x)}$ and $\overline{c(y)}$ a total of l times. Moreover, since any correspondence $(\bar{x}, \bar{y})$ between x and y is also a correspondence between $c(x)$ and $c(y)$, it follows that $\mathrm{DTW}(c(x), c(y)) \le \mathrm{DTW}(x, y)$.

Inefficient Protocol via Hashing. Suppose Alice and Bob are guaranteed that $\mathrm{DTW}(x, y) \le n/\alpha$, and that the maximum run-length l satisfies $l < \alpha$. Then it suffices for Alice and Bob to compute $\mathrm{DTW}(c(x), c(y))$; and for this it suffices for Bob to be able to reconstruct $c(x)$. The claim is that from a random hash of $c(x)$ of length $O(n/\alpha \log n)$ bits, given $c(y)$, Bob can reconstruct $c(x)$. Indeed, given that $\mathrm{DTW}(c(x), c(y)) \le n/\alpha$, and given that the runs in $c(x)$ and $c(y)$ are all of length one, one can verify that there must be an optimal correspondence $(\overline{c(x)}, \overline{c(y)})$ between $c(x)$ and $c(y)$ such that $\overline{c(y)}$ is obtained from $c(y)$ by extending at most n/α runs. Since there are $n^{O(n/\alpha)}$ ways to choose which runs in $c(y)$ are extended, and since there are then $n^{O(n/\alpha)}$ ways to choose the new lengths to which those runs are extended, it follows that there are only $n^{O(n/\alpha)}$ options for $\overline{c(y)}$. Moreover, because $\overline{c(x)}$ and $\overline{c(y)}$ differ in at most n/α positions, for a given option of $\overline{c(y)}$ there are only $n^{O(n/\alpha)} \cdot |\Sigma|^{O(n/\alpha)} = n^{O(n/\alpha)}$ options for $\overline{c(x)}$ and thus for $c(x)$. Since starting from $c(y)$, there are only $n^{O(n/\alpha)}$ options for $c(x)$, meaning that a $O(n/\alpha \log n)$-bit hash allows Bob to recover $c(x)$ with high probability.

Efficiency via Edit Distance Sketch. In addition to requiring that $\mathrm{DTW}(x, y) \le n/\alpha$ and $l < \alpha$, the above protocol is inefficient since Bob needs to enumerate over all possibilities of $c(x)$ and compute the hash value of each. Exploiting the fact that $c(x)$ and $c(y)$ contain only runs of length one, we prove that $\mathrm{DTW}(c(x), c(y))$ is within a constant factor of the edit distance between $c(x)$ and $c(y)$. This means that Alice can instead invoke the edit-distance communication protocol of [21] of size $O(n/\alpha \log n \log \alpha)$, which allows Bob to efficiently recover $c(x)$ using the fact that the edit distance between $c(x)$ and $c(y)$ is $O(n/\alpha)$.

Handling Heavy Hitters. The arguments presented so far require that x and y contain no runs of length greater than α. We call such runs *heavy hitters*. To remove this restriction, a key observation is that there can be at most n/α heavy hitters. Therefore Alice can communicate to Bob precisely which runs are heavy hitters in x using $O(n/\alpha \log n)$ bits. The players then proceed as before: Alice collapses her input x to $c(x)$ by removing consecutive duplicates, and Bob collapses his input y to $c(y)$ by removing consecutive duplicates. We still have $\mathrm{DTW}(c(x), c(y)) \le \mathrm{DTW}(x, y)$ since any correspondence between x and y is a correspondence between $c(x)$ and $c(y)$. Thus, as before, Bob can reconstruct $c(x)$ whenever $\mathrm{DTW}(x, y) \le n/\alpha$. Now, though, it could be that $\mathrm{DTW}(x, y) > \alpha\, \mathrm{DTW}(c(x), c(y))$ because of the positions in $c(x)$ and $c(y)$ that occur more than α times. However, Bob uses his knowledge of the locations and values of the heavy hitters, together with $c(x)$, to create a string x' formed from x by collapsing runs of length less than α, and not doing anything to runs of length at least α. Now by computing $\mathrm{DTW}(x', y)$, Bob obtains a α-approximation for $\mathrm{DTW}(x, y)$, since any correspondence between x' and y gives rise to a correspondence between x and y by duplicating each letter α times.

Having handled the heavy hitters, the only remaining requirement by our protocol is that the distances between letters in x and y be zero and one. Thus we arrive at the following:

▶ **Proposition 6** (Protocol over Hamming Space). *Consider* DTW *over a metric space* Σ *of polynomial size with distances zero and one. Then for* $p = 1 - \text{poly}(n^{-1})$*, there is an efficient* p*-accurate one-way communication protocol for* α*-DTW over* $\Sigma^{\leq n}$ *which uses* $O\left(n\alpha^{-1} \cdot \log \alpha \cdot \log n\right)$ *bits. Moreover, for any* $\delta \in (0, 1)$*, there is an inefficient* $(1 - \delta)$*-accurate protocol for* α*-DTW*$(\Sigma^{\leq n})$ *using space* $O(n\alpha^{-1} \cdot \log n + \log \delta^{-1})$ *for any* $\delta \in (0, 1)$.

Note that our protocol is constructive in that it actually allows for y to build a correspondence between x and y satisfying the desired approximation bounds.

In generalizing to DTW over arbitrary metric spaces, we will use our protocol over Hamming Space as a primitive. Moreover, we will exploit the fact that it can be used to solve a slightly more sophisticated problem which we call *bounded* α*-DTW*:

▶ **Definition 7** (Bounded α-DTW). *In the bounded* α*-DTW*$(\Sigma^{\leq n})$ *problem, Alice and Bob are given strings* x *and* y *in* $\Sigma^{\leq n}$*. The goal for Bob is:*

- *If* $\text{DTW}_0(x, y) \leq n/\alpha$*, solve* α-DTW *on* (x, y).
- *If* $\text{DTW}_0(x, y) > n/\alpha$*, either solve* α-DTW *on* (x, y)*, or return "Fail".*

A crucial observation is that Proposition 6 continues to hold without modification if the alphabet Σ has arbitrary distances and our goal is to solve the bounded α-DTW problem.

Extending Distance Range via HSTs. The result for the bounded α-DTW problem allows for Bob to either determine an α-approximation for $\text{DTW}(x, y)$, or to determine that $\text{DTW}(x, y) > n/\alpha$. As a result the algorithm can be used to distinguish between $\text{DTW}(x, y) \leq n/\alpha$ an $\text{DTW}(x, y) > n$. One issue though is that the argument cannot distinguish between larger distances, such as for example between the cases $\text{DTW}(x, y) \leq n$ and $\text{DTW}(x, y) > n\alpha$. A key idea for resolving this issue is to first consider the DTW problem over a 2-hierarchically well-separated tree metric (HST), and then use the embedding of [17] to embed an arbitrary finite metric of polynomial size and aspect ratio into such a metric. A 2-hierarchically well-separated tree metric is defined as the shortest path metric on a tree whose nodes are elements of Σ and whose edges have positive weights for which on any root-to-leaf path, the weights are geometrically decreasing by a factor of 2. Since the weights decrease geometrically, for convenience we define pairwise distances in the tree metric to be the maximum edge length on the tree path between the nodes, a notion of distance which coincides with the sum of edge lengths up to a constant factor.

Suppose the points in Σ correspond to a 2-hierarchically well-separated tree metric and we wish to distinguish between whether $\text{DTW}(x, y) \leq nr/\alpha$ or $\text{DTW}(x, y) > nr$. A crucial idea is what we call the r-simplification $s_r(x)$ of a string x, which replaces each character p_i in x with its highest ancestor in the tree reachable via edges of weight at most $r/4$. A key property is that $\text{DTW}(s_r(x), s_r(y)) \leq \text{DTW}(x, y)$, since for two points ℓ_1, ℓ_2 in x, y, respectively, either they each get replaced with the same point in the r-simplifications of x and y, or the maximu-length edge on a path between ℓ_1 and ℓ_2 is the same before and after r-simplification. Notice that if a point in $s_r(x)$ is not equal to a point in $s_r(y)$, then their distance is at least $r/4$, by the definition of an r-simplification. Combining the preceding two observations, if $\text{DTW}(x, y) \leq nr/\alpha$, then $\text{DTW}(s_r(x), s_r(y)) \leq nr/\alpha$ and there is a correspondence for which $s_r(x)$ and $s_r(y)$ disagree in at most $4n/\alpha$ positions. On the other hand, since we only "collapse" edges of weight at most $r/4$, we have that if $\text{DTW}(x, y) > nr$, then $\text{DTW}(s_r(x), s_r(y)) > nr/2$, since the optimal correspondence has length at most $2n$.

It follows that the cases of $\text{DTW}(x, y) \leq nr/\alpha$ and $\text{DTW}(x, y) > nr$, correspond with the cases of $\text{DTW}(s_r(x), s_r(y)) \leq nr/\alpha$ and $\text{DTW}(s_r(x), s_r(y)) > nr/2$, and moreover that when $\text{DTW}(s_r(x), s_r(y)) \leq nr/\alpha$, there is an optimal correspondence for which $s_r(x)$ and

$s_r(y)$ disagree in at most $4n/\alpha$ positions. Thus we can use our protocol for the α-bounded DTW problem to figure out which case we are in, for a given r. This gives a protocol for distinguishing between whether $\mathrm{DTW}(x, y) \leq nr/\alpha$ or $\mathrm{DTW}(x, y) > nr$.

In order to obtain an α-approximation for $\mathrm{DTW}(x, y)$, the rough idea now is to run the above protocol multiple times in parallel as r varies in powers of 2, and then to find the smallest value of r for which the protocol declares $\mathrm{DTW}(x, y) \leq nr$. This works as long as points are taken from a 2-hierarchically well-separated tree metric. In order to extend the result to hold over arbitrary finite metrics of polynomial size and aspect ratio, the final piece is the embedding ϕ of [17], which embeds any polynomial size metric Σ into a 2-hierarchically well-separated tree metric for which for all $a, b \in \Sigma$, $d(a, b) \leq d(\phi(a), \phi(b))$ and $\mathbf{E}(d(\phi(a), \phi(b))) = O(\log n)d(a, b)$. This "lopsided" guarantee is sufficient for us since it ensures in any correspondence the sum of distances after performing the embedding will not shrink, while for a single fixed optimal correspondence, by a Markov bound the sum of distances after performing the embedding will not increase by more than an $O(\log n)$ factor with constant probability. Putting the pieces together we are able to obtain an efficient 2/3-accurate one-way communication protocol for α-DTW using $O(n/\alpha \log \alpha \log^3 n)$ bits. Formally, we arrive at the following theorem:

▶ **Theorem 8** (Main Upper Bound). *Let Σ be a metric space of size and aspect ratio polynomial in n. Then there is an efficient 2/3-accurate one-way communication protocol for α-DTW over Σ with space complexity $O\left(n\alpha^{-1} \cdot \log \alpha \cdot \log^3 n\right)$ and an inefficient 2/3-accurate one-way protocol with complexity $O\big(n\alpha^{-1} \cdot \log^3 n\big)$.*

Optimizing in the Case of Natural Numbers. We can further optimize the logarithmic factors in our upper bound when the underlying alphabet Σ is, for example, the natural numbers and $d(a, b) = |a - b|$. We handle the case $\mathrm{DTW}(x, y) \leq n/\alpha$ as before. However, for larger values of $\mathrm{DTW}(x, y)$, we take a different approach.

We first explain the case of distinguishing $\mathrm{DTW}(x, y) \leq n$ versus $\mathrm{DTW}(x, y) > \alpha n$. The idea is to impose a randomly shifted grid of side length $\alpha/4$, and to round each point in x and y down to the nearest smaller grid point, resulting in strings x' and y'. Define a *short edge* in a correspondence to be an edge of cost at most $\alpha/4$, and otherwise call the edge a *long edge*. We assume w.l.o.g. that any correspondence has length at most $2n$.

Suppose first $\mathrm{DTW}(x, y) \leq n$, and consider an optimal correspondence. We will show that the effect of rounding is such that with probability at least 2/3, $\mathrm{DTW}(x', y') \leq O(n)$. First we consider what effect rounding has on the short edges. The expected number of short edges with endpoints that get rounded to different grid points is at most

$$\sum_{\text{short edge length } l} \frac{l}{\alpha/4} \leq \frac{4\,\mathrm{DTW}(x, y)}{\alpha}.$$

Each such edge has its length increased by at most $\alpha/4$ after rounding, and so the expected contribution of short edges to the correspondence after rounding is at most $O(\mathrm{DTW}(x, y))$. Since each long edge has its length increase by at most an additive $\alpha/4$, and its original length is at least $\alpha/4$, its contribution changes by at most a constant factor, so the total contribution of long edges after rounding is $O(\mathrm{DTW}(x, y))$. Hence, when $\mathrm{DTW}(x, y) \leq n$, with probability at least 2/3 after rounding, we have $\mathrm{DTW}(x', y') = O(n)$.

Next suppose $\mathrm{DTW}(x, y) > n\alpha$, and consider any correspondence. The total change in the cost of the correspondence that can result from the rounding procedure is at most $2n \cdot \alpha/4$, since there are at most $2n$ edges in total. Consequently the effect of rounding is such that $\mathrm{DTW}(x', y') > n\alpha/2$.

It follows that when comparing the cases of $\mathrm{DTW}(x, y) \leq n$ and $\mathrm{DTW}(x, y) > n\alpha$, there is an $\Omega(\alpha)$-factor gap between $\mathrm{DTW}(x', y')$ in the two cases. Further, after rounding to grid points, all non-equal points have distance at least $\alpha/4$, and so if $\mathrm{DTW}(x', y') \leq n$, then there is a correspondence on which they differ in at most $O(n/\alpha)$ positions. Thus our protocol for bounded α-DTW can be applied to distinguish between the two cases. A similar approach can be used to distinguish between $\mathrm{DTW}(x, y) \leq rn/\alpha$ and $\mathrm{DTW}(x, y) > rn$ in general, and this can then be used to solve α-DTW similarly as for 2-hierarchically well-separated tree metrics above. We save roughly a $\log n$ factor here because we do not incur the $\log n$ factor distortion of embedding an arbitrary metric into a tree metric.

We remark that our algorithm in the 1-dimensional natural number case uses a similar grid snapping as used in [15] for their nearest neighbor search algorithm for Frechét distance. Recently, Bringmann (personal communication) obtained a sketch for Frechét distance which builds upon the ideas in [15] and uses $O(n/\alpha)$ bits. To the best of our knowledge, these techniques do not yield nontrivial results for Dynamic Time Warping, however.

A Unified Approach. To unify the argument for 2-hierarchically well-separated tree metrics and the natural numbers, we recall the definition of a σ-separable metric space. A δ-bounded partition of a metric space (Σ, d) is a partition such that the diameter of each part is at most δ. A distribution over partitions is then called σ-separating if for all $x, y \in \Sigma$, the probability that x and y occur in different parts of the partition is at most $\sigma \cdot d(x, y)/\delta$. We say Σ is σ-separable if for every $\delta > 0$, there exists a σ-separating probability distribution over δ-bounded partitions of Σ. One can also define an efficient notion of this, whereby the distribution over partitions is efficiently sampleable.

By adapting our argument for the natural numbers to σ-separable metrics of polynomial size and aspect ratio, we obtain an efficient 2/3-accurate protocol for α-DTW with bit complexity $O(\sigma n/\alpha \log^3 n \log\log\log n)$, where the $\log\log\log n$ comes from minor technical subtitles. For general metrics, it is known that $\sigma = O(\log n)$, while for the natural numbers, $\sigma = O(1)$. Consequently, our result for σ-separable metrics captures both the result obtained using HSTs (up to a factor of $\log\log\log n$) as well as the optimization for the natural numbers. Moreover, the theorem allows for space savings over many additional metrics, such as low-dimensional integer vectors equipped with ℓ_p-norms, metrics with bounded doubling dimension, etc., all of which have $\sigma \ll O(\log n)$ [13, 30, 31]. The general result we arrive at is captured formally in the following theorem:

▶ **Theorem 9** (Extended Main Upper Bound). *Let (Σ, d) be a metric space of size and aspect ratio* $\mathrm{poly}(n)$. *Suppose that (Σ, d) is efficiently σ-separable for some $1 \leq \sigma \leq O(\log n)$. Then there is an efficient 2/3-accurate one-way communication protocol for α-DTW$(\Sigma^{\leq n})$ with space complexity $O\left(\sigma n \alpha^{-1} \cdot \log \alpha \cdot \log^2 n \cdot \log\log\log n\right)$ and an inefficient 2/3-accurate one-way protocol with space complexity $O\big(\sigma n \alpha^{-1} \cdot \log^2 n \cdot \log\log\log n\big)$.*

The proof closely follows that for the natural numbers, where instead of our randomly shifted grid, we use a random δ-bounded partition. If we are trying to distinguish $\mathrm{DTW}(x, y) \leq nr/\alpha$ versus $\mathrm{DTW}(x, y) > nr$, then we set $\delta = \Theta(r)$. Just like for the grid, where we "snapped" points to their nearest grid point, we now snap points to a representative point in each part of the partition, obtaining two new sequences $\tilde{x}$ and $\tilde{y}$. By using shared randomness, the representative in each part can be agreed upon without any communication. Just like in the grid case, we show that if $\mathrm{DTW}(x, y) \leq nr/\alpha$, then for the optimal correspondence, in expectation its cost increases only by a constant factor after snapping. On the other hand, if $\mathrm{DTW}(x, y) > nr$, then we show that for every correspondence, its cost decreases only by a constant factor. The key difference is that now the expected number of short edges with endpoints occurring in different parts of the partition is at most $\frac{\sigma \cdot \mathrm{DTW}(x,y)}{\delta}$.

Complexity Lower Bounds

The simplest of our lower bounds comes from a reduction from a randomized 1-way communication lower bound for indexing over large alphabets [22]. In this problem, Alice is given a string s in $\mathcal{U}^r$ for some universe $\mathcal{U}$ and length parameter r, and Bob is given a character $a \in \mathcal{U}$ and an index $j \in [r]$. The goal is for Bob to decide if $s_j = a$ with probability at least $1 - 1/|\mathcal{U}|$. It is known if Alice sends a single message to Bob, then there is an $\Omega(r \log_2 |\mathcal{U}|)$ lower bound. By reducing this large-alphabet indexing problem to α-DTW when $r = n/\alpha$. To perform the reduction, Alice's input string $s = s_1, \ldots, s_{n/\alpha} \in \mathcal{U}^{n/\alpha}$ is mapped to the string $x = (s_1, 1), (s_2, 2), \ldots, (s_{n/\alpha}, n/\alpha)$. Bob's inputs of $a \in \mathcal{U}$ and $j \in [r]$ are mapped to an input string $y = (a, j), (a, j), \ldots$ in which the character (a, j) is repeated n times. If $s_j = a$, then $\mathrm{DTW}(x, y) = n/\alpha - 1$ (due to the $n/\alpha - 1$ characters of x that do not get matched with an equal-value letter); otherwise $\mathrm{DTW}(x, y) \geq n$ (due to the fact that none of the letters in y can be correctly matched). This gives a reduction to α-DTW as desired. Using this we have an $\Omega(n/\alpha \cdot \log n)$ lower bound for $(1 - 1/n)$-accurate α-DTW, provided the alphabet size $|\Sigma|$ is say, at least n^2. Thus we have the following theorem:

▶ **Theorem 10** (Tight Bound Over Hamming Space). *Consider $1 \leq \alpha \leq n$, and consider the generalized Hamming distance over a point-set Σ with Σ of polynomial size $n^{1+\Omega(1)}$. For $p \geq 1 - 1/|\Sigma|^{-1}$, $\mathrm{CC}_p(\alpha\text{-}DTW(\Sigma^{\leq n}) = \Theta[n\alpha^{-1} \cdot \log n]$.*

In order to obtain a nearly tight lower bound for arbitrary finite metric spaces, we construct a more intricate lower bound of $\Omega(n/\alpha)$ which holds whenever $|\Sigma| \geq 3$. For convenience, we describe the argument for the case of $\Sigma = \{0, 1, 2\}$ below. The lower bound is achieved via a reduction from the Index problem in which Alice has $s \in \{0, 1\}^t$, Bob has an $i \in [t]$, and Bob would like to output s_i with probability at least $2/3$. The randomized 1-way communication complexity of this problem is $\Omega(t)$. We instantiate $t = \Theta(n/\alpha)$. For each s_j, if $s_j = 1$, Alice creates a string $Z(1)$ of length 3α consisting of α 0s, followed by α 1s, followed by α 2s; and if $s_j = 0$, Alice creates a string $Z(0)$ of length $2\alpha + 1$ consisting of α 0s, followed by a single 1, followed by α 2s. She then concatenates $Z(s_1), Z(s_2), \ldots, Z(s_t)$ into a single string x of length n. Bob, who is given an index $i \in [t]$, creates the string $y = (012)^{i-1}(02)(012)^{t-i}$; that is, we have the length-3 string 012 repeated $i - 1$ times, then the string 02, followed by the string 012 repeated $t - i$ times. (We call each piece of the form (012) and (01) a *block*.) Notice that if $s_i = 0$, then $\mathrm{DTW}(x, y) = 1$, since the single 1 in $Z(s_i)$ can match to either the 0 or 2 in the (02) block of Bob's string y. On the other hand, if $s_i = 1$, the entire run of α 1s in $Z(s_i)$ has to appear somewhere in the correspondence and cannot match to the 0 or 2 in the i-th piece of Bob's string, without incurring a cost of α. So these α 1s must either "travel" to blocks $j > i$ in y or blocks $j < i$ in y. Suppose, without loss of generality, most of these α 1s are matched to a block $j > i$. This has a ripple effect, since it causes the α 2s in the i-th block to also have to travel to a block $j > i$. While this is possible, it then means the α 0s in the $(i + 1)$-st block must travel to a block even larger than j, etc. Eventually, we run out of blocks to match the elements in Alice's string to since there are $t - i$ blocks in her string that need to be matched to fewer than $t - i$ blocks in Bob's string. This ultimately forces $\mathrm{DTW}(x, y) \geq \alpha$, completing the reduction from the Index problem to α-DTW. The extension of this argument to arbitrary Σ establishes that our upper bound for general metric spaces is optimal up to a polylogarithmic factor:

▶ **Theorem 11** (General Lower Bound). *Let $\Sigma = \{a, b, c\}$ be three letters with a two-point distance function $d : \Sigma \times \Sigma \to \mathbb{R}_+$, not necessarily satisfying the triangle inequality. Consider $1 \leq \alpha \leq n$. Then $CC_{0.1}\left[\mathrm{DTEP}_r^\alpha(\Sigma^{\leq n})\right] = \Omega(n/\alpha)$.*

Our communication protocols are non-linear, and we conclude our lower bounds by observing that linear sketches must have size $\Omega(n)$. (See the extended paper [36].)

▶ **Theorem 12** (Linear Sketching Lower Bound). *Consider $1 \leq \alpha \leq n$. Then any 0.1-error linear sketch for α-DTW on $\{0,1,2\}^{4n}$ has space complexity $\Omega(n)$.*

4 The bounded αDTW problem

Recall that in the Bounded α-DTW problem, the goal for Bob is: If $\mathrm{DTW}_0(x,y) \leq n/\alpha$, solve α-DTW on (x,y); and if $\mathrm{DTW}_0(x,y) > n/\alpha$, either solve α-DTW on (x,y), or return "Fail". In this section, we formally present the protocol for bounded α-DTW. We then use this to give tight bounds on the one-way communication complexity of α-DTW over generalized Hamming Space. The protocols in this section are constructive in that, in addition to estimating $\mathrm{DTW}(x,y)$, Bob is also able to build a correspondence between x and y. The protocol for Bounded α-DTW forms the core for the communication protocol over arbitrary metrics. The full protocol over arbitrary metrics is given in the extended paper [36].

In order to design an efficient one-way communication scheme for bounded α-DTW, we will use what we refer to as the *K-document exchange* problem as a primitive. Here, Alice and Bob are given strings x and $y \in \Sigma^n$. The goal for Bob is: If $\mathrm{ed}(x,y) \leq K$, recover the string x; and if $\mathrm{ed}(x,y) > K$, either recover x or return "Fail".

The K-document exchange problem has been studied extensively [32, 23, 8, 12, 21]. The one-way communication protocol of [21] efficiently solves K-document exchange using $O(K \log(n/K) \cdot \log n)$ bits with high probability. This can be slightly improved at the cost of being no longer time-efficient using the protocol of [32], which achieves accuracy $1-\delta$ for any $\delta \in (0,1)$ by having Alice simply hash her string to a $\Theta(K \cdot \log n + \log \delta^{-1})$-bits.

The K-document exchange problem concerns edit distance rather than DTW. Nonetheless, in designing a sketch for DTW, the K-document exchange problem will prove useful due to a convenient relationship between edit distance and DTW over generalized Hamming space.

▶ **Lemma 13** (DTW_0 Approx. Edit Dist.). *Let x, y be strings of length at most n with letters from any metric space, and suppose that neither string contains any runs of length greater than one. Then $\mathrm{DTW}_0(x,y) \leq \mathrm{ed}(x,y) \leq 3\,\mathrm{DTW}_0(x,y)$.*

Proof. We first show that $\mathrm{DTW}_0(x,y) \leq \mathrm{ed}(x,y)$. A sequence of edits from x to y can be viewed as consisting of insertions in each of x and y, as well as substitutions. One can create expansions $\overline{x}$ and $\overline{y}$ of x and y, respectively, by extending runs by one in each place where the sequence of edits would have performed an insertion. The Hamming distance between $\overline{x}$ and $\overline{y}$ is then at most the length of the sequence of edits. Hence $\mathrm{DTW}_0(x,y) \leq \mathrm{ed}(x,y)$.

Next we show that $\mathrm{ed}(x,y) \leq 3\,\mathrm{DTW}_0(x,y)$. Consider an optimal correspondence $(\overline{x}, \overline{y})$ between x and y. Without loss of generality, we may assume that whenever two runs in $\overline{x}$ and $\overline{y}$ overlap, at least one of them has length only one. (Indeed, otherwise both runs could have been reduced in size by one at no cost to DTW.) Therefore, any run of length k in $\overline{x}$ must overlap k distinct runs in $\overline{y}$, and thus must incur at least $(k-1)/2$ Hamming differences. On the other hand, because the run is length k, the expansion of the run can be simulated by $k-1$ insertions. Therefore, $\overline{x}$ and $\overline{y}$ can be constructed from x and y through at most $2\,\mathrm{DTW}_0(x,y)$ edits. Hence, $\mathrm{ed}(x,y) \leq 3\,\mathrm{DTW}_0(x,y)$. ◀

We now present an efficient one-way communication scheme for bounded α-DTW.

▶ **Proposition 14** (Protocol for Bounded DTW). *Consider* DTW *over a metric space* Σ *of polynomial size. Then for* $p = 1 - \mathrm{poly}(n^{-1})$*, there is an efficient* p*-accurate one-way communication protocol for bounded* α*-DTW over* $\Sigma^{\leq n}$ *which uses* $O\left(n\alpha^{-1} \cdot \log \alpha \cdot \log n\right)$ *bits. Moreover, for any* $\delta \in (0,1)$*, there is an inefficient* $(1-\delta)$*-accurate protocol for bounded* α*-DTW*$(\Sigma^{\leq n})$ *using space* $O(n\alpha^{-1} \cdot \log n + \log \delta^{-1})$ *for any* $\delta \in (0,1)$*.*

Proof. We assume without loss of generality that α and n/α are integers. Let $x \in \Sigma^{\leq n}$ be a string given to Alice, and $y \in \Sigma^{\leq n}$ be a string given to Bob. Alice can construct a string x' by taking each run in x which is of length less than α and reducing its length to one. Notice that $\mathrm{DTW}(x', y) \leq \mathrm{DTW}(x, y)$ trivially and that $\mathrm{DTW}(x, y) < \alpha\, \mathrm{DTW}(x', y)$ because any correspondence between x' and y can be turned into a correspondence between x and y by duplicating every letter in the original correspondence $\alpha - 1$ times. Thus if Alice could communicate x' to Bob, then Bob could solve α-DTW.

In an attempt to communicate x' to Bob, Alice constructs a list L consisting of the pairs (i, l_i) for which the i-th run in x is of length $l_i \geq \alpha$. Alice then sends L to Bob. Notice that $|L| \leq n/\alpha$, and thus can be communicated with $O(\frac{n}{\alpha} \log n)$ bits. Moreover, if Alice defines x'' to be x except with every run reduced to length one, then x' can be recovered from x'' and L. Therefore, if Alice could further communicate x'' to Bob, then Bob could solve α-DTW.

In an attempt to communicate x'' to Bob, Alice invokes the one-way communication protocol of [21] for the $3n/\alpha$-document exchange problem. She sends Bob the resulting sketch s of size $O(n/\alpha \cdot \log \alpha \log n)$ bits which is correct with probability at least p. Bob defines y'' to be y with each run reduced to length one and uses the sketch s along with y'' in order to try to recover x''. If Bob is able to use s to recover a value for x'', then he can correctly solve α-DTW with high probability. If Bob is unable to use s to recover a value for x'', then Bob may conclude with high probability that $\mathrm{ed}(x'', y'') > 3n/\alpha$. Because $\mathrm{ed}(x'', y'') \leq 3\,\mathrm{DTW}_0(x'', y'')$ by Lemma 13 and because $\mathrm{DTW}_0(x'', y'') \leq \mathrm{DTW}_0(x, y)$, we have that $n/\alpha < \mathrm{DTW}_0(x, y)$. It follows that in this case Bob can correctly return "Fail".

Rather than using the efficient one-way communication protocol of [21], Alice could instead invoke the protocol of [32] in which she sends Bob a hash of x'' using $O(n\alpha^{-1} \cdot \log n + \log \delta^{-1})$ and Bob is able to then inefficiently recover x'' correctly with probability at least $1 - \delta$. This gives an inefficient $(1-\delta)$-accurate protocol which uses $O(n\alpha^{-1} \cdot \log n + \log \delta^{-1})$ bits. ◀

We now prove a tight communication bound for α-DTW over generalized Hamming space.

▶ **Theorem 15** (Theorem 10 Restated). *Consider* $1 \leq \alpha \leq n$*, and consider the generalized Hamming distance over a point-set* Σ *with* Σ *of polynomial size* $n^{1+\Omega(1)}$*. For* $p \geq 1 - 1/|\Sigma|^{-1}$*, the* p*-accurate one-way communication complexity of* α*-DTW*$(\Sigma^{\leq n})$ *is* $\Theta[n\alpha^{-1} \cdot \log n]$*.*

Proposition 14 implies the desired upper bound (via the inefficient protocol). In order to prove Theorem 15, it therefore suffices to prove the lower bound. To do this, we first introduce a problem with high one-way communication complexity.

▶ **Lemma 16** ([22], Theorem 3.1). *Define* $(n, \mathcal{S})$*-SET as follows. Alice gets an* n*-element set* $S \subseteq \mathcal{S}$ *and Bob gets a character* $a \in \mathcal{S}$*. The goal is for Bob to determine whether* $a \in S$*. Let* $p \geq 1 - \frac{1}{|\mathcal{S}|}$*. Then* $\mathrm{CC}_p((n, \mathcal{S})\text{-}SET) \geq \Omega(n \log(|S|/n))$*.*

Proof of Theorem 15. As described above, it suffices to show the lower bound. To this end, we reduce $(n/\alpha, \Sigma)$-SET to α-DTW for strings of length n. Suppose Alice is given $S \subseteq \Sigma$ of size n/α and Bob is given the character $a \in \Sigma$. Then Alice can compute x to be the concatenation of the elements of S in an arbitrary order. Alice will use the resulting string $x \in \Sigma^{\leq n}$ as an input for the α-DTW problem. Bob can then define y to be the character a repeated n times. Notice that if $a \in S$ then $\text{DTW}(x, y) = n/\alpha - 1$, whereas if $a \notin S$ then $\text{DTW}(x, y) = n$. By Lemma 16, this reduction establishes that for $p \geq 1 - \frac{1}{|\Sigma|}$ the p-accurate one-way communication complexity of α-DTW is at least $\Omega(n\alpha^{-1} \cdot \log(\alpha|\Sigma|/n)) = \Omega(n\alpha^{-1} \cdot \log n)$, since $|\Sigma| = n^{1+\Omega(1)}$. ◀

References

1 John Aach and George M Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6):495–508, 2001.

2 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 59–78. IEEE, 2015.

3 Pankaj K Agarwal, Kyle Fox, Jiangwei Pan, and Rex Ying. Approximating dynamic time warping and edit distance for a pair of point sequences. *arXiv preprint*, 2015. arXiv:1512.01876.

4 Ghazi Al-Naymat, Sanjay Chawla, and Javid Taheri. Sparsedtw: A novel approach to speed up dynamic time warping. In *Proceedings of the Eighth Australasian Data Mining Conference-Volume 101*, pages 117–127. Australian Computer Society, Inc., 2009.

5 Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David Woodruff. Efficient sketches for earth-mover distance, with applications. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 324–330. IEEE, 2009.

6 Nurjahan Begum, Liudmila Ulanova, Jun Wang, and Eamonn J. Keogh. Accelerating Dynamic Time Warping Clustering with a Novel Admissible Pruning Strategy. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 49–58, 2015.

7 D. Belazzougui and Q. Zhang. Edit Distance: Sketching, Streaming and Document Exchange. *ArXiv e-prints*, 2016. arXiv:1607.04200.

8 Djamal Belazzougui. Efficient deterministic single round document exchange for edit distance. *arXiv preprint*, 2015. arXiv:1511.09229.

9 Donald J. Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop, Seattle, Washington, July 1994. Technical Report WS-94-03*, pages 359–370, 1994.

10 Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 79–97. IEEE, 2015.

11 EG Caiani, A Porta, G Baselli, M Turiel, S Muzzupappa, F Pieruzzi, C Crema, A Malliani, and S Cerutti. Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume. In *Computers in Cardiology 1998*, pages 73–76. IEEE, 1998.

12 Diptarka Chakraborty, Elazar Goldenberg, and Michal Kouckỳ. Streaming algorithms for embedding and computing edit distance in the low distance regime. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 712–725. ACM, 2016.

13 Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha, and Serge Plotkin. Approximating a finite metric by a small number of tree metrics. In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 379–388. IEEE, 1998.

14 Alexander De Luca, Alina Hang, Frederik Brudy, Christian Lindner, and Heinrich Hussmann. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 987–996. ACM, 2012.

15 Anne Driemel and Francesco Silvestri. Locality-Sensitive Hashing of Curves. In *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, pages 37:1–37:16, 2017.

16 Ioannis Z Emiris and Ioannis Psarros. Products of Euclidean metrics and applications to proximity questions among curves. *arXiv preprint*, 2017. `arXiv:1712.06471`.

17 Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

18 Omer Gold and Micha Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. *arXiv preprint*, 2016. `arXiv:1607.05994`.

19 Daniel S. Hirschberg. A Linear Space Algorithm for Computing Maximal Common Subsequences. *Commun. ACM*, 18(6):341–343, 1975.

20 Utku Irmak, Svilen Mihaylov, and Torsten Suel. Improved single-round protocols for remote file synchronization. In *INFOCOM*, pages 1665–1676, 2005.

21 Utku Irmak, Svilen Mihaylov, and Torsten Suel. Improved single-round protocols for remote file synchronization. *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, 3:1665–1676 vol. 3, 2005.

22 Thathachar S Jayram and David P Woodruff. Optimal bounds for Johnson-Lindenstrauss transforms and streaming problems with subconstant error. *ACM Transactions on Algorithms (TALG)*, 9(3):26, 2013.

23 Hossein Jowhari. Efficient communication protocols for deciding edit distance. In *European Symposium on Algorithms*, pages 648–658. Springer, 2012.

24 Eamonn J. Keogh. Exact Indexing of Dynamic Time Warping. In *VLDB 2002, Proceedings of 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China*, pages 406–417, 2002.

25 Eamonn J. Keogh and Michael J. Pazzani. Scaling up Dynamic Time Warping to Massive Dataset. In *Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD '99, Prague, Czech Republic, September 15-18, 1999, Proceedings*, pages 1–11, 1999.

26 Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*, pages 285–289, 2000.

27 Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 596–605. ACM, 1995.

28 Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv preprint*, 2010. `arXiv:1003.4083`.

29 Mario E Munich and Pietro Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 108–115. IEEE, 1999.

30 Assaf Naor. Probabilistic clustering of high dimensional norms. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 690–709. SIAM, 2017.

31 Ofer Neiman. On Stochastic Decompositions of Metric Spaces.

32 Alon Orlitsky. Interactive communication: Balanced distributions, correlated files, and average-case complexity. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 228–238. IEEE, 1991.

33 François Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn J. Keogh. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowl. Inf. Syst.*, 47(1):1–26, 2016.

34 Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

35 Yasushi Sakurai, Masatoshi Yoshikawa, and Christos Faloutsos. FTW: fast similarity search under the time warping distance. In *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA*, pages 326–337, 2005.

36 Braverman Vladimir, Moses Charikar, William Kuszmaul, David P. Woodruff, and Liu F. Yang. The One-Way Communication Complexity of Dynamic time Warping Distance. *arXiv preprint*, 2019. `arXiv:1903.03520`.

37 Rex Ying, Jiangwei Pan, Kyle Fox, and Pankaj K Agarwal. A simple efficient approximation algorithm for dynamic time warping. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 21. ACM, 2016.

38 Yunyue Zhu and Dennis Shasha. Warping indexes with envelope transforms for query by humming. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 181–192. ACM, 2003.