Approximating Edit Distance in Near-Linear Time

Alexandr Andoni
MIT
andoni@mit.edu

Krzysztof Onak*
MIT
konak@mit.edu

ABSTRACT

We show how to compute the edit distance between two strings of length n up to a factor of $2^{\tilde{O}(\sqrt{\log n})}$ in $n^{1+o(1)}$ time. This is the first sub-polynomial approximation algorithm for this problem that runs in near-linear time, improving on the state-of-the-art $n^{1/3+o(1)}$ approximation. Previously, approximation of $2^{\tilde{O}(\sqrt{\log n})}$ was known only for *embedding* edit distance into ℓ_1, and it is not known if that embedding can be computed in less than a quadratic time.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms

Algorithms

Keywords

approximation algorithms, edit distance, metric embeddings

1. INTRODUCTION

The *edit distance* (or *Levenshtein distance*) between two strings is the number of insertions, deletions, and substitutions needed to transform one string into the other [17]. This distance is of fundamental importance in several fields such as computational biology and text processing/searching, and consequently, problems involving edit distance were studied extensively (cf. [21], [11], and references therein). In computational biology, for instance, edit distance and its slight variants are the most elementary measures of dissimilarity for, say, genomic data, and thus improvements on edit distance algorithms have the potential of major impact.

The basic problem is to compute the edit distance between two strings of length n over some alphabet. The text-book dynamic programming runs in $O(n^2)$ time (cf. [8] and references therein). This was only slightly improved by Masek and Paterson [18] to $O(n^2/\log^2 n)$ time for constant-size alphabets. Their result from 1980 remains the best algorithm to this date.

Since near-quadratic time is too costly when working on large datasets, practitioners tend to rely on faster heuristics (cf. [11], [21]). This leads to the question of finding fast algorithms with provable guarantees, specifically: can one *approximate* the edit distance between two strings in near-linear time [12, 3, 2, 4, 10, 9, 22, 14, 15]?

Prior results on approximate algorithms[1]**.** A $\sqrt{n}$-approximation algorithm that runs in linear time immediately follows from the $O(n + d^2)$-time exact algorithm of Myers [20], where d is the edit distance between the input strings. Subsequent research improved the approximation first to $O(n^{3/7})$, and then to $O(n^{1/3+o(1)})$, due to, respectively, Bar-Yossef, Jayram, Krauthgamer, and Kumar [2], and Batu, Ergün, and Sahinalp [4].

A *sublinear* time algorithm was obtained by Batu, Ergün, Kilian, Magen, Raskhodnikova, Rubinfeld, and Sami [3]. Their algorithm distinguishes the cases when the distance is $O(n^{1-\epsilon})$ vs. $\Omega(n)$ in time[2] $\tilde{O}(n^{1-2\epsilon}+n^{(1-\epsilon)/2})$ for any $\epsilon > 0$. Note that their algorithm cannot distinguish distances, say, $O(n^{0.1})$ vs. $\Omega(n^{0.9})$.

On a related front, in 2005, the breakthrough result of Ostrovsky and Rabani gave an *embedding* of the edit distance metric into ℓ_1 with $2^{\tilde{O}(\sqrt{\log n})}$ distortion [22]. This result vastly improved related applications, namely nearest neighbor search and sketching. However it did not have implications for the original problem of computing edit distance in sub-quadratic time. In particular, to the best of our knowledge it is not known whether it is possible to compute their embedding in less than a quadratic time.

The best approximation to this date remains the 2006 result of Batu, Ergün, and Sahinalp [4], achieving $n^{1/3+o(1)}$ approximation. Even for $n^{2-\epsilon}$ time, their approximation is $n^{\epsilon/3+o(1)}$.

Our result. We obtain approximation $2^{\tilde{O}(\sqrt{\log n})}$ in near-linear time. This is the first sub-polynomial approximation algorithm for computing the edit distance between two strings running in strongly subquadratic time.

*Supported by a Symantec research fellowship, NSF grant 0728645, and NSF grant 0732334.

STOC'09, May 31–June 2, 2009, Bethesda, Maryland, USA.

[1]We make no attempt at presenting a complete list of results for restricted problems, such as average case edit distance, weakly-repetitive strings, bounded distance regime, or related problems, such as pattern matching/nearest neighbor, sketching. However, for a very thorough survey, if only slightly outdated, see [21].

[2]We use $\tilde{O}(f(n))$ to denote $f(n) \cdot \log^{O(1)} f(n)$.

THEOREM 1.1. *We can compute the edit distance between two strings $x, y \in \{0,1\}^n$ up to a factor of $2^{\tilde{O}(\sqrt{\log n})}$ in $n \cdot 2^{\tilde{O}(\sqrt{\log n})}$ time.*

Our result immediately extends to a sublinear-time algorithm as well. In this scenario, the goal is to compute the distance between two strings x, y of the same length n in $o(n)$ time. For this problem, for any $\alpha < \beta \leq 1$, we can distinguish distance $O(n^\alpha)$ from distance $\Omega(n^\beta)$ in $O(n^{\alpha+2(1-\beta)+o(1)})$ time. We describe this application in Appendix A.

Before describing our general approach and the techniques used, we first introduce a few definitions. Readers familiar with Earth-Mover Distance (EMD), product spaces (specifically min-product spaces), tree/graph metrics, and the difference between oblivious and non-oblivious embeddings may skip the next section.

1.1 Preliminaries and Notation

We write $\mathrm{ed}(x, y)$ to denote the edit distance between strings x and y. We use the notation $[n] = \{1, 2, 3, \dots n\}$. For a string x, a substring starting at i, of length m, is denoted $x[i : i+m-1]$. Whenever we say *with high probability* (w.h.p.) throughout the paper, we mean "with probability $1 - 1/p(n)$", where $p(n)$ is a sufficiently large polynomial function of the input size.

Embeddings. For a metric (M, d_M), and another metric (X, ρ), an embedding $\phi : M \to X$ has distortion $\gamma \geq 1$ if, for any $x, y \in M$, we have $d_M(\phi(x), \phi(y)) \leq \rho(x, y) \leq \gamma \cdot d_M(\phi(x), \phi(y))$.

Embedding ϕ is *oblivious* if it is randomized and, for any subset $S \subset M$ of size n, the distortion guarantee holds for all pairs $x, y \in S$ with high probability. The embedding ϕ is *non-oblivious* if it holds for a specific set S (i.e., ϕ is allowed to depend on S).

Metrics. We define *thresholded Earth-Mover Distance*, denoted TEMD_t for a fixed threshold $t > 0$, as the following distance on subsets A and B of size $s \in \mathbb{N}$ of some metric $\mathcal{M} = (M, d_M)$:

$$\mathrm{TEMD}_t(A, B) = \tfrac{1}{s} \min_{\tau : A \to B} \sum_{a \in A} \min\left\{d_M(a, \tau(a)), t\right\} \qquad (1)$$

where τ ranges over all bijections between sets A and B. TEMD_∞ is the simple Earth-Mover Distance (EMD). We will always use $t = s$ and thus drop the subscript t; i.e., $\mathrm{TEMD} = \mathrm{TEMD}_s$.

A *graph (tree) metric* is a metric induced by a connected weighted graph (tree) G, where the distance between two vertices is the length of the shortest path between them. We denote by TM an arbitrary tree metric.

Semimetric spaces. We define a semimetric to be a pair (M, d_M) that satisfies all the properties of a metric space except the triangle inequality. A γ-*near metric* is a semimetric (M, d_M) such that there exists some metric (M, d^*_M) (satisfying the triangle inequality) with the property that, for any $x, y \in M$, we have that $d^*_M(x, y) \leq d_M(x, y) \leq \gamma \cdot d^*_M(x, y)$.

Product spaces. A *sum-product over a metric* $\mathcal{M} = (M, d_M)$, denoted $\bigoplus^k_{\ell_1} \mathcal{M}$, is a derived metric over the set M^k, where the distance between two points $x = (x_1, \dots x_k)$ and $y = (y_1, \dots y_k)$ is equal to

$$d_{1,M}(x, y) = \sum_{i \in [k]} d_M(x_i, y_i).$$

Analogously, a *min-product over* $\mathcal{M} = (M, d_M)$, denoted $\bigoplus^k_{\min} \mathcal{M}$, is a semimetric over M^k, where the distance between two points $x = (x_1, \dots x_k)$ and $y = (y_1, \dots y_k)$ is

$$d_{\min,M}(x, y) = \min_{i \in [k]} \left\{d_M(x_i, y_i)\right\}.$$

We also slightly abuse the notation by writing $\bigoplus^k_{\min} \mathrm{TM}$ to denote the min-product of k tree metrics (that could differ from each other).

1.2 Techniques

Our starting point is the Ostrovsky-Rabani embedding. For strings x, y, as well as for all substrings σ of specific lengths, we compute vectors v_σ in low-dimensional ℓ_1, such that the distance between two such vectors approximates the edit distance between the associated (sub-)strings. In this respect, these vectors can be seen as an embedding of the considered strings into ℓ_1 of *polylogarithmic dimension*. Unlike the Ostrovsky-Rabani embedding, however, our embedding is *non-oblivious*, in the sense that the vectors v_σ are computed given all the relevant strings σ. In contrast, Ostrovsky and Rabani give an *oblivious* embedding $\phi_n : \{0,1\}^n \to \ell_1$ such that $\|\phi_n(x) - \phi_n(y)\|_1$ approximates $\mathrm{ed}(x, y)$. However, the obliviousness comes at a high price: their embedding requires a high dimension, of order $\Omega(n)$, and a high computation time, of order $\Omega(n^2)$ (even when allowing randomized embedding, and a high probability of a correct answer). We further note that reducing the dimension of this embedding seems unlikely as suggested by the results on impossibility of dimensionality reduction within ℓ_1 [7, 6, 16]. Nevertheless, perhaps not surprisingly, we reuse the general recursive approach of the Ostrovsky-Rabani embedding.

The heart of our algorithm is a near-linear time algorithm that, given a sequence of low-dimensional vectors $v_1, \dots v_n \in \ell_1$, and an integer $s < n$, constructs new vectors $q_1, \dots q_m \in \ell_1^{O(\log^2 n)}$, where $m = n - s + 1$, with the following property. For all $i, j \in [m]$, the value $\|q_i - q_j\|_1$ approximates the Earth-Mover Distance (EMD)[3] between the sets $A_i = \{v_i, v_{i+1}, \dots v_{i+s-1}\}$ and $A_j = \{v_j, v_{j+1}, \dots v_{j+s-1}\}$. To accomplish this (non-oblivious) embedding, we proceed in two stages. First, we embed (obliviously) the EMD distance into a *min-product of ℓ_1's* of low dimension. In other words, for a set A, we associate a matrix $L(A)$ such that the EMD distance between sets A and B is approximated by $\min_r \sum_t |L(A)_{rt} - L(B)_{rt}|$. Min-products help us simultaneously on two fronts: one is that we can apply a *weak* dimensionality reduction in ℓ_1, using the Cauchy projections, and the second one enables us to accomplish a low-dimensional EMD embedding itself (the latter reason turns out to be the most important). Our embedding $L(\cdot)$ is not only low-dimensional, but it is also *linear*, allowing us to compute vectors $L(A_i)$ in near-linear time by performing one pass over the sequence $v_1, \dots v_n$. Linearity is crucial here as even the total size of A_i's is $\sum_i |A_i| = (n - s + 1) \cdot s$, which can be as high as $\Omega(n^2)$, and so processing each A_i separately is infeasible.

In the second stage, we show how to embed a set of n points lying in a low-dimensional min-product of ℓ_1's back into a low-dimensional ℓ_1 with only a small distortion. We note that this is not possible in general, with any distortion,

[3] In fact, our algorithm does this for thresholded EMD, TEMD, but the technique is precisely the same.

as our points do not even form a metric. We show that this is possible when we assume that the semi-metric induced by the set of points actually approximates some metric (in our case, the min-product approximates some EMD metric). The embedding from this stage starts by embedding a min-product of ℓ_1's into a low-dimensional min-product of tree metrics. We further embed the latter into a n-point metric supported by a shortest-path metric of a *sparse* graph. Finally, we observe that we can implement Bourgain's embedding on a sparse graph metric in *near-linear time*. These last two steps make our embedding non-oblivious.

2. SHORT OVERVIEW OF THE OSTROVSKY-RABANI EMBEDDING

We now briefly describe the embedding of Ostrovsky and Rabani [22]. Some notions introduced here are used in the next section.

The embedding of Ostrovsky and Rabani is recursive. For a fixed m, they construct the embedding of edit distance over strings of length m using the embedding of edit distance over strings of shorter length $l \le m/2^{\sqrt{\log m}}$. It is readily seen that the number of recursion levels is $O(\sqrt{\log m}\log\log m)$. We denote their embedding of length-m strings by $\phi_m : \{0,1\}^m \to \ell_1$, and let d_m^{OR} be the resulting distance: $d_m^{\mathrm{OR}}(x,y) = \|\phi_m(x) - \phi_m(y)\|_1$. For two strings $x, y \in \{0,1\}^m$, the embedding is such that $d_m^{\mathrm{OR}} = \|\phi_m(x) - \phi_m(y)\|_1$ approximates an "idealized" distance $d_m^*(x,y)$, which itself approximates the edit distance between x and y.

Before describing the "idealized" distance d_m^*, we introduce some notation. Partition x into $b = 2^{\sqrt{\log m}}$ blocks called $x^{(1)}, \dots x^{(b)}$ of length $l = m/b$. Next, fix $j \in [b]$ and $s \le l$. Consider the set of all substrings of $x^{(j)}$ of length $l - s + 1$, embed each recursively, and let $S_j^s(x) \subset \ell_1$ be the set of resulting vectors (note that $|S_i^s| = s$). Formally,

$$S_j^s(x) = \left\{\phi_{l-s+1}(x[(j-1)l+z : (j-1)l+z+l-s]) \mid z \in [s]\right\}.$$

Taking ϕ_{l-s+1} as given (and thus also the sets $S_j^s(x)$ for all x), define the new "idealized" distance d_m^* approximating the edit distance between strings $x, y \in \{0,1\}^m$ as

$$d_m^*(x,y) = \sum_{j=1}^{b} \sum_{\substack{f \in \mathbb{N} \\ s = 2^f \le l}} \mathrm{TEMD}(S_j^s(x), S_j^s(y)) \qquad (2)$$

where TEMD is the thresholded Earth-Mover Distance defined in Eqn. (1). Using the terminology from the preliminaries, the distance function d_m^* can be viewed as the distance function of the sum-product of TEMDs, i.e., $\bigoplus_{\ell_1}^{b} \bigoplus_{\ell_1}^{O(\log m)} \mathrm{TEMD}$, and the embedding into this product space is attained by the natural identity map (using sets S_j^s).

The key idea behind this distance d_m^* is that, as Ostrovsky and Rabani essentially show, as long as one can approximate d_m^* at each step up to a factor of $\log^{O(1)} m$, the final distortion is at most $2^{\tilde{O}(\sqrt{\log m})}$. Specifically, suppose that at each step one approximates d_m^* up to distortion $\alpha \ge 2$. Then, if we denote by $\tau(m)$ the distortion for strings of length up to m, the analysis of Ostrovsky and Rabani proves that the distortion satisfies the recurrence

$$\tau(m) \le (\alpha \log m)^{O(1)} \cdot \left(\tau\left(m/2^{\sqrt{\log m}}\right) + 2^{\sqrt{\log m}}\right). \qquad (3)$$

Thus, to complete a step of the recursion, it is sufficient to embed the metric $\bigoplus_{\ell_1}^{b} \bigoplus_{\ell_1}^{O(\log m)} \mathrm{TEMD}$ into ℓ_1 with a small distortion α. Indeed, Ostrovsky and Rabani show how to embed a relaxed version of TEMD into ℓ_1 with $\alpha = O(\log n)$ distortion, yielding the desired embedding, which approximates d_m^* up to $\alpha = O(\log m)$ distortion at each level. Plugging $\alpha = O(\log m)$ in Eqn. (3), they obtain $\tau(m) = 2^{\tilde{O}(\sqrt{\log m})}$. The required dimension is $\tilde{O}(m)$.

3. PROOF OF MAIN THEOREM

We now describe our general approach. Fix $x \in \{0,1\}^n$. For each substring σ of x we construct a low-dimensional vector v_σ such that, for any substrings σ, τ of the same length, the edit distance between σ and τ is approximated by the ℓ_1 distance between the vectors v_σ and v_τ. We note that the embedding is non-oblivious: to construct vectors v_σ's we need to know *all* the substrings of x in advance (akin to Bourgain's embedding guarantee). We also note that computing such vectors is enough to solve the problem of approximating the edit distance between two strings, x and y. Specifically, we apply this procedure to the string $x' = x \circ y$, the concatenation of x and y, and then compute the ℓ_1 distance between the vectors corresponding to x and y, substrings of x'.

More precisely, for each length $m \in W$, for some set $W \subset [n]$ specified later, and for each substring $x[i : i+m-1]$, where $i = 1, \dots n-m+1$, we compute a vector $v_i^{(m)} \in \mathbb{R}^k$, where $k = O(\log^2 n)$. The construction is inductive: to compute vectors $v_i^{(m)}$, we use vectors $v_i^{(l)}$ for $l \ll m$ and $l \in W$. The general approach of our construction is based on the analysis of the recursive step of Ostrovsky and Rabani. In particular, our vectors $v_i^{(m)} \in \ell_1$ will also approximate the d_m^* distance, defined in Eqn. (2) (with redefined sets S_i^s). The main new challenge is to process one level (vectors $v_i^{(m)}$ for a fixed m) in near-linear time. Besides the computation time itself, a fundamental difficulty in applying the approach of Ostrovsky and Rabani directly is that their embedding would give a much higher dimension k, proportional to $O(m)$. Thus, if we were to use their embedding, even storing all the vectors would take quadratic space.

To overcome this last difficulty, we settle on embedding non-obliviously the set of substrings $x[i : i+m-1]$ for $i \in [n-m+1]$ under the "ideal" distance d_m^* with $\log^{O(1)} n$ distortion (formally, under the distance d_m^* from Eqn. (2), when $S_j^s(x[i : i+m-1]) = \{v_{i+(j-1)l+z-1}^{(l-s+1)} \mid z \in [s]\}$ for $l = m/2^{\sqrt{\log n}}$). Existentially, we know that there exist vectors $v_i^{(m)} \in \mathbb{R}^k$, for $k = O(\log^2 n)$, such that $\|v_i^{(m)} - v_j^{(m)}\|_1$ approximate $d_m^*(x[i : i+m-1], x[j : j+m-1])$ for all i, j, by Bourgain's embedding [5]. We show that we can also compute these $v_i^{(m)}$'s efficiently for all $i \in [n-m+1]$.

The main building block is the following theorem. It shows how to approximate the TEMD distance for the desired sets S_j^s.

THEOREM 3.1. *Fix $n, M \in \mathbb{N}$ and $s \in [n]$. Suppose we have n vectors $v_1, \dots v_n$ in $\{-M, \dots M\}^\alpha$ for $\alpha = O(\log^2 n)$. Define sets $A_i = \{v_i, v_{i+1}, \dots v_{i+s-1}\}$, for $i = 1, \dots n-s+1$.*

Let $k = O(\log^2 n)$. We can compute (randomized) vectors $q_i \in \ell_1^k$ for $i \in [n-s+1]$ such that for any $i, j \in [n-s+1]$, w.h.p., we have

$$\mathrm{TEMD}(A_i, A_j) \le \|q_i - q_j\|_1 \le \mathrm{TEMD}(A_i, A_j) \cdot \log^{O(1)} n.$$

Furthermore, computing all vectors q_i takes $\tilde{O}(n)$ time.

To map the statement of this theorem to the above description, we mention that, for each l from a specific set of integers, we apply the theorem to vectors $\left(v_i^{(l-s+1)}\right)_{i\in[n-l+s]}$ for each $s = 1, 2, 4, \ldots l$.

We prove Theorem 3.1 in the subsequent sections. For now, we show how it implies the main theorem, Theorem 1.1.

PROOF OF THEOREM 1.1. We start by concatenating y to the end of x; we will work with the new version of x only. Let $k = O(\log^2 n)$ and $b = 2^{\sqrt{\log n}}$. We construct vectors $v_i^{(m)} \in \mathbb{R}^k$ for $m \in W$, where $W \subset [n]$ is a carefully chosen set of size $2^{\tilde{O}(\sqrt{\log n})}$. Namely, W is a minimal set such that: $n \in W$, and, for each $i \in W$ with $i \geq b$, we have that $i/b - 2^j + 1 \in W$ for all integers $j \leq \lfloor \log i/b \rfloor$. It is easy to show by induction that the size of W is $2^{O(\sqrt{\log n}\log\log n)}$.

For each $m \in W$ such that $m \leq 2^{2\sqrt{\log n}}$, we set $v_i^{(m)}$ to be equal to $h_m(x[i : i + m - 1])$, where $h_m : \{0,1\}^m \to \{0,1\}^k$ is a randomly chosen function. It is readily seen that $\|v_i^{(m)} - v_j^{(m)}\|_1$ is a $2^{2\sqrt{\log n}}$ approximation to $\mathrm{ed}(x[i : i + m - 1], x[j : j + m - 1])$ for each $i, j \in [n - m + 1]$.

For $m \in W$ such that $m > 2^{2\sqrt{\log n}}$, we proceed as follows. Let $l = m/b$. First we construct vectors approximating TEMD on sets $A_i^{m,s} = \{v_{i+z}^{(l-s+1)} \mid z = 0, \ldots s - 1\}$, where $s = 1, 2, 4, 8, \ldots l$ and $i \in [n - l + s]$. In particular, for fixed $s \in [l]$ equal to a power of 2, we apply Theorem 3.1 to the set of vectors $\left(v_i^{(l-s+1)}\right)_{i\in[n-l+s]}$ obtaining vectors $\left(q_i^{(m,s)}\right)_{i\in[n-l+1]}$. Theorem 3.1 guarantees that, for each $i, j \in [n - l + 1]$, the value $\|q_i^{(m,s)} - q_j^{(m,s)}\|_1$ approximates $\mathrm{TEMD}(A_i^{m,s}, A_j^{m,s})$ up to a factor of $\log^{O(1)} n$. We can then use these vectors $q_i^{(m,s)}$ to obtain the vectors $\tilde{v}_i^{(m)} \in \mathbb{R}^{2^{O(\sqrt{\log n})}}$ that approximate the "idealized" distance d_m^* for substrings $x[i : i+m-1]$, for $i \in [n-m+1]$. Specifically, we let the vector $\tilde{v}_i^m$ be a concatenation of vectors $q_{i+(j-1)l}^{(m,s)}$ over all values of s, powers of 2 less than l, and $j \in [b]$:

$$\tilde{v}_i^{(m)} = \left(q_{i+(j-1)l}^{(m,s)}\right)_{s=2^f \leq l, f\in\mathbb{N}}^{j\in[b]}.$$

Then, the vectors $\tilde{v}_i^{(m)}$ approximate the distance d_m^*, as defined in Eqn. (2), with the sets $S_j^s(x[i : i + m - 1])$, for $i \in [n - m + 1]$ and $j \in [b]$, taken as

$$\begin{aligned} S_j^s(x[i : i + m - 1]) &= A_{i+(j-1)l}^{m,s} \\ &= \left\{v_{i+(j-1)l+z}^{(l-s+1)} \mid z = 0, \ldots s - 1\right\}. \end{aligned}$$

The vectors $\tilde{v}_i^{(m)}$ already satisfy our property: $\|\tilde{v}_i^{(m)} - \tilde{v}_j^{(m)}\|_1$ approximates the edit distance between $x[i : i+m-1]$ and $x[j : j+m-1]$ for all $i, j \in [n-m+1]$. The only reason we are not done is that $\tilde{v}_i^{(m)}$ have dimension $b\log^{O(1)} n = 2^{O(\sqrt{\log n})}$. Although it would be fine even if we just used $\tilde{v}_i^{(m)}$ recursively (and this would not change bounds, up to constants in $O(\cdot)$), we can still reduce the dimension further down to $O(\log^2 n)$ by using lemmas we develop later, namely Lemmas 3.3, 3.4, 3.6.

The algorithm finishes by outputting $\|v_1^{(n)} - v_{n+1}^{(n)}\|$, which is an approximation to the edit distance between $x[1 : n]$ and $x[n + 1 : 2n] = y$. The approximation factor of $2^{\tilde{O}(\sqrt{\log n})}$ follows from Eqn. (3) for $\alpha = \log^{O(1)} n$. The total running time is $O(|W| \cdot n \cdot b \cdot \log^{O(1)} n) = n \cdot 2^{\tilde{O}(\sqrt{\log n})}$. □

3.1 Proof of Theorem 3.1

The proof proceeds in two stages. In the first stage we show an embedding of TEMD into a low-dimensional space. Specifically, we show an (oblivious) embedding of TEMD into a *min-product* of ℓ_1's. Recall that min-product of ℓ_1, denoted $\bigoplus_{\min}^l \ell_1^k$, is a semi-metric where the distance between two l-by-k vectors $x, y \in \mathbb{R}^{l\times k}$ is $d_{\min,1}(x, y) = \min_{i\in[l]} \{\sum_{j\in[k]} |x_{i,j} - y_{i,j}|\}$. Our min-product of ℓ_1's has dimensions $l = O(\log n)$ and $k = O(\log^2 n)$. The min-product can be seen as emerging from two separate sources: one from the embedding of TEMD into ℓ_1 (of initially high-dimension), and another from a *weak* dimensionality reduction in ℓ_1, using Cauchy projections. Furthermore, our embedding, denoted λ, is linear in the sets A: $\lambda(A) = \sum_{a\in A} \lambda(\{a\})$. The linearity allows us to compute the embedding of sets A_i in a streaming fashion: the embedding of A_{i+1} is obtained from the embedding of A_i with $\log^{O(1)} n$ additional processing. This stage appears in Section 3.1.1.

In the second stage, we show that, given a set of n points in min-product of ℓ_1's, we can embed these points into low-dimensional ℓ_1 with $O(\log n)$ distortion. The time required is near-linear in n and the dimensions of the min-product of ℓ_1's. To accomplish this step, we embed the min-product of ℓ_1's into a min-product of tree metrics.

Next, we show that n points in the low-dimensional min-product of tree metrics can be embedded into a graph metric supported by a *sparse* graph. We note that this is in general not possible, with any (even non-constant) distortion. We show that this is possible when we assume that our subset of the min-product of tree metrics approximates some actual metric (in our case, the min-product approximates the TEMD metric). Finally, we observe that we can implement Bourgain's embedding in near-linear time on a sparse graph metric. This stage appears in Section 3.1.2.

We conclude with the proof of Theorem 3.1 in Section 3.1.3.

3.1.1 Embedding EMD into min-product over ℓ_1

In the next lemma, we show how to embed TEMD into a min-product of ℓ_1's of low dimension. Moreover, when the sets A_i are obtained from a sequence of vectors $v_1, \ldots v_n$, by taking $A_i = \{v_i, \ldots v_{i+s}\}$, we can compute the embedding in near-linear time.

LEMMA 3.2. *Fix $n, M \in \mathbb{N}$ and $s \in [n]$. Suppose we have n vectors $v_1, \ldots v_n$ in $\{-M, \ldots M\}^\alpha$ for $\alpha = O(\log n)$. Consider the sets $A_i = \{v_i, v_{i+1}, \ldots v_{i+s-1}\}$, for $i \in [n - s + 1]$.*

Let $k = O(\log^3 n)$. We can compute (randomized) vectors $q_i \in \ell_1^k$ for $i \in [n - s + 1]$ such that for any $i, j \in [n - s + 1]$, we have that

$$\Pr\left[\|q_i - q_j\|_1 \leq \mathrm{TEMD}(A_i, A_j) \cdot O(\log^2 n)\right] \geq 0.1$$

and $\|q_i - q_j\|_1 \geq \mathrm{TEMD}(A_i, A_j)$ w.h.p. The computation takes $\tilde{O}(n)$ time.

Thus, we can embed the TEMD *metric over sets A_i into $\bigoplus_{\min}^l \ell_1^k$, for $l = O(\log n)$, such that the distortion is $O(\log^2 n)$ w.h.p. The computation time is $\tilde{O}(n)$.*

PROOF. First we show how to embed TEMD metric over the sets A_i into ℓ_1 of dimension $h = \tilde{O}(M^\alpha)$. For this purpose, we use a slight modification of the embedding of [1] (it

can also be seen as a strengthening of the TEMD embedding of Ostrovsky and Rabani).

The embedding of [1] constructs $m = O(\log s)$ embeddings ψ_i, each of dimension $h = \tilde{O}(M^\alpha)$, and then the final embedding is just the concatenation $\psi = \psi_1 \circ \psi_2 \ldots \circ \psi_m$. For $i = 1, \ldots m$, we impose a randomly shifted grid of side-length $R_i = 2^{i-2}$. Then ψ_i has a coordinate for each cell and the value of that coordinate, for a set A, is equal to the number of points from A falling into the corresponding cell. Now, if we scale ψ up by $\Theta(\log n)$, Theorem 3.1 from [1] says that the vectors $q'_i = \psi(A_i)$ satisfy the condition that, for any $i, j \in [n - s + 1]$, we have: 1) $\mathbb{E}\left[\|q'_i - q'_j\|_1\right] \leq \mathrm{TEMD}(A_i, A_j) \cdot O(\log^2 n)$ and 2) $\|q'_i - q'_j\|_1 \geq \mathrm{TEMD}(A_i, A_j)$ w.h.p. Thus, the vectors q'_i satisfy the promised properties except they have a high dimension.

To reduce the dimension of q'_i's, we apply a weak ℓ_1 dimensionality reduction via 1-stable (Cauchy) projections. Namely, we pick a random matrix P of size $k = O(\log^3 n)$ by mh, the dimension of ψ, where each entry is distributed according to a Cauchy distribution, which has probability distribution function $f(x) = \frac{1}{\pi} \cdot \frac{1}{1+x^2}$. Now define $q_i = P \cdot q'_i \in \ell_1^k$. Standard properties of ℓ_1 dimensionality reduction guarantee that the vectors q_i satisfy the properties promised in the lemma statement, after an appropriate rescaling (cf. Theorem 5 of [13] with $\epsilon = 1/2$, $\gamma = 1/6$, and $\delta = n^{-O(1)}$).

It remains to show that we can compute the vectors q_i in $\tilde{O}(n)$ time. For this, we note that the resulting embedding $P \cdot \psi(A)$ is linear, namely $P \cdot \psi(A) = \sum_{a \in A} P \cdot \psi(\{a\})$. Thus, we can use the idea of a sliding window over the stream $v_1, \ldots v_n$ to compute $q_i = P \cdot \psi(A_i)$ iteratively. Specifically, note that

$$\begin{aligned} q_{i+1} &= P \cdot \psi(A_{i+1}) = P \cdot \psi(A_i \cup \{v_{i+s}\} \setminus \{v_i\}) \\ &= q_i + P \cdot \psi(\{v_{i+s}\}) - P \cdot \psi(\{v_i\}). \end{aligned}$$

Since we can compute $P \cdot \psi(\{v_i\})$, for any i, in $\log^{O(1)} n$ time, we conclude that the total time to compute q_i's is $O(n \cdot \log^{O(1)} n)$.

Finally, we show how we obtain an efficient embedding of TEMD into min-product of ℓ_1's.

We apply the above procedure $l = O(\log n)$ times. Let $q_i^{(z)}$ be the resulting vectors, for $i \in [n - s + 1]$ and $z \in [l]$. The embedding of a set A_i is the concatenation of the vectors $q_i^{(z)}$, namely $Q_i = (q_i^{(1)}, q_i^{(2)}, \ldots q_i^{(l)}) \in \bigoplus_{\min}^l \ell_1^k$. The Chernoff bound implies that w.h.p., for any $i, j \in [n-s+1]$, we have that

$$d_{\min,1}(Q_i, Q_j) = \min_{z \in [l]} \|q_i^z - q_j^z\| \leq \mathrm{TEMD}_s(A_i, A_j) \cdot O(\log^2 n).$$

Also, $d_{\min,1}(Q_i, Q_j) \geq \mathrm{TEMD}_s(A_i, A_j)$ trivially. Thus the vectors Q_i are an embedding of the TEMD metric on A_i's into $\bigoplus_{\min}^l \ell_1^k$ with distortion $O(\log^2 n)$ w.h.p. □

3.1.2 Embedding of min-product over ℓ_1 into low-dimensional ℓ_1

In this section, we show how, given n points $Q_1, \ldots Q_n$ in the semi-metric space $\bigoplus_{\min}^l \ell_1$, we can embed them into ℓ_1 of dimension $O(\log^2 n)$ with distortion $\log^{O(1)} n$. This embedding works under the assumption that the semi-metric on $Q_1, \ldots Q_n$ is a $\log^{O(1)} n$ approximation of some metric. We start by showing that we can embed a min-product of ℓ_1's into a min-product of tree metrics.

LEMMA 3.3. *Fix $n, M \in \mathbb{N}$ such that $M = n^{O(1)}$. Consider n vectors $v_1, \ldots v_n$ in $\bigoplus_{\min}^l \ell_1^k$, where each coordinate of each v_i lies in the interval $\{-M, \ldots, M\}$. We can embed these vectors into a min-product of $O(l \log^2 n)$ tree metrics, i.e., $\bigoplus_{\min}^{O(l \log^2 n)} \mathrm{TM}$, incurring distortion $O(\log n)$ w.h.p. The computation time is $\tilde{O}(kln)$.*

PROOF. We consider all thresholds 2^t, for $t \in \{0, 1, \ldots, \log M\}$. For each threshold 2^t, and for each coordinate of the min-product (i.e., ℓ_1^k), we create $O(\log n)$ tree metrics. Each tree metric is independently created as follows. We again use randomly shifted grids. Specifically, we define a hash function $h : \ell_1^k \to \mathbb{Z}^k$ as

$$h(x_1, \ldots, x_k) = \left(\left\lfloor \frac{x_1 + u_1}{2^t} \right\rfloor, \left\lfloor \frac{x_2 + u_2}{2^t} \right\rfloor, \ldots, \left\lfloor \frac{x_k + u_k}{2^t} \right\rfloor \right),$$

where each u_t is chosen at random from $[0, 2^t)$. We create each tree metric so that the nodes corresponding to the points hashed by f to the same value are at distance 2^t (this creates a set of stars), and each pair of points that are hashed to different values are at distance $2M$ (we connect the roots of the stars). It is easy to prove that for two points $x, y \in \ell_1^k$, the following holds

$$1 - \frac{\|x - y\|_1}{2^t} \leq \Pr_h[h(x) = h(y)] \leq e^{-\|x-y\|_1/2^t}.$$

By the Chernoff bound, if $x, y \in \ell_1^k$ are at distance at most 2^t, they will be at distance at most 2^t in one of the tree metrics that we have created w.h.p. This proves that our embedding is unlikely to expand by more than a constant factor.

On the other hand, let v_i and v_j be two input vectors at distance greater than 2^t. The probability that they are at distance smaller than $2^t / c \log n$ in any of the $O(l \log^2 n)$ tree metrics, is at most n^{-c+1} for any $c > 0$, by union bound. □

We now show that we can embed a subset of the min-product of tree metrics into a graph metric, assuming the subset is close to a metric.

LEMMA 3.4. *Consider a semi-metric $\mathcal{M} = (X, \xi)$ of size n in $\bigoplus_{\min}^l \mathrm{TM}$ for some $l \in \mathbb{N}$, where each tree metric in the product is of size $O(n)$. Suppose $\mathcal{M}$ is a γ-near metric (i.e., it is equal to a metric up to a factor γ). Then we can embed $\mathcal{M}$ in a connected weighted graph with $O(nl)$ edges with distortion γ in $O(nl)$ time.*

PROOF. We consider l separate trees each on $O(n)$ nodes, corresponding to each of l dimensions of the min-product. We identify the nodes of trees that correspond to the same point in the min-product. The graph we obtain has at most $O(nl)$ edges. Denote the shortest-path metric it spans by $\mathcal{M}' = (V, \rho)$, and denote our embedding by $\phi : X \to V$. Clearly, for each pair u, v of points in M, we have $\rho(\phi(u), \phi(v)) \leq \xi(u, v)$. If the distance between two points shrinks after embedding, there is a sequence of points $w_0 = u, w_1, \ldots, w_{k-1}, w_k = v$ such that $\rho(\phi(u), \phi(v)) = \xi(w_0, w_1) + \xi(w_1, w_2) + \cdots + \xi(w_{k-1}, w_k)$. Because $\mathcal{M}$ is a γ-near metric, there exists a metric $\xi^\star : X \times X \to [0, \infty)$, such that $\xi^\star(x, y) \leq \xi(x, y) \leq \gamma \cdot \xi^\star(x, y)$, for all $x, y \in X$. Therefore,

$$\begin{aligned} \rho(\phi(u), \phi(v)) &= \sum_{i=0}^{k-1} \xi(w_i, w_{i+1}) \geq \sum_{i=0}^{k-1} \xi^\star(w_i, w_{i+1}) \\ &\geq \xi^\star(w_0, w_k) = \xi^\star(u, v) \geq \xi(u, v)/\gamma. \end{aligned}$$

Hence, the distortion is bounded by γ. □

We now show how to embed the shortest-path metric of a graph into a low dimensional ℓ_1-space in time near-linear in the graph size. For this purpose, we implement Bourgain's embedding [5] in near-linear time. We use the following version of Bourgain's embedding, which follows from the analysis in [19].

LEMMA 3.5 (BOURGAIN'S EMBEDDING [19]). *Let $\mathcal{M} = (X, \rho)$ be a finite metric on n points. There is an algorithm that computes an embedding $f : X \to \ell_1^k$ of $\mathcal{M}$ into ℓ_1^k for $k = O(\log^2 n)$ such that, with high probability, for each $u, v \in X$, we have $\rho(u, v) \le \|f(u) - f(v)\|_1 \le \rho(u, v) \cdot O(\log n)$.*

Specifically, for coordinate $i \in [k]$ of f, the embedding associates a nonempty set $A_i \subseteq X$ such that $f(u)_i = \rho(u, A_i) = \min_{a \in A} \rho(u, a)$. Each A_i is samplable in linear time.

LEMMA 3.6. *Consider a connected graph $G = (V, E)$ on n nodes with m edges and a weight function $w : E \to (0, \infty)$. There is a randomized algorithm that embeds G into $\ell_1^{O(\log^2 n)}$ with distortion $O(\log n)$ w.h.p., in time $\tilde{O}(m)$.*

PROOF. Let $\psi : V \to \ell_1^{O(\log^2 n)}$ be the embedding given by Lemma 3.5. We note that, for any nonempty subset $A \subseteq V$, we can compute $\rho(v, A)$ for all $v \in V$ by Dijkstra's algorithm in $\tilde{O}(m)$ time. The total running time is thus $\tilde{O}(m)$. □

3.1.3 Finalization of the proof of Theorem 3.1

The proof of Theorem 3.1 results from applying the Lemmas 3.2, 3.3, 3.4, and 3.6 in order.

In some cases to properly apply a lemma, we need to assume that all our coordinates are integers. Since the number of dimensions is always at most polylogarithmic, this can be done by multiplying each coordinate by the same polylogarithmic factor and rounding to the nearest integer.

Acknowledgment

The authors thank Piotr Indyk for helpful discussions, and Robert Krauthgamer, Sofya Raskhodnikova, Ronitt Rubinfeld, and Rahul Sami for early discussions on near-linear algorithms for edit distance.

4. REFERENCES

[1] A. Andoni, P. Indyk, and R. Krauthgamer. Earth mover distance over high-dimensional spaces. In *Proc. of SODA*, pages 343–352, 2008.

[2] Z. Bar-Yossef, T. S. Jayram, R. Krauthgamer, and R. Kumar. Approximating edit distance efficiently. In *Proc. of FOCS*, pages 550–559, 2004.

[3] T. Batu, F. Ergün, J. Kilian, A. Magen, S. Raskhodnikova, R. Rubinfeld, and R. Sami. A sublinear algorithm for weakly approximating edit distance. In *Proc. of STOC*, pages 316–324, 2003.

[4] T. Batu, F. Ergün, and C. Sahinalp. Oblivious string embeddings and edit distance approximations. In *Proc. of SODA*, pages 792–801, 2006.

[5] J. Bourgain. On Lipschitz embedding of finite metric spaces into Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[6] B. Brinkman and M. Charikar. On the impossibility of dimension reduction in ℓ_1. In *Proc. of FOCS*, 2003.

[7] M. Charikar and A. Sahai. Dimension reduction in the ℓ_1 norm. In *Proc. of FOCS*, pages 551–560, 2002.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[9] G. Cormode. *Sequence Distance Embeddings*. Ph.D. Thesis. University of Warwick, 2003.

[10] G. Cormode, M. Paterson, S. C. Sahinalp, and U. Vishkin. Communication complexity of document exchange. In *Proc. of SODA*, pages 197–206, 2000.

[11] D. Gusfield. *Algorithms on strings, trees, and sequences*. Cambridge University Press, Cambridge, 1997.

[12] P. Indyk. Algorithmic aspects of geometric embeddings (tutorial). *Proc. of FOCS*, pages 10–33, 2001.

[13] P. Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. *J. ACM*, 53(3):307–323, 2006. Previously appeared in FOCS'00.

[14] S. Khot and A. Naor. Nonembeddability theorems via fourier analysis. *Math. Ann.*, 334(4):821–852, 2006. Preliminary version appeared in FOCS'05.

[15] R. Krauthgamer and Y. Rabani. Improved lower bounds for embeddings into l_1. In *Proc. of SODA*, pages 1010–1017, 2006.

[16] J. Lee and A. Naor. Embedding the diamond graph in L_p and dimension reduction in L_1. *Geometric and Functional Analysis (GAFA)*, 14(4):745–747, 2004.

[17] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals (in russian). *Doklady Akademii Nauk SSSR*, 4(163):845–848, 1965. Appeared in English as: V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10(8), 707–710, 1966.

[18] W. J. Masek and M. Paterson. A faster algorithm computing string edit distances. *J. Comput. Syst. Sci.*, 20(1):18–31, 1980.

[19] J. Matousek. *Lectures on Discrete Geometry*. Springer, 2002.

[20] E. W. Myers. An $O(ND)$ difference algorithm and its variations. *Algorithmica*, 1(2):251–266, 1986.

[21] G. Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88, 2001.

[22] R. Ostrovsky and Y. Rabani. Low distortion embedding for edit distance. *J. ACM*, 54(5), 2007. Preliminary version appeared in STOC'05.

A. A SUBLINEAR TIME ALGORITHM

THEOREM A.1. *Let α and β be two constants such that $0 \le \alpha < \beta \le 1$. There is an algorithm that distinguishes pairs of strings with edit distance $O(n^\alpha)$ from those with distance $\Omega(n^\beta)$ in time $n^{\alpha+2(1-\beta)+o(1)}$.*

PROOF. Let $f(n) = 2^{\tilde{O}(\sqrt{\log n})}$ be a non-decreasing function that bounds the approximation factor of the algorithm given by Theorem 1.1. Let $b = \frac{n^{\beta-\alpha}}{f(n)\cdot\log n}$. We partition the input strings x and y into b blocks, denoted x_i and y_i for $i \in [b]$, of length n/b each.

If $\mathrm{ed}(x, y) = O(n^\alpha)$, then $\max_i \mathrm{ed}(x_i, y_i) \le \mathrm{ed}(x, y) = O(n^\alpha)$. On the other hand, if $\mathrm{ed}(x, y) = \Omega(n^\beta)$, then $\max_i \mathrm{ed}(x_i, y_i) \ge \mathrm{ed}(x, y)/b = \Omega(n^\alpha \cdot f(n) \cdot \log n)$. Moreover, the number of blocks i such that $\mathrm{ed}(x_i, y_i) \ge \mathrm{ed}(x, y)/2b = \Omega(n^\alpha \cdot f(n) \cdot \log n)$ is at least

$$\frac{\mathrm{ed}(x, y) - b \cdot \mathrm{ed}(x, y)/2b}{n/b} = \Omega(n^{\beta-1} \cdot b).$$

Therefore, we can tell the two cases apart with constant probability by sampling $O(n^{1-\beta})$ pairs of blocks (x_i, y_i) and checking if any of the pairs is at distance $\Omega(n^\alpha \cdot f(n) \cdot \log n)$. Since for each such pair of strings, we only have to tell edit distance $O(n^\alpha)$ from $\Omega(n^\alpha \cdot f(n) \cdot \log n)$, we can use the algorithm of Theorem 1.1. We amplify the probability of success of that algorithm in the standard way by running it $O(\log n)$ times. The total running time of the algorithm is $O(n^{1-\beta}) \cdot O(\log n) \cdot (n/b)^{1+o(1)} = O(n^{\alpha+2(1-\beta)+o(1)})$. □