Low Distortion Embeddings for Edit Distance *

Rafail Ostrovsky[†] Yuval Rabani[‡]

August 19, 2005

Abstract

We show that $\{0,1\}^d$ endowed with edit distance embeds into ℓ_1 with distortion $2^{O\left(\sqrt{\log d \log\log d}\right)}$. We further show efficient implementations of the embedding that yield solutions to various computational problems involving edit distance. These include sketching, communication complexity, nearest neighbor search. For all these problems, we improve upon previous bounds.

Category and subject Descriptors
F.2.2 Analysis of Algorithms and Problem Complexity, Nonnumerical Algorithms and Problems – *computations on discrete structures, pattern matching*

General Temrs
Algorithms,Theory.

Keywords Metric spaces, low distortion embeddings, edit distance.

[*] Preliminary. The proceedings version appeared at STOC 2005, Copyrigtht 2005 ACM 1-58113-960-8/05/0005.

[†] Computer Science Department, University of California at Los Angeles, 90095, USA. Email: `rafail@cs.ucla.edu`. Part of this work was done while at the Institute for Pure and Applied Mathematics (IPAM). Supported in part by a gift from Teradata, Intel equipment grant, NSF Cybertrust grant, OKAWA research award and B. John Garrick Foundation.

[‡] Computer Science Department, Technion — Israel Institute of Technology, Haifa 32000, Israel. Email: `rabani@cs.technion.ac.il`. Part of this work was done while visiting the Institute for Pure and Applied Mathematics (IPAM) at UCLA. Supported in part by Israel Science Foundation grant number 52/03 and by United States-Israel Binational Science Foundation grant number 02-00282.

1 Introduction

Given two metric spaces (X_1, d_1) and (X_2, d_2), an *embedding* $\varphi : (X_1, d_1) \hookrightarrow (X_2, d_2)$ has *distortion* c if and only if distances are preserved up to a factor of c (and uniform scaling). Easy to compute low distortion embeddings are extremely useful in computer science. Simply put, in many applications, if we can embed with small distortion a metric space which we do not understand well into some other metric space for which we do have efficient algorithms, then such an embedding provides an efficient algorithm for the original metric space. On a more fundamental level, studying embeddings of different metric spaces is a way to learn about the structure of these metric spaces and it has numerous implications in combinatorial optimization, discrete mathematics, functional analysis, and other areas.

In this paper we study the *edit distance* metric: Given two strings over a finite character alphabet, the *edit distance* (also known as *Levenshtein distance* [10]) measures the minimum number of character insertions, deletions, and substitutions needed to transform one string into the other. Edit distance plays a central role in genomics, text processing, web applications, and other areas. In particular, fast estimation of edit distance and efficient search according to the edit distance are the most investigated and used algorithms in computational biology. In this paper, we show that edit distance embeds in ℓ_1 with relatively small distortion. More specifically we show that $\{0, 1\}^d$ endowed with edit distance embeds into ℓ_1 with distortion $2^{O\left(\sqrt{\log d \log \log d}\right)}$. Note that edit distance is well-defined even on strings over a larger alphabet, as well as on strings of varying length. Our results trivially extend to larger alphabet, and they can be applied to variable length strings using standard padding arguments. We omit the discussion on these extensions from this extended abstract.

Furthermore, we show that our embedding can in fact be made *efficient*, thus implying improved algorithms for a number of problems, including sketching and approximate nearest neighbor search.

Given two d-bit strings, the best known running time to compute the exact edit distance, due to Masek and Paterson [11], is $O(d^2/\log d)$ (there is an easy quadratic-time dynamic programming algorithm). For approximating the edit distance, Batu et al. [3] show an algorithm that runs in time $O(d^{\max(\alpha/2, 2\alpha-1)})$ and can distinguish between edit distance $O(d^\alpha)$ and edit distance $\Omega(d)$. The best approximation achieved by a (nearly) linear time algorithm is the $d^{3/7}$ result of Bar-Yossef et al. [2]. If the edit distance metric is modified to allow "block operations" (i.e., swapping arbitrarily large blocks as a single operation), then the resulting *block edit* metric can be embedded into ℓ_1 with distortion $O(\log d \log^* d)$ [6, 12, 5]). Andoni et al. [1] showed that edit distance can not be embedded into ℓ_1 with distortion less then 3/2. With regard to the possibility of embedding of edit distance into ℓ_1 with small distortion, they say:

> *"So far, however such a result seems quite elusive. This raises the possibility that a low-distortion embedding might not be possible."*

We show an embedding into ℓ_1 with distortion $2^{O\left(\sqrt{\log d \log \log d}\right)}$. (Notice that distortion d

is trivial.) It is also worth pointing out that our paper provides a theoretical foundation to the experimentally successful idea of Broder et al. [4] of estimating similarity between documents or web pages by looking at sets of “shingles” (substrings) covering the document. Our methods (as well as other constructions and results in [3, 7, 2]) can be considered as a refinement of the original approach of [4].

A notion related to embedding is the *sketching* model. In this model we compute, for any string x, a *sketch* (i.e., a small “fingerprint”) of x which is far smaller then the length of x and yet we can estimate the distance between two strings x and y by examining their sketches. Sketching is related to multi-scale dimension reduction methods and approximate nearest neighbor search [9, 8], to streaming algorithms, and to communication complexity of document exchange [6]. The sketching model is well understood for Hamming distance (and implicitly for ℓ_1), see [9]. For edit distance, Bar-Yossef et al. [2] show how to compute a constant size sketch that can distinguish between edit distance at most k and edit distance at least $(kd)^{2/3}$ for any $k \leq \sqrt{d}$. Our embedding results can be used to produce constant size sketches that can distinguish between edit distance at most k and edit distance at least $2^{O\left(\sqrt{\log d \log\log d}\right)} \cdot k$ for all feasible values of k.

Another important problem is that of approximate nearest neighbor search algorithms. Given a database of n points in an underlying metric space, we want to pre-process the database and provide a search algorithm which, given a query point, finds a database point which is close to the query point. There is a vast literature on this subject. We restrict our attention to some of the theoretical work where the pre-processing cost is polynomial in the input size (even for high dimensional data; for d-bit strings the input size is nd) and the search cost is polynomial in the size of the query and in $\log n$. Kushilevitz et al. [9] and Indyk and Motwani [8] consider databases in ℓ_1, ℓ_2, and the Hamming cube. Their search algorithms retrieve a database point at distance at most $1 + \epsilon$ times the minimum. Muthukrishnan and Sahinalp [12] show how to extend this result to block edit distance. Indyk [7] gives a solution for edit distance where the search can return a point at distance at most d^ϵ times the minimum, for any $\epsilon > 0$. The Bar-Yossef et al. paper [2] gives similar bounds with a better pre-processing performance. Our embedding results imply a solution where the search returns a point at distance at most $2^{O\left(\sqrt{\log d \log\log d}\right)}$ times the minimum.

We defer the discussion of other applications to the full version of the paper.

2 Preliminaries

We denote by $[i, j]$ the set $\{i, i + 1, \ldots, j\}$ and we denote by $[j]$ the set $[1, j]$. (If $j < i$ then $[i, j]$ is the empty set.) Let $x \in \{0, 1\}^*$. We denote by $|x|$ the length of x. For $i \in [|x|]$ we denote by x_i the i-th character in x. For $i, j \in [|x|]$ we define $x[i, j] = x_i x_{i+1} x_{i+2} \ldots x_j$. (If $j < i$ then $x[i, j]$ is the empty string.) When we denote a set of strings, we usually intend (unless otherwise specified) that multiple copies of the same string are counted as different elements of the set. Thus, the simplified notation $\{x^1, x^2, \ldots, x^n\}$ is used for the set

$$\left\{\left(x^j, k\right) :\ j \in [n] \wedge k = \left|\left\{i \in [j-1] :\ x^i = x^j\right\}\right|\right\}.$$

For $s \in \mathbb{N}$, we put

$$\mathrm{shifts}(x, s) = \{x[1, |x| - s + 1], x[2, |x| - s + 2], \ldots, x[s, |x|]\}$$

Let $x, y \in \{0,1\}^*$. We denote by xy the concatenation of x followed by y. We denote by $\mathrm{ed}(x, y)$ the edit distance between x and y, which is the minimum number of insert, delete, and substitute operations needed to convert x to y (or vice versa). For x, y with $|x| = |y|$, we denote by $\mathcal{H}(x, y)$ the Hamming distance between x and y (i.e., the number positions i such that $x_i \neq y_i$). For a set X and $s \in \mathbb{N}$, we denote by $\binom{X}{s}$ the set of subsets of X of cardinality s. Let $x, y \in \{0,1\}^*$. Consider an optimal sequence of edit operations converting x into y. Any such sequence is equivalent to a function $f_{x,y} : [0, |x| + 1] \to [0, |y| + 1] \cup \{\varepsilon\}$ with the following properties.

1. $f_{x,y}(0) = 0$ and $f_{x,y}(|x| + 1) = |y| + 1$.
2. $\forall i \in [|x|]$, $f_{x,y}(i) \in [|y|] \cup \{\varepsilon\}$.
3. $\forall i, j \in [|x|]$ such that $i < j$ and $f_{x,y}(i), f_{x,y}(j) \neq \varepsilon$, it holds that $f_{x,y}(i) < f_{x,y}(j)$.

The interpretation of $f_{x,y}$ as a sequence of edit operations is as follows. Having $f_{x,y}(i) = \varepsilon$ corresponds to deleting x_i. If there is no i such that $f_{x,y}(i) = j$ that corresponds to inserting y_j. If $j = f_{x,y}(i) \in [|y|]$ and $x_i \neq y_j$ then that corresponds to substituting y_j for x_i. The extension of $f_{x,y}$ to 0 and $|x| + 1$ is useful in some of the calculations below. Notice that for $j \in [0, |y| + 1]$ we may put

$$f_{y,x}(j) = f_{x,y}^{-1}(j) = \begin{cases} i & \exists i \in [0, |x| + 1],\ f_{x,y}(i) = j; \\ \varepsilon & \text{otherwise.} \end{cases}$$

The following facts are trivial.

Fact 1. $\mathrm{ed}(x, y) \geq ||x| - |y||$.

Fact 2. $\mathrm{ed}(x, y) \geq |\{j :\ f_{x,y}(j) = \varepsilon\}| + \left|\left\{j :\ f_{x,y}^{-1}(j) = \varepsilon\right\}\right|$.

Fact 3. $|\{j :\ f_{x,y}(j) = \varepsilon\}| \geq |x| - |y|$.

Fact 4. *Let $i, i' \in [|x|]$ and $j, j' \in [|y|]$ satisfy the following conditions: $i \leq i'$; $j \leq j'$; $f_{x,y}(i) = j$; $\forall i'' \in [i + 1, i']$, $f_{x,y}(i'') = \varepsilon$; and $\forall j'' \in [j + 1, j']$, $f_{x,y}^{-1}(j'') = \varepsilon$. Then,*

$$\mathrm{ed}(x, y) = \mathrm{ed}(x[1, i'], y[1, j']) + \mathrm{ed}(x[i' + 1, |x|], y[j' + 1, |y|]).$$

For $x, y \in \{0,1\}^*$, $\mathrm{ed}(x, y)$ can be estimated roughly by comparing substrings of x and y, as the following lemmas quantify.

Lemma 5. *Let $x, y \in \{0,1\}^*$ such that $|x| \le |y|$, and let $b \in \mathbb{N}$, $b < |x|$. Then, there exists an injection*

$$f : [|x| - b + 1] \to [|y| - b + 1]$$

such that

$$\begin{aligned} |\{i \in [|x| - b + 1] &: \operatorname{ed}(x[i, i + b - 1], y[f(i), f(i) + b - 1]) \\ &> 2\operatorname{ed}(x, y)\}| \le \operatorname{ed}(x, y). \end{aligned}$$

Proof. Let $I = \{i \in [|x| - b + 1] : \ f_{x,y}(i) \in [|y| - b + 1]\}$. Put

$$i_{\max} = \max\{i \in [|x| - b + 1] :$$

$$\forall j \in [i], \ f_{x,y}(j) \in [|y| - b + 1] \vee f_{x,y}(j) = \varepsilon\}.$$

By Fact 4,

$$\operatorname{ed}(x, y) = \operatorname{ed}(x[1, i_{\max}], y[1, |y| - b + 1]) +$$

$$+ \operatorname{ed}(x[i_{\max} + 1, |x|], y[|y| - b + 2, |y|]).$$

We have that

$$\begin{aligned} & |\{i \in [|x| - b + 1] : \ f_{x,y}(i) \in [|y| - b + 2, |y|]\}| \\ = \ & |\{i \in [i_{\max} + 1, |x| - b + 1] : \ f_{x,y}(i) \neq \varepsilon\}| \\ \le \ & |x| - b + 1 - i_{\max} \\ \le \ & |y| - b + 1 - i_{\max} \\ \le \ & \left|\left\{j \in [|y| - b + 1] : \ f^{-1}_{x,y}(j) = \varepsilon\right\}\right|, \end{aligned}$$

where the last inequality follows from Fact 3. Using Fact 2,

$$\begin{aligned} |I| \ = \ & |\{i \in [|x| - b + 1] : \ f_{x,y}(i) \neq \varepsilon\}| \\ & - |\{i \in [|x| - b + 1] : \ f_{x,y}(i) \in [|y| - b + 2, |y|]\}| \\ = \ & |x| - b + 1 - |\{i \in [|x| - b + 1] : \ f_{x,y}(i) = \varepsilon\}| \\ & - |\{i \in [|x| - b + 1] : \ f_{x,y}(i) \in [|y| - b + 2, |y|]\}| \\ \ge \ & |x| - b + 1 - |\{i \in [|x| - b + 1] : \ f_{x,y}(i) = \varepsilon\}| \\ & - \left|\left\{j \in [|y| - b + 1] : \ f^{-1}_{x,y}(j) = \varepsilon\right\}\right| \\ \ge \ & |x| - b + 1 - \operatorname{ed}(x[1, i_{\max}], y[1, |y| - b + 1]) \\ \ge \ & |x| - b + 1 - \operatorname{ed}(x, y). \end{aligned}$$

For every $i \in I$, put $f(i) = f_{x,y}(i)$ and extend f arbitrarily to $[|x| - b + 1]$. We show that for every $i \in I$, $\operatorname{ed}(x[i, i + b - 1], y[f(i), f(i) + b - 1]) \le 2\operatorname{ed}(x, y)$. Let

$$g(i) = \max\{i + b - 1, \max\{i' \in [i, |x|] :$$

$$\exists j \in [f(i), f(i)+b-1],\ f_{x,y}(i') = j\}\},$$

and let

$$h(i) = \max\{f(i)+b-1, \max\{j \in [f(i), |y|] :$$
$$\exists i' \in [i, i+b-1],\ f_{x,y}(i') = j\}\}.$$

Notice that either $g(i) = i+b-1$ or $h(i) = f(i)+b-1$. Moreover, $g(i)-(i+b-1) \leq \mathrm{ed}(x,y)$ and $h(i)-(f(i)+b-1) \leq \mathrm{ed}(x,y)$. Clearly, $\mathrm{ed}(x[i,g(i)], y[f(i),h(i)]) \leq \mathrm{ed}(x,y)$. Therefore,

$$\begin{aligned} & \mathrm{ed}(x[i,i+b-1], y[f(i), f(i)+b-1]) \\ \leq\ & \mathrm{ed}(x[i,g(i)], y[f(i),h(i)]) + g(i) - (i+b-1) + \\ & +h(i) - (f(i)+b-1) \\ \leq\ & 2\,\mathrm{ed}(x,y), \end{aligned}$$

as required. □

Let $b, d, s \in \mathbb{N}$ with $\frac{d}{b} \in \mathbb{N}$ and $s < b$.

Lemma 6. *For every $x, y \in \{0,1\}^d$ there exists a sequence $k_1, k_2, \ldots, k_{d/b}$ satisfying*

$$\sum_{i=1}^{d/b} k_i \leq 2\,\mathrm{ed}(x,y),$$

such that for every $i \in [d/b]$ there exists a bijection

$$\tau_i : \mathrm{shifts}\,(x[(i-1)b+1, ib], s) \to$$
$$\mathrm{shifts}\,(y[(i-1)b+1, ib], s)$$

such that

$$|\{z \in \mathrm{shifts}\,(x[(i-1)b+1, ib], s) :\ \mathrm{ed}(z, \tau_i(z)) > k_i\}|$$
$$\leq \mathrm{ed}(x,y).$$

Proof. Let $i \in [d/b]$. Put

$$a_i^x = \max\{(i-1)b+1, \max\{j \in [d+1] :$$
$$f_{x,y}(j-1) \in [0, (i-1)b]\}\},$$
$$b_i^x = \min\{ib, \min\{j \in [d] :\ f_{x,y}(j+1) > ib\}\},$$
$$a_i^y = \max\{(i-1)b+1, \max\{j \in [d+1] :$$
$$f_{x,y}^{-1}(j-1) \in [0, (i-1)b]\}\},$$

and

$$b_i^y = \min\left\{ib, \min\left\{j \in [d] :\ f_{x,y}^{-1}(j+1) > ib\right\}\right\}.$$

By Fact 4,

$$\begin{aligned}\mathrm{ed}(x,y) = \mathrm{ed}(x[1,a_i^x-1],y[1,a_i^y-1])+ \\ +\,\mathrm{ed}(x[a_i^x,b_i^x],y[a_i^y,b_i^y]) + \mathrm{ed}(x[b_i^x+1,d],y[b_i^y+1,d]).\end{aligned}$$

Moreover, by Fact 1,

$$\mathrm{ed}(x[1,a_i^x-1],y[1,a_i^y-1]) \geq |a_i^x-a_i^y|$$

and

$$\mathrm{ed}(x[b_i^x+1,d],y[b_i^y+1,d]) \geq |b_i^x-b_i^y|\,.$$

We assume without loss of generality that $b_i^x - a_i^x \leq b_i^y - a_i^y$. Notice that if $z \in \mathrm{shifts}(x[(i-1)b+1,ib],s)$ (or $z \in \mathrm{shifts}(y[(i-1)b+1,ib],s)$) then $|z| = b-s+1$. By Lemma 5, there exists an injection $f : [b_i^x - a_i^x - b + s + 1] \to [b_i^y - a_i^y - b + s + 1]$ such that $|\{j \in [b_i^x-a_i^x-b+s+1] : \mathrm{ed}(x[a_i^x+j-1,a_i^x+j+b-s-1],y[a_i^y+f(j)-1,a_i^y+f(j)+b-s-1]) > 2\,\mathrm{ed}(x[a_i^x,b_i^x],y[a_i^y,b_i^y])\}| \leq \mathrm{ed}(x[a_i^x,b_i^x],y[a_i^y,b_i^y])$. For every $j \in [b_i^x - a_i^x - b + s + 1]$ set $\tau_i(x[a_i^x+j-1,a_i^x+j+b-s-1]) = y[a_i^y+f(j)-1,a_i^y+f(j)+b-s-1]$ and extend τ_i to the rest of shifts $(x[(i-1)b+1,ib],s)$ arbitrarily, and set $k_i = 2\,\mathrm{ed}(x[a_i^x,b_i^x],y[a_i^y,b_i^y])$. Now,

$$\begin{aligned} & |\{z \in \mathrm{shifts}\,(x[(i-1)b+1,ib],s) :\ \mathrm{ed}(z,\tau_i(z)) > k_i\}| \\ \leq\ & \mathrm{ed}(x[a_i^x,b_i^x],y[a_i^y,b_i^y]) + \max\{a_i^x,a_i^y\} - \\ & -((i-1)b+1) + ib - \min\{b_i^x,b_i^y\} \\ \leq\ & \mathrm{ed}(x[a_i^x,b_i^x],y[a_i^y,b_i^y]) + \mathrm{ed}(x[1,a_i^x-1],y[1,a_i^y-1]) + \\ & +\,\mathrm{ed}(x[b_i^x+1,d],y[b_i^y+1,d]) \\ =\ & \mathrm{ed}(x,y).\end{aligned}$$

Moreover,

$$\begin{aligned}\sum_{i=1}^{d/b} k_i\ &\leq\ 2\sum_{i=1}^{d/b}\mathrm{ed}(x[a_i^x,b_i^x],y[a_i^y,b_i^y]) \\ &\leq\ 2\,\mathrm{ed}(x,y).\end{aligned}$$

This completes the proof. □

3 The Embedding

In this section we prove our main result, the following upper bound on the distortion of embedding edit distance into ℓ_1. The embedding given in this section ignores computational efficiency. In the next section we present an efficient implementation of the embedding.

Theorem 7. *There exists a universal constant $c > 0$ such that for every $d \in \mathbb{N}$ there exists an embedding*
$\varphi : \big(\{0,1\}^d, \mathrm{ed}\big) \hookrightarrow \ell_1$ with distortion at most $2^{c\sqrt{\log d \log\log d}}$.

We first present an informal description of the embedding. Let $x \in \{0,1\}^d$ be any string. We partition x into $2^{\sqrt{\log d \log\log d}}$ disjoint substrings of (approximately) the same length. We refer to these substrings as *blocks*. Let $x^1, x^2, \ldots$ denote the blocks. We consider the sets $\text{shifts}(x^i, s)$ for s ranging over the non-negative powers of $\log d$ that are below the block length. Given $x, y \in \{0,1\}^d$, define the distance between $\text{shifts}(x^i, s)$ and $\text{shifts}(y^i, s)$ to be the minimum cost perfect matching between the two sets, where the cost of an edge between two elements is their edit distance, truncated above s. This is a metric on sets of strings that are much shorter than x. Ideally, we would like to embed this metric into ℓ_1. The edit distance-preserving embedding into ℓ_1 would then consist of concatenating the scaled embeddings of $\text{shifts}(x^i, s)$ for all blocks i and all values of s. However, a good embedding of $\text{shifts}(x^i, s)$ seems to be too strong an inductive hypothesis. Therefore, we inductively embed the strings in $\text{shifts}(x^i, s)$ into ℓ_1 and redefine the edge costs for the matching to be the truncated ℓ_1 distance between the embedded strings. We embed this metric over sets of strings into ℓ_1. This embedding is not necessarily low distortion. The following lemma, which may be of independent interest, states the properties of this embedding.

Lemma 8. *For every $\epsilon > 0$ and for every $d, s, t \in \mathbb{N}$ that satisfy $\ln(s/\epsilon) \le t \le d$ there is a mapping $\psi : \binom{\{0,1\}^d}{s} \to \ell_1$ such that for every $A, B \in \binom{\{0,1\}^d}{s}$,*

$$\|\psi(A) - \psi(B)\|_1 \le \frac{1}{s} \cdot \min_{\sigma} \left\{ \sum_{x \in A} \min\{t, 2\mathcal{H}(x, \sigma(x)) \ln(s/\epsilon)\} \right\}$$

(where the first minimum is taken over all bijections $\sigma : A \to B$), and furthermore if for all $x \in A$ and $y \in B$, $\mathcal{H}(x,y) \ge t$, then

$$\|\psi(A) - \psi(B)\|_1 \ge (1-\epsilon)t.$$

Proof. Put $b = \frac{d \cdot \ln(s/\epsilon)}{t}$. Consider the function $\chi : \binom{\{0,1\}^d}{s} \to \{0,1\}^{(2d)^b}$, which is defined as follows. Let $A = \{x^1, x^2, \ldots, x^s\} \in \binom{\{0,1\}^d}{s}$. To set $\chi(A)$, we generate a coordinate for every sequence $I = (i_1, i_2, \ldots, i_b) \in [d]^b$ and for every string $z \in \{0,1\}^b$. We set $\chi(A)_{I,z} = 1$ iff there is $j \in \{1, 2, \ldots, s\}$ such that $z = x^j_{i_1} x^j_{i_2} \cdots x^j_{i_b}$. For all $A \in \binom{\{0,1\}^d}{s}$ put $\psi(A) = \frac{t}{2sd^b}\chi(A)$.
[1]

Let $A, B \in \binom{\{0,1\}^d}{s}$. Let $\sigma : A \to B$ be a bijection that minimizes

$$\sum_{x \in A} \min\{t, 2\mathcal{H}(x, \sigma(x)) \ln(s/\epsilon)\}.$$

[1] That is, we consider multisets: for every I, take the multiset of samples from all the strings in A (or B), and add a counter to each string in the multiset, so the indices for the vector ψ are strings plus counters. Alternatively, we can use other encodings, such as we can index ψ with strings and the entries will have a value which is the counter.

Let $x \in A$. Let $I \in [d]^b$ be chosen uniformly at random. Then, for $z = x_{i_1} x_{i_2} \cdots x_{i_b}$, $\chi(A)_{I,z} = 1$. On the other hand,

$$\begin{aligned}
\Pr[\chi(B)_{I,z} = 0] &\leq \Pr\left[\bigvee_{j=1}^{b} \{x_{i_j} \neq \sigma(x)_{i_j}\}\right] \\
&= 1 - \Pr\left[\bigwedge_{j=1}^{b} \{x_{i_j} = \sigma(x)_{i_j}\}\right] \\
&= 1 - \left(1 - \frac{\mathcal{H}(x, \sigma(x))}{d}\right)^b \\
&\leq 1 - e^{-\frac{2b\mathcal{H}(x,\sigma(x))}{d}} \\
&= 1 - e^{-\frac{2\mathcal{H}(x,\sigma(x)) \ln(s/\epsilon)}{t}} \\
&\leq \min\left\{1, \frac{2\mathcal{H}(x, \sigma(x)) \ln(s/\epsilon)}{t}\right\}.
\end{aligned}$$

Thus we get

$$\begin{aligned}
& \|\psi(A) - \psi(B)\|_1 \\
= \; & \frac{t}{2sd^b} \sum_{I \in [d]^b} \sum_{z \in \{0,1\}^b} |\chi(A)_{I,z} - \chi(B)_{I,z}| \\
= \; & \frac{t}{2s} \cdot \sum_{x \in A} \Pr\left[\chi(B)_{I,z} = 0 \mid z = x_{i_1} x_{i_2} \cdots x_{i_b}\right] \\
& + \frac{t}{2s} \cdot \sum_{x \in B} \Pr\left[\chi(A)_{I,z} = 0 \mid z = x_{i_1} x_{i_2} \cdots x_{i_b}\right] \\
\leq \; & \frac{t}{s} \cdot \sum_{x \in A} \min\left\{1, \frac{2\mathcal{H}(x, \sigma(x)) \ln(s/\epsilon)}{t}\right\} \\
= \; & \frac{1}{s} \cdot \sum_{x \in A} \min\{t, 2\mathcal{H}(x, \sigma(x)) \ln(s/\epsilon)\}.
\end{aligned}$$

On the other hand, assuming that $\min\{\mathcal{H}(x, y) : \; x \in A \wedge y \in B\} \geq t$,

$$\begin{aligned}
\Pr[\chi(B)_{I,z} = 1] &\leq s \cdot \left(1 - \frac{\mathcal{H}(x, \sigma(x))}{d}\right)^b \\
&\leq s \cdot \left(1 - \frac{t}{d}\right)^b \\
&= s \cdot \left(1 - \frac{t}{d}\right)^{\frac{d \ln(s/\epsilon)}{t}} \\
&\leq s \cdot e^{-\ln(s/\epsilon)} \\
&= \epsilon.
\end{aligned}$$

Therefore,

$$\begin{aligned}
& \|\psi(A)-\psi(B)\|_1 \\
= \;& \frac{t}{2sd^b}\sum_{I\in[d]^b}\sum_{z\in\{0,1\}^b}|\chi(A)_{I,z}-\chi(B)_{I,z}| \\
= \;& \frac{t}{2s}\cdot\sum_{x\in A}\Pr\left[\chi(B)_{I,z}=0 \mid z=x_{i_1}x_{i_2}\cdots x_{i_b}\right] \\
& +\frac{t}{2s}\cdot\sum_{x\in B}\Pr\left[\chi(A)_{I,z}=0 \mid z=x_{i_1}x_{i_2}\cdots x_{i_b}\right] \\
= \;& \frac{t}{2s}\cdot\sum_{x\in A}\left(1-\Pr\left[\chi(B)_{I,z}=1 \mid z=x_{i_1}x_{i_2}\cdots x_{i_b}\right]\right) \\
& +\frac{t}{2s}\cdot\sum_{x\in B}\left(1-\Pr\left[\chi(A)_{I,z}=1 \mid z=x_{i_1}x_{i_2}\cdots x_{i_b}\right]\right) \\
\geq \;& \frac{t}{2s}\cdot 2(1-\epsilon)s \\
= \;& (1-\epsilon)t,
\end{aligned}$$

completing the proof. □

For given values d, s and for $t=s$, $\epsilon=\frac{1}{2}$, we denote by $\psi_{d,s}$ the embedding ψ of Lemma 8.

Proof of Theorem 7. The proof is by induction on d. Clearly, the claim is true for d sufficiently small. We therefore assume that the claim holds for all strings of length less than d. For $d'<d$, let $\varphi_{d'}$ denote the embedding constructed for strings in $\{0,1\}^{d'}$. Let $x\in\{0,1\}^d$. For $j\in\mathbb{N}$ let $s_j=(\log d)^j$. Put $i_{\max}=2^{\sqrt{\log d\log\log d}}$ and put $j_{\max}=\min\left\{j\in\mathbb{N}:\ s_j\geq\frac{d}{i_{\max}\cdot\log d}\right\}$. For $j\in[0,j_{\max}]$, put $d_j=\frac{d}{i_{\max}}-s_j+1$. For $i\in[i_{\max}]$ and $j\in[0,j_{\max}]$, put $A_{i,j}(x)=\mathrm{shifts}(x[(i-1)d/i_{\max}+1, id/i_{\max}], s_j)$, and put

$$B_{i,j}(x)=\left\{\varphi_{d_j}(y):\ y\in A_{i,j}(x)\right\}.$$

Finally, define the vector $\varphi_d(x)$ whose coordinates are indexed by I,z,i,j as follows.

$$(\varphi_d(x))_{I,z,i,j}=\left(\psi_{d_j,s_j}(B_{i,j})\right)_{I,z}.$$

(Notice that we implicitly assume, for simplicity, that $i_{\max}$ divides d, and we partition x into $i_{\max}$ blocks of length $d/i_{\max}$ each. The assumption can be removed by increasing the length of some of the blocks by 1. To simplify the notation, we ignore this issue in this extended abstract.)

Consider two strings $x,y\in\{0,1\}^d$, $x\neq y$. By Lemma 6, there is a sequence $k_1,k_2,\ldots,k_{i_{\max}}$ satisfying

$$\sum_{i=1}^{i_{\max}}k_i\leq 2\,\mathrm{ed}(x,y), \tag{1}$$

such that for every $i \in [i_{\max}]$ and for every $j \in [0, j_{\max}]$ there exists a bijection $\tau : A_{i,j}(x) \to A_{i,j}(y)$ such that

$$|\{z \in A_{i,j}(x) : \ \mathrm{ed}(z, \tau(z)) > k_i\}| \leq \mathrm{ed}(x, y). \tag{2}$$

Now, if $s_j < \mathrm{ed}(x, y)$ then, trivially, by Lemma 8,

$$\|\psi_{d_j,s_j}(B_{i,j}(x)) - \psi_{d_j,s_j}(B_{i,j}(y))\|_1 \leq s_j. \tag{3}$$

Otherwise, if $s_j \geq \mathrm{ed}(x, y)$, then by Equation (2) and the induction hypothesis,

$$\left|\left\{z \in B_{i,j}(x) : \ \mathcal{H}(z, \tau(z)) > \|\varphi_{d_j}\|_{\mathrm{Lip}} \cdot k_i\right\}\right| \leq \mathrm{ed}(x, y).$$

Therefore, by Lemma 8,

$$\begin{aligned}
&\|\psi_{d_j,s_j}(B_{i,j}(x)) - \psi_{d_j,s_j}(B_{i,j}(y))\|_1 \\
\leq\ & \frac{1}{s_j} \cdot \min_{\sigma} \left\{ \sum_{z \in B_{i,j}(x)} \min\{s_j, 2\mathcal{H}(z, \sigma(z)) \cdot \ln(2s_j)\} \right\} \\
\leq\ & \frac{1}{s_j} \cdot \sum_{z \in B_{i,j}(x)} \min\{s_j, 2\mathcal{H}(z, \tau(z)) \cdot \ln(2s_j)\} \\
\leq\ & \frac{1}{s_j} \cdot (\mathrm{ed}(x, y) \cdot s_j + s_j \cdot 2\mathcal{H}(z, \tau(z)) \cdot \ln(2s_j)) \\
\leq\ & \mathrm{ed}(x, y) + 2\|\varphi_{d_j}\|_{\mathrm{Lip}} \cdot k_i \cdot \ln(2s_j).
\end{aligned} \tag{4}$$

Summing Equations (3) and (4), over i, j, and using Condition (1), we get

$$\begin{aligned}
&\|\varphi_d(x) - \varphi_d(y)\|_1 \\
=\ & \sum_{i=1}^{i_{\max}} \sum_{j=0}^{j_{\max}} \left\|\psi_{d_j,s_j}(B_{i,j}(x)) - \psi_{d_j,s_j}(B_{i,j}(y))\right\|_1 \\
\leq\ & i_{\max} \cdot \sum_{j:\ s_j < \mathrm{ed}(x,y)} s_j + i_{\max} \cdot (j_{\max} + 1) \cdot \mathrm{ed}(x, y) + \\
& + \left(2\|\varphi_{d_{j_{\max}}}\|_{\mathrm{Lip}} \cdot \frac{\log^2 d}{\log\log d}\right) \cdot \sum_{i=1}^{i_{\max}} k_i \\
\leq\ & \left(2i_{\max} \cdot \frac{\log d}{\log\log d} + 4\|\varphi_{d_{j_{\max}}}\|_{\mathrm{Lip}} \cdot \frac{\log^2 d}{\log\log d}\right) \cdot \mathrm{ed}(x, y).
\end{aligned}$$

Thus, we obtain the recurrence relation

$$\|\varphi_d\|_{\mathrm{Lip}} \leq 4\|\varphi_{d_{j_{\max}}}\|_{\mathrm{Lip}} \cdot \frac{\log^2 d}{\log\log d} + 2i_{\max} \cdot \frac{\log d}{\log\log d},$$

so $\|\varphi_d\|_{\mathrm{Lip}}$ (as a function of d) satisfies the conditions of Lemma 14 (in the appendix), and therefore the recurrence solves to $\|\varphi_d\|_{\mathrm{Lip}} = 2^{O(\sqrt{\log d \log\log d})}$.

We now proceed to bound $\|\varphi_d^{-1}\|_{\mathrm{Lip}}$. Define a sequence $j_1, j_2, \ldots, j_{i_{\max}}$ as follows. For $i \in [1, i_{\max}]$, if $x[(i-1)d/i_{\max}+1, id/i_{\max}] = y[(i-1)d/i_{\max}+1, id/i_{\max}]$ then put $j_i = -1$, otherwise put

$$\begin{gathered} j_i = \max\{j \in [0, j_{\max}] : \\ \forall z \in A_{i,j}(x) \forall z' \in A_{i,j}(y), \mathrm{ed}(z,z') \geq \|\varphi_{d_j}^{-1}\|_{\mathrm{Lip}} \cdot s_j\}. \end{gathered}$$

Put $s_{j_{\max}+1} = \frac{d}{i_{\max}}$. Let $I = \{i \in [1, i_{\max}] : \; j_i \geq 0\}$. Clearly,

$$\mathrm{ed}(x,y) \leq \sum_{i \in I} \left(\|\varphi_{d_{j_i+1}}^{-1}\|_{\mathrm{Lip}} + 2 \right) \cdot s_{j_i+1}.$$

On the other hand, consider $z \in A_{i,j}(x)$ and $z' \in A_{i,j}(y)$. If $\mathrm{ed}(z,z') \geq \|\varphi_{d_j}^{-1}\|_{\mathrm{Lip}} \cdot s_j$, then by the induction hypothesis $\mathcal{H}(\varphi(z), \varphi(z')) \geq s_j$. Therefore, by Lemma 8,

$$\begin{aligned} & \|\varphi_d(x) - \varphi_d(y)\|_1 \\ \geq \; & \sum_{i \in I} \left\| \psi_{d_{j_i}, s_{j_i}}(B_{i,j_i}(x)) - \psi_{d_{j_i}, s_{j_i}}(B_{i,j_i}(y)) \right\|_1 \\ \geq \; & \frac{1}{2} \sum_{i \in I} s_{j_i} \\ \geq \; & \frac{1}{2 \log d} \cdot \sum_{i \in I} s_{j_i+1} \\ \geq \; & \frac{1}{2 \left(\|\varphi_{d_{j_i+1}}^{-1}\|_{\mathrm{Lip}} + 2 \right) \log d} \cdot \mathrm{ed}(x,y). \end{aligned}$$

Thus we get the recurrence

$$\|\varphi_d^{-1}\|_{\mathrm{Lip}} \leq 2 \|\varphi_{d/i_{\max}}^{-1}\|_{\mathrm{Lip}} \cdot \log d + 4 \log d,$$

which solves to $\|\varphi_d^{-1}\|_{\mathrm{Lip}} = 2^{O(\sqrt{\log d \log\log d})}$, by Lemma 14.

4 Implementation, Applications

The embedding described in the previous section is computationally inefficient. An efficient implementation of the embedding is derived from the following algorithmic version of Lemma 8.

Lemma 9. *There exists a probabilistic polynomial time algorithm ψ that for every $\delta > 0$, for every $d, s \in \mathbb{N}$ that satisfy $\ln(2s) \leq s \leq d$, and for every $A \in \binom{\{0,1\}^d}{s}$ computes $\psi(A) = \psi(A, d, s, \delta) \in \ell_1^{O(s^2 \log(1/\delta))}$ such that for every $A, B \in \binom{\{0,1\}^d}{s}$ the following holds with probability at least $1 - \delta$:*

$$\|\psi(A) - \psi(B)\|_1 \leq \frac{1}{s} \cdot \min_{\sigma} \left\{ \sum_{x \in A} \min\{s, 6\mathcal{H}(x, \sigma(x)) \ln(2s)\} \right\}$$

(where the first minimum is taken over all bijections $\sigma : A \to B$), and furthermore if for all $x \in A$ and $y \in B$, $\mathcal{H}(x,y) \geq s$, then

$$\|\psi(A) - \psi(B)\|_1 \geq \frac{s}{3}.$$

Proof sketch. Put $b = \frac{d \ln(2s)}{s}$, as in the proof of Lemma 8 (with $t = s$ and $\epsilon = \frac{1}{2}$). Instead of averaging over all $I \in [d]^b$, we take a sample of $O(s \log(1/\delta))$ independent, uniformly distributed, I-s. Using standard large deviation bounds, for any $A, B \in \binom{\{0,1\}^d}{s}$,

$$\Pr\left[\|\psi(A) - \psi(B)\|_1 > \frac{1}{s} \cdot \right.$$

$$\left. \cdot \min_{\sigma}\{\sum_{x \in A} \min\{s, 6\mathcal{H}(x, \sigma(x)) \ln(2s)\}\}\right] < \frac{\delta}{2},$$

and if for all $x \in A$ and $y \in B$, $\mathcal{H}(x,y) \geq s$, then

$$\Pr\left[\|\psi(A) - \psi(B)\|_1 < \frac{s}{3}\right] < \frac{\delta}{2}.$$

We still have the problem of indexing, for every sample point I, all possible $z \in \{0,1\}^b$. Notice, however, that for any $A \in \binom{\{0,1\}^d}{s}$ and $I \in [d]^b$, the number of possible values of z that actually appear is s. Hashing $\{0,1\}^b$ into a domain of size $O(s)$ is sufficient with high probability. (Notice that $\|\psi(A) - \psi(B)\|_1$ cannot increase due to the hashing.) □

Lemma 9 implies the following algorithmic version of Theorem 7.

Theorem 10. *There exists a polynomial time algorithm φ that for every $\delta > 0$, for every $d \in \mathbb{N}$, and for every $x \in \{0,1\}^d$ computes $\varphi(x) = \varphi(x, d, \delta) \in \ell_1^{O(d^2 \log(d/\delta))}$, such that for every $x, y \in \{0,1\}^d$, with probability at least $1 - \delta$,*

$$2^{-O(\sqrt{\log d \log\log d})} \cdot \mathrm{ed}(x,y) \leq$$

$$\leq \|\varphi(x) - \varphi(y)\|_1 \leq$$

$$\leq 2^{O(\sqrt{\log d \log\log d})} \cdot \mathrm{ed}(x,y).$$

The proof of Theorem 10 follows closely the proof of Theorem 7, and is omitted from this version of the paper. Notice that in order to embed $\{0,1\}^d$ into ℓ_1, we need to take $\delta \ll 4^{-d}$ to ensure that all $\binom{2^d}{2}$ distances are preserved with high probability.

Theorem 10 can be used to solve many data processing problems that involve edit distance, by first embedding the data into ℓ_1, and then applying previous results for ℓ_1 data. We discuss several examples to illustrate the issue. Our first example is a sketching algorithm, which is a basic building block for other tasks.

Theorem 11. *There are universal constants $c, \alpha, \beta > 0$, $\alpha < \beta$, such that the following holds. There is a probabilistic polynomial time algorithm s that for every $\delta > 0$, for every $d, t \in \mathbb{N}$, and for every $x \in \{0,1\}^d$ computes a sketch $s(x) = s(x, d, t, \delta) \in \{0,1\}^{O(\log(1/\delta))}$ such that for every $x, y \in \{0,1\}^d$ with probability at least $1 - \delta$ the following holds. If $\mathrm{ed}(x,y) \leq t$ then $\mathcal{H}(s(x), s(y)) \leq \alpha \log(1/\delta)$, and if $\mathrm{ed}(x,y) \geq 2^{c\sqrt{\log d \log\log d}} \cdot t$ then $\mathcal{H}(s(x), s(y)) \geq \beta \log(1/\delta)$.*[2]

Proof. To compute $s(x)$, use Theorem 10 to embed x in ℓ_1, then use the ℓ_1 sketching algorithm implied in [9]. □

Corollary 12. *There is $c > 0$ such that for every $\delta > 0$, for every $n, t \in \mathbb{N}$, there is a a one-round public-coin probabilistic two-party communication protocol that on input $x, y \in \{0,1\}^n$ exchanges $O(\log(1/\delta))$ bits and with probability at least $1 - \delta$ outputs 1 if $\mathrm{ed}(x,y) \leq t$ and 0 if $\mathrm{ed}(x,y) \geq 2^{c\sqrt{\log n \log\log n}} \cdot t$.*

Proof. Suppose Alice gets x and Bob gets y. Alice computes $s(x)$, Bob computes $s(y)$ (on the same random string), then they exchange these bits. They output 1 if and only if $\mathcal{H}(s(x), s(y)) \leq \alpha \log(1/\delta)$. By Theorem 11, the protocol succeeds with probability at least $1 - \delta$. □

Another obvious application is approximate nearest neighbor search. In the nearest neighbor search problem, a database consisting of n points in an ambient distance space must be pre-processed, and a search algorithm must find a database point closest to an input point, using the pre-processed database. In the approximation version, the search algorithm may return a database point at distance somewhat larger than the minimum. Using approximate nearest neighbor algorithms for points in ℓ_1 [9, 8], we get the following theorem.

Theorem 13. *There is a probabilistic pre-processing algorithm D and a search algorithm N that satisfy the following conditions with high probability. On any input $X \subset \binom{\{0,1\}^d}{n}$, the pre-processing algorithm D computes in polynomial time a pre-processed database $D(X) = D(X, n, d)$. Using $D(X)$, on any input $q \in \{0,1\}^d$, the search algorithm N finds in time polynomial in d and $\log n$ a database point $N(q) \in X$ such that for every $x \in X$, $\mathrm{ed}(q, N(q)) \leq 2^{O(\sqrt{\log d \log\log d})} \cdot \mathrm{ed}(q, x)$.*

Proof. Let φ be the algorithm from Theorem 10, taking $\delta \ll 2^{-2d}$. (So, φ embeds $\left(\{0,1\}^d, \mathrm{ed}\right)$ into $\ell_1^{O(d^3)}$.) To pre-process the database X, apply the ℓ_1 pre-processing algorithm of [9] to the database $\{\varphi(x) : x \in X\}$. To search a query q, apply the ℓ_1 search algorithm of [9] to $\varphi(q)$. □

[2] The comparison of $s(x)$ and $s(y)$ is done using the same outcome of the algorithm's coin tosses in both cases.

References

[1] A. Andoni, M. Deza, A. Gupta, P. Indyk, and S. Raskhodnikova. Lower bounds for embedding edit distance into normed spaces. In *Proc. SODA 2003*, pages 523–526.

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

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

[4] A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the Web. In *Proc. of the 6th Int'l World Wide Web Conf.*, 1997, pages 391–404.

[5] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. In *Proc. SODA 2002*, pages 667–676.

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

[7] P. Indyk. Approximate nearest neighbor under edit distance via product metrics. In *Proc. SODA 2004*, pages 646–650.

[8] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. STOC 1998*, pages 604–613.

[9] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM Journal on Computing*, 30(2):457–474, 2000. (Preliminary version appeared in *Proc. STOC 1998*.)

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

[11] W.J. Masek and M.S. Paterson. A faster algorithm for computing string edit distnace. *Journal of Computer and Systems Sciences*, 20(1):18-31, 1980.

[12] S. Muthukrishnan and S.C. Sahinalp. Approximate nearest neighbors and sequence comparison with block operations. In *Proc. STOC 2000*, pages 416–424.

Appendix

Lemma 14. *Consider a non-negative function $f : \mathbb{N} \to \mathbb{R}$ that satisfies the following. There exist $n_0 \in \mathbb{N}$ and constants $\alpha, \beta > 0$ such that for every $n > n_0$*

$$f(n) \le f\left(\frac{n}{2^{\sqrt{\log n \log\log n}}}\right) \cdot (\log n)^{\alpha} + 2^{\beta\sqrt{\log n \log\log n}}.$$

Then, there exists a constant $c > 0$ such that for every $n \in \mathbb{N}$, $f(n) \leq 2^{c\sqrt{\log n \log\log n}}$.

Proof. The proof is by induction on n. For the base case, if we take a sufficiently large constants c and $n_1 \geq n_0$, then for all $n \leq n_1$, $f(n) \leq 2^{c\sqrt{\log n \log\log n}}$, and for all $n > n_1$,

$$2^{c\sqrt{(\log n - \sqrt{\log n \log\log n})\log\log n}} \cdot (\log n)^{\alpha} \geq 2^{\beta\sqrt{\log n \log\log n}}.$$

So, let $n > n_1$ and assume that the claim holds for all $n' < n$. We have

$$\begin{aligned}
\log f(n) &\leq \log\left(f\left(\frac{n}{2^{\sqrt{\log n \log\log n}}}\right) \cdot (\log n)^{\alpha} + \right. \\
&\quad \left. + 2^{\beta\sqrt{\log n \log\log n}}\right) \\
&\leq \log\left(2^{c\sqrt{(\log n - \sqrt{\log n \log\log n})\log\log n}} \cdot (\log n)^{\alpha} + \right. \\
&\quad \left. + 2^{\beta\sqrt{\log n \log\log n}}\right) \\
&\leq \log\left(2 \cdot 2^{c\sqrt{(\log n - \sqrt{\log n \log\log n})\log\log n}} \cdot \right. \\
&\quad \cdot (\log n)^{\alpha}) \\
&= c\sqrt{(\log n - \sqrt{\log n \log\log n})\log\log n} + \\
&\quad + \alpha \log\log n + 1 \\
&= c\sqrt{\log n \log\log n} \cdot \left(\sqrt{1 - \sqrt{\frac{\log\log n}{\log n}}} + \right. \\
&\quad \left. + \frac{\alpha}{c}\sqrt{\frac{\log\log n}{\log n}} + \frac{1}{c\sqrt{\log n \log\log n}}\right) \\
&\leq c\sqrt{\log n \log\log n} \cdot \left(1 - \frac{1}{2}\sqrt{\frac{\log\log n}{\log n}} + \right. \\
&\quad \left. + \frac{\alpha}{c}\sqrt{\frac{\log\log n}{\log n}} + \frac{1}{c\sqrt{\log n \log\log n}}\right) \\
&\leq c\sqrt{\log n \log\log n},
\end{aligned}$$

provided that c and n_1 are sufficiently large. □