Big Data: Final report

Contents

Chapter 1

Preliminaries

In the preliminaries chapter, we introduce several terminologies and notations which will be used throughout this report.

Definition 1. *For two strings $x, y \in \Gamma^*$, $x \oplus y$ is the concatenation of x and y. For two embedding $f, g : D \to l_1$, we denote $f \oplus g$ as the embedding such that maps x into $f(x) \oplus f(y)$.*

Naturally, for a series of embedding $\{f_i\}, f_i : D \to l_1$, we can define $\bigoplus_i f_i$ as well.

TEMD(thresholded version of EMD) is a key ingredient in the OR construction. We introduce its definition.

Definition 2. *Threshold earth mover distance(TEMD) on $(\{0,1\}^d)^s$ with threshold t is defined as*

$$\mathrm{TEMD}_t(A, B) = \frac{1}{s} \min_{\sigma} \sum_{a \in A} \min(t, |a - \sigma_a|_1)$$

.

In which σ is a bijection from $A \to B$.

We also introduce the original EMD here:

Definition 3. *Earth mover distance(EMD) on $(\{0,1\}^d)^s$ is defined as*

$$\mathrm{EMD}(A, B) = \frac{1}{s} \min_{\sigma} \sum_{a \in A} |a - \sigma_a|_1$$

.

In which σ is a bijection from $A \to B$.

Remark 1. *It can be interpreted as the smallest work needed to transfer a set of elements to another set of elements, and work is measured by the underlying ground metric.*

Definition 4. *$B(x, r)$ means the set $\{y \mid dist(x, y) \leq r\}$.*

OR is an abbreviation for Ostrovsky and Rabani[**?**].

Chapter 2

Survey on the existed result: Upper bound

In this chapter, we survey the results for upper bound for the embedding into l_1.

For embedding the general edit distance into l_1, no progress has ever been done since the OR algorithm in 2006, which has distortion $2^{O(\sqrt{\log d \log\log d})}$[**?**].

But people do have made several progress on related problems. First of all, the construction in OR's original paper needs $O(n^2)$ time, which is highly non-practical, so we will also introduce a practical non-oblivious algorithm which runs in $2^{O(\sqrt{\log d \log\log d})}$ time[**?**].

Since it might be hard to embed the whole edit distance, we may want to embed some of its subset first. In 2006, it has been proved that we can embed the Ulam metric(the edit distance over non-repetitive strings) into l_1 with distortion $O(\log d)$[**?**].

Another direction is that we know insertion and deletion cause us lots of trouble, what if we only allowed free cycle shift to the string x(the so-called shift metric) to make our life easier? In[**?**], an $O(\log^2 d)$ embedding for shift metric is introduced.

Finally, since we need those embeddings because the low distortion embedding into l_1 will bring us efficient algorithm for several important task, we can switch our goal a little bit by constructing a metric which is both easy to construct the embedding and has good structure for designing the algorithm. In [**?**], they introduce a so-called product metric which admits efficient algorithms for several important task and showed the Ulam metric can be embedded in it with constant distortion.

2.1 The OR embedding for general edit distance

In this section we will first present the essences of the OR's $2^{O(\sqrt{\log d \log\log d})}$ embedding, and then analyze it to show that why this embedding is tight and why it is hard to improve it trivially.

Before start the short overview of the OR's algorithm, it is fruitful to point out several key properties of it:

- It is an overall recursive algorithm, which builds the embedding for $\{0,1\}^d$ from smaller embedding for length $\leq d/2^{\sqrt{\log d \log\log d}}$.
- It is an overall block-based algorithm, each block is independent when constructing the mapping. Or more generally, it is a local algorithm. Each bit-flip will only influence a very small part of the result.
- It is constant norm embedding, every string with the same length will have the same norm after the embedding.

Essences of the OR construction

The original paper is full of details, but the key idea is not very complicated.
We can recognize it indeed has two part:

- Using modified TEMD(threshold earth move distance) and embedding for smaller substrings to construct a middle metric M, which approximates edit distance well if the embedding for smaller parts works well.
- Embedding M(TEMD indeed) into l_1(in a relaxed sense) with small distortion, indeed, it might be hard to embed TEMD directly with $O(\log d)$ distortion, but a relaxed embedding is enough, in which the expansion is bounded absolutely, but the contraction are bounded only for a specific case(but this case is enough for bounding the overall contraction because of the property of edit distance).

Now, denote the embedding of a string of length d as φ_d. For a string $x \in \{0,1\}^d$, we decompose it into $B = 2^{\sqrt{\log d \log\log d}}$ parts with equal length(we can assume $|x|$ is a multiplier of B by a simple padding.). Suppose $x = \bigoplus_i x_i$.

It uses a multiple scale scanning method to construct the embedding recursively.
Let $i_{max} = B, s_j = (\log d)^j, d_j = \frac{d}{i_{max}} - s_j + 1$, $j_{max} = \min\{j \in \mathbb{N} \ : \ s_j \geq \frac{d}{i_{max} \log d}\}$.
Now, denote $A_{i,j}(x) = \text{shifts}(x_i, s_j)$ in the original paper. Let $S_{i,j}(x) = \{\varphi_{d_j}(z) \mid z \in A_{i,j}(x)\}$.
Consider the following new metric M.

$$M(x,y) = \sum_{i=1}^{i_{max}} \sum_{j=0}^{j_{max}} 2\ln(2s_j)\text{TEMD}_{t/2\ln(2s_j)}(S_{i,j}(x), S_{i,j}(y)).$$

It can be showed that M approximates the edit distance with distortion:

$$O(\log d)(D_{d/B} + B).$$

In which D_d is the resulting distortion for $\{0,1\}^d$.

Then, they show a simple method which embeds $TEMD$ with distortion $O(\log d)$ (Actually, it is not a strict embedding, but works for the cases that are needed). Optimize the overall recurrence, we obtain:

$$D_d = 2^{O(\sqrt{\log d \log\log d})}.$$

Before analyze the algorithm, we give a brief remark on this algorithm.

Remark of OR's algorithm

OR's algorithm is a very sophisticated construction. It combines many excellent ideas into a wonderful result.

In this subsection, we try to explain the idea and the intuition behind this excellent algorithm, which might be important to make progress upon it. Otherwise, we will just know nothing except it works.

From the previous section, we know that construct an embedding directly is hard, so is naturally to adopt a standard idea, which assumes we have embedding for smaller strings with lower distortion, and build embedding for larger strings upon it.

If we want to use recursive embedding, it is natural to use the embedding for continuous substrings, but a problem comes as how to find those substrings with small edit distance? Lemma 5 in the original paper gives a insight on this, namely, by using the optimal edit sequence, we can find a pairing between substrings in x and y which keep small distance($2\text{ed}(x, y)$) almost everywhere.

Now, denote $S(x, f) = \text{shifts}(x, f)$.

Then, it becomes interesting as how to utilize the existence of such a pairing to construct embedding? Existence of this pairing can be used to bound the expansion of such embedding. But how can we turn this into an embedding for l_1? TEMD! TEMD comes naturally, the existence of the pairing tells us: with small $ed(x, y)$, $\text{TEMD}(S(x, f), S(y, f))$ is small too for every f.

Then, how to bound the contraction? What we want is that: with big $ed(x, y)$, $\text{TEMD}(S(x, f), S(y, f))$ is big too. A careful analysis show that it will be big only if the minimum edge cost from $S(x, f)$ to $S(y, f)$ is on the same magnitude with f. But we don't know x and y beforehand, so a way to deal with it is simply sample f on different scale, and one of them will hit that cost as that cost is decreasing when f is increasing.

Finally, it makes use of Lemma 6 and constructs the whole embedding.

Analysis of OR construction

After studying the OR's construction, the following questions come to our mind: is the distortion bound tight? Can we improve some part of it to construct better embedding?

In the following, we first give an affirmative answer to the first question, and then summarize our thoughts on how to improve the OR construction.

$2^{O(\sqrt{\log n \log\log n})}$ is tight for this algorithm.

We will show the tight is bound by giving a example that reaches the bound.

This is indeed a very trivial fact, because it is a homogeneous block-based algorithm(homogeneous means that for each block, the embedding is the same). Suppose it divides the string into B parts, and use ϕ to map the blocks, for $x = \bigoplus_i x_i$, the mapping is:

$$\varphi(x) = \bigoplus_i \phi(x_i).$$

Then we can see for string $x = \bigoplus_i x_i$ and $y = \bigoplus_i y_i$, the distance after mapping is:

$$\sum_i |\phi(x_i) - \phi(y_i)|_1.$$

Now, suppose the length of each part is m, consider the following strings:

$$S_0 = 01010101\ldots, S_1 = 10101010\ldots$$

and $|S_0| = |S_1| = m$.

Now consider $x_1 = S_0 0^{m(B-1)}$ and $y_1 = S_1 0^{m(B-1)}$(0^N means N duplications of 0). Clearly $ed(x_1, y_1) = ed(S_0, S_1) = 2$.

Also, consider $x_2 = S_0S_1S_0S_1\ldots(\text{B repeats}), y_2 = S_1S_0S_1S_0\ldots(\text{B repeats})$. Clearly $ed(x_2, y_2) = 2$.

But note that after the mapping, we have:

$$|\varphi(x_2) - \varphi(y_2)| = B|\phi(S_0) - \phi(S_1)| = B|\varphi(x_1) - \varphi(y_1)|.$$

It means the distortion is at least $B = 2^{\sqrt{\log d \log\log d}}$ for $\{0,1\}^d$.

Why it is hard to improve the OR's construction trivially

Indeed, there are three important ingredients in OR's algorithm.
First, let us look at its bound for $|\varphi_d|_{Lip}$.

$$|\varphi_d|_{Lip} \leq \frac{\log d}{\log\log d}(4|\varphi_{d/B}|\log d + 2B).$$

- Block-dividing, when approximate $ed(x,y)$ with M, dividing B block will create distortion $O(B)$.
- Multi-scale scanning in M, leaves a gap of $\log d$, and duplication of $\log\log d$.
- Map M, or indeed relaxed TEMD into l_1, impose distortion $\log d$.

Also, consider the bound for $|\varphi_d^{-1}|_{Lip}$.
Look at the analysis, we can see that the distortion here comes from the gap between the multi-scale scanning, it is indeed:

$$|\varphi_d^{-1}|_{Lip} \leq 2(|\varphi_{d/B}^{-1}|_{Lip} + 2)gap = 2(|\varphi_{d/B}^{-1}|_{Lip} + 2)\log d.$$

From this, we can immediately realise that the multi-scale scanning with gap $\log d$ gives an improvement of $\log\log d$ in the bound for $|\varphi_d|_{Lip}$. But anyway, it does not affect the overall complexity($2^{O(\sqrt{\log d \log\log d})}$) if we looks at it carefully, it can only makes the constant c in $2^{c\sqrt{\log d \log\log d}}$ smaller. (But anyway, it is a polynomial improvement.)
Note that, in order to get better embedding, we need to improve part 2 and part 3 at the same time.
Now, let us discuss the potential improvement for OR's algorithm for these three ingredients.

1.Block-dividing

From the analysis of its tightness, we already show that any block-dividing approach will suffer a distortion of B. If we insist in dividing the blocks first, it is impossible to reduce the distortion here.

2.Multi-scale scanning

The first question is, why we need multi-scale scanning here? The key point of it is that when upper bounding $|\varphi_d^{-1}|_{Lip}$. denote $M_{i,j} = \min_{a\in S_{i,j}(x), b\in S_{i,j}(y)} ed(x,y)$. TEMD$(S_{i,j}(x), S_{i,j}(y))$ gives a meaningful upper bound for $ed(x_i, y_i)$ only when s_j and $M_{i,j}$ are roughly the same. So indeed we need to sample s_j with different magnitude in this method inherently. And the truth is, without Multi-scale scanning, we don't know any method to obtain a meaningful bound on the contraction.
Another remark is that when the multi-scale gap is g, its contribution to the $|\varphi_d|_{Lip}$ is $\frac{\log d}{\log g}$, make it bigger can give some minor improvement in the bound for $|\varphi_d|_{Lip}$. But this won't affect anything. Indeed, make g to be a constant C or any polynomial of $\log d$ gives the same overall complexity.

3.Embedding relaxed TEMD into l_1

This part is quite technical, but indeed we conjecture it is tight as witnessed by the random case.

Key point detection-based method

From the above, we can see that it is hard to improve OR's construction trivially(Of course it should be that, otherwise why the original authors haven't done it?)

The idea and the intuition

Indeed, we can see that the most important part in OR is part 1: Block-dividing, the part 2 and part 3 all works for it. So an appealing idea is to design a different way to build the embedding recursively.

Note that, block embedding is oblivious in the sense it divides the block oblivious to the content of the strings, every strings are divided by the same way. An advantage of it is the natural alignment between blocks make it easier to design embedding. But it also has disadvantages for losing many global informations. So what if we use a non-oblivious method here?

2.2 Fast non-oblivious embedding for all substrings of particular string

We know that OR's construction gives an embedding works for every string, but implement this embedding is quite slow, even probabilistic version needs $O(n^2)$ time.

But generally, we don't need to find embedding works for every string, we just need embedding for strings we are interested(like sketching model or approximation algorithm), can we make use of it to speed up the embedding process?

This paper[**?**] gives an affirmative answer to this question, the main result from it is:

Theorem 1. *Given string x, we can compute an embedding into l_1 for substring of it in time $O(n2^{O(\sqrt{\log d \log\log d})})$, and this embedding has distortion $2^{O(\sqrt{\log d \log\log d})}$ for every substrings of x.*

Look at the the details in OR construction, we can realize that the computational bottleneck is the embedding from TEMD to l_1, if we can improve that part, we will obtain a better result.

One thing to notice is that suppose we have computed the embedding for TEMD for strings of length s in interval $[l, r]$, suppose we move to $[l+1, r+1]$, we only change the set by one insertion and one deletion, if we can come up with an good embedding for TEMD which utilize that to speed that procedure up, we will obtain a faster algorithm.

The algorithm itself is quite complicated and full of details, which make use of several important results and technique in embedding(some of them indeed reminds me of the big data streaming algorithm). So we won't cover its details here, if you're interested, you can refer the original paper.

But the key part of it is a linear probabilistic embedding for EMD into l_1. Since it is a linear embedding, a streaming fashion algorithm can be used to compute the embedding for every contiguous block of the whole string very fast.

And the non-oblivious part is using the Bourgain's theorem to approximate the shortest path metric.

Application

This embedding can be used to approximate the edit distance between two string x and y, concatenates x and y into a single string, compute the embeddings for its first half and second half, and return the l_1 distance between them is enough.

2.3 The $O(\log d)$ distortion embedding for Ulam metric

In this section we present the main part for the $O(\log d)$ embedding for the Ulam metric and gives our remark of it. Although it is a quite strong result, we don't think the method can be applied to the general embedding.

Let Γ be the alphabet set(note that the distortion doesn't depend on the size of the alphabet.)

The word "permutation" in the original paper might be misleading, this embedding actually works for every string with no duplication, it is not required that every alphabet occurred in the string.

Embedding

The embedding is simple and short, let $p \in \Gamma^d$ and p is a permutation. The embedding f maps p into $l_1^{\binom{|\Gamma|}{2}}$, for each $a, b < \Gamma$ and $a < b$:

if both $a, b \in p$:

$$f(p)_{a,b} = \frac{1}{|p^{-1}(a) - p^{-1}(b)|}.$$

Otherwise:

$$f(p)_{a,b} = 0.$$

Proof the correctness

To bound its expansion, it is sufficient to prove it for each pair of strings with edit distance 1.

With a little elementary calculation, it can be shown the expansion is $O(\log d)$.

Also, the contraction is $O(1)$ here, but the proof is quite long, highly non-trivial and exploits several clever ideas, it based on a technique which randomly partition one sequence into two subsequences.

Remark

It is a pretty strong result for Ulam metric, but the key part of its proof is based on the fact that there are no repetitions in the string. After thinking about it carefully, one can realize that it makes no sense to down sample a string in $\{0, 1\}^n$ as did in the proof.

2.4 The $O(\log^2 d)$ embedding for shift metric

In the previous section, we review a partial result which gives a good embedding for a subset of the general string. But for the general string, things are much worse. So, one idea is that we design embedding for some weak edit distance first.

In edit distance metric, we notice that the insertion and deletion operation caused us most troubles. So what if we only allow cycle shift on the string? To be precise, consider the shortest metric on the following graph(shift metric), from one point $x \in \{0,1\}^d$:

- substitute one position cost 1.
- shift x cyclically one step cost 0.

In the survey for the lower bound part, the technique in that can also be used to prove the lower bound for the shift metric is also $O(\log d)$.

In [**?**], a $O(\log^2 d)$ embedding for shift metric are devised, suggesting that cycle shift is far more weak than the general insertion and deletion.

In this section, we will briefly review the construction and the proof of this embedding, and gives our remark on that.

The construction

Let $cyc(x)$ be the set of all rotations of the string x, so that $|cyc(x)| = d$.

Then we can see that the distance between x and y in the shift metric is just:

$$\min_{a \in cyc(x), b \in cyc(y)} H(a,b).$$

in which $H(a,b)$ is the hamming distance between a and b.

Obverse that this is just the earth mover distance for $cyc(x)$ and $cyc(b)$.

So it suffices to give a good embedding for the earth mover distance over the multi-sets of size d over $\{0,1\}^d$. Actually, this paper mainly talks about the embedding for the earth mover distance, this is just a side product of that embedding.

Let $\Delta \in \mathbb{N}$, for $x, y \in [\Delta]^d$, let $H(x,y)$ be the hamming distance between them, which is number of the positions i such that $x_i \neq y_i$.

We have the following theorem:

Theorem 2. *Let $d, \Delta > 0$, there exists an embedding ψ of EMD over $[\Delta]^d$ for sets with size s with distortion $O(\log s \cdot \log(d\Delta))$.*

The construction goes with 2 steps, in the first step, it gives a randomized embedding for the hamming distance over $[\Delta]^d$ to the weighted hamming distance H_w with very lower dimension of $O(\log(d\Delta))$(kind of dimensional reduction!), then it embeds H_w into a tree metric with constant distortion, since EMD over tree metric can be embedded into l_1 with constant distortion, it completes the construction.

We omit the details here since it is long and complicated, and not quite relevant for our following discussion, if you are interested, you can check the original paper.

Remark

Since EMD is a key ingredient in the OR construction, it is definitely fruitful to study how to improve it. But also we should notice that improving EMD along won't improve the whole complexity of OR's algorithm.

Indeed, we don't need a whole embedding for TEMD in order to make the OR's construction works. And after reading the construction in this paper, we don't think it is connected to the general edit distance in any sense. But anyway, it is a good result.

2.5 The constant distortion Embedding for Ulam metric into a computational nice metric

We know that l_1 is indeed a mathematically nice metric, the metric itself is simple and elegant, but it also causes troubles for designing the embedding algorithm. Since what we really want from an embedding is the the power to design efficient algorithm for several important task, we can make our life easier by embedding the edit distance into an metric which might be more complicated than l_1, but still admit efficient algorithms.

In [**?**], a notation of product-metric is introduced, in that paper, they design an constant distortion embedding to a specific product-metric, and then show that metric admits efficient algorithm for nearest neighbor, sketching algorithm and edit distance estimation.

Despite it is very interesting in its own right, as it get rid of the embedding into l_1, it is not closely related to our project. So in this section, we will just briefly review the construction and the proof for this embedding, and give our remark on their contribution.

The construction

Definition 5. *Let $M_1, M_2, \ldots, M_k$ be k metric, let $\bigoplus_{l_p}^k M$ be the metric such that the distance between two point x, y is $\left(\sum_{i=1}^k M_i(x,y)^p\right)^{1/p}$. We define $\bigoplus_{l_\infty}^k M$ and $\bigoplus_{l_2^2}^k M$ in the same way.*

It is called product metric because it builds one metric on the results of other metrics, in this paper, the product metric which are used are $\bigoplus_{l_2^2}^d \bigoplus_{l_\infty}^{O(\log d)} {l_1}^{2d}$.

Now, let the length of permutations be d, for one permutation P, let $P[i]$ denotes its i-th element, and for convenience, let $P[-i] = -i$ and extent the alphabet to $\{-d+1, \ldots, 0, 1, \ldots, d\}$, let $P^{-1}(a)$ be the position of a in permutation P. Denote $P_{a,k}$ be the set of $\{P[P^{-1}(a)], P[P^{-1}(a)-1], P[P^{-1}(a)-2], \ldots, P[P^{-1}(a)-k+1]\}$.

Let $\varphi_{a,k}$ to be the incidence vector scaled by $\frac{1}{2k}$ in $\{0, \frac{1}{2k}\}^{2d}$ for $P_{a,k}$. Let K be every integer powers of $(1+\gamma)$ which are smaller than d, then let $\varphi_a = \bigoplus_{l_\infty}^{k\in K} \varphi_{a,k}$(a bit abuse of notation, but is fine here.)

Finally, the whole embedding $\varphi(P) = \bigoplus_{l_2^2}^{a\in[d]} \varphi_a$.

And we have the following theorem:

Theorem 3. *φ has constant distortion.*

The proof is based one the following fact:

Lemma 1. *Let P, Q be two permutations, and let $0 < \delta < \frac{1}{2}$. Let T_δ be the set containing all symbols $a \in [d]$ for which there exists $k \in [d]$ such that the symmetric difference $|P_{a,k} \triangle Q_{a,k}| > 2\delta k$. Then:*

$$\frac{1}{2} ed(P,Q) \le |T_\delta| \le \frac{4}{\delta} ed(P,Q).$$

Based on that lemma, we can prove φ has constant distortion in a very straightforward way.

Another lemma states that such product metric admits efficient algorithm for NNS(nearest neighbor search):

Lemma 2. *For every $k, l, m > 1$ and $\varepsilon > 0$, the metric $\bigoplus_{l_2^2}^k \bigoplus_{l_\infty}^l {l_1}^m$. admits an NNS scheme achieving approximation $O(\varepsilon^{-3} \log\log n)$, query time $(klm)^{O(1)} n^\varepsilon$, and space $(klm)^{O(1)} n^{2+\varepsilon}$.*

Combining these two results, we obtain a sub polynomial NNS algorithm with approximation $O(\log\log n)$, it can be strengthened to $O(\log\log d)$ by other techniques which we omit here.

Remark

This result is certainly worth our noticing, after reading this paper, as this only talks about Ulam metric, we begin to think is it possible to construct some product metric so that the result for general edit distance space can also be improved?

Note that indeed, such a product metric comes from some combinatorial estimation scheme of the edit distance. Of course, the l_1 embedding is also a combinatorial estimation scheme, but in the product metric, we have far more tools than in l_1, for example, since we have l_∞, we can take the maximum without loss any thing, in l_1, we need to add them up and loss of factor the number of addend.

So if we can find something like the lemma 1, we can also design an good algorithm for l_1. But the sad thing is that, those combinatorial estimation scheme for Ulam metric rely entirely on the non-repetitive properties of the string. With out this assumption, we really don't know how to design an estimation scheme, I think if even it is possible, it won't be a straightforward one like those for Ulam metric, it has to make use of recursive design.

Chapter 3

Survey the existed result: the lower bound

In this chapter we will survey the existed results for this problem on the lower bound side. Includes the essences of the several proofs for the lower bound. We also emphasis the intuitions behind their proof, and analyze their limitations and suggest the potential approaches to strength the lower bound.

The first non-trivial lower bound is that the distortion for embedding into l_1 is at least 1.5 [**?**] in 2003, which eliminates the possibility for an isometric embedding. Then in 2007, It had been strengthened to $\Omega(\log n)$[**?**].

After that, although no progress has been done for the lower bound for embedding into the l_1, there are lower bound has been proven for other metric. We know that l_2^2(the square of l_2) is stronger than l_1 in the sense that l_1 can be embedded isometrically into it. In 2009, It has been proved the lower bound for embedding into l_2^2 is at $\Omega(\frac{\log d}{\log \log d})$[**?**]. Indeed, this result is a communication complexity result, which is very powerful and interested in its own right.

3.1 The 1.5 lower bound

In this section, we will introduce the essences of the technique of the 1.5 lower bound from [**?**]. And gives our remark on it.

This proof is quite simple, since we don't know how to apply the existing lower bound technique to obtain over the whole edit distance graph. What about extract a subgraph from the whole graph which we know how to prove lower bound?

It contains two parts, each part is straightforward:

- The edit distance graph over $\{0,1\}^{2n}$ contains the shortest distance metric over $K_{2,n}$ as a subgraph.
- The metric family $\{K_{2,n}\}$ has embedding lower bound 1.5.

Then the lower bound comes naturally.

The first part is proved by a simple construction:

Lemma 3. *The edit distance over $\{0,1\}^{2n}$ contains $K_{2,n}$ as a subgraph.*

Proof. Let $A_1 = (01)^n, A2 = (01)^{n-1}$. And B_i to be the string that deleting the i-th zero from A_1.

Then we can see:

- $ed(A_1, A_2) = 2$.
- $\forall i, j\ ed(A_i, B_j) = 1$.
- $\forall i, j\ ed(B_i, B_j) = 2$.

Which is identical to $K_{2,n}$ if we treat A_1, A_2 as the left part and $B_1, B_2, \ldots, B_n$ as the right part. □

To prove the second part, we need two previous result:

Lemma 4. *l_1 can be isometrically embedded into the square of l_2(denote it as l_2^2 later). l_2^2 means that two points x, y have distance $\|x-y\|_2^2$. Which means the lower bound for l_2^2 will imply the lower bound for l_1.*

Lemma 5. *Any point set X in l_2^2 satisfy the so called negative-type inequality: Let $X = \{x_1, x_2, \ldots x_n\}$ in l_2^2, for any $b_1, b_2, \ldots, b_n \in \mathbb{R}$ such that $\sum_i b_i = 0$. We have:*

$$\sum_{i,j} b_i b_j \|x_i - x_j\|_2^2 \leq 0.$$

Now, let f be the embedding from $K_{2,n}$ to l_2^2. With out loss of generality, suppose after the embedding, the distance will be no shorter than the original one(we can apply a global shift to achieve this). Let $a_1, a_2 = \frac{n}{2}$ and $b_i = 1$ for any i, then $a_1 + a_2 + \sum_{i=1}^{n} b_i = 0$.

Let $c = max_{i,j}\|f(A_i) - f(B_j)\|_2^2$. . Then we have:

$$\begin{aligned}
&a_1 a_2 ed(A_1, A_2) + \sum_{1 \leq i < j \leq n} b_i b_j ed(B_i, B_j) + \sum_{i=1}^{2}\sum_{j=1}^{n} a_i b_j c \leq 0 \\
\Rightarrow &2\left(\frac{n}{2}\right)^2 + \binom{n}{2} \cdot 2 \leq 2 \cdot n \left(\frac{n}{2}\right) \cdot c \\
\Rightarrow &c \geq \frac{3}{2} - \frac{1}{n}.
\end{aligned}$$

Since n can be arbitrarily large, we conclude that the graph family $\{K_{2,n}\}$ can not be embedded into l_2^2 for distortion better than 1.5

Remark

First it can be shown that $K_{2,n}$ is possible to be embedded in l_1 with distortion 1.5, thus the bound is tight for this technique.

We can see that in this proving, nearly everything about the complexity of the edit distance graph is untouched. Of course $K_{2,n}$ is far more simple than the whole graph.

But still, how can we improve this lower bound? With some human ingenuity, it might be possible to find some other subgraph in edit distance with better unembeddablity result in the edit distance graph.

But this kind of method is very naive in the sense that it discard many information about the whole graph. So generally we don't think this kind of proof will leads us to lower bound better than $\Omega(\log n)$.

3.2 The $O(\log n)$ lower bound

The proof of the lower bound is pretty straight forward.
It indeed makes use of a proof framework. We will first present the framework here.
Let τ and η be two arbitrary probability distribution over $\{0,1\}^n \times \{0,1\}^n$. If:

$$\mathbb{E}_\tau[ed(x,y)] \leq \alpha \cdot \mathbb{E}_\eta[ed(x,y)].$$

And for every boolean function $f : \{0,1\}^n \to \{0,1\}$, we have:

$$\mathbb{E}_\tau|f(x) - f(y)| \geq \beta \cdot \mathbb{E}_\eta|f(x) - f(y)|.$$

Lemma 6. *Any embedding for edit distance from $\{0,1\}^n$ to l_1 must have distortion $\frac{\beta}{\alpha}$, if the above requirements are satisfied,*

Proof. Any embedding φ from $\{0,1\}^n \to l_1$ can be written as a convex combination of finite number of boolean function: $\varphi = \sum_{i=1}^N a_i f_i$ in Which $a_i \geq 0$ and $f_i : \{0,1\}^n \to \{0,1\}$. Such that $|\varphi(x) - \varphi(y)|_1 = \sum_{i=1}^N a_i |f_i(x) - f_i(y)|$.
Then we know that for any embedding φ, we also have:

$$\mathbb{E}_\tau|\varphi(x) - \varphi(y)| \geq \beta \cdot \mathbb{E}_\eta|\varphi(x) - \varphi(y)|.$$

Then we have:

$$\frac{\mathbb{E}_\tau|\varphi(x) - \varphi(y)|}{\mathbb{E}_\tau[ed(x,y)]} \geq \frac{\beta}{\alpha}\frac{\mathbb{E}_\eta|\varphi(x) - \varphi(y)|}{\mathbb{E}_\eta[ed(x,y)]}.$$

Suppose φ has distortion D and :

$$ed(x,y) \leq |\varphi(x) - \varphi(y)| \leq D \cdot ed(x,y).\ \forall x, y$$

Then we know that:

$$D \geq \frac{\mathbb{E}_\tau|\varphi(x) - \varphi(y)|}{\mathbb{E}_\tau[ed(x,y)]} \geq \frac{\beta}{\alpha}\frac{\mathbb{E}_\eta|\varphi(x) - \varphi(y)|}{\mathbb{E}_\eta[ed(x,y)]} \geq \frac{\beta}{\alpha}.$$

Which concludes the proof. □

Then we define two distributions τ and η on $\{0,1\}^n \times \{0,1\}^n$.

- η is the uniform distribution over all the pairs.
- To define τ, let $cyc(x)$ be the string obtained from x shifted to the left (take the front character and put it at the end). Let $S = \{(x, cyc(x)) \mid x \in \{0,1\}^n\}$, and τ_S is the uniform distribution over S. And let H be the set of pair which have hamming distance exactly 1, and let τ_H be the uniform distribution over T. Then let $\tau = \frac{1}{2}(\tau_S + \tau_H)$.

Note that τ only consists of string pairs which are very close.
Then, since $E_\tau[ed(x,y)]$ is $O(1)$, and $E_\eta[ed(x,y)]$ is $\Omega(n)$, we know that.

$$\mathbb{E}_\tau[ed(x,y)] \leq O(\frac{1}{n}) \cdot \mathbb{E}_\eta[ed(x,y)].$$

By methods in the boolean function analysis(we omit the details here since it is tedious and irrelevant for our following discuss), we can prove that:
for every boolean function $f : \{0,1\}^N \to \{0,1\}$, we have:

$$\mathbb{E}_\tau|f(x) - f(y)| > \Omega(\frac{\log d}{d}) \cdot \mathbb{E}_\eta|f(x) - f(y)|.$$

Limitation of this proof technique

First, note that there exists such function f that:

$$\mathbb{E}_\tau |f(x) - f(y)| < O(\frac{\log d}{d}) \cdot \mathbb{E}_\eta |f(x) - f(y)|.$$

The f is quite simple, fix some $k = \Theta(\log d)$, $f(x) = 1$ if 0^k is a substring of x (we allow it to wrap around x).

We should notice that the boolean function is indeed a tool here, what it really says is:

Fact 1. *The graph induced by the distribution τ has edge expansion $\Theta(\frac{\log d}{d})$.*

So, a naturally idea to improve this result is to consider another distribution τ' on edges such that we can prove a stronger expansion result. Since indeed τ only makes use of the cycle shift and the substitution operations, which are far from capturing the whole complexity of the edit distance space.

This seems plausible, but notice that what this proof technique proved is indeed a "expected distortion" lower bound, which is stronger than the worst case distortion. Recall that in the previous chapter, we have the following results.

Lemma 7. *There exists an $O(\log^2 d)$ embedding φ for subset of string $V \subset \{0,1\}^d$ such that $\frac{V}{2^d}$ is $1 - o(1)$.*

Extends φ into an embedding for $\{0,1\}^d$ by map strings not in V into a fixed vector.

Then we can see that, for any distribution τ if:

$$\Pr_{(i,j)\sim\eta} |i \in V \wedge j \in V| = 1 - o(1).$$

Then we can see that:

$$\frac{\mathbb{E}_\tau |\varphi(x) - \varphi(y)|}{\mathbb{E}_\tau [ed(x,y)]} = O(\log^2 d).$$

Which means that, there do exist embedding whose expected distortion is $O(\log^2 d)$, thus, if τ's "mass" is concentrated in V, we can not prove lower bound better than $O(\log^2 d)$ by this technique.

Generally, we believe that there exists embedding better than $O(\log^2 d)$ for random string, specifically, we have the following conjecture:

Conjecture 1. *For every d, There exists $O(\log d)$ embedding for $1 - o(1)$ fractions of strings in $\{0,1\}^d$.*

Conclusion

Our observation here suggest that in order to prove better lower bound for this problem, it seems plausible to construct some distribution concentrate on those highly self repeated strings(Recall that we have good embedding for $O(\log d)$-repetition-free string), or equivalently, prove the graph consisted or those strings has a very high edge expansion.

To be specific, this graph should have two feature:

1. It also has edges representing insertion and deletion.
2. Its vertices are highly self repeated.

But it is extremely hard to prove an edge expansion result on this graph, since we introducing the insertion and deletion here, it becomes unclear how to adopt the ideas from boolean analysis, and it is also unclear how the self repentance can be utilized in the proof.

Remark

It is really a big progress from 1.5 to $\Omega(\log n)$. So it is worthwhile to give our remark and our understanding of such a great improvement.

The first observation is that both proof extract a sub-metric from the whole graph and prove an property for that subgraph which imply the lower bound for the whole graph.

The 1.5 lower bound did it in an naive way by extracting $K_{2,n}$, since $K_{2,n}$ is far more simple than the whole graph, it is easy to prove the lower bound. But the result will be weak too.

The $\Omega(\log n)$ lower bound did it in a more clever way, what it indeed extracted is the so called shift metric, but is has its own limitation too, consider the shortest metric on the following graph, from one point $x \in \{0,1\}^n$:

- substitute one position cost 1.
- shift x cyclically one step cost 0.

It is easy to see that the proof remain valid for shift metric too, so shift metric also has the $\Omega(\log n)$ lower bound. But the shift metric can be embedded with distortion $O(\log^2 n)$, which suggests it is still less complex than the original graph.

And to prove the result for $K_{2,n}$, we only use some trivial result. But to prove the result for the shift metric, we need some involved result in boolean function analysis.

Indeed, the more complex the subgraph we extract, the harder to prove the result. Mankind are stuck here mainly because we have no tool to analyze the arbitrary insertion and deletion operations, but without them the subgraph are too simple in the sense that it carries few informations.

3.3 The communication complexity based lower bound

Another idea is that: we study the embedding into l_1 because that embedding will imply a good algorithm for some problem. What if we turn this around and prove a lower bound for such an algorithm? Then it will also imply a lower bound for the embedding itself.

So, what can be done with an embedding? The most interesting application is the so-called sketching model. In the sketching model, there is a randomize algorithm sk such that for any $x, y \in \{0,1\}^n$, only with $sk(x), sk(y)$, w.h.p. we can approximate the edit distance $ed(x, y)$.

It is important since the following lemma[**?**]:

Lemma 8. *For l_2^2, or any l_2^p for $p \geq 1$, there is a sketching algorithm sk with $O(1)$ output length and approximate the distance with $O(1)$ distortion w.h.p.*

Which immediately imply that:

Corollary 1. *If there is an embedding from $\{0,1\}^n$ with distortion D into l_1 or $l_2^p (p \geq 1)$, then there is a sketching algorithm sk with $O(1)$ output length and approximate the edit distance with distortion $O(D)$.*

Proof. We can just apply the sketching algorithm for l_1(or l_2^p) after the embedding. □

Also, a sketching algorithm will also imply a communication protocol for estimating the edit distance.

We define the decisional version for the approximation protocol:

Definition 6. $\mathrm{CC}_{R,\alpha,d}$ *is the randomized communication complexity for the following problem: for any two string $x, y \in \{0,1\}^d$, deciding whether $ed(x, y) < R/\alpha$ or $ed(x, y) > R$.*

In this paper, the following result are proved:

Theorem 4. *Let $d > 1$ and $\alpha = \alpha(d) > 1$. Then, there exists constants $c > 0$ and $0 < c_1 < c_2 < 1$, such that for all R satisfying $d^{c_1} \leq R \leq d^{c_2}$, $c \cdot \mathrm{CC}_{R,\alpha,d} + log(\alpha \log \alpha) \geq \log \log d$.*

Then, since a D distortion embedding will imply an $O(1)$ length and $O(D)$ approximation sketching algorithm, which will then imply an $O(1)$ communication protocol for $O(D)$ approximation. If D is not $\Omega(\frac{\log d}{\log \log d})$. Plug it in the previous theorem, we can get a contradiction. Then we can prove the following very strong result:

Corollary 2. *Any embedding into metric which admits a $O(1)$ length, $O(1)$ distortion randomized sketching algorithm will have distortion at least $\Omega(\frac{\log d}{\log \log d})$.*

The details of the proof are very complex, basically, to prove the randomize communication complexity, a common method from Yao's minimax theorem is to design a hard distribution over the input, and prove that every deterministic algorithm will do badly on it. The distribution over (x, y) is roughly pick x at uniform, and obtain y by first apply an random noise N_ρ on x, and then perform a permutation drawn from an distribution D_t on the coordinates of it. We omit the details here since it is quite complex and involved, you can refer to the original paper if you're interested.

Remark

Despite that this work is very interesting and implies really a strong lower bound result. But we can see that the lower bound for the communication complexity result is indeed a much more powerful result than the lower bound for the distortion of the embedding.

But since we do have many tools in communication complexity, can we utilise them to prove a good result for it? It might be an approach worth to try, but we believe it will be very tough as indicated by the length of this proof.

Chapter 4

Failed attempts and the reason

In this chapter we will summarize our attempts on this problem, and in the meantime, explaining why our attempts failed and the insight we got from our failure.

There are three attempts which we have made:

- A naive idea, build the embedding from random walk.
- Recursively construct the embedding from local embedding.
- Merge the existed good embeddings.

Although none of them leads to significant result, we have understood that why they won't work, and obtained some side products(embedding for sparse strings and small local of random string).

4.1 A naive idea of graph random walk

The idea and the intuition

The first idea comes from the intuition that we may use the underlying graph directly to construct an embedding.

Namely, the edit distance graph can be seen as a shortest path metric on the following graph G:

- Vertex set $V = \{0,1\}^n \bigcup \{0,1\}^{n-1}$, each vertex represent a 01 string of length n or length $n-1$.
- Edge set $E = \{(x,y) \mid ed(x,y) = 1, x, y \in V\}$.

The first intuition is that, if we perform a random walk start from a vertex x, then we will get a distribution on vertex set V, since a distribution is in l_1, it is an embedding. Now, if two vertex x and y are close, it seems plausible that their distribution will be close too, if vertex x and y are far apart, their distribution will of course be far apart. Then if we can work out a clever random walk scheme, then it might be able to construct a good embedding.

For this, we need some property of this graph, we conjecture that the spectrum gap of this graph λ might be the right parameter. Because we know that a good expander is the worst case for the embedding, if we can prove that G has spectrum gap $\Theta(\frac{1}{\log |V|})$, then maybe we can make use of it and construct a better embedding.

Why this failed

To test this idea, we find that calculating things on paper are quite hard, so we first run some experiment.

The first observation is that indeed random walk distribution is roughly a function of the distance between them for most case.

But the experiment shows that for two random string x, y, for another random string z, $ed(x-z)-ed(y-z)$ looks like a normal distribution, which means that nearly every random string z whose distance are nearly same to both x and y, from this it is impossible to construct a sub-polynomial embedding.

Another remark is: it can be shown that if a embedding purely based on distance, then it is indeed a convex combination of many "Ball cut". A ball cut $B(x,r)$ is a indicator function for whether a point y is within distance r to x.

For a good embedding to work, we want that the ball cut $B(x,r)$ is stable in the sense that if we move one step in the graph, we have a small probability to "cross" the cut. Otherwise, since the total embedding is just the convex combination of those cuts, if we move one step in the graph, the distance after the embedding will also change dramatically, which is of course a bad embedding.

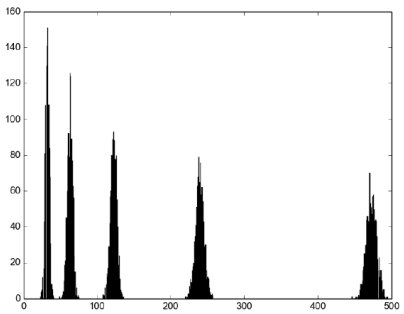
Proposition 1. *Let x be randomly chosen, x_0 be obtained by performing a random edit operation on x. Let $N(x,L)$ be the set of strings that is within L edit distance of x. Then for $L = O(\sqrt[3]{n})$, w.h.p. $|N(x,L) \bigcap N(x_0,L)| = o(1)|N(x,L) \bigcup N(x_0,L)|$.*

Insight and conclusion

Although this method failed, we indeed get some insight in this problem.

Edit distance graph is very complex and have very high dimension

Fact 2. *The distribution of the edit distance of two random strings is plotted in the following figure. $n = 100, 200, 400, 800, 1600, 1000$ samples are used. The mean is estimated to be $0.3n$ and the std grows logarithmically.*



And since the graph is of very high dimension, a ball's "mass" is of course concentrated on its "shell", which means that even two balls are very close, their intersection is negligible.

Embedding's distortion is connected to the unstableness of single bit boolean function.

The second one is that from our argument, it can be shown that if the embedding's every support(boolean function which constitute the embedding.) are unstable, then of course the whole embedding is bad too, this observation is quite crucial for the two sense:

- (Upper bound) For designing the algorithm of construction, it is important to understand what boolean function(or more simply, a 'bit') is stable under random edit operation.
- (Lower bound) To prove the lower bound. It suffices to prove there is no boolean function which are stable under some random edit operation. In fact, this is the method which is used in the lower bound paper.

4.2 Recursively algorithm from embedding for the local embedding

Since we have not succeeded in designing a better algorithm for building the embedding from the existed embedding for small strings. An naturally idea is to change the induction based. The professor gives us an idea which can be summarized as follows:

Idea 1. *Fix $\{0,1\}^d$, suppose there exists an embedding which has distortion D for every two string $x, y \in \{0,1\}^d$ such that $ed(x,y) \leq r$(we don't care about its distortion for pairs that $ed(x,y) > r$), find a method which builds embedding for larger r based on that.*

Definition 7. *For the continence, we call the above embedding r-local embedding.*

If there exists an intelligent way to do it, then we can build a good embedding for the whole space.

The connection between the previous idea

Before starting to summarize our thoughts on this idea, it is important to note what is the difference between it and the previous method.

First we have the following trivial proposition.

Proposition 2. *An embedding with distortion D in ball $B(x,r)$ for any x induces an embedding for $\{0,1\}^r$ with distortion D.*

Proof. We fix every string of x except the last r character, then it translates to an embedding for $\{0,1\}^r$ with distortion D. □

This means that, the following four induction basis are in increasing level of power(with fixed r and D and N):

1. There exists an embedding with distortion D for $\{0,1\}^r$.
2. For some $x_0 \in \{0,1\}^N$, there exists an embedding with distortion D for $B(x_0, r)$.
3. For every $x \in \{0,1\}^N$, there exists an embedding with distortion D for $B(x,r)$.
4. There exists an embedding with distortion D for every two string $x, y \in \{0,1\}^d$ such that $ed(x,y) \leq r$.

From this, we can see what we indeed want here is making a stronger induction base in order to prove a stronger result, it seems quite plausible indeed.

Naive ideas and counterexamples

The first naive idea is we just construct the embedding without the knowledge of the edit distance metric. It looks like not reliable, and it is indeed the case:

Fact 3. *With any L and N, there exists graph with size $O(L \cdot N)$ and has a L-local embedding with distortion 1, but the whole graph has an embedding lower bound $\Omega(\log N)$.*

Proof. We construct a expander of size N which has embedding lower bound $\Omega(\log N)$, then we replace each node in this expander by a tree of size L(with unit edge cost), and each edge in this expander by a long edge with cost $L + 1$.

Note then by simply concatenation every tree's embedding, we get a L-local embedding with distortion 1(Note that tree can be isometrically embeds into l_1). But the overall structure of this graph is still an expanders, which means the whole graph has an embedding lower bound $\Omega(\log N)$. □

Why it is hard to design such an algorithm

After a lot of effort on that problem, we believe this is extremely hard for such a construction. Of course, proving a method will not work at all is indeed a lower bound result, which is very hard in generally, but we provide our thoughts and intuitions to argue why this method seems unlikely to provide us with something new for designing the embedding.

In order for the construction to work, suppose we need to extend the embedding from L-local embedding φ to L'-local embedding φ'.

Then every recursive construction will work like this:

1. Input $x \in \{0,1\}^N$
2. Extract many strings from x, denote them as set $S(x)$.
3. Build the embedding for x from $\varphi(S(x))$.

Then, it is natural to consider the following idea:

Idea 2. *Since we should make use of L-local embedding φ, we want for every x and y, almost every pairs between $S(x)$ and $S(y)$ have distance $\leq L$.*

Consider this seriously, this means that for every x, $S(x)$ are within a small ball $B(x_0, L)$.

Now, how can we extract strings from x which are in $B(x_0, L)$, note this means based on x, we modify at most L positions in x_0, it is really ridiculous if those positions are not contiguous, since otherwise the distance will be disturbed by the content of x_0, and it is highly inconceivable that there exists an magic x_0 whose content can help us do the embedding!

But then, it is just equivalent to build embedding for shorter strings, which we have already talked about, and thus

Ok, the above idea is not that good, requiring every pair between every $S(x)$ and $S(y)$ seems too strong a requirement, what about we group them first?

Idea 3. *We group strings in $S(x)$ into groups $S_1(x), S_2(x), \ldots, S_N(x)$ first, and to utilize the L-local embedding φ, we want for each i, each x and y, almost every pairs between $S_i(x)$ and $S_i(y)$ have distance $\leq L$.*

For the same reason above, now it is indeed required that for each x, $S_i(x) \subseteq B(x_i, L)$. In which $x_i \in \{0,1\}^N$. Now, we are asking that whether there will be such a sequence of $\{x_i\}$, whose content can magically help us build the embedding? If it is false, then it will also degenerate to the case which we build embeddings from shorter strings. We believe even this is possible, it will be extremely hard to find.

Ok, so what about relaxing the requirement that for each i, every pair between $S_i(x)$ and $S_i(y)$ should have distance $\leq L$, what if like in OR's construction, use TEMD to describe the distance between $S_i(x)$ and $S_i(y)$?

This sounds reasonable, but indeed, since the base metric in the TEMD must be based on embedding φ(Otherwise where to use it?), and φ provide NO GUARANTEE for strings with distance $> L$, their embedding can even be equal, it will simply destroy the TEMD.

4.3 Merge the existed embedding

In the above section we illustrate why it is hard to build the embedding based on the local embedding. However, during the process, we also obtain some interesting results, which show there are good embedding for some specific subset of the strings. Note that the first two results are not discovered by anyone before, so at least we improve the knowledge of human here.

This inspired us to study whether or not it is possible to merge those embeddings and obtain a very good global embedding.

We first summarize our results here, then discuss the possibility of such an approach.

- For every N, there exists an $2^{O(\sqrt{\log r \log\log r})}$ embedding for 01 strings with at most r ones and length N. (The distortion is independent with the length.)
- For every N, there exists an $2^{O(\sqrt{\log r \log\log r})}$ for ball $B(x, r)$ for $1 - o(1)$ fraction of $x \in \{0, 1\}^N$.
- There exists an $O(\log^2 d)$ embedding for $1 - o(1)$ fraction of strings in $\{0, 1\}^N$.

Those results indeed sound quite exciting, not that we already have very good embedding which works for almost every strings, and we also have a method which can build a very good embedding for a small ball. So if we can cover those "exception" strings with some embedding balls, and design a method to merge all those embedding into a single one with low distortion, then we are done!

However, after a careful analysis, we find this direction is also unlikely to work.

To start with, we illustrate how to build such embeddings, and then discussing why it is pretty hard to achieve the goal and merge those embeddings.

Embedding for sparse string

The basic idea here is to modify the OR's construction, since it is rather important discovery, we will cover its details here.

Preliminaries

To start with, first we can safely assume each string has the same number of ones.

Suppose $N > r$(otherwise it is meaning less), for every $x \in 0, 1^N$, we first append $2N$ zeros at the back of it, then we change several last zeros into one, so that the whole string has r ones. Let this map be τ.

Denote one(x) to be the number of ones in x, then we can see $ed(\tau(x), \tau(y)) = ed(x, y) + |\text{one}(x) - \text{one}(y)|$. Since $|\text{one}(x) - \text{one}(y)| \leq ed(x, y)$.

$$ed(x, y) \leq ed(\tau(x), \tau(y)) \leq 2ed(x, y).$$

Which means τ is a constant distortion embedding, then we can just apply the embedding for string with length $3N$ and fixed number of ones r.

And for the simplicity of analysis, we forbid the substitution operation, but since it can be replaced by one deletion and one insertion, this will only cause distortion 2.

The intuition and the organization of the proof

Now, for strings a and b, suppose a's ones positions are $a_1, a_2, \ldots, a_r$, and b's are $b_1, b_2, \ldots, b_r$ respectively.

Our intuition for the construction is, note that $ed(a, b) \leq 2r$, because of course we can erase every ones from both side, so in order for $ed(a, b)$ to be small, the edit sequence must 'pair' many a_i and b_j.

This suggests that we can adopt the methods in OR's construction.

In the following, we first give the modified version of Lemma 5 and Lemma 6 in the original paper and prove its correctness. Then based on that, we construct our embedding which is a modified version of the original OR's embedding. Then we proved it has desired expansion and contraction. At last, we discuss the implication of this result.

New Lemma 5 and Lemma 6

Let us recap lemma 5 first:

Lemma 9 (Old Lemma 5)**.** *Let $x, y \in \{0, 1\}^*$ such that $|x| \leq |y|$, and let $b \in N$, $b < |x|$. Let $P(x, b)$ denotes the multi-set of every contiguous substring of length b in x. Then there exists an injection $\sigma : P(x, b) \to P(y, b)$ such that:*

$|\{ed(u, \sigma(u)) > 2ed(x, y)|\ u \in P(x, b)\}| \leq ed(x, y)$.

In order to give the modified Lemma 5, we introduce some definition first:

Definition 8. *Let $a \in \{0, 1\}^N$ with exactly r ones, we denote $a_1, a_2, \ldots, a_r$ as the positions of those ones. Also, denote $a^{i,j}$ as follow, we first cut the contiguous substring $[a_i, a_j]$ from a, then we append zeros at the back of it so it has length N, it has $j - i + 1$ ones in total.*

Lemma 10 (New Lemma 5)**.** *Let $x, y \in \{0, 1\}^N$ such that they both have exactly r ones and $b \in N$, $b < r$. Let $Q(x, b)$ denotes the multi-set $\{x^{i,i+b-1} \mid i \in [1, r-b+1]\}$. Then there exists an injection $\sigma : Q(x, b) \to Q(y, b)$ such that:*

$|\{ed(u, \sigma(u)) > 2ed(x, y)|\ u \in Q(x, b)\}| \leq ed(x, y)$.

Proof. The proof is the same with the original proof. Note that we delete at most $ed(x, y)$ ones from both sides, which means that the rest ones are paired in the optimal edit sequence.

From this pairing we can easily construct the injection σ(map the unpaired ones arbitrarily), and it is straightforward to prove it satisfied the requirement. □

Then comes to Lemma 6, we recap the old lemma 6 here:

Lemma 11 (Old Lemma 6)**.** *Let $x, y \in \{0,1\}^N$, let $b, s \in N$ such that $\frac{N}{b} \in N$ and $s \le b$.*

Let $x_i = x[(i-1)b+1, ib-1], y_i = y[(i-1)b+1, ib-1]$.

Then there exists numbers $k_1, k_2, \ldots, k_{N/b}$ such that:

$$\sum_{i=1}^{N/b} k_i \le 2ed(x,y).$$

And there exists bijection τ_i from $P(x_i, s)$ to $P(y_i, s)$(P are defined in the old lemma 5) for each $i \in [1, N/b]$ such that:

$$|\{ed(u, \tau_i(u)) > k_i \mid u \in P(x_i, s)\}| \le ed(x,y).$$

The new Lemma 6 is just rewriting the above one with the new definition.

Lemma 12 (New Lemma 6)**.** *Let $x, y \in \{0,1\}^N$, both with exactly r ones. Let $b, s \in N$ such that $\frac{r}{b} \in N$ and $s \le b$.*

Let $x_i = x^{(i-1)b+1, ib-1}, y_i = y^{(i-1)b+1, ib-1}$.

Then there exists numbers $k_1, k_2, \ldots, k_{N/b}$ such that:

$$\sum_{i=1}^{N/b} k_i \le 2ed(x,y).$$

And there exists bijection τ_i from $Q(x_i, s)$ to $Q(y_i, s)$(Q are defined in the new lemma 5) for each $i \in [1, N/b]$ such that:

$$|\{ed(u, \tau_i(u)) > k_i \mid u \in Q(x_i, s)\}| \le ed(x,y).$$

Proof. Still in the same way in the original proof of Lemma 6, it is straightforward to prove this one. □

The embedding

We denote φ_d be the embedding for $\{0,1\}^N$ with d ones.

Now, suppose $x \in \{0,1\}^N$ and there are exactly r ones in x. To construct φ_r, first let $B = 2^{\sqrt{\log r \log\log r}}$, with out loss of generality, assume r is a multiple of B here.

We partition the ones in x into B blocks.

Let $d = \frac{r}{B}$, consider the i-th block which consists of d ones. And let $a_1, a_2, \ldots, a_d$ be the positions for the ones in the i-th block.

Let $s_j = 2^j$, $d_j = d - s_j + 1$.

Let $A_{i,j}(x) = \{a^{k,k+d_j-1} \mid k \in [1, s_j]\}$, $B_{i,j}(x) = \{(a_k, a^{k,k+d_j-1}) \mid k \in [1, s_j]\}$. And $S_{i,j}(x) = \{u \oplus \varphi_{d_j}(v) \mid (u,v) \in B_{i,j}(x)\}$. Note they are both multi-set.

Note $a^{l,r}$ here means x^{a_l, a_r}, which is a substring from x.

Then we embeds the multi-set $S_{i,j}(x)$ for $j \in [0, \log d]$ into l_1 by the same method in the original paper's Lemma 8. Denote the results by $\Psi_{i,j}(x)$ And concatenates them all to obtain the embedding for the first block.

For every block, we apply the same method, and then we concatenate the embeddings for each block to obtain the final embeddings for the string.

Proof for correctness

Before giving the proof, it is fruitful to recall the Lemma 8 in the original paper.

Expansion

Now, let us bound its expansion.

There are one thing to be noticed, every 'good pair' in τ_i for each i is indeed part of the global optimal edit sequence, which is pretty clear from the proof for Lemma 6. Which means that the difference between their absolute positions $\leq ed(x, y)$.

Consider two strings $x, y \in \{0, 1\}^N$ both with exactly r ones. By lemma 6, there exists $B = 2^{\sqrt{\log r \log \log r}}$ numbers $k_1, k_2, \ldots, k_B$ number such that:

$$\sum_{i=1}^{B} k_i \leq 2ed(x, y).$$

And for each i, there exists a bijection $\tau_i : A_{i,j}(x) \to A_{i,j}(y)$ such that:

$$|\{ed(z, \tau_i(z)) > k_i \mid z \in A_{i,j}(x)\}| \leq ed(x, y).$$

If $s_j < ed(x, y)$, then trivially we know:

$$\|\Psi_{i,j}(x) - \Psi_{i,j}(y)\| \leq s_j \leq ed(x, y).$$

Otherwise, from the induction hypothesis and the previous fact, we know:

$$\|\{\|z - \tau_i(z)\|_1 > \|\varphi\|_{d_j} \cdot k_i + ed(x, y) \mid z \in S_{i,j}(x)\}\| \leq ed(x, y).$$

Therefor, by Lemma 8:

$$\begin{aligned}
\|\Psi_{i,j}(x) - \Psi_{i,j}(y)\| &\leq \frac{1}{s_j} \min_{\sigma} \left\{ \sum_{z \in S_{i,j}(x)} \min(s_j, 2\|z - \sigma(z)\|_1 \ln(2s_j)) \right\} \\
&\leq \frac{1}{s_j} \left\{ \sum_{z \in S_{i,j}(x)} \min(s_j, 2\|z - \tau_i(z)\|_1 \ln(2s_j)) \right\} \\
&\leq \frac{1}{s_j} \Big(ed(x, y) \cdot s_j + s_j \cdot (\|\varphi_{d_j}\|_{Lip} \cdot k_i + ed(x, y)) \cdot \ln(2s_j) \Big). \\
&\leq 2 \log d \cdot ed(x, y) + 2\|\varphi_{d_j}\|_{Lip} \cdot k_i \cdot \ln(2s_j).
\end{aligned}$$

Sum them up for each i, j, we can get a relationship:

$$\|\varphi_r\|_{Lip} \leq O(\log d^2)(\|\varphi_{r/B}\|_{Lip} + B).$$

Which solves to $\|\varphi_r\|_{Lip} \leq 2^{O(\sqrt{\log r \log \log r})}$.

Contraction

Now, let us bound the contraction.

Let $M_{i,j}$ be:

$$\min_{(a,b) \in B_{i,j}(x), (a', b') \in B_{i,j}(y)} \Big(|a - a'| + ed(b, b') \Big).$$

Let $I = \{M_{i,0} \neq 0 \mid i \in [B]\}$.

For each $i \in I$, pick an arbitrary j_i, then we can prove that:

$$\sum_{i \in I} 2M_{i,j_i} + 4s_{j_i} \geq ed(x, y).$$

Indeed, it suffices to construct a way with at most $\sum_{i \in I} 2M_{i,j_i} + 4s_{j_i}$ steps which can change x to y.

It is quite simple, suppose $M_{i,j}$'s arguments are (a_i, b_i) and (a'_i, b'_i).

For i-th block, if $i \notin I$, then we do nothing because they are already aligned. Otherwise with $4s_{j_i}$ we can delete replace other ones in the i-th block with zeros, except b_i's in x and b'_i's in y.

Then we just align a_i and a'_i for each i, change b_i into b'_i by $ed(b_i, b'_i)$ step. The first one can be done in $|a_i - a'_i|$, the second one may introduce new misalignment at most $ed(b_i, b'_i)$, but we can use another $ed(b_i, b'_i)$ to recover it. After that, we have changed x into y.

Then, with a similar analysis in the original proof, we can show the contraction is also indeed $2^{O(\sqrt{\log d \log\log d})}$.

Implication

Note that, this indeed means that the distortion in OR's construction only depends on the number of ones, which is a quite counterintuitive result.

Embedding for neighborhood of a random string

Embedding for $1 - o(1)$ fraction string

In this section, we present an embedding which works for $1 - o(1)$ fractions of $\{0,1\}^n$ with distortion $O(\log^2 n)$. Since this results make use of the embedding for the Ulam metric, we will also recap the embedding for Ulam metric.

The intuition here is that it suffices to came up with an algorithm work well for random string, and from a random string it is possible to extract a feature of $O(\log d)$ length which are enough to identify the position, and turn this problems into a problem about permutation.

t-repetition free string

Definition 9. *Denote the fold of the string x with size t as $F(x,t) = x[1 \dots t]x[2 \dots t{+}1] \dots x[|x|{-}t{+}1 \dots |x|]$. Note it is a new string with alphabet $\sum = \{0,1\}^t$.*

Definition 10. *A t-repetition free string is a string x such that every elements in $FOLD(x,t)$ are distinct.*

With a simple probabilistic method, we can show that the probability of a random string to be t-repetition is very high.

Lemma 13. *Pick a random string x in $\{0,1\}^d$ uniformly, the probability that x is $k \log d$-repetition free is at least $1 - d^{2-k}$.*

Proof. Let $i < j$, denote $E_{i,j}$ to be the events such that $x[i \dots i+t-1] = x[j \dots j+t-1]$, let us compute $\Pr[E_{i,j}]$.

The first case is that $[i \dots i+t-1]$ and $[j \dots j+t-1]$ don't intersect, then $\Pr[E_{i,j}] = 2^{-t}$ obviously.

The second case is that they intersect, note that in this case, when $x[i \dots j-1]$ are fixed, the whole contents of $x[j \dots j+t-1]$ are also fixed since $x[i \dots i+t-1] = x[j \dots j+t-1]$, so the probability is still $\Pr[E_{i,j}] = 2^{-t}$.

The let $U = \bigcup E_{i,j}$, note $\Pr[U] \le d^2 2^{-t} = 2^{-t+2\log d}$. And the complement of U is the event that the random string is t-repetition free, so we know the probability is at least:

$$1 - \Pr[U] \ge 2^{-k\log d + 2\log d} = 1 - d^{2-k}.$$

□

Small distortion embedding for t-repetition free string

For a t-repetition free string x, note that $F(x,t)$ is indeed a permutation.

First it is straightforward to show that the mapping $F : \{0,1\}^d \to \{\{0,1\}^t\}^{d-t+1}$ has distortion t.

Also, we know that for a permutation(indeed, for strings with distinct characters), there exists an embedding to l_1 with distortion $O(\log d)$.

Combine those two embedding, we obtain an embedding for $k \log d$-repetition free string with distortion $O(k \log^2 d)$.

And the above argument shows there are at most d^{2-k} fractions of strings are not $k \log d$-repetition free, for fixed $k > 2$, d^{2-k} is $o(1)$ in terms of d.

To summarize, we have $O(\log^2 d)$ embedding for a $1 - o(1)$ fractions of the strings.

Method for merging existed embeddings for Balls

We will first discuss the general results we have got for merging embeddings, which are quite trivial, then we will discuss why this is hard and unlikely to be done.

Merge a ball and one point

Suppose we have embedding φ with distortion D for a ball $B(x,r)$, and we want to put another point y into it.

Then find y's nearest point in $B(x,r)$ y_0, we map y into $\varphi(y_0) \oplus dist(y,y_0)$, and map every point in $x_0 \in B(x,r)$ into $\varphi(x_0) \oplus 0$.

It can be proved that the distortion for the new embedding φ' for $B(x,r) \bigcup \{y\}$ has distortion at most $2D+1$.

For $a, b \in B(x,r)$, the distortion of course won't change.

For $x \in B(x,r)$ and y.

Note the new distance after φ' is:

$$E = dist(y,y_0) + \|\varphi(x) - \varphi(y_0)\|_1.$$

From the distortion for φ, we know that:

$$\|\varphi(x) - \varphi(y_0)\|_1 \le dist(x,y_0) \le D \cdot \|\varphi(x) - \varphi(y_0)\|_1.$$

Then we have:

$$dist(y,y_0) + dist(x,y_0) \le E \le dist(y,y_0) + D \cdot dist(x,y_0)$$

We also know that:

$$\max\Big(|dist(y,y_0) - dist(y_0,x)|, dist(y,y_0)\Big) \le dist(x,y) \le dist(y,y_0) + dist(y_0,x).$$

Then:

$$\frac{dist(y,y_0)+dist(y_0,x)}{dist(y,y_0)+dist(y_0,x)} \le \frac{E}{dist(x,y)} \le \frac{dist(y,y_0)+D\cdot dist(x,y_0)}{\max\Big(|dist(y,y_0)-dist(y_0,x)|, dist(y,y_0)\Big)} \le (2D+1).$$

Merge two balls

Suppose we have two embeddings for $B(x,r)$ and $B(y,r)$ such that $dist(x,y) = 1$. And we want to merge them.

From that we first get an embedding for $I = B(x,r) \bigcap B(y,r)$, then we embed each point x outside I with the same embedding for its nearest point in I, and add an unique coordinate with 1 for him, then we can see it only cause a constant factor and work for $B(x,r) \bigcup B(y,r)$.

Why it is hard to merge embeddings?

First, we should notice that although we have got embedding for $1 - o(1)$ fractions of strings, the $o(1)$ of 2^d is still very huge.

Indeed, suppose we fill the first part of a string by $01010101\ldots$, and the second part is arbitrary, then of course this string can not be covered by the $1 - o(1)$ fractions since it has $\Omega(d)$ self-repetition, and since the second part is arbitrary, embedding for those $o(1)$ part will induce an embedding for $\{0,1\}^{d/2}$. So indeed:

Fact 4. *building embedding for this $o(1)$ part is at least hard as building embedding for every string.*

Another crucial observation is that, consider change one t-repetition free string x into another one y, if we apply the change from left to right, the middle string will be a prefix of string y concatenated by a suffix of string x, note that this middle string itself has very limited repetition, any t-substrings will occur at one times in the prefix and one times in the suffix, only the part at the position of the concatenation will occur many repetition, but that part is very small.

Also, consider change one highly self repetition string x into another y, we can see the middle string will also be highly self repetition string(Note that we have not defined highly self repetition string, but any reasonable definition will suffice).

So here comes the observation:

Observation 1. *In the edit distance graph, two part "random string" and "highly self repetition string" are highly separated, not only their have no intersection, furthermore in order to traveling between one part by the shortest path, one don't need go to the other side.*

It suggests that the embedding for "random string" set might be useless for building the embedding for the core.