Streaming algorithms for embedding and computing edit distance in the low distance regime

Diptarka Chakraborty
joint work with Elazar Goldenberg and Michal Koucký

7 November, 2015
Outline

1. Introduction
   - Problem Definition
   - Motivation

2. Result on Embedding

3. Applications
   - Document Exchange Problem
   - Other Applications

4. Embedding Algorithm
   - Description
   - Result
   - Proof Sketch

5. Computing Edit Distance
   - Comparison with Previous Work
   - Finding Alignment with small cost
   - Proof Sketch
1 Introduction
   - Problem Definition
   - Motivation

2 Result on Embedding

3 Applications
   - Document Exchange Problem
   - Other Applications

4 Embedding Algorithm
   - Description
   - Result
   - Proof Sketch

5 Computing Edit Distance
   - Comparison with Previous Work
   - Finding Alignment with small cost
   - Proof Sketch
Problem Definition

Suppose we are given two strings $x, y \in \{0, 1\}^*$

- **Edit distance**, denoted by $\Delta_e(x, y)$, is defined as the minimum number of insertion, deletion and bit flip operations needed for converting from $x$ to $y$. 

---

Diptarka Chakraborty, diptarka@cse.iitk.ac.in

Streaming algorithms for embedding and computing edit distance.
Problem Definition

Suppose we are given two strings $x, y \in \{0, 1\}^*$

- **Edit distance**, denoted by $\Delta_e(x, y)$, is defined as the minimum number of insertion, deletion and bit flip operations needed for converting from $x$ to $y$.

- **Hamming distance**, denoted by $\Delta_H(x, y)$, is defined as the minimum number of bit flip operations needed.
Problem Definition

Suppose we are given two strings $x, y \in \{0, 1\}^*$

- **Edit distance**, denoted by $\Delta_e(x, y)$, is defined as the minimum number of insertion, deletion and bit flip operations needed for converting from $x$ to $y$.

- **Hamming distance**, denoted by $\Delta_H(x, y)$, is defined as the minimum number of bit flip operations needed.

The problem is to find a *randomized embedding* from edit metric to Hamming metric with small *distortion factor*. 
Problem Definition

Define a map $f : \{0, 1\}^n \rightarrow \{0, 1\}^{l(n)}$ s.t. $\forall x, y \in \{0, 1\}^n$, 

$$\beta \cdot \Delta_e(x, y) \leq \Delta_H(f(x), f(y)) \leq \alpha \cdot \Delta_e(x, y)$$

where distortion factor is $\phi_d = \alpha / \beta$. 

Remark: Previous best known bound by Jowhari '12: $\phi_d \leq O(\log n \log^* n)$
Problem Definition

- Define a map $f : \{0, 1\}^n \rightarrow \{0, 1\}^{l(n)}$ s.t. $\forall x, y \in \{0, 1\}^n$,
  \[ \beta \cdot \Delta_e(x, y) \leq \Delta_H(f(x), f(y)) \leq \alpha \cdot \Delta_e(x, y) \]
  where distortion factor is $\phi_d = \alpha / \beta$.
- Now instead consider randomized map, i.e.,
  $f : \{0, 1\}^n \times \{0, 1\}^r \rightarrow \{0, 1\}^{l(n)}$ s.t. above holds w.h.p.
Define a map \( f : \{0, 1\}^n \rightarrow \{0, 1\}^{l(n)} \) s.t. \( \forall x, y \in \{0, 1\}^n, \)
\[
\beta \cdot \Delta_e(x, y) \leq \Delta_H(f(x), f(y)) \leq \alpha \cdot \Delta_e(x, y)
\]
where distortion factor is \( \phi_d = \alpha / \beta \).

Now instead consider randomized map, i.e., \( f : \{0, 1\}^n \times \{0, 1\}^r \rightarrow \{0, 1\}^{l(n)} \) s.t. above holds w.h.p.

**Remark:** Previous best known bound by Jowhari '12: \( \phi_d \leq O(\log n \log^* n) \)
Motivation

- From computational perspective, problems on Hamming distance are somehow easier than that on edit distance.
- Embedding provides us power to use results from the world of Hamming metric.
From computational perspective, problems on Hamming distance are somehow easier than that on edit distance.

Embedding provides us power to use results from the world of Hamming metric.

Applications include document exchange problem under edit metric, designing sketching protocol for gap-edit distance, approximately nearest neighbor search.
1. Introduction
   - Problem Definition
   - Motivation

2. Result on Embedding

3. Applications
   - Document Exchange Problem
   - Other Applications

4. Embedding Algorithm
   - Description
   - Result
   - Proof Sketch

5. Computing Edit Distance
   - Comparison with Previous Work
   - Finding Alignment with small cost
   - Proof Sketch
Result on Embedding

There exists a mapping $f : \{0, 1\}^n \times \{0, 1\}^{6n} \rightarrow \{0, 1\}^{3n}$ which satisfies the following conditions:

1. For every $x, y$, $\Delta_e(x, y)/2 \leq \Delta_H(f(x, r), f(y, r))$ with probability at least $1 - \exp(-\Omega(n))$.
2. For every $x, y$, $\Delta_H(f(x, r), f(y, r)) \leq O(\Delta_e(x, y)^2)$ with probability at least $2/3$.
3. Given $f(x, r)$ and $r$, it is possible to decode back $x$ with probability $1 - \exp(-\Omega(n))$.

Moreover, both the mapping $f$ and its decoding (given $f(x, r)$ and $r$) take linear time and can be performed in a streaming fashion.
Result on Embedding

There exists a mapping $f : \{0, 1\}^n \times \{0, 1\}^{6n} \rightarrow \{0, 1\}^{3n}$ which satisfies the following conditions:

1. For every $x, y$, $\Delta_e(x, y)/2 \leq \Delta_H(f(x, r), f(y, r))$ with probability at least $1 - \exp(-\Omega(n))$. 

Diptarka Chakraborty, diptarka@cse.iitk.ac.in

Streaming algorithms for embedding and computing edit distance
Result on Embedding

There exists a mapping $f : \{0, 1\}^n \times \{0, 1\}^{6n} \to \{0, 1\}^{3n}$ which satisfies the following conditions:

1. For every $x, y$, $\Delta_e(x, y)/2 \leq \Delta_H(f(x, r), f(y, r))$ with probability at least $1 - \exp(-\Omega(n))$.

2. For every $x, y$, $\Delta_H(f(x, r), f(y, r)) \leq O(\Delta_e(x, y))^2$ with probability at least $2/3$.
There exists a mapping $f : \{0, 1\}^n \times \{0, 1\}^{6n} \rightarrow \{0, 1\}^{3n}$ which satisfies the following conditions:

1. For every $x, y$, $\Delta_e(x, y)/2 \leq \Delta_H(f(x, r), f(y, r))$ with probability at least $1 - \exp(-\Omega(n))$.

2. For every $x, y$, $\Delta_H(f(x, r), f(y, r)) \leq O(\Delta_e(x, y))^2$ with probability at least $2/3$.

3. Given $f(x, r)$ and $r$, it is possible to decode back $x$ with probability $1 - \exp(-\Omega(n))$. 

Moreover, both the mapping $f$ and its decoding (given $f(x, r)$ and $r$) take linear time and can be performed in a streaming fashion.
There exists a mapping \( f : \{0, 1\}^n \times \{0, 1\}^{6n} \rightarrow \{0, 1\}^{3n} \) which satisfies the following conditions:

1. For every \( x, y \), \( \Delta_e(x, y)/2 \leq \Delta_H(f(x, r), f(y, r)) \) with probability at least \( 1 - \exp(-\Omega(n)) \).

2. For every \( x, y \), \( \Delta_H(f(x, r), f(y, r)) \leq O(\Delta_e(x, y))^2 \) with probability at least \( 2/3 \).

3. Given \( f(x, r) \) and \( r \), it is possible to decode back \( x \) with probability \( 1 - \exp(-\Omega(n)) \).

Moreover, both the mapping \( f \) and its decoding (given \( f(x, r) \) and \( r \)) take linear time and can be performed in a streaming fashion.
Introduction

- Problem Definition
- Motivation

Result on Embedding

Applications

- Document Exchange Problem
- Other Applications

Embedding Algorithm

- Description
- Result
- Proof Sketch

Computing Edit Distance

- Comparison with Previous Work
- Finding Alignment with small cost
- Proof Sketch
Applications

- Computing edit distance (nontrivial)
- Document exchange problem under edit metric
- Designing sketching protocol for gap-edit distance
- Approximately nearest neighbor search
Document Exchange Problem

- Alice and Bob hold two strings $x$ and $y$ respectively.
- Bob’s task is to
  - decide whether $\Delta_e(x, y) > k$
  - otherwise report $x$ correctly.
Document Exchange Problem

Use the following protocol (shared randomness) to solve:

- Alice and Bob compute \( f(x, r) \) and \( f(y, r) \) respectively (linear time)

- Use protocol for Hamming metric by Porat and Lipsky '07 (uses \( O(k^2 \log n + n \log n) \) time and \( O(k^2 \log n) \) bits to be transmitted)

- Bob will learn \( f(x, r) \) and then decode (linear time)

- Use algorithm by Landau et al. to decide whether \( \Delta_e(x, y) \leq k \) (\( O(n + k^2) \) time)
Use the following protocol (shared randomness) to solve:

- Alice and Bob compute \( f(x, r) \) and \( f(y, r) \) respectively (linear time)
- Use protocol for Hamming metric by Porat and Lipsky ’07 (uses \( O(k^2 \log n + n \log n) \) time and \( O(k^2 \log n) \) bits to be transmitted)
- Bob will learn \( f(x, r) \) and then decode (linear time)
- Use algorithm by Landau et al. to decide whether \( \Delta_e(x, y) \leq k \) (\( O(n + k^2) \) time)

**Remark:** Previous best known bound by Jowhari ’12: \( O(n \log n + k^2 \log^2 n) \) on time and \( O(k \log^2 n \log^* n) \) on number of bits to be transmitted
Similarly,

- solves $k$ vs. $ck^2$ gap-edit distance, for some $c > 0$ using constant size sketches
- return a point within the distance $O(k)$ times that of the closest one
1 Introduction
   • Problem Definition
   • Motivation

2 Result on Embedding

3 Applications
   • Document Exchange Problem
   • Other Applications

4 Embedding Algorithm
   • Description
   • Result
   • Proof Sketch

5 Computing Edit Distance
   • Comparison with Previous Work
   • Finding Alignment with small cost
   • Proof Sketch
The basic scheme is as follows:

- Pick a sequence of random functions $h_1, \cdots, h_{3n} : \{0, 1\} \rightarrow \{0, 1\}$
- Maintain a pointer $i$ for current position on input and initially set to 1
- In time $t \leq 3n$, append output by $x_i$ and increment $i$ by $h_t(x_i)$
- If $i$ exceeds $n$, append zeros
Recall the Result

There exists a mapping $f : \{0, 1\}^n \times \{0, 1\}^{6n} \rightarrow \{0, 1\}^{3n}$ which satisfies the following conditions:

1. For every $x, y$, $\Delta_e(x, y)/2 \leq \Delta_H(f(x, r), f(y, r))$ with probability at least $1 - \exp(-\Omega(n))$.

2. For every $x, y$, $\Delta_H(f(x, r), f(y, r)) \leq O(\Delta_e(x, y))^2$ with probability at least $2/3$.

3. Given $f(x, r)$ and $r$, it is possible to decode back $x$ with probability $1 - \exp(-\Omega(n))$.

Moreover, both the mapping $f$ and its decoding (given $f(x, r)$ and $r$) take linear time and can be performed in a streaming fashion.
Proof Sketch

- Use Chernoff bound to show that \( i \) will exceed \( n \) within time \( 3n \) except with probability \( 2^{-\Omega(n)} \)
- If exceed, then use the similar algorithm to decode
Proof Sketch

- Use Chernoff bound to show that $i$ will exceed $n$ within time $3n$ except with probability $2^{-\Omega(n)}$
- If exceed, then use the similar algorithm to decode
- Lower bound follows from decoding algorithm
Proof Sketch

- Use Chernoff bound to show that $i$ will exceed $n$ within time $3n$ except with probability $2^{-\Omega(n)}$
- If exceed, then use the similar algorithm to decode
- Lower bound follows from decoding algorithm
- To prove upper bound, reduce the problem to a well-known problem on random walk
1 Introduction
   • Problem Definition
   • Motivation

2 Result on Embedding

3 Applications
   • Document Exchange Problem
   • Other Applications

4 Embedding Algorithm
   • Description
   • Result
   • Proof Sketch

5 Computing Edit Distance
   • Comparison with Previous Work
   • Finding Alignment with small cost
   • Proof Sketch
## Comparison with Previous Work

<table>
<thead>
<tr>
<th>Authors</th>
<th>Time</th>
<th>Space</th>
<th>Approx. Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>WF74</td>
<td>$O(n^2)$</td>
<td>$O(n)$</td>
<td>Exact</td>
</tr>
<tr>
<td>LMS98</td>
<td>$O(n + k^2)$</td>
<td>$O(n)$</td>
<td>Exact</td>
</tr>
<tr>
<td>PP08</td>
<td>$O(n + k^2)$</td>
<td>$O(n)$</td>
<td>Exact</td>
</tr>
<tr>
<td>LMS98</td>
<td>$O(n)$</td>
<td>$O(n)$</td>
<td>$\sqrt{n}$</td>
</tr>
<tr>
<td>AKO10</td>
<td>$n^{1+\epsilon}$ (randomized)</td>
<td>$O(n)$</td>
<td>$(\log n)^{O(1/\epsilon)}$</td>
</tr>
<tr>
<td>Saha14</td>
<td>$O(n)$ (randomized)</td>
<td>$O(\log n)$</td>
<td>$O(k)$</td>
</tr>
<tr>
<td><strong>This paper</strong></td>
<td>$O(n + k^6)$ (randomized)</td>
<td>$O(k^6)$</td>
<td>Exact</td>
</tr>
</tbody>
</table>
Finding Alignment with small cost

- Maintain two pointers \( i_x, i_y \) for \( x \) and \( y \) respectively
- If \( x_{i_x} = y_{i_y} \), set \( a(i_x) = i_y \)
- Else with probability \( 1/2 \), set \( s(i_x) = D \) and increment \( i_x \)
- With remaining probability increment only \( i_y \)
- Stop if both \( i_x \) and \( i_y \) reach \( n + 1 \)
- If \( i_y = n + 1 \) and \( i_x < n \), set \( a(i_x), \ldots, a(n) = D \)
Proof Sketch

- Compute an alignment of cost $ck^2$
- Compute kernels of size $O(k^6)$ for each of the inputs
- Run known algorithm, say PP08, on those kernels
Proof Sketch

- Compute an alignment of cost $ck^2$
- Compute kernels of size $O(k^6)$ for each of the inputs
- Run known algorithm, say PP08, on those kernels
- To boost probability, re-run the embedding and kernelization if alignment is too costly
- Use pre-computed kernel to avoid re-reading input
THANK YOU!!!