Review on CGK Embedding

The paper describes two methods:

Embedding This embeds the 2 strings x and y into $f(x,r)$ and $f(y,r)$ using a random string r which creates $3n$ hash functions $h_1, h_2, ..., h_{3n} : \{0,1\} \to \{0,1\}$. $\therefore$ $f(x,r) : \{0,1\}^n \times \{0,1\}^{6n} \to \{0,1\}^{3n}$. We compare and get the Hamming distance of $f(x,r)$ and $f(y,r)$ and claim that:

$$\frac{1}{2} \cdot \Delta_e(x,y) \leq \Delta_H(f(x,r), f(y,r)) \leq O((\Delta_e(x,y))^2)$$

with probability atleast 2/3.

Kernelization This converts x and y into x' and y' such that $\Delta_e(x,y) = \Delta_e(x',y')$. It uses the following 2 methods:

- **Deflation** We increase the length of x and y and keep the EditDistance same. We can prove that there exists a substring w in x and y of the form $w = p^r$ where p is the periodicity of w and $r > 2$. Somewhere in w, the strings x and y would align wrt EditDistance, as w is same in both the strings x and y, all the bits after would be aligned too. We can add p sometime after the initial alignment index such that $w' = p^{r+1}$ and $\Delta_e(x,y) = \Delta_e(x',y')$
- **Shrinkage** Similar to the above case, we can remove p bits form w and the EditDistance would remain the same. In case of **Shrinkage**, we define $s = K + 2k + (k+1)(t+1)$ and reduce w by keeping the first s and last s bits.

Decompose $x = u_0 w_1 u_1 ... w_l u_l$ and $y = v_0 w_1 v_1 ... w_l v_l$, deflate and shrink each w_i to get the desired x' and y'.

Note: this paper only compares 2 binary strings with low edit distance $O(n^{1/6})$, where n is the length of both the strings.

Theorems and Proofs

Theorem (**Theorem 4** in [CGK16])**.** *The mapping $f : \{0,1\}^n \times \{0,1\}^{6n} \to \{0,1\}^{3n}$ computed by Algorithm 1 satisfies the following condtions:*

1. *For every $x \in \{0,1\}^n$, given $f(x,r)$ and r, it is possible to decode back x with probability $1 - exp(-\Omega(n))$*

2. *For every $x, y \in \{0,1\}^n$, $\Delta_e(x,y)/2 \leq \Delta_H(f(x,r), f(y,r))$ with probability at least $1 - exp(-\Omega(n))$*

3. *For every positive constant c and every $x, y \in \{0,1\}^n$, $\Delta_H(f(x,r), f(y,r)) \leq c \cdot (\Delta_e(x,y))^2$ with probability at least $1 - \frac{12}{\sqrt{c}}$*

Proof.

1. we can decode back x if we are given $f(x,r)$ and r only if the value of i from Algorithm 1 is $n+1$ at the end of $f(x,r)$. As this would mean that we have traversed through all the bits of x.
   Consider that we have infinite hash functions $h_1, h_2, ... : \{0,1\} \to \{0,1\}$.
   Consider for each x_i we embed it using hash functions $h_k, ..., h_l$. As we embed x_i for all $h_k, ..., h_l$, it implies that $h_k(x_i) = ... = h_{l-1}(x_i) = 0$ and $h_l(x_i) = 1$.
   Hence, we can interpret the given condition as n geometric distributions, where the total number of trials required is less than $3n$.

   For every grometric distribution, $p = 1/2$.
   Define, X_i = Number of hash functions used to embed x_i, and all the X_i are i.i.d

   $$\begin{aligned} E[X_i] &= 2 \\ \therefore E[X] &= 2n \end{aligned}$$

   Using Equation (6) from [HR90]

   $$\begin{aligned} \Pr(S \geq (1+\epsilon)m) &\leq e^{-\epsilon^2 m/3} \\ (1+\epsilon) &= 3/2 \\ \epsilon &= 1/2 \\ \Pr(X > 3n) &\leq e^{-\frac{(1/2)^2 2n}{3}} \\ &\leq e^{-n/6} \\ \therefore \Pr(X < 3n) &\geq 1 - e^{-n/6} \end{aligned}$$

2. Consider the i value mentioned in Algorithm 1 is $n+1$ for both x and y. Let $l = \Delta_H(f(x,r), f(y,r))$, then we need to apply atmost l edit operations to x to get y.
   Except, when atmost the last l bits of y are 0 and align with the padded

0s of x. (The paper [CGK16], does not mention the last bits of x here, but I think that maximum of l bits from either x and y could be aligned with the padded 0s of the other).
Hence, $\Delta_e(x,y) \leq 2l$.
As per out initial assumption, this is possible only when the i value reach $n+1$ for both x and y.

$$\begin{aligned}
\Pr(X < 3n \cap Y < 3n) &= \Pr(X < 3n) \cdot \Pr(Y < 3n) --- (X and Y are independent events) \\
&= (1 - e^{-n/6}) \cdot (1 - e^{-n/6}) \\
&= 1 - 2e^{-n/6} + e^{-n/3} \\
&\approx 1 - 2e^{-n/6} \\
&= 1 - e^{-(n/6 - log 2)} \\
&= 1 - e^{(-\Omega(n))}
\end{aligned}$$

3. This can be proved by combining **Lemma 4.2** and **Proposition 3.2**.
   Lemma 4.2:

$$\Pr(\Delta_H(f(x,r), f(y,r)) \leq l) \geq \sum_{t=0}^{l} q(t, \Delta_e(x,y))$$

For our case, $l = c \cdot (\Delta_e(x,y))^2$

$$\Pr(\Delta_H(f(x,r), f(y,r)) \leq c \cdot (\Delta_e(x,y))^2) \geq \sum_{t=0}^{c \cdot (\Delta_e(x,y))^2} q(t, \Delta_e(x,y))$$

From **Proposition 3.2**, $\displaystyle\sum_{t=0}^{l} q(t,k) \geq 1 - \frac{12k}{\sqrt{l}}$

$$\begin{aligned}
\Pr(\Delta_H(f(x,r), f(y,r)) \leq c \cdot (\Delta_e(x,y))^2) &\geq 1 - \frac{12\Delta_e(x,y)}{\sqrt{c \cdot (\Delta_e(x,y))^2}} \\
&\geq 1 - \frac{12}{\sqrt{c}}
\end{aligned}$$

□

Lemma (Lemma 4.2 in [CGK16]**).** *Let $x, y \in \{0,1\}^n$ be of edit distance $\Delta_e(x,y) = k$. Let $q(t,k)$ be the probability that a random walk on the integer line starting from the origin visits the point k at time t for the first time. Then for any $l > 0$, $\Pr(\Delta_H(f(x,r), f(x,y)) \leq l) \geq \sum_{t=0}^{l} q(t,k)$ where the probability is over the choice of r.*

Proof. [*My interpretation*] k is the Edit Distance $\Delta_e(x,y)$.
Imagine a timeline, we start at 0 and have a marker at k.
Our methodology for moving on the timeline is as follows:

1. $i_x(t)$ and $i_y(t)$ are the indices of x and y being embedded at time t. Our position on the timeline would be $i_x(t) - i_y(t)$ at any given time. Since the Edit Distance $= k$, the value of $i_x(t) - i_y(t)$ should be less than k at all times.

2. Let $d_t = i_x(t) - i_y(t)$,

   (a) when $x_{ix(t)} = y_{iy(t)}$, then $h_t(x_{ix(t)}), h_t(y_{iy(t)}) = (0,0)$ or $(1,1)$ $\implies$ value of d_t does not change.

   (b) when $x_{ix(t)} \neq y_{iy(t)}$, then $h_t(x_{ix(t)}), h_t(y_{iy(t)}) = (0,0),(0,1),(1,0),(1,1)$ $\implies$ value of d_t changes only when $x_{ix(t)} \neq y_{iy(t)}$ $\implies$ d_t changes only when it is contributing to the Hamming Distance. $\implies$ Hamming Distance could be interpreted as our movement on the timeline.

3. Ignoring the steps when $x_{ix(t)} = y_{iy(t)}$, the probability that we move $+1$ steps is $1/4$, -1 steps is $1/4$ and stay where we are is $1/2$.

4. Hence, the **Lemma** statement can be interpreted as: The Probability that we take atmost l steps to reach k is greater than the probability that we reach k for the first time within l steps.

$\square$

References

[CGK16] Diptarka Chakraborthy, Elazar Goldenberg, and Michal Koucký. Streaming algorithms for embedding and computing edit distance in the low distance regime. *STOC '16*, pages 712–725, 2016.

[HR90] T. Hagerup and C. Rüss. A Guided Tour of Chernoff Bounds. *Information Processing Letters*, 33:305–308, 1989/90.