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. y differs from x only in the parts where $f(x,r) \neq f(y,r)$.
$f(x,r)_j \neq f(y,r)_j \implies x_{i_x(j)} \neq y_{i_y(j)}$
If $l = \Delta_H(f(x,r), f(y,r))$, then $x_{i_x(j)} \neq y_{i_y(j)}$ in atmost l places.

Define the following edit operations:

$$
\begin{aligned}
h_j(f(x,r)_j), h_j(f(y,r)_j) &= (0,0) && \text{, ignore} \\
&= (0,1) && \text{, } \textit{insert} \\
&= (1,0) && \text{, } \textit{delete} \\
&= (1,1) && \text{, } \textit{replace}
\end{aligned}
$$

We ignore at (0,0) as $(j+1)^{th}$ index would have the same values.
Hence, the number of edit operations required would be less than or equal to the hamming distance.
Except, we need to consider the case where 0s at the end of x or y coincide with the trailing 0s, in this case, the actual bit flips required would be $2l$.
Hence, $\Delta_e(x,y) \leq 2 * \Delta_H(f(x,r), f(y,r))$.

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

$$
\begin{aligned}
\Pr(\Delta_H(f(x,r), f(y,r)) \leq l) &\geq \sum_{t=0}^{l} q(t, \Delta_e(x,y)) \\
\text{For our case, } l = c \cdot (\Delta_e(x,y))^2 & \\
\Pr(\Delta_H(f(x,r), f(y,r)) \leq c \cdot (\Delta_e(x,y))^2) &\geq \sum_{t=0}^{c \cdot (\Delta_e(x,y))^2} q(t, \Delta_e(x,y))
\end{aligned}
$$

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. *Example:*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46		H-Values					
x	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0																																		0	1		0	1
y	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1																																	1		0	30	1	1
f(x,r)	1	1	1	0	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0			2		0	31	1	0
f(y,r)	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0			3		1	32		0
ix	1	1	1	2	3	4	5	5	6	6	6	6	6	6	7	8	8	8	8	8	8	8	9	9	9	10	11	11	12	12	13	14	14	14	15	15	16	16	16	16	16	16	16	16	16			4	1		33		0
iy	1	1	1	2	3	4	5	5	5	5	5	5	5	5	6	6	6	6	6	6	6	6	7	8	8	9	10	10	11	11	12	12	12	12	13	13	14	15	15	16	16	16	16	16	16			5		1	34		1
ix-iy	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2	2	2	1	1	0	0	0	0	0	0	-	Particle moment	6	1	1	35	0	
																																																7	0	0	36	1	
y=x0	1	0	1	0	1	1	0	1	0	0	0	1	0	0	1																											Edit operations:						8	1	0	37	1	
x1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	1				i1	4	insert																				insert, 3,3						9		0	38		0
x2	1	0	1	1	0	1	0	1	0	1	0	0	0	1	0	0	1			i2	7	insert																				insert, 5,6						10		0	39		1
x3	1	0	1	1	0	1	0	1	1	1	0	0	0	1	0	0	1			i3	9	replace																				replace, 6,8						11		0	40		
x4	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	0	1			i4	12	replace																				replace, 9, 11						12		0	41		
x5	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0	1				i5	15	delete																				delete, 13,15						13		0	42		
x6=x	1	0	1	1	0	1	0	1	1	1	0	1	0	1	0					i6	15	delete																				delete, 14,15						14		1	43		
						insert									insert								replace						replace						delete*2													15	1	0	44		
ix	1	1	1	2	3	4	5	5	6	6	6	6	6	6	7	8	8	8	8	8	8	8	9	9	9	10	11	11	12	12	13	14	14	14	15	15	16	16	16	16	16	16	16	16	16			16		0	45		
dj	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	-	Goalspot	17		0			
																																																18		0			
ix-iy	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2	2	2	1	1	0	0	0	0	0	0			19		0			
dj	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0			20		0			
dj=ix-iy	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1			21		0			
																																																22		1			
f(x,r)	1	1	1	0	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0			23	1	0			
f(y,r)	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0			24		0			
f(x,r)=f(y,r)	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1			25		1			
																																																26	1	1			
																																																27	0				
dj=ix-iy	1	1	1	1	1				1	1	1	1	1	1		1	1	1	1	1	1	1	1									1	1	1						1	1	1	1	1	1			28	1				
f(x,r)=f(y,r)	1	1	1	1	1	-	-	-	1	1	1	1	1	1	-	1	1	1	1	1	1	1	0	-	-	-	-	-	-	-	-	1	1	1	-	-	-	-	-	1	1	1	1	1	1			29	0	0			

1. In the above image, we have the value of x, y, $f(x,r)$ and $f(y,r)$ (based on the H-values mentioned on the right side).
   The values i_x and i_y are the indices of x and y for the given f values.

2. **Particle Moment:** Every step taken by the particle is defined as $h_j(x_{i_x(j)}) - h_j(y_{i_y(j)})$, for j^{th} value of $f(x,r)$ and $f(y,r)$.
   Now, $\sum_{k=0}^{j} h_j(x_{i_x(k)}) = i_x(j) - 1$ and $\sum_{k=0}^{j} h_j(y_{i_y(k)}) = i_y(j) - 1$, as the i value changes only when $h() = 1$.
   Hence, $h_j(x_{i_x(j)}) - h_j(y_{i_y(j)}) = i_x(j) - i_y(j)$.
   Hence, the position of particle at time j is $i_x(j) - i_y(j)$.

3. **Goalspot:** We define $y = x^{(0)}, x^{(1)}, ..., x^{(k)} = x$ as a series of k strings such that the first string is y and the last string is x and $\Delta_e(x^{(l-1)}, x^{(l)}) = 1$.
   $\therefore$, in the above image, each $x^{(l)}$ is mentioned. The inserted bits are colored yellow, replaced bits are colored blue and deleted bits are colored red. The edit operation from $x^{(1)}, x^{(6)}$ are: insert, insert, replace, replace, delete, delete. The first index $(i^{(l)})$ where $x^{(l)}$ differs from $x^{(l-1)}$ is 4, 7, 9, 12, 15, 15 respectively.
   We define d_j(goalspot) as:
   $d_0 = 0$
   when $i_x(j) = i^{(l)}$ for the first time, if the corresponding edit operation is:

$$\begin{aligned} \text{insert} &\implies \text{increment } d_j \text{ by } 1 \\ delete &\implies \text{decrement } d_j \text{ by } 1 \\ otherwise &\implies \text{do not change } d_j \end{aligned}$$

4. Idle state:
$$d_j = i_x(j) - i_y(j) \Longrightarrow x_{i_x(j)} = y_{i_y(j)} \tag{1}$$
When $d_j = i_x(j) - i_y(j)$, the x and y strings are aligned at that point $\Longrightarrow x_{i_x(j)} = y_{i_y(j)} \Longrightarrow f(x,r)_j = f(y,r)_j$
x and y would continue to be in alignment (defined as the idle state) unless and until d_j does not change, i.e. we do not encounter an edit operation.

5. We can verify from the above image, when $d_j = x_{i_x(j)} - y_{i_y(j)}$ then $f(x,r)_j = f(y,r)_j$.
Except, when we have replace operation (the cell marked red in the last row). Because in case of replace, x and y are aligned, but their values are different. Ergo, the system won't be in an idle state.

Kennel Struggle:

1. **Kennel** $\equiv$ Goalspot.
2. **Cat** $\equiv$ Particle. Cat has a random moment {-1,0,+1} with probability {1/4,1/2,1/4}.
3. **Dog** $\equiv$ Edit distance. The dog is incharge of moving the kennel. The actions involve moving the kennel to the left($\equiv$ Deletion), moving the kennel to the right($\equiv$ Insertion), barking ($\equiv$ Bit flip). The Dog disappears after k steps. **Note:** When the Cat reaches the Kennel, the dog has to make a move because Cat reaching the Kennel *equiv* Particle=Goalspot, referring to 1, this would mean that the system is in idle state and would continue to remain in idle state unless the value of d_j does not change.

Success Scenario: Cat reaches an empty Kennel.
Probability of Cat reaching an empty Kennel in l steps is least if the dog uses the following strategy:

1. Move the Kennel to the left by 1 position
2. Wait until the cat reaches the Kennel
3. Move the Kennel to left by 1 position

As Dog has k moves, the above process will repeat k times.

The probability that the Cat reaches position 1 for the first time in t moves is given by $q(t,1)$. Hence for k steps, we have $\sum_{t_1+t_2+\ldots+t_k<l}(q(t_1,1)q(t_2,1)\ldots q(t_k,1))$. Hence,

$$\Pr(\text{Cat reaches empty Kennel in atmost} l \text{ steps}) \geq \sum_{t_1+t_2+\ldots+t_k \leq l}(q(t_1,1)q(t_2,1)\ldots q(t_k,1))$$

$$\Pr(\text{Cat reaches empty Kennel in atmost} l \text{ steps}) \geq \sum_{t \leq l} q(t,k)$$

As the Particle moment is defined by $h_j(x_{i_x(j)}) - h_j(y_{i_y(j)})$, the Particle would move only when $x_{i_x(j)} \neq y_{i_y(j)}$

$$\Pr(\Delta_H(f(x,r), f(y,r)) \leq l) \geq \sum_{t \leq l} q(t,k)$$

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

□

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.