Approximate Similarity Search Under Edit Distance Using Locality Sensitive Hashing

Algorithm

We consider a random underlying function ρ which takes an alphabet (Σ) and the length of the output string ($|s|$) as the parameters and returns random numbers $(r1, r2)$. $\therefore$ ρ would have $|\Sigma| \times |s|$ keys which would all return 2 random values in [0,1).
We randomly choose a value of $p \leq 1/3$.
d is the length of the maximum string and n is the total number of strings.

How to Hash:

1. **Calculate p_a and p_r:**

$$p_a = \sqrt{\frac{p}{1+p}}$$

$$p_r = \frac{\sqrt{p}}{\sqrt{1+p} - \sqrt{p}}$$

2. **Initialise** $i = 0$ and $s =$ ””.

3. **Hashing:** while $i < |x|$ and $|s| < \frac{8d}{1-p_a} + 6 \log n$, we get the values $(r1, r2)$ from $\rho(x_i, |s|)$:

   (a) if $r1 \leq p_a$, **hash-insert**: append $\perp$ to s

   (b) if $r1 > p_a$ and $r2 \leq p_r$, **hash-replace**: append $\perp$ to s and increment i

   (c) if $r1 > p_a$ and $r2 > p_r$, **hash-match**: append x_i to s and increment i.

Analysis

Note: we consider only the case where $h_\rho(x) = h_\rho(y)$ as they would belong to the same bucket. We ignore the cases where $h_\rho(x) \neq h_\rho(y)$, as the grid walk in this case would end up at STOP node.
Note: if $h_\rho(x) = h_\rho(y)$, then, hash-match occurs only when $x_{i_x} = y_{i_y}$ as hash-math inserts the actual alphabet (not $\perp$) in which case both $\tau_k(x)$ and $\tau_k(y)$

would have same operation.

For every value in transcript (τ) of x and y, we can define the edit distance operation as:

When $x_{i_x} \neq y_{i_y}$

$\tau_k(x)$	$\tau_k(y)$	ED Operation
hash-insert	hash-insert	loop
hash-insert	hash-replace	insert
hash-replace	hash-insert	delete
hash-replace	hash-replace	replace

When $x_{i_x} = y_{i_y}$

$\tau_k(x)$	$\tau_k(y)$	ED Operation
hash-insert	hash-insert	loop
hash-replace	hash-replace	match
hash-match	hash-match	match

Hence, given any two string x and y, if $h_\rho(x) = h_\rho(y)$, then we can find a path from $(0,0)$ to $(|x|,|y|)$ in the Edit Distance table which can be derived from the above table. And applying these operations on x, we would get the string y.

Note: in [McC21], they have considered a "STOP" state in the Edit Distance table which is visited in case of inconsistency.

Lemmas and Proofs

Lemma (Lemma 6). *For any string x of length d, $\Pr_\rho(\tau(x,\rho)$ is complete$) \geq 1 - 1/n^2$.*

Proof. Recall that a transcript $\tau(x,\rho)$ is complete if $|\tau(x,\rho)| < 8d/(1-p_a) + 6\log n$. If the transcript contains l insert operations, $|\tau(x,\rho)| \leq d + l$ since the maximum length of a string is d.
We check the bounds for the probability of $l > 7d/(1-p_a) + 6\log n$.

Consider success to be when we do not have hash-insert operations. This behaves like a geometric progression with probability $= (1-p_a)$. The expected number of hash-inserts we need to get an operation which is not a hash-insert would be $\frac{1}{1-p_a}$. Hence,

$$E[l] = \frac{d}{1-p_a}$$

The relevant Chernoff bound is:

$$\begin{aligned} \Pr(X \geq (1+\delta)E[X]) &\leq \left(\frac{e^\delta}{(1+\delta)^(1+\delta)}\right)^{E[X]} \\ &= e^{[\delta-(1+\delta)\ln(1+\delta)]*E[X]} \end{aligned} \quad (1)$$

We will be manipulating Equation (1) in the sequel.
To calculate $\Pr(l > 7d/(1-p_a) + 6\log n)$ with $E[X] = d/(1-p_a)$, we set $\delta = 6 + \frac{6(1-p_a)\log n}{d}$.
We know that $\delta - (1+\delta)\ln(1+\delta) \leq -\delta^2/3$, when $0 \leq \delta \leq 1$ [HR90].
However, our value of $\delta = 6 + \frac{6(1-p_a)\log n}{d} > 6$, hence we cannot use the above bound. Instead, we use the fact that:

$$\delta - (1+\delta)\ln(1+\delta) \leq -\delta/3$$

when $\delta > 1$.
Substituting this in Equation (1) and using the fact that $\frac{6d}{(1-p_a)} > 0$, we get:

$$\begin{aligned} \Pr(l \geq (1+\delta)E[X]) &\leq e^{-\delta E[X]/3} \\ &< e^{-(6d/(1-p_a)+6\log n)/3} \\ &< e^{-(6\log n)/3} \\ &= 1/n^2 \end{aligned}$$

Hence, $\Pr(l < 7d/(1-p_a) + 6\log n) > 1 - 1/n^2$

□

Lemma (Lemma 7). *Consider a walk through $G(x,y)$ which at step i takes the edge with label corresponding to $g_i(x,y,\rho)$. Assume k is such that the prefix $g(x,y,\rho)[k]$ of length k is alive. Then after k steps, the walk arrives at node $(i(x,k,\rho), i(y,k,\rho))$.*

Proof. Intuitively:
If the gridwalk $g(x,y,\rho)$ is derived from transcript τ, then at the iteration k, the position would be at $(i(x,k,\rho), i(y,k,\rho))$, i.e. the respective pointers of x and y (provided that we do not enounter "STOP" in the process, i.e., the gridwalk is alive). □

Lemma (Lemma 8). *Let x and y be any two strings, and ρ be any underlying function where both $\tau(x,\rho)$ and $\tau(y,\rho)$ are complete.*
Then $h_\rho(x) = h_\rho(y)$ if and only if $g(x,y,\rho)$ is alive. Furthermore, if $h_\rho(x) = h_\rho(y)$ then the path defined by $g(x,y,\rho)$ reaches node $(|x|,|y|)$.

Proof. Intuitively:
$g(x,y,\rho)$ is not alive only when it goes to the stop node, this happens only when one of the hash values is a hash-match and the other is not. As hash-match records the value of x_i, and if $h_\rho(x) = h_\rho(y)$, then both the hashes would have hash-match. Also, if it is alive, as we end each string with \$, it has to go to the very end, hence it will reach $(|x|,|y|)$ and the hash values would be equal. □

Lemma (Lemma 9). *Let x and y be any two strings, and for any $k < 8d/(1-p_a) + 6\log n$ Let E_k be the event that $i(x,k,\rho) < |x|$, $i(y,k,\rho) < |y|$, and*

$x_{i(x,k,\rho)} \neq y_{i(y,k,\rho)}$. *Then if* $\Pr_\rho[E_k] > 0$*, the following four conditional bounds hold:*

$$\Pr_\rho[g_k(x,y,\rho) = loop|E_k] = p_a^2$$

$$\Pr_\rho[g_k(x,y,\rho) = delete|E_k] = p_a(1-p_a)p_r$$

$$\Pr_\rho[g_k(x,y,\rho) = insert|E_k] = p_a(1-p_a)p_r$$

$$\Pr_\rho[g_k(x,y,\rho) = replace|E_k] = (1-p_a)^2 p_r^2$$

Proof.

$$\Pr_\rho(\tau_k(x,\rho) = hash - insert|E_k) = p_a$$

$$\Pr_\rho(\tau_k(x,\rho) = hash - replace|E_k) = (1-p_a)p_r$$

$$\Pr_\rho(\tau_k(x,\rho) = hash - match|E_k) = (1-p_a)(1-p_r)$$

Loop operation occurs when $\tau_k(x,\rho)$ is hash-insert and $\tau_k(y,\rho)$ is hash-insert. Similarly, delete occurs when we have hash-replace-hash-insert, insert occurs when we have hahd-insert-hash-replace and replace occurs when we have hash-replace-hash-replace. Multiplying the probabilities, we get the above values.
Note: $p_r = p_a/(1-p_a)$, hence, $p_r(1-p_a) = p_a$. We can write $p_a(1-p_a)p_r = p_a^2$ and $(1-p_a)^2p_r^2 = p_a^2$. Substituting the values above, we get, $\Pr_\rho[g_k(x,y,\rho) = loop|E_k] = \Pr_\rho[g_k(x,y,\rho) = delete|E_k] = \Pr_\rho[g_k(x,y,\rho) = insert|E_k] = \Pr_\rho[g_k(x,y,\rho) = replace|E_k] = p_a^2$ □

Lemma (Lemma 11). *Let x and y be two strings that do not contain \$. Then if $ED(x,y) = r$,*

1. *there exists a transformation T of length r that solves $x \cdot \$$ and $y \cdot \$$*
2. *there does not exist any transformation T' of length $< r$ that solves $x \cdot \$$ and $y \cdot \$$*

Proof. ED can be transformed into transformation.
ED of x and y is same as the ED of $x \cdot \$$ and $y \cdot \$$. AS ED is the minimum distance between the strings, we cannot find a transformation of length less than that. □

Note: T is the edit operations from the Edit Distance whereas $\mathcal{T}$ is obtained from the Grid Walk by removing loop and match.
For each transformation, we find the first index which differs from y and apply the transformation there. We generate r such strings in the process.

Lemma (Lemma 12). *Let x and y be two distinct strings and let $T = \mathcal{T}(x,y,\rho)$. Then $h_\rho(x) = h_\rho(y)$ if and only if T solves x and y.*

Proof. If $T = \mathcal{T}(x, y, \rho)$ solves x and y, then the gridwalk $g(x, y, \rho)$ starts at $(0, 0)$ and ends at$(|x|, |y|)$, this is possible only when $h_\rho(x) = h_\rho(y)$
If $h_\rho(x) = h_\rho(y)$, then $\mathcal{T}(x, y, \rho)$ would be derived from the gridwalk $g(x, y, \rho)$ and hence it would solve x and y. $\square$

Lemma (**Lemma 13**). *For any $\$ -$ terminal strings x and y, let T be a transformation of length t that is valid for x and y. Then*

$$p^t - 2/n^2 \leq \Pr_\rho(T \text{ is a prefix of } \mathcal{T}(x, y, \rho)) \leq p^t$$

Proof.

1. **To prove** $\Pr_\rho(T$ **is a prefix of** $\mathcal{T}(x, y, \rho)) \leq p^t$**:**
   Define G_T as: all the transformations T_g which contain T as a prefix, G_T is the set of all the gridwalks such that T_g are derived from G_T.
   $\implies$ G_T is alive(as G_T does not contain STOP).
   To calculate the $\Pr(g(x, y, \rho) \in G_T)$
   Induction approach:
   (a) **Base Step:** for $t = 0$, all the transforms are valid, hence, $\Pr(g(x, y, \rho) \in G_T) = 1$.
   (b) **Inductive Hypothesis:** $\sum \Pr_\rho(g(x, y, \rho)[t-1]$ is a prefix of $G_{T'}) = p^{(t-1)}$ *where $G_{T'}$ is a set of all the transformations T_g with the last operation removed.*
   (c) **Inductive Step:**
   Let the last operation be σ.
   $g(x, y, \rho) = g'(x, y, \rho) \cdot \sigma \cdot \{loop, match\}^* \cdot \{loop\}^*$
   Define:
   $g'' = g'(x, y, \rho) \cdot \sigma,$
   $g''' = g'' \cdot \{loop, match\}^*,$
   $g(x, y, \rho) = g''' \cdot \{loop\}^*$
   The Probability can be defined as:
   $\Pr_\rho(g(x, y, \rho) \in G_T) = \Pr_\rho(g'(x, y, \rho) \in G_{T'}) \cdot \Pr(g'' \in G_T | g'(x, y, \rho) \in G_{T'}) \cdot \Pr(g''' \in G_T | g'' \in G_T) \cdot \Pr_\rho(g(x, y, \rho) \in G_T | g''' \in G_T)$
   Note: we can directly multiply the probabilities as all the values are subset of the respective conditions.
   Now,
   $\Pr_\rho(g'(x, y, \rho) \in G_{T'}) = p^{t-1}$ — *from Inductive hypothesis*
   $\Pr(g'' \in G_T | g'(x, y, \rho) \in G_{T'}) = p_a^2$ — *as from Lemma 9, all the operations have a probability of p_a^2.*
   $\Pr(g''' \in G_T | g'' \in G_T) = 1$ — *as G_T does not contain "STOP" and we are done with the last operation, the only option left is "LOOP" or "MATCH" if $x_{i_x} = y_{i_y}$ then we have "MATCH", o.w. "LOOP".*
   $\Pr_\rho(g(x, y, \rho) \in G_T | g''' \in G_T) = \frac{1}{1-p_a^2}$ — *as addind loop operations is like a Poisson distribution with the probability of success$=1 - p_a^2$*
   Multiplying all, we get: $\Pr_\rho(g(x, y, \rho) \in G_T) = p^{t-1} \frac{p_a^2}{1-p_a^2} = p^t$

2. To prove $p^t - 2/n^2 \leq \Pr_\rho(T \text{ is a prefix of } \mathcal{T}(x,y,\rho))$:
   T is a prefix of $\mathcal{T}(x,y,\rho)$ if $g(x,y,\rho) \in G_T$ and $g(x,y,\rho)$ is complete, i.e., the transcripts $\tau(x,\rho)$ and $\tau(y,\rho)$ are complete:

$$\begin{aligned}\Pr(T \text{ is a prefix of } \mathcal{T}(x,y,\rho)) = &\Pr(g(x,y,\rho) \in G_T) + \Pr(g(x,y,\rho) \text{ is complete}) \\ &- \Pr(g(x,y,\rho) \in G_T \cap g(x,y,\rho) \text{ is complete})\end{aligned}$$

$$\begin{aligned}\Pr(g(x,y,\rho) \in G_T) &= p^t\text{—from point 1} \\ \Pr(g(x,y,\rho) \text{ is complete}) &= 1 - \Pr(g(x,y,\rho) \text{ is not complete}) \\ &= 1 - \Pr(\tau(x,\rho) \text{ is not complete} | \tau(y,\rho) \text{ is not complete}) \\ &\geq 1 - (\Pr(\tau(x,\rho) \text{ is complete}) + \Pr(\tau(y,\rho) \text{ is complete})) \\ &\qquad \text{—using union bound} \\ &= 1 - (\frac{1}{n^2} + \frac{1}{n^2}) \text{ —from Lemma 6} \\ &= 1 - \frac{2}{n^2}\end{aligned}$$

$$\begin{aligned}\Pr(g(x,y,\rho) \in G_T \cap g(x,y,\rho) \text{ is complete}) &\leq 1 \\ -\Pr(g(x,y,\rho) \in G_T \cap g(x,y,\rho) \text{ is complete}) &\geq -1\end{aligned}$$

   Adding all, we get:

$$\Pr(T \text{ is a prefix of } \mathcal{T}(x,y,\rho)) \geq p^t - \frac{2}{n^2}$$

□

Bounds on Collision Probability:

Lemma (*Combining* **Lemma 14** *and* **Lemma 15**:)**.** *If x and y satisfy $ED(x,y) \leq r$, then* $\Pr_\rho(h_\rho(x) = h_\rho(y)) \geq p^r - \frac{2}{n^2}$.
If x and y satisfy $ED(x,y) \geq cr$, then $\Pr_\rho(h_\rho(x) = h_\rho(y)) \leq (3p)^{cr}$.

Proof. We get the lower bound from **Lemma 13**, if $ED(x,y) \leq r$, then there exists a transformation T of size less than r. The probability that T is a prefix of $\mathcal{T}(x,y,\rho) \geq p^t - \frac{2}{n^2}$.
For the upper bound, let $\mathcal{T}$ be the set of all the transformations that solve x and y, then $\Pr_{h \in H}(h(x) = h(y)) = \sum_{T \in \mathcal{T}} p^{|T|}$.
We can imagine the transformations in form of a trie with each node having 3 children for insert delete and replace and the minimum depth of any leaf is cr. For all the nodes with $depth > cr$, we merge the children into the parent node, so the probability value which was earlier $3p^i$ for the 3 child nodes would now

be p^{i-1} which is obviously greater as $p < 1/3$. Hence,

$$\begin{aligned}\Pr_{h\in H}(h(x)=h(y)) &= \sum_{T\in\mathcal{T}} p^{|T|} \\ &\leq \sum_{T\in\mathcal{T}} p^{cr} \\ &= (3p)^{cr} \text{—as we have } 3^{cr} \text{ leaves}\end{aligned}$$

$\square$

References

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

[McC21] S. McCauley. Approximate similarity search under edit distance using locality-sensitive hashing. In *ICDT 2021*, pages 21:1 – 21:22, 2021.