Path histogram distance and complete subtree histogram distance for rooted labelled caterpillars

Taiga Kawaguchi\textsuperscript{a}, Takuya Yoshino\textsuperscript{b} and Kouichi Hirata\textsuperscript{b}

\textsuperscript{a}Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology, Fukuoka, Japan; \textsuperscript{b}Department of Artificial Intelligence, Kyushu Institute of Technology, Iizuka, Fukuoka, Japan

\section*{ABSTRACT}
A rooted labelled caterpillar (a caterpillar, for short) is a rooted labelled unordered tree transformed to a path after removing all the leaves in it. In this paper, we discuss two histogram distance between caterpillars. One is a path histogram distance as an $L_1$-distance between the histograms of paths from the root to every leaf and another is a complete subtree histogram distance as an $L_1$-distance between the histograms of complete subtrees for every node. While the latter is always a metric for general trees, the former is not a metric. In this paper, we show that, for caterpillars, the path histogram distance is always a metric, simply linear-time computable and incomparable with the edit distance. Furthermore, we give experimental results for caterpillars in real data of comparing the path histogram distance and the complete subtree histogram distance with the isolated-subtree distance as the most general tractable variation of the edit distance.

\section*{1. Introduction}

An efficient comparison of tree-structured data such as HTML and XML data for web mining, DNA and glycan data for bioinformatics, phrase trees for natural language processing, and so on, is one of the important tasks for data mining. The tree-structured data are often regarded as rooted labelled unordered trees (trees, for short). The most famous distance measure between trees is the edit distance (Tai, 1979).

The edit distance is formulated as the minimum cost of edit operations, consisting of a substitution, a deletion and an insertion, applied to transform from a tree to another tree. Unfortunately, it is known that the problem of computing the edit distance between trees is MAX SNP-hard (Zhang & Jiang, 1994). Furthermore, this statement also holds even if trees are binary (Hirata, Yamamoto, & Kuboyama, 2011).

In order to achieve an efficient comparison of trees, many variations of the edit distance have developed as more structurally sensitive distances (cf., Kuboyama, 2007; Yoshino & Hirata, 2017). Almost variations are metrics and the problem of computing them is tractable as cubic-time computable (Yamamoto, Hirata, & Kuboyama, 2013; Yoshino & Hirata, 2017; Hirata, 2014).
In particular, the isolated-subtree distance (or constrained distance) (Zhang, 1996) is the most general tractable variation of the edit distance (Yoshino & Hirata, 2017).

On the other hand, the restriction of the form of trees provides another direction to an efficient comparison of trees. A caterpillar (cf., Gallian, 2007) is a tree transformed to a path after removing all the leaves in it. Whereas the caterpillars are very restricted and simple, there are many caterpillars in a real dataset, see Table 1 in Section 4. Recently, Muraka, Yoshino, and Hirata (2018) have designed the algorithm to compute the edit distance for caterpillars in cubic time under the unit cost function.

Hence, whereas both directions provide tractable metrics for the comparison of trees, they have not achieved high efficiency as linear-time computable. By focusing on such efficiency, the histogram distances based on local information (Aratsu, Hirata, & Kuboyama, 2008; Kailing, Kriegel, Schönauer, & Seidl, 2004; Li, Wang, Li, & Gao, 2013) are known as the constant-factor lower bounding distances of the edit distance. Whereas we can compute them more efficiently as linear-time computable in almost cases than the variations of the edit distance, none of them is a metric.

In order to achieve the comparison of trees with preserving both the efficiency and the metricity, in this paper, we investigate a path histogram distance and a complete subtree histogram distance and restrict trees to caterpillars.

A path histogram of a tree consists of pairs of a path from the root to a leaf and its frequency in the tree. Then, a path histogram distance for two trees is an \( L_1 \)-distance between the path histograms for the two trees. Note that the path histogram distance is not a metric for trees in general.

A complete subtree histogram distance is an \( L_1 \)-distance between histograms consisting of complete subtrees in two trees (Akutsu, Fukagawa, Halldórsson, Takasu, & Tanaka, 2013). The complete subtree histogram is computable in linear time and always a metric, which is also a metric for trees. It is always greater than or equal to the tree edit distance in general (Akutsu et al., 2013), that is, it is comparable with the tree edit distance. As an extreme case, there exist two caterpillars (two paths with the same length such that the labels of leaves are different) such that the tree edit distance between them is one, but the complete subtree histogram distance is the number of nodes in two caterpillars.

Hence, in this paper, we investigate the path histogram distance for caterpillars, together with the complete subtree histogram distance. First, we show that the path histogram distance is always a metric for caterpillars. Also, we show that the path histogram distance can be computed in linear time and it is incomparable with the edit distance (and its variations) and the complete subtree histogram distance.

**Table 1.** The number of caterpillars.

| dataset  | #cat (c) | #data | %     |
|----------|----------|-------|-------|
| N-glycans| 514      | 2142  | 23.996|
| all-glycans | 8005    | 10,704| 74.785|
| CSLOGS   | 41,592   | 59,691| 69.679|
| dblp     | 5,154,295| 5,154,530| 99.995|
| SwissProt| 6804     | 50,000| 13.608|
| TPC-H    | 86,805   | 86,805| 100.000|
| Auction  | 0        | 37    | 0     |
| Nasa     | 0        | 2430  | 0     |
| Auction− | 259      | 259   | 100.000|
| Nasa−    | 21,245   | 27,921| 76.089|
Furthermore, we give experimental results for caterpillars in real data, such as N-glycans and all of the glycans (we refer to all-glycans) from KEGG, CSLOGS, dblp, SwissProt, TPC-H, Auction and Nasa from UW XML Repository datasets represented by Table 1 in Section 4. Then, we compare the path histogram distance and the complete subtree histogram distance with the isolated-subtree distance for running time, distributions and scatters.

2. Preliminaries

A tree \(T\) is a connected graph \((V, E)\) without cycles, where \(V\) is the set of vertices and \(E\) is the set of edges. We denote \(V\) and \(E\) by \(V(T)\) and \(E(T)\). The size of \(T\) is \(|V|\) and denoted by \(|T|\). We sometime denote \(v \in V(T)\) by \(v \in T\). We denote an empty tree \((\emptyset, \emptyset)\) by \(\emptyset\). A rooted tree is a tree with one vertex \(r\) chosen as its root. We denote the root of a rooted tree \(T\) by \(r(T)\).

Let \(T\) be a rooted tree such that \(r = r(T)\) and \(u, v, w \in T\). We denote the unique path from \(r\) to \(v\), that is, the tree \((V', E')\) such that \(V' = \{v_1, \ldots, v_k\}\), \(v_1 = r\), \(v_k = v\) and \((v_i, v_{i+1}) \in E'\) for every \(i\) \((1 \leq i \leq k - 1)\), by \(U_P(v)\). The parent of \(v\) \((v \neq r)\) is its adjacent vertex on \(U_P(v)\) and the ancestors of \(v\) \((v \neq r)\) are the vertices on \(U_P(v) - \{v\}\). We say that \(u\) is a child of \(v\) if \(v\) is the parent of \(u\) and \(u\) is a descendant of \(v\) if \(v\) is an ancestor of \(u\). We call a vertex with no children a leaf and denote the set of all the leaves in \(T\) by \(lv(T)\).

Let \(T\) be a tree \((V, E)\) and \(v\) a vertex in \(T\). A complete subtree of \(T\) at \(v\), denoted by \(T[v]\), is a tree \(T' = (V', E')\) such that \(r(T') = v\), \(V' = \{u \in V \mid u \leq v\}\) and \(E' = \{(u, w) \in E \mid u, w \in V'\}\). The degree of \(v\), denoted by \(d(v)\), is the number of children of \(v\) and the degree of \(T\), denoted by \(d(T)\), is \(\max\{|d(v)\mid v \in T\}\). The height of \(v\), denoted by \(h(v)\), is \(\max\{|U_P(v)| \mid w \in lv(T[u])\}\), and the height of \(T\), denoted by \(h(T)\), is \(\max\{h(v) \mid v \in T\}\).

We use the ancestor orders \(<\) and \(\leq\), that is, \(u < v\) if \(v\) is an ancestor of \(u\) and \(u \leq v\) if \(u < v\) or \(u = v\). We say that \(w\) is the least common ancestor of \(u\) and \(v\), denoted by \(u \sqcup v\), if \(u \leq w\), \(v \leq w\) and there exists no vertex \(w' \in T\) such that \(w' \leq w\), \(u \leq w'\) and \(v \leq w'\).

For \(v \in T\), \(pre(v)\) (resp., \(post(v)\)) denotes the occurrence order of \(v\) in the preorder (resp., postorder) traversal of all the vertices in \(T\). Then, we say that \(u\) is to the left of \(v\) in \(T\) if \(pre(u) \leq pre(v)\) and \(post(u) \leq post(v)\). We say that a rooted tree is ordered if a left-to-right order among siblings is given; unordered otherwise. We say that a rooted tree is labelled if each vertex is assigned a symbol from a fixed finite alphabet \(\Sigma\). For a vertex \(v\), we denote the label of \(v\) by \(l(v)\), and sometimes identify \(v\) with \(l(v)\). In this paper, we call a rooted labelled unordered tree a tree simply.

We say that a tree \(C\) such that \(r = r(C)\) is a caterpillar (cf., Gallian, 2007) if \(C\) is transformed to a path \(U_P(v)\) for some \(v \in C\) after removing \(lv(C)\). For a caterpillar \(C\), we call the remained path \(U_P(v)\) a backbone of \(C\) and denote it by \(bb(C)\). It is obvious that \(V(C) = V(bb(C)) \cup lv(C)\). We can determine whether or not a tree \(T\) is a caterpillar in \(O(|T|)\) time.

Next, we introduce an edit distance and a Tai mapping.

Definition 2.1(Edit operations (T'ai, 1979)): The edit operations of a tree \(T\) are defined as follows (Figure 1):

1. Substitution: Change the label of the vertex \(v\) in \(T\).
2. Deletion: Delete a vertex \(v\) in \(T\) with parent \(v'\), making the children of \(v\) become the children of \(v'\). The children are inserted in the place of \(v\) as a subset of the children.
Substitution \((v \mapsto w)\)

Deletion \((v \mapsto \varepsilon)\)

Insertion \((\varepsilon \mapsto v)\)

**Figure 1.** Edit operations for trees.

...of \(v'\). In particular, if \(v\) is the root in \(T\), then the result applying the deletion is a forest consisting of the children of the root.

(3) **Insertion:** The complement of deletion. Insert a vertex \(v\) as a child of \(v'\) in \(T\) making \(v\) the parent of a subset of the children of \(v'\).

Let \(\varepsilon \in \Sigma\) denote a special blank symbol and define \(\Sigma_{\varepsilon} = \Sigma \cup \{\varepsilon\}\). Then, we represent each edit operation by \((l_1 \mapsto l_2)\), where \((l_1, l_2) \in (\Sigma_{\varepsilon} \times \Sigma_{\varepsilon} \setminus \{(\varepsilon, \varepsilon)\})\). The operation is a substitution if \(l_1 \neq \varepsilon\) and \(l_2 \neq \varepsilon\), a deletion if \(l_2 = \varepsilon\), and an insertion if \(l_1 = \varepsilon\). For vertices \(v\) and \(w\), we also denote \((l(v) \mapsto l(w))\) by \((v \mapsto w)\). We define a cost function \(\gamma(\Sigma_{\varepsilon} \times \Sigma_{\varepsilon} \setminus \{(\varepsilon, \varepsilon)\}) \mapsto \mathbb{R}^+\) on pairs of labels. We often constrain a cost function \(\gamma\) to be a metric, that is, \(\gamma(l_1, l_2) \geq 0\), \(\gamma(l_1, l_2) = 0\) iff \(l_1 = l_2\), \(\gamma(l_1, l_2) = \gamma(l_2, l_1)\) and \(\gamma(l_1, l_3) \leq \gamma(l_1, l_2) + \gamma(l_2, l_3)\). In particular, we call the cost function that \(\gamma(l_1, l_2) = 1\) if \(l_1 \neq l_2\) a unit cost function.

**Definition 2.2 (Edit distance (Tai, 1979)):** For a cost function \(\gamma\), the cost of an edit operation \(e = l_1 \mapsto l_2\) is given by \(\gamma(e) = \gamma(l_1, l_2)\). The cost of a sequence \(E = e_1, \ldots, e_k\) of edit operations is given by \(\gamma(E) = \sum_{i=1}^{k} \gamma(e_i)\). Then, an edit distance \(\tau_{\text{TAI}(T_1, T_2)}\) between trees \(T_1\) and \(T_2\) is defined as follows:

\[
\tau_{\text{TAI}(T_1, T_2)} = \min \left\{ \gamma(E) \middle| E \text{ is a sequence of edit operations transforming } T_1 \text{ to } T_2 \right\}.
\]

**Definition 2.3 (Tai mapping (1979)):** Let \(T_1\) and \(T_2\) be trees. We say that a triple \((M, T_1, T_2)\) is a Tai mapping (a mapping, for short) from \(T_1\) to \(T_2\) if \(M \subseteq V(T_1) \times V(T_2)\) and every pair \((v_1, w_1)\) and \((v_2, w_2)\) in \(M\) satisfies the following condition:

(1) \(v_1 = v_2\) iff \(w_1 = w_2\) (one-to-one condition).
(2) \(v_1 \leq v_2\) iff \(w_1 \leq w_2\) (ancestor condition).

We will use \(M\) instead of \((M, T_1, T_2)\) when there is no confusion denote it by \(M \in \mathcal{M}_{\text{TAI}(T_1, T_2)}\). Furthermore, we denote the set \(\{v \in T_1 \mid (v, w) \in M\}\) by \(M_{|1}\) and \(\{w \in T_1 \mid (v, w) \in M\}\) by \(M_{|2}\).
Let $M$ be a mapping from $T_1$ to $T_2$. Then, the cost $\gamma(M)$ of $M$ is given as follows:

$$\gamma(M) = \sum_{(v, w)\in M} \gamma(v, w) + \sum_{v\in M_1} \gamma(v, e) + \sum_{w\in M_2} \gamma(e, w).$$

Trees $T_1$ and $T_2$ are isomorphic, denoted by $T_1 \equiv T_2$, if there exists a mapping $M \in \mathcal{M}_{\text{TAI}}(T_1, T_2)$ such that $|M_1| = |M_2| = 0$ and $\gamma(M) = 0$.

**Theorem 2.4 (Tai, 1979):**

$$\tau_{\text{TAI}}(T_1, T_2) = \min\{\gamma(M) | M \in \mathcal{M}_{\text{TAI}}(T_1, T_2)\}.$$

**Theorem 2.5:** The following statements hold.

1. For trees $T_1$ and $T_2$, the problem of computing $\tau_{\text{TAI}}(T_1, T_2)$ is MAX SNP-hard (Zhang & Jiang, 1994). This statement holds even if both $T_1$ and $T_2$ are binary, the maximum height of $T_1$ and $T_2$ is at most 3 or the cost function is the unit cost function (Akutsu et al., 2013; Hirata et al., 2011).

2. For caterpillars $C_1$ and $C_2$, we can compute $\tau_{\text{TAI}}(C_1, C_2)$ in $O(h^2 \lambda^3)$ time, where $h = \max\{h(C_1), h(C_2)\}$ and $\lambda = \max\{|lv(C_1)|, |lv(C_2)|\}$. Furthermore, if we adopt the unit cost function, then we can compute it in $O(h \lambda^3)$ time (Muraka et al., 2018).

Finally, we introduce an isolated-subtree mapping and an isolated-subtree distance as the variations of the Tai mapping and the edit distance.

**Definition 2.6 (Isolated-subtree mapping and distance (Zhang, 1996)):** Let $T_1$ and $T_2$ be trees and $M \in \mathcal{M}_{\text{TAI}}(T_1, T_2)$. We say that $M$ is an isolated-subtree mapping, denoted by $M \in \mathcal{M}_{\text{ILST}}(T_1, T_2)$, if $M$ satisfies the following condition:

$$\forall (v_1, w_1)(v_2, w_2)(v_3, w_3) \in M(v_3 < v_1 \sqcup v_2 \Leftrightarrow w_3 < w_1 \sqcup w_2).$$

Furthermore, we define an isolated-subtree distance $\tau_{\text{ILST}}(T_1, T_2)$ as follow:

$$\tau_{\text{ILST}}(T_1, T_2) = \min\{\gamma(M) | M \in \mathcal{M}_{\text{ILST}}(T_1, T_2)\}.$$

It is obvious that $\mathcal{M}_{\text{ILST}}(T_1, T_2) \subseteq \mathcal{M}_{\text{TAI}}(T_1, T_2)$ and then $\tau_{\text{TAI}}(T_1, T_2) \leq \tau_{\text{ILST}}(T_1, T_2)$. In contrast to Theorem 2.5, the following theorem also holds.

**Theorem 2.7 (cf., Yamamoto et al., 2013):** Let $T_1$ and $T_2$ be trees. Then, we can compute $\tau_{\text{ILST}}(T_1, T_2)$ in $O(n^2 d)$ time, where $n = \max\{|T_1|, |T_2|\}$ and $d = \min(d(T_1), d(T_2))$.

It is known that $\tau_{\text{ILST}}$ is the most general tractable variation of $\tau_{\text{TAI}}$ (Yoshino & Hirata, 2017).

### 3. Path histogram distance and complete subtree histogram distance for caterpillars

In this section, we formulate a path histogram distance and a complete subtree histogram distance between caterpillars.

First, we introduce a path histogram distance. Let $T$ be a tree such that $r = r(T)$. Then, for $v \in lv(T)$, we regard the path $P = UP_r(v)$ such that $V(P) = \{v_1, \ldots, v_k\}$, $v_1 = r$, $v_k = v$ and $(v_i, v_{i+1}) \in E(P)$ for every $i$ ($1 \leq i \leq k - 1$) as a string $l(v_1) \cdots l(v_k)$ on $\Sigma$ and denote it by
For a tree \( T \), a path histogram \( H_{p(T)} \) of \( T \) consists of pairs \( (s, f(s, T)) \) for every \( s \in S(T) \).

For trees \( T_1 \) and \( T_2 \), a path histogram distance \( \delta_{p(T_1, T_2)} \) between \( T_1 \) and \( T_2 \) is defined as an \( L_1 \)-distance between \( H_{p(T_1)} \) and \( H_{p(T_2)} \), that is,

\[
\delta_{p(T_1, T_2)} = \sum_{s \in S(T_1) \cup S(T_2)} |f(s, T_1) - f(s, T_2)|.
\]

For \( \lambda = |\nu(T)| \), it is obvious that \( |H_{p(T)}| \leq \lambda \) and \( \sum_{s \in S(T)} f(s, T) = \lambda \).

Next, we introduce a complete subtree histogram distance. We denote the set \( \{\nu[v] \mid v \in T\} \) of all the complete subtrees in \( T \) by \( \mathcal{C}(T) \). For \( c \in \mathcal{C}(T) \), the number of occurrences of \( c \) in \( T \) by \( f(c, T) \).

For \( n = |T| \), it is obvious that \( |\mathcal{H}_{\mathcal{C}(T)}| \leq n \) and \( \sum_{c \in \mathcal{C}(T)} f(c, T) = n \). As the properties of \( \delta_{\mathcal{C}_s} \), the following theorem holds.

**Theorem 3.3 (cf., Akutsu et al., 2013):** Let \( T_1 \) and \( T_2 \) be trees.

1. \( \delta_{\mathcal{C}_s} \) is a metric.
2. We can compute \( \delta_{\mathcal{C}_s(T_1, T_2)} \) in \( O(|T_1| + |T_2|) \) time.
3. \( \tau_{\mathcal{C}_s(T_1, T_2)} \leq \delta_{\mathcal{C}_s(T_1, T_2)} \).

In the remainder of this section, we discuss the properties of the path histogram distance \( \delta_{p} \). Start the following example.

**Example 3.4** For trees, \( \delta_{p} \) is not a metric in general. For trees \( T_1 \) and \( T_2 \) in Figure 2, it holds that \( \delta_{p(T_1, T_2)} = 0 \) (because \( H_{p(T_1)} = \{(aba, 2)\} \)) but \( T_1 \neq T_2 \).

![Figure 2](image-url)
In Figure 2, $T_1$ is a caterpillar but $T_2$ is not. If both trees are caterpillars, then the following theorem holds.

**Theorem 3.5** For caterpillars, $\delta_r$ is always a metric.

**Proof.** By the definition, it is sufficient to show that $\delta_r(C_1, C_2) = 0$ iff $C_1 \equiv C_2$ for caterpillars $C_1$ and $C_2$. In other words, it is sufficient to show that we can transform $C$ from $\mathcal{H}_r(C)$ uniquely.

Let $C$ be a caterpillar such that $r = r(C)$. Suppose that the backbone $bb(C)$ of $C$ consists of vertices $v_1, \ldots, v_k$ such that $v_1 = r$. We denote a string $l(v_1) \cdots l(v_k)$ representing $bb(C)$ by $s(C)$. Then, for every leaf $v \in lv(C)$, $s_r(v)$ is of the form $l(v_1) \cdots l(v_k)$ such that $v = v_k$ and $l(v_1) \cdots l(v_{k-1})$ is a prefix of $s(C)$. Hence, consider the following procedure:

First, select the longest string $s = l(v_1) \cdots l(v_n)$ in $\mathcal{H}_r(C)$ and set $v_1, \ldots, v_{n-1}$ to a backbone. Next, for every $(s, f(s, C)) \in \mathcal{H}_r(C)$ such that $s = l(v_1) \cdots l(v_k)$ $(1 \leq k \leq n)$, set $v_k$ to $f(s, C)$ children of $v_{k-1}$ in $bb(C)$.

Since every caterpillar has just one backbone, the above procedure constructs a caterpillar $C$ from $\mathcal{H}_r(C)$ uniquely.

**Theorem 3.6** We can compute $\delta_r(C_1, C_2)$ in $O(|C_1| + |C_2|)$ time.

**Proof.** Let $C$ be a caterpillar such that $r = r(C)$ and suppose that $bb(C)$ consists of vertices $r = v_1, \ldots, v_n$. Then, repeat the following procedure from $i=1$ to $n$:

For every leaf $v$ which is a child of $v_i$, store $s = l(v_1) \cdots l(v_i)l(v)$ and the number of leaves with the label $l(v)$ as the children of $v_i$ as $f(s, C)$.

Since $V(C) = V(bb(C)) \cup lv(C)$ and $V(bb(C)) \cap lv(C) = \emptyset$, the repetition traverses every vertex in $C$ just once, so we can compute $\mathcal{H}_r(C)$ in $O(|C|)$ time.

**Theorem 3.7** There exist caterpillars $C_1$ and $C_2$ satisfying the following statements. These statements also hold even if $\tau_{\text{ILST}}$ is replaced with $\tau_{\text{ILST}}$.

1. $\tau_{\text{ILST}}(C_1, C_2) = 1$ but $\delta_r(C_1, C_2) = O(\lambda)$, where $\lambda = \max\{|lv(C_1)|, |lv(C_2)|\}$.
2. $\delta_r(C_1, C_2) = 2$ but $\tau_{\text{ILST}}(C_1, C_2) = O(n)$, where $n = \max\{|C_1|, |C_2|\}$.

**Proof.** The caterpillars $C_1$ and $C_2$ that are isomorphic but just labels of the roots are different satisfy statement 1. On the other hand, the paths $C_1$ and $C_2$ with the same length such that all the labels of vertices in $C_1$ is $a$ and all the labels of vertices in $C_2$ are $b$ satisfy the statement 2.

Hence, $\tau_{\text{ILST}}$ or $\tau_{\text{ILST}}$ and $\delta_r$ are incomparable metrics between caterpillars.

**Example 3.8** Note that $\tau_{\text{ILST}} \leq \tau_{\text{ILST}}$ for caterpillars in general. For caterpillars $C_1$ and $C_2$ in Figure 2, it holds that $1 = \tau_{\text{ILST}}(C_1, C_2) < \tau_{\text{ILST}}(C_1, C_2) = 3$. Also, since $\mathcal{H}_r(C_1) = \{|ab, 3\}$ and $\mathcal{H}_r(C_2) = \{|ab, 2\}$, it holds that $\delta_r(C_1, C_2) = 4$.

**Theorem 3.9** There exist caterpillars $C_1$ and $C_2$ satisfying the following statements:
(1) \( \delta_{\text{CS}}(C_1, C_2) = 1 \) but \( \delta_{\text{P}}(C_1, C_2) = O(\lambda) \), where \( \lambda = \max\{|l(C_1)|, |l(C_2)|\} \).
(2) \( \delta_{\text{ILST}}(C_1, C_2) = 1 \) but \( \delta_{\text{CS}}(C_1, C_2) = O(n) \), where \( n = \max\{|C_1|, |C_2|\} \).

**Proof.** Let \( C_1 \) be a star, that is, \( bb(C_1) \) consists of a single vertex, and \( C_2 \) the caterpillar obtained by adding new vertex connected to the root of \( C_1 \). Then, since \( S(C_1) \cap S(C_2) = \emptyset \), it holds that \( \delta_{\text{P}}(C_1, C_2) = O(\lambda) \). On the other hand, it is obvious that \( \delta_{\text{CS}}(C_1, C_2) = 1 \), so the statement 1 holds. Furthermore, the same caterpillars \( C_1 \) and \( C_2 \) in the proof of Theorem 3.7.2 also satisfy the statement 2.

Hence, \( \delta_{\text{CS}} \) and \( \delta_{\text{ILST}} \) are also incomparable metrics between caterpillars.

Since \( 0 \leq \delta_{\text{P}}(C_1, C_2) \leq \lambda_1 + \lambda_2 \) and \( 0 \leq \delta_{\text{CS}}(C_1, C_2) \leq n_1 + n_2 \), where \( \lambda_i = |l(C_i)| \) and \( n_i = |T_i| \), we use the normalized versions \( \delta_{\text{P}}^* \) and \( \delta_{\text{CS}}^* \) of \( \delta_{\text{P}} \) and \( \delta_{\text{CS}} \), that is, \( \delta_{\text{P}}^*(C_1, C_2) = \delta_{\text{P}}(C_1, C_2)/(\lambda_1 + \lambda_2) \) and \( \delta_{\text{CS}}^*(C_1, C_2) = \delta_{\text{CS}}(C_1, C_2)/(n_1 + n_2) \). Additionally, since \( 0 \leq \tau_{\text{ILST}}(C_1, C_2) \leq n_1 + n_2 \), we use the normalized version \( \tau_{\text{ILST}}^* \) of \( \tau_{\text{ILST}} \), that is, \( \tau_{\text{ILST}}^*(T_1, T_2) = \tau_{\text{ILST}}(T_1, T_2)/(n_1 + n_2) \).

### 4. Experimental results

In this section, we give experimental results of comparing \( \delta_{\text{P}} \) and \( \delta_{\text{CS}} \) with \( \tau_{\text{ILST}} \) (and comparing \( \delta_{\text{P}}^* \) and \( \delta_{\text{CS}}^* \) with \( \tau_{\text{ILST}}^* \)) for caterpillars.

We use N-glycans and all of the glycans (we refer to all-glycans) N-glycans and all of the glycans (we refer to all-glycans) from KEGG, CSLOGS, dblp, Auction and Nasa from UW XML Repository datasets. Table 1 illustrates the number of caterpillars in the datasets, denoted by \#cat, and the number of data, denoted by \#data.

Note that, in the dataset TPC-H, the number of different caterpillars is just 8, so we will use the 8 caterpillars, referring TPC-H, in the following. Furthermore, For \( D \in \{\text{Auction, Nasa}\} \), \( D^- \) denotes the trees obtained by deleting the root for every tree in \( D \). Since one tree in \( D \) produces some trees in \( D^- \), the total number of trees in \( D^- \) is greater than that of \( D \).

We deal with caterpillars for N-glycans, all-glycans, CSLOGS, the selected 0.1% (= 5154) caterpillars from the largest one in dblp (we refer to dblp\(_{0.1}\)), SwissProt, TPC-H, Auction and Nasa. Table 2 illustrates the information of such caterpillars. Here, \((a, b, c)\) means that \( a, b \) and \( c \) are the minimum, the maximum and the average number.

In the following, we denote the number of caterpillars (\#cat), the number of vertices (\#vertex), the degree, the height, the number of leaves (\#leaves) and the number of labels (\#labels) by \( c, n, d, h, \lambda \) and \( \beta \), respectively.

Table 3 illustrates the running time to compute \( \delta_{\text{P}} \), \( \delta_{\text{CS}} \), \( \tau_{\text{ILST}} \) and \( \tau_{\text{TAI}} \) for all the pairs in caterpillars in Table 2. Here, we assume that a cost function in \( \tau_{\text{ILST}} \) and \( \tau_{\text{TAI}} \) is a unit cost

| dataset     | \#vertex (n) | degree (d) | height (h) | \#leaves (\lambda) | \#labels (\beta) |
|-------------|--------------|------------|------------|---------------------|------------------|
| N-glycans   | (6,15);6,40  | (1,3);1,84 | (1,9);4,22 | (1,7);2,18          | (2,8);4,50       |
| all-glycans | (1,24);5,74  | (0,5);1,49 | (0,15);3,02| (1,14);1,72         | (1,9);2,84       |
| CSLOGS      | (2,404);5,84 | (1,403);3,05| (1,70);2,20| (1,403);3,64        | (2,168);3,18     |
| dblp\(_{0.1}\)| (30,751);41,74| (28,750);40,73| (1,12);1,01| (28,750);40,73      | (6,13);10,61     |
| SwissProt   | (17,89);35,10| (12,59);24,96| (2,2);2    | (15,87);33,10       | (10,23);16,79    |
| TPC-H       | (4,7);8,62   | (3,16);7,63 | (1,11);1   | (3,16);7,63         | (4,17);8,63      |
| Auction     | (1,15);4,29  | (0,12);3   | (0,2);0,71 | (1,13);3,57         | (1,15);4,29      |
| Nasa        | (1,37);2,50  | (0,34);1,13| (0,4);0,86 | (1,35);1,64         | (1,5);2,21       |
function. Also we refer the result in Muraka, Yoshino, and Hirata (2019) to the running time of \(\tau_{\text{TAI}}\). The computer environment is that CPU is Intel Xeon E51650 v3 (3.50 GHz), RAM is 1 GB and OS is Ubuntu Linux 14.04 (64bit).

Table 3 shows that, except TPC-H\(^{\circ}\), it holds that \(\delta_p<\delta_c<\tau_{\text{ILST}}<\tau_{\text{TAI}}\) for running time. The running time of \(\delta_p\) and \(\delta_c\) is large if both \(n\) and \(c\) are large. On the other hand, the running time of \(\tau_{\text{ILST}}\) (and then \(\tau_{\text{TAI}}\)) is large if both \(n\) and \(d\) (or \(\lambda\)), rather than \(c\), are large.

For the ratios of \(\tau_{\text{ILST}}/\delta_p\) and \(\tau_{\text{ILST}}/\delta_c\), \(\tau_{\text{ILST}}/\delta_p\) is smaller than 10 and \(\tau_{\text{ILST}}/\delta_c\) is smaller than 7 for N-glycans, all-glycans, CSLOGS, Auction\(^{-}\) and Nasa\(^{-}\). On the other hand, the ratios are extremely larger for dblp\(_{0.1}\) and SwissProt than others. The reason is that the caterpillars in dblp\(_{0.1}\) and SwissProt have much larger values of \(n\), \(d\), \(\lambda\) and a smaller value of \(h\) than other datasets.

Next, we discuss the distributions and scatters for the normalized distances of \(\delta_{p^*}\), \(\delta_{c^*}\) and \(\tau_{\text{ILST}^*}\).

Figure 3 illustrates the distributions of the normalized distances \(\delta_{p^*}\) (long dashed line), \(\delta_{c^*}\) (solid line) and \(\tau_{\text{ILST}^*}\) (short dashed line) for pairs of caterpillars in N-glycans, all-glycans, CSLOGS, dblp\(_{0.1}\), SwissProt and Nasa\(^{-}\). Here, we omit the distributions of TPC-H\(^{\circ}\) and Auction\(^{-}\), because the value of \(\delta_p\) and \(\delta_{c^*}\) in TPC-H\(^{\circ}\) has just one value of 1 and those in Auction\(^{-}\) has just two values of 0 and 1.

Figure 3 shows that (1) the distribution of \(\delta_{p^*}\) is larger than that of \(\delta_{c^*}\), for N-glycans and all-glycans, (2) the distributions of all the distances concentrate near to 1 for CSLOGS and dblp\(_{0.1}\) and (3) the distributions of \(\delta_{p^*}\), \(\delta_{c^*}\) and \(\tau_{\text{ILST}^*}\) are independent.

Figure 4 and 5 illustrate the scatter charts of the normalized distances \(\delta_{p^*}\), \(\delta_{c^*}\) and \(\tau_{\text{ILST}^*}\) for N-glycans, all-glycans and CSLOGS, and those for dblp\(_{0.1}\), SwissProt and Nasa\(^{-}\).

Here, \(d_1/d_2\) (\(d_1\), \(d_2\) \(\in\) \{\(\delta_{p^*}\), \(\delta_{c^*}\), \(\tau_{\text{ILST}^*}\)\}) denotes the scatter charts between the number of pairs of caterpillars with the value of the distance \(d_2\) pointed at the x-axis and that with the value of the distance \(d_1\) pointed at the y-axis. Also \(cc\) denotes the correlation coefficients between \(d_1\) and \(d_2\).

Figures 4 and 5 show that the plots concentrate at \(x=1\) for the cases of \(\tau_{\text{ILST}^*/\delta_{p^*}}\) and \(\tau_{\text{ILST}^*/\delta_{c^*}}\) in N-glycans, all-glycans, CSLOGS and Nasa\(^{-}\) and for the case of \(\tau_{\text{ILST}^*/\delta_{p^*}}\) in dblp\(_{0.1}\) and the plots concentrate at \(y=1\) for the cases of \(\delta_{p^*/\delta_{c^*}}\) in N-glycans, all-glycans, CSLOGS, dblp\(_{0.1}\) and Nasa\(^{-}\). Just in SwissProt, the plots concentrate no axis.

Also, the correlation coefficient of \(\tau_{\text{ILST}^*/\delta_{c^*}}\) has the highest value in N-glycans, all-glycans and dblp\(_{0.1}\), whereas the correlation coefficient of \(\delta_{p^*/\delta_{c^*}}\) has the highest value in CSLOGS, SwissProt and Nasa\(^{-}\).

Furthermore, consider the correlation coefficients in Figures 4 and 5.
The correlation coefficient of $d_{P^*/d_{CS^*}}$ in N-glycans and those of all the cases in all-glycans are small, that is, less than 0.5. Also the correlation coefficient of neither $t_{ILST^*/d_{P^*}}$ nor $t_{ILST^*/d_{CS^*}}$ in N-glycans is large, that is, less than 0.6. Then, no pairs of distances are interrelated.

Figure 3. The distribution of $d_{P^*}$, $d_{CS^*}$ and $t_{ILST^*}$.

(1) The correlation coefficient of $\delta_{P^*}/\delta_{CS^*}$ in N-glycans and those of all the cases in all-glycans are small, that is, less than 0.5. Also the correlation coefficient of neither $\tau_{ILST^*/\delta_{P^*}}$ nor $\tau_{ILST^*/\delta_{CS^*}}$ in N-glycans is large, that is, less than 0.6. Then, no pairs of distances are interrelated.
The correlation coefficient of $\tau_{ILST}^{*}/\delta_P^{*}$ in CSLOGS is much larger than those of $t_{ILST}^{*}/d_P^{*}$ and $t_{ILST}^{*}/d_{CS}^{*}$, where the former is greater than 0.7 but the latter is less than 0.3. Also, the correlation coefficient of $t_{ILST}^{*}/d_P^{*}$ in dblp0.1 is much larger than those of $t_{ILST}^{*}/d_P^{*}$ and $\delta_{ILST}^{*}/\delta_{CS}^{*}$, where the former is greater than 0.9 but the latter is less than 0.6. Then, the distances of $\delta_P^{*}$ and $\delta_{CS}^{*}$ in CSLOGS and those of $\tau_{ILST}^{*}$ and $\delta_{CS}^{*}$ in dblp0.1 are interrelated.

Figure 4. The scatter charts of $d_P^{*}$, $d_{CS}^{*}$ and $\tau_{ILST}^{*}$ for N-glycan, all-glycan and CSLOGS.
(3) All of the correlation coefficients in SwissProt are large, that is, greater than 0.95. Also, all of the correlation coefficients in Nasa are also large, that is, greater than 0.85. Then, all the distances of $d_{P^*}, d_{CS^*}$ and $t_{ILST^*}$ are highly interrelated.

5. Conclusion and future works

In this paper, we have introduced the path histogram distance and shown that it is a
metric, linear-time computable and an incomparable metric with the edit distance and
its variations. Also, by introducing the complete subtree histogram distance as another metrics, we have given experimental results for caterpillars in real data to compare running time, distributions and scatter charts between the path histogram distance $\delta_p$, the complete subtree histogram distance $\delta_{CS}$ and the isolated-subtree distance $\tau_{ILST}$.

As a result, whereas $\delta_p$ and $\tau_{ILST}$ are incomparable in Theorem 3.7 and $\delta_p$ and $\delta_{CS}$ are incomparable in Theorem 3.9, the experimental results show there exist cases such that their distances are interrelated.

In this paper, we have just referred the result of computing $\tau_{TAI}$ for caterpillars (Muraka et al., 2019) in Table 3. Then, it is a future work to compare $\delta_p$ and $\delta_{CS}$ with $\tau_{TAI}$, instead of $\tau_{ILST}$, in more detail. Also it is an important future work to compare $\delta_p$ and $\delta_{CS}$ with the vertical and horizontal distances (Muraka et al., 2019) to approximate $\tau_{TAI}$ for caterpillars.

As another metrics between trees, Kawaguchi, Yoshino, and Hirata (2018) have introduced an earth mover’s distance (Rubner, Tomasi, & Guibas, 2007) for trees. Since their formulation is based on the complete subtree histogram, it is a future work to formulate the earth mover’s distance for caterpillars based on the path histogram if possible, and compare it with the path histogram distance.

Notes

1. Kyoto Encyclopedia of Genes and Genomes, http://www.kegg.jp/.
2. http://www.cs.rpi.edu/zaki/www-new/pmwiki.php/Software/Software.
3. http://dblp.uni-trier.de/.
4. http://aiweb.cs.washington.edu/research/projects/xmltk/xmldata/www/repository.html.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is partially supported by Grant-in-Aid for Scientific Research (KAKENHI) [grant numbers 17H00762, 16H02870 and 16H01743] from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

Akutsu, T., Fukagawa, D., Halldórsson, M. M., Takasu, A., & Tanaka, K. (2013). Approximation and parameterized algorithms for common subtrees and edit distance between unordered trees. Theoretical Computer Science, 470, 10–22.

Aratsu, T., Hirata, K., & Kuboyama, K. (2008). Sibling distance for rooted labeled trees. JSAI PAKDD’08 post-workshop proceeding, LNAI (Vol. 5433, pp. 99–110), Springer.

Gallian, J. A. (2007). A dynamic survey of graph labeling. The Electronic Journal of Combinatorics, 14, DS6.

Hirata, K., Yamamoto, Y., & Kuboyama, T. (2011). Improved MAX SNP-hard results for finding an edit distance between unordered trees. Proceeding CPM’11, LNCS (Vol. 6661, pp. 402–415) Springer.

Kailing, K., Kriegel, H-P., Schönauer, S., & Seidl, T. (2004). Efficient similarity search for hierarchical data in large databases. Proceeding EDBT’04 (pp. 676–693) : LNCS (Vol. 2992, pp.676–693), Springer.
Kawaguchi, T., Yoshino, T., & Hirata, K. (2018). Earth mover’s distances between rooted labeled unordered trees formulated from complete subtrees. Proceeding ICPRAM’18 (revised selected papers), LNCS (Vol. 11351, pp. 65–88) Springer.

Kuboyama, T. (2007). Matching and learning in trees (PhD thesis). University of Tokyo.

Li, F., Wang, H., Li, J., & Gao, H. (2013). A survey on tree edit distance lower bound estimation techniques for similarity join on XML data. SIGMOD Record, 43, 29–39.

Muraka, K., Yoshino, T., & Hirata, K. (2018). Computing edit distance between rooted labeled caterpillars. Proceeding FedCSIS’18 (pp. 245–252).

Muraka, K., Yoshino, T., & Hirata, K. (2019). Vertical and horizontal distances to approximate edit distance for rooted labeled caterpillars. Proceeding ICPRAM’19 (pp. 590–597): SCITEPRESS – Science and Technology Publications, Lda.

Rubner, Y., Tomasi, C., & Guibas, L. J. (2007). The earth Mover’s distance as a metric for image retrieval. International Journal of Computer Vision, 40, 99–121.

Tai, K.-C. (1979). The tree-to-tree correction problem. Journal of the ACM, 26, 422–433.

Yamamoto, Y., Hirata, K., & Kuboyama, T. (2013). Tractable and intractable variations of unordered tree edit distance. International Journal of Foundations of Computer Science, 25, 307–329.

Yoshino, T., & Hirata, K. (2017). Tai mapping hierarchy for rooted labeled trees through common subforest. Theory of Computing Systems, 60, 759–783.

Zhang, K. (1996). A constrained edit distance between unordered labeled trees. Algorithmica, 15, 205–222.

Zhang, K., & Jiang, T. (1994). Some MAX SNP-hard results concerning unordered labeled trees. Information Processing Letters, 49, 249–254.