情報処理学会論文誌 Vol.59 No.1 1–8 (Jan. 2018)

根付きラベル付き木の編集距離を近似する先祖パス子孫ラベルヒストグラム距離

阿部 大祐[1,†1,a)] 平田 耕一[1,†1,b)]

受付日 2024年2月9日, 採録日 2024年2月26日

概要：根付きラベル付き無順序木 (以後, 単に木) を比較する距離として, 本論文では, 任意の頂点 v に対する, 根から v の親へのパス, v のラベル, v の子孫のラベルの集合からなる三つ組のヒストグラム L_1 距離である先祖パス子孫ラベル (Ancestral-Path Descendant-Labels, 以後 APDL) ヒストグラム距離を導入する. そして, APDL 距離が木の頂点数の線形時間で計算でき, かつ, メトリックになることを示す. また, APDL ヒストグラムの部分ヒストグラムを利用したヒストグラム距離を導入する. さらに, 実データにおける APDL ヒストグラム距離とその部分ヒストグラム距離, および, 編集距離の変種である孤立部分木距離, LCA 保存距離, トップダウン距離の分布を比較する.

キーワード：グラフアルゴリズム, グラフ理論, データマイニング

Ancestral-Path Descendant-Labels Histogram Distance for Rooted Labeled Trees

Daisuke Abe[1,†1,a)] Kouichi Hirata[1,†1,b)]

Received: February 9, 2024, Accepted: February 26, 2024

Abstract: In order to compare rooted labeled trees (trees, for short), in this paper, we introduce an ancestral-path descendant-labels (APDL, for short) histogram distance as the L_1 distance between histograms of triples consisting of the path from the root to v, the label of v and the multiset of labels in the descendant of v for every vertex v. Then, we show that the APDL histogram distance can be computed in linear time with respect to the number of vertices in trees and it is a metric. Also we introduce the histogram distances based on the parts of the APDL histogram. Furthermore, we compare their distances with the isolated-subtree distance, the LCA-preserving distance and the top-down distance as variations of the tree edit distance for real data.

Keywords: Graph Algorithm, Graph theory, Data mining

1. はじめに

Web マイニングに対する HTML や XML データなどの半構造データ, またバイオインフォマティクスにおける RNA や糖鎖データといった木構造データの比較はデータマイニングにおける重要なタスクの一つである. これらの木構造データは**根付きラベル付き無順序木** (rooted labeled unordered tree, 以後単に木という) として表され, この木を比較するためには木同士の類似度や非類似度を計算することが必要である. 木の非類似度を表す距離として最も有名なものの一つに**編集距離** (edit distance)[9] がある.

編集距離は, 頂点の削除, 挿入, 置換からなる 3 つの編集操作によって, 一方の木から他方の木へ変換する際に必要な編集操作の最小コストとして定義される. この編集距離の計算は MAX SNP 困難となり [13], 現実的に計算することができない.

[1] 情報処理学会
IPSJ, Chiyoda, Tokyo 101–0062, Japan
[†1] 現在，九州工業大学
Presently with Kyushu Institute of Technology
[a)] abe.daisuke456@mail.kyutech.jp
[b)] hirata@ai.kyutech.ac.jp

その困難性を回避するために, 孤立部分木距離 [11], LCA 保存距離 [10,14], トップダウン距離 [3,8,12] などの編集距離の変種が提案されているが, これらの計算には頂点数 n と最小次数 d に対して $O(n^2d)$ 時間かかる [11,14]. 一方, 編集距離の定数下限を持つ**ヒストグラム距離** [1,2,4,5,6,7] は, その多くが線形時間で計算可能であるが, メトリックにならないことが多い.

そこで本論文では, 編集距離よりも高速で計算することができ, かつ, メトリックになる距離として, 新たに**先祖パス子孫ラベル** (ancestal-path descendent-labels, 以下, APDL と略す) **ヒストグラム距離**を導入する. そして, 実データを用いて APDL ヒストグラム距離を含むさまざまなヒストグラム距離, および, 編集距離の変種を計算し, それらの間の関係や実計算時間を考察する.

本論文の構成は以下の通りである. まず 2 節では, 本論文で使用する木と編集距離とその変種について説明し, APDL ヒストグラム距離をはじめとするいくつかのヒストグラム距離を導入する. 3 節では, APDL ヒストグラム距離の性質について議論する, 特に, 木から APDL ヒストグラム距離を線形時間で構築するアルゴリズム, および, APDL ヒストグラムから木を一意に構成するアルゴリズムを設計する. 4 節では, 作成したプログラムを実データに適用し, 実行時間や孤立部分木距離, 他のアルゴリズム距離などとの関係について考察する. 最後に 5 節では, まとめと今後の課題について述べる.

2. 準備

本節では, 木と編集距離とその変種, および, APDL ヒストグラム距離を含むいくつかのヒストグラム距離について導入する.

2.1 根付きラベル付き無順序木

サイクルを持たない連結グラフを**木** (tree) という. 木 $T = (V, E)$ に対して, $V(T) = V$, $E(T) = E$ と表し, $v \in V(T)$ を単に $v \in T$ で表す. また, $|V(T)|$ を $|T|$ と表す.

根付き木 (rooted tree) とは木 T から任意の頂点 r を根とした木のことであり, このとき, $r = r(T)$ と表す. また, 根付き木 T の各頂点 v に対して, v から $r(T)$ への一意に決まるパスを $P_T[v]$ とする. すなわち, $P_T[v] = (\{v_1, \ldots, v_k\}, \{(v_i, v_{i+1}) | 1 \leq i \leq k-1\})$, ただし, $v_1 = r(T)$ であり, $v_k = v$ である. このとき, $P_T[v]$ を, パスをなす頂点のラベル列 $l(v_1) \cdots l(v_k)$ とみなすことができる.

任意の $v \in T$ に対して, $P_T[v]$ 上の v に隣接する頂点を v の**親** (parent) といい, $par(v)$ と表す. また, $P_T[v] \setminus \{v\}$ 上の頂点を v の**先祖** (ancestor) という. ここで, $P_T[r] = (\{r\}, \emptyset)$ とする.

頂点 u が頂点 v の先祖であるとき, $u < v$ と表し, $u < v$ もしくは $u = v$ のとき $u \leq v$ と表す. また, 頂点 v が頂点 u の親であるとき, u は v の**子** (child) といい, v が u の祖先であれば u は v の**子孫** (descendant) であるという. 頂点 v のすべての子の集合を $ch(v)$ と表す.

任意の $v \in T$ に対して, $h(v) = |P_T[v]|$ を v の**高さ** (height) といい, $h(T) = \max\{h(v) \mid v \in T\}$ を T の**高さ** (height) という. また, $d(v) = |ch(v)|$ を v の**次数** (degree) といい, $d(T) = \max\{d(v) \mid v \in T\}$ を T の**次数** (degree) という.

同じ親を持つ二つの頂点を**兄弟** (sibiling) という. 子を持たない頂点を**葉** (leaf) という. 葉でない頂点を**内部頂点** (internal vertex) という. また, $u, v \in T$ に対して, $u \leq w$ かつ $v \leq w$ であり, $u \leq w'$, $v \leq w'$ かつ $w' \leq w$ となる w' が存在しないとき, w を u と v の**最近共通先祖** (least common ancestor, LCA) といい, $u \sqcup v$ と表す.

根付き木 $T = (V, E)$ と頂点 $v \in T$ に対して, $r(S') = v, V' = \{w \in V \mid w \leq v\}, E' = \{(u, w) \in E \mid u, w \in V'\}$ となるような根付き木 $S' = (V', E')$ を v を根とする T の**完全部分木** (complete subtree) といい, $T[v]$ と表す. また, $T[v]$ から v を削除した森を $T(v)$ と表す.

根付き木のうち, 兄弟間の左から右への順序が与えられている木を**順序木** (ordered tree) といい, そうでない木を**無順序木** (unordered tree) という. 有限アルファベット Σ の文字が木 T の各頂点にラベルを割り当てられている木を**ラベル付き木** (labeled tree) といい, 頂点 v に割り当てられているラベルを $l(v)$ として表す. 本論文では, 根付きラベル付き無順序木を対象として扱うため, これを単に木という. なお, 実際に計算する場合には, 根付きラベル付き無順序木を, ある兄弟順序に基づいた根付きラベル付き順序木として扱う.

根付きラベル付き順序木 T に対して, T の頂点 v とその子である $v_1, ..., v_n$ に対して, v を走査した後で $T[v_k](1 \leq k \leq i)$ を順に再帰的に走査することによって得ることができる頂点列を $T[v]$ の**先行順走査** (preorder traversal) という. 同様に, $T[v_k](1 \leq k \leq i)$ を順に走査した後, v を走査することによって得ることができる頂点列を $T[v]$ の**後行順走査** (postorder traversal) という. ここで, 左から右の兄弟順序の下での先行順走査の反転は, 右から左の兄弟順序の下での後行順走査となる.

2 つの頂点の集合を比較するために, 本論文ではラベルの重集合を用いる. Σ 上の**重集合** (multiset) S は Σ から自然数 N への写像 $S : \Sigma \to N$ として定義される. 重集合 S の**要素数** (cardinality) を $|S|$ と表し, $\sum_{a \in \Sigma} S(a)$ と定義する. 通常の集合と同じように, すべての $a \in \Sigma$ に対して $S(a) = 0$ となる**空の重集合** (empty multiset) を $\emptyset$ と表す. Σ 上の 2 つの重集合 S_1 と S_2 に対して, すべての $a \in \Sigma$ に対して $S1(a) \leq S2(a)$ となるとき, S_1 は S_2 の**部分重集合** (sub-multiset) であるといい, $S_1 \subseteq S_2$ と表す. また, す

べての $a \in \Sigma$ に対して, S_1 と S_2 の**和** (sum) $S_1 \oplus S_2$ を $(S_1 \oplus S_2)(a) = S_1(a) + S_2(a)$ を満たす重集合とし, **差** (difference) $S_1 \ominus S_2$ を $(S_1 \ominus S_2)(a) = \max\{S_1(a) - S_2(a), 0\}$ を満たす重集合と定義する.

また, 本論文では Σ 上の文字列, すなわち, $s \in \Sigma^*$ を利用する. ここで, $\varepsilon \in \Sigma^*$ は空語である. $s \in \Sigma^*$ に対して, $|s|$ は s の長さ, $s[i](1 \leq i \leq |s|)$ は s の i 番目の文字, $s[i,j](i \leq i \leq j \leq |s|)$ は s の部分文字列 $s[i] \cdots s[j]$ を表す. また, $s \in \Sigma^*$ と $a \in \Sigma$ に対して, $s \cdot a$ で s に a を連接して得られる文字列を表す.

2.2 編集距離とその変種

木を比較する距離である編集距離は, 以下の編集操作によって定式化される.

定義 1. (編集操作 [9]). 以下の 3 つの操作を, 木 T の**編集操作** (edit operation) という.

(1) **置換** (substitution): 頂点 v のラベルを別のラベルに変更する.
(2) **削除** (deletion): 木 T が持つ, v' を親に持つ頂点 $v \in T$ を削除し, v の子を新たに v' の子にする.
(3) **挿入** (insertion): 削除の逆にあたる操作であり, 頂点 v を $v' \in T$ の子として挿入し, v' の子の部分集合の親を v に変える.

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ とする. ここで, ラベルの組 $(l_1, l_2) \in (\Sigma_\varepsilon \times \Sigma_\varepsilon) \setminus \{(\varepsilon, \varepsilon)\}$ に対して, 編集操作を $(l_1 \mapsto l_2)$ と表す. $l_1 \neq \varepsilon$ かつ $l_2 \neq \varepsilon$ のときは置換, $l_2 = \varepsilon$ のときは削除, $l_1 = \varepsilon$ のときは挿入をそれぞれ表す. このとき, 2 つのラベル間の**コスト関数** (cost function) を $\gamma : (\Sigma_\varepsilon \times \Sigma_\varepsilon - \{(\varepsilon, \varepsilon)\}) \to \mathbf{R}^+$ と定義する. 特に, $\gamma(l_1, l_2) = 0 \Leftrightarrow l_1 = l_2$, $\gamma(l_1, l_2) = 1 \Leftrightarrow l_1 \neq l_2$ が成り立つコスト関数を**単一コスト関数** (unit cost function) という.

定義 2. (編集距離 [9]). $e = (l_1 \mapsto l_2)$ となる編集操作 e のコストをコスト関数 γ を用いて, $\gamma(e) = \gamma(l_1, l_2)$ と表す. 編集操作列 $E = e_1, e_2, ..., e_n$ のコストを $\gamma(E) = \sum_{i=1}^{n} \gamma(e_i)$ とするとき, 木 T, S の間の**編集距離** $\tau_{\text{TAI}}(T, S)$ を以下のように定義する.

$$\tau_{\text{TAI}}(T, S) = \min\{\gamma(E) \mid \text{E は T から S を得るための編集操作列 }\}.$$

次に, 編集距離と関係が深い Tai マッピングを導入する.

定義 3. (Tai マッピング [9]). T_1, T_2 を木とし, $M \subseteq V(T_1) \times V(T_2)$ とする. このとき, 任意の $(u_1, v_1), (u_2, v_2) \in M$ が以下の条件をすべて満たすとき, 3 つ組 (M, T_1, T_2) を**無順序 Tai マッピング** (unordered Tai mapping) という.

(1) $u_1 = u_2 \Leftrightarrow v_1 = v_2$ (一対一対応)
(2) $u_1 \leq u_2 \Leftrightarrow v_1 \leq v_2$ (先祖関係保存)

以後, (M, T_1, T_2) を単にマッピングといい, $M \in \mathcal{M}_{\text{TAI}}(T_1, T_2)$ と表す.

定理 1. 以下が成立する.

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

次に, マッピングを制限することによって得られるマッピングの変種と, それらの最小コストとして定式化される編集距離の変種を導入する.

定義 4. T_1, T_2 を木とし, $M \in \mathcal{M}_{\text{TAI}}(T_1, T_2)$ を T_1 と T_2 のマッピングとする. また, $M \setminus \{(r(T_1), r(T_2))\}$ を M^- と表す.

(1) 以下の条件を満たす M を**孤立部分木マッピング**(isolated-subtree mapping)[11]といい, $M \in \mathcal{M}_{\text{ILST}}(T_1, T_2)$ と表す.

$$\forall (u_1, v_1), (u_2, v_2), (u_3, v_3) \in M (u_3 < u_1 \sqcup u_2 \Leftrightarrow v_3 < v_1 \sqcup v_2).$$

このとき, **孤立部分木距離** (isolated-subtree distance) $\tau_{\text{ILST}}(T_1, T_2)$ を以下のように定義する.

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

(2) 以下の条件を満たす M を **LCA 保存マッピング** (LCA-preserving mapping) または**次数 2 マッピング** (degree-2 mapping)[10,14] といい, $M \in \mathcal{M}_{\text{LCA}}(T_1, T_2)$ と表す.

$$\forall (u_1, v_1), (u_2, v_2) \in M^- ((u_1 \sqcup u_2, v_1 \sqcup v_2) \in M).$$

このとき, **LCA 保存距離** (LCA-preserving distance) $\tau_{\text{LCA}}(T_1, T_2)$ を以下のように定義する.

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

(3) 以下の条件を満たす M を**トップダウンマッピング** (top-down mapping)[3,8,12] といい, $M \in \mathcal{M}_{\text{TOP}}(T_1, T_2)$ と表す.

$$\forall (u, v) \in M^- ((par(u), par(v)) \in M).$$

このとき, **トップダウン距離** (top-down distance) $\tau_{\text{TOP}}(T_1, T_2)$ を以下のように定義する.

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

定理 2. T_1, T_2 を木とする. このとき, 以下が成り立つ.

(1) $\tau_{\text{TAI}}(T_1, T_2)$ を計算する問題は MAX SNP 困難である [13].

(2) $\tau_{\mathrm{LIST}}(T_1,T_2)$, $\tau_{\mathrm{LCA}}(T_1,T_2)$, $\tau_{\mathrm{TOP}}(T_1,T_2)$ は $O(n^2d)$ 時間で計算可能である, ここで, $n=\max\{|T_1|,|T_2|\}$, $d=\min\{d(T_1),d(T_2)\}$ である [11,14].

(3) $\tau_{\mathrm{TAI}}(T_1,T_2) \le \tau_{\mathrm{LIST}}(T_1,T_2) \le \tau_{\mathrm{LCA}}(T_1,T_2) \le \tau_{\mathrm{TOP}}(T_1,T_2)$ である [10,11,14].

2.3 ヒストグラム距離

2 つの木 T_1 と T_2 に対する**ヒストグラム距離** (histogram distance) $\delta(T_1,T_2)$ は, パターン Pat に基づく T_1 と T_2 の 2 つのヒストグラム間の L_1 距離として定式化される. パターン Pat に関する頂点 $v\in T$ が存在するとき, Pat は木 T に**出現する** (occur) という. 木 T 中に出現するパターン Pat の**頻度** (frequency) を $f(Pat,T)$ と表す. そして, T 中のパター ン Pat の**ヒストグラム** (histogram) $\mathcal{H}(Pat,T)$ はパターン Pat と頻度 $f(Pat,T)$ の組からなる $(Pat,f(Pat,T))$ として構成される.

各 $v\in T$ に対して, v のラベル $l(v)$, v の兄弟の重集合 $\widetilde{ch(v)}$, 根となる頂点 $v_1=r(T)$ から $v=v_k$ へのパス $P_T(v)=[v_1,...,v_k]$, v を根とする完全部分木 $T[v]$, および $T(v)$ に出現するラベルの重集合 $\widetilde{T(v)}$ をパターン Pat の構成要素として利用する. ラベルのアルファベット Σ に対して $P_T(v)$ を $l(v_1)\cdots l(v_k)\in\Sigma^*$ の文字列として表し, $\widetilde{ch(v)}$ と $\widetilde{T(v)}$ を $1\le i\le n$ に対して $ch(v)(a_i)=k_i$, $\widetilde{T(v)}(a_i)=k_i$ となるときに文字列 $a_1^{k_1}\cdots a_n^{k_n}\in\Sigma^*$ として表す. 特に, v が T の葉のとき, $ch(v)$ と $T(v)$ を ε で表す.

定義 5. T を木とし, $v\in T$ とする. このとき, $L(v)=\langle l(v)\rangle$, $S(v)=\langle l(v),\widetilde{ch(v)}\rangle$, $AP(v)=\langle P_T(v),l(v)\rangle$, $DL(v)=\langle l(v),\widetilde{T(v)}\rangle$, $CS(v)=\langle l(v),T[v]\rangle$, $APS(v)=\langle P_T(v),l(v),\widetilde{ch(v)}\rangle$, $APDL(v)=\langle P_T(v),l(v),\widetilde{T(v)}\rangle$ を, それぞれ, T における v の**ラベル (L) パターン** (label (L) pattern), **兄弟 (S) パターン** (sibling (S) pattern), **祖先パス** (AP) **パターン** (ancestral-path (AP) pattern), **子孫ラベル** (DL) **パターン** (descendant-labels (DL) pattern), **完全部分木** (CS) **パターン** (complete subtree (CS) pattern), **祖先パス兄弟** (APS) **パターン** (ancestral-path sibiling (APS) pattern), **祖先パス子孫ラベル** ($APDL$) **パターン** (ancestral-path descendant-labels (APDL) pattern) という.

定義 6. T, T_1 ,T_2 を木とし,
$Pat\in\{L,S,AP,DL,CS,APS,APDL\}$ とする. このとき, $\{Pat(v)\mid v\in T\}$ を $Pat(T)$ と表す. T における Pat のヒストグラムを Pat **ヒストグラム** (Pat histogram) といい, $\mathcal{H}_{Pat}(T)$ と表す. また, Pat **ヒストグラム距離** (Pat histogram distance)$\delta_{Pat}(T_1,T_2)$ を $\mathcal{H}_{Pat}(T_1)$ と $\mathcal{H}_{Pat}(T_2)$ の L_1 距離と定義する.

命題 1. T_1, T_2 を木とし, $n=\max\{|T_1|,|T_2|\}$, $h=\max\{h(T_1),h(T_2)\}$ とする.

(1) $\delta_L(T_1,T_2)$ は $O(n)$ 時間で計算することができ, $\delta_L(T_1,T_2)\le 2\cdot\tau_{\mathrm{TAI}}(T_1,T_2)$ となるが, δ_L はメトリックではない [5,7].

(2) $\delta_S(T_1,T_2)$ は $O(n)$ 時間で計算することができ, $\delta_S(T_1,T_2)\le 5\cdot\tau_{\mathrm{TAI}}(T_1,T_2)$ となるが, δ_L はメトリックではない [2].

(3) δ_{AP} はメトリックではない [6].

(4) $\delta_{CS}(T_1,T_2)$ は $O(n\log n)$ 時間で計算することができ, $\tau_{\mathrm{TAI}}(T_1,T_2)\le\delta_{CS}(T_1,T_2)\le(2h+2)\cdot\tau_{\mathrm{TAI}}(T_1,T_2)$ となり, δ_{CS} はメトリックである [1,4].

3. APDL ヒストグラム距離

本節では, 主に $APDL$ ヒストグラム距離の性質について議論する.

例 1. 図 1 の木 T_1 と T_2 を考える. このとき, T_1 と T_2 の $APDL$ ヒストグラム $\mathcal{H}_{APDL}(T_1)$ と $\mathcal{H}_{APDL}(T_2)$ は表 1 のようになる. したがって, $\delta_{APDL}(T_1,T_2)=4$ となる. なお, $T_1\not\equiv T_2$ だが $\mathcal{H}_{APS}(T_1)=\mathcal{H}_{APS}(T_2)$ となるので, δ_{APS} はメトリックではない.

表 1 $\mathcal{H}_{APDL}(T_1)$ と $\mathcal{H}_{APDL}(T_2)$ の APDL ヒストグラム

$\mathcal{H}_{APDL}(T_1)$		$\mathcal{H}_{APDL}(T_2)$	
$(\langle\varepsilon,a,a^4b^5\rangle,1)$	$(\langle ab,a,b\rangle,1)$	$(\langle\varepsilon,a,a^4b^5\rangle,1)$	$(\langle ab,a,b\rangle,1)$
$(\langle aba,a,\varepsilon\rangle,1)$	$(\langle a,b,a^2b\rangle,1)$	$(\langle aba,a,\varepsilon\rangle,1)$	$(\langle a,b,ab\rangle,1)$
$(\langle ab,a,ab\rangle,1)$	$(\langle aba,b,\varepsilon\rangle,2)$	$(\langle ab,a,ab\rangle,1)$	$(\langle aba,b,\varepsilon\rangle,2)$
$(\langle a,b,a^2b^2\rangle,1)$	$(\langle ab,b,a\rangle,1)$	$(\langle a,b,a^3b^2\rangle,1)$	$(\langle ab,b,a\rangle,1)$
$(\langle abb,a,\varepsilon\rangle,1)$		$(\langle abb,a,\varepsilon\rangle,1)$	

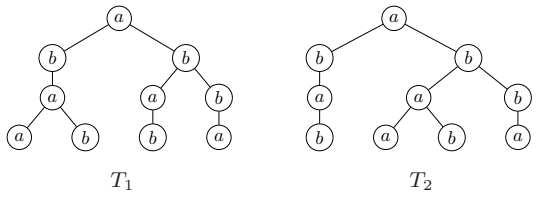


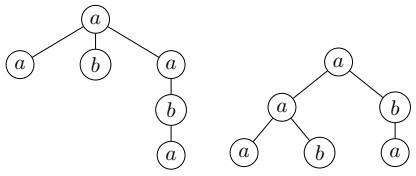
図 1 例 1 の木 T_1 と T_2

例 2. 図 2 の木 T_3 と T_4 を考える. このとき, T_1 と T_2 の $APDL$ ヒストグラム $\mathcal{H}_{APDL}(T_3)$ と $\mathcal{H}_{APDL}(T_4)$ は表 2 のようになる. したがって, $\delta_{APDL}(T_3,T_4)=8$ となる. なお, $T_3\not\equiv T_4$ だが $\mathcal{H}_{DL}(T_3)=\mathcal{H}_{DL}(T_4)$ となるので, δ_{DL} はメトリックではない.

例 1 と例 2 より, 以下の命題が成り立つ.

表 2 $\mathcal{H}_{APDL}(T_3)$ と $\mathcal{H}_{APDL}(T_4)$ の APDL ヒストグラム

$\mathcal{H}_{APDL}(T_3)$		$\mathcal{H}_{APDL}(T_4)$	
$(\langle\varepsilon, a, a^3b^2\rangle, 1)$	$(\langle aab, a, \varepsilon\rangle, 1)$	$(\langle\varepsilon, a, a^3b^2\rangle, 1)$	$(\langle aa, a, \varepsilon\rangle, 1)$
$(\langle a, a, ab\rangle, 1)$	$(\langle a, a, \varepsilon\rangle, 1)$	$(\langle a, a, ab\rangle, 1)$	$(\langle ab, a, \varepsilon\rangle, 1)$
$(\langle aa, b, a\rangle, 1)$	$(\langle a, b, \varepsilon\rangle, 1)$	$(\langle a, b, a\rangle, 1)$	$(\langle aa, b, \varepsilon\rangle, 1)$



T_3 T_4

図 2 例 2 の木 T_3 と T_4

命題 2. δ_{APS} と δ_{DL} はメトリックではない.

木 T に対して, アルゴリズム 1 の TREE2APDL は, 木 T から T の APDL ヒストグラムを構成するアルゴリズムである. このアルゴリズムは, まず 3 行目の for ループで木を後行順で走査しながら頂点 i のラベル $L[i]$ と子孫のラベルの重集合 $C[i]$ を格納し, 次に 9 行目の for ループで木を右から左の兄弟順とみなした先行順で走査しながら頂点 i の先祖パス $P[i]$ を格納している. そして, 14 行目の for ループでそれらをヒストグラムとしてまとめている.

procedure TREE2APDL(T)
/* T: 木 */
/* $[v_1, ..., v_n]$: 後行順走査における T の頂点の文字列 */
/* $C[h], S[i]$: Σ 中のラベルの重集合 $P[i]$ */
/* str : $\Sigma*$ 中の文字列 */
1 **for** $h = 1$ ***to*** $h(T)$ **do** $C[h] \leftarrow \emptyset$;
2 $h_0 \leftarrow 0$;
3 **for** $i = 1$ ***to*** n **do**
4 $h_i \leftarrow h(v_i)$; $L[i] \leftarrow l(v_i)$;
5 **if** $h_i \geq h_{i-1}$ **then** $S[i] \leftarrow \varepsilon$;
6 **else** $S[i] \leftarrow C[h_{i-1}]$; $C[h_i] \leftarrow C[h_i] \oplus C[h_{i-1}]$; $C[h_{i-1}] \leftarrow \emptyset$;
7 $C[h_i] \leftarrow C[h_i] \oplus \{l(v_i)\}$;
8 $h_{n+1} \leftarrow -1$; $str \leftarrow \varepsilon$;
9 **for** $i = n$ ***downto*** 1 **do**
10 $h_i \leftarrow h(v_i)$;
11 **if** $h_i \leq h_{i+1}$ **then** $str \leftarrow str[1, h_i]$;
12 $P[i] \leftarrow str$; $str \leftarrow str \cdot l(v_i)$;
13 $\mathcal{H} \leftarrow \emptyset$;
14 **for** $i = 1$ ***to*** n **do**
15 **if** $(\langle P[i], L[i], S[i]\rangle, k) \in \mathcal{H}$ **then** k++;
16 **else** $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\langle P[i], L[i], S[i]\rangle, 1)\}$;
17 **return** $\mathcal{H}$;

Algorithm 1: TREE2APDL.

定理 3. 木 T に対して, アルゴリズム 1 は T を 2 回走査するだけで $\mathcal{H}_{APDL}(T)$ を $O(|T|)$ 時間で計算することができる. したがって, 木 T_1 と T_2 に対して, $\delta_{APDL}(T_1, T_2)$ は $O(|T_1| + |T_2|)$ 時間で計算できる.

アルゴリズム 1 とは逆に, アルゴリズム 2 の APDL2TREE は, APDL ヒストグラムから T を構成するアルゴリズムである. ここで, *APDL* ヒストグラム $\mathcal{H}$ から得られる $\widehat{\mathcal{H}}$ は, $(\langle p, l(v), S\rangle, f) \in \mathcal{H}$ に対して $\widehat{\mathcal{H}}(\langle p, l(v), S\rangle) = f$ と定義される重集合である.

アルゴリズム 2 では, 2 行目で T の根を設定した後, 4 行目の for ループで, 深さを 1 ずつ増やして, その深さの頂点を配置しながら幅優先で木を構築している. ここで, APDL ヒストグラムの先祖パスの長さがその頂点の深さとなり, また, 7 行目でそれぞれの子孫ラベルの整合性を保つように深さごとの頂点を配置している.

procedure APDL2TREE($\mathcal{H}$)
/* $\mathcal{H} = \mathcal{H}_{APDL}(T)$ */
1 $n \leftarrow \max\{|p| \mid (\langle p, l, S\rangle, k) \in \mathcal{H}\}$;
/* $n = h(T)$ */
2 **set** r as the root of T for $(\langle\varepsilon, l(r), \widetilde{T(r)}\rangle, 1) \in \mathcal{H}$;
3 $\widehat{\mathcal{H}_0} \leftarrow \{\langle\varepsilon, l(r), \widetilde{T(r)}\rangle\}$;
4 **for** $h = 1$ ***to*** n **do**
5 $\mathcal{H}_h \leftarrow \{(\langle p, l(v), S\rangle, f) \in \mathcal{H} \mid |p| = h\}$;
6 **foreach** $\langle p_0, l(v_0), S_0\rangle \in \widehat{\mathcal{H}_{h-1}}$ **do**
7 **select** $\langle p_1, l(v_1), S_1\rangle, \ldots, \langle p_k, l(v_k), S_k\rangle \in \widehat{\mathcal{H}_h}$ such that $S_0 = (\{l(v_1)\} \oplus S_1) \oplus \cdots \oplus (\{l(v_k)\} \oplus S_k)$;
8 **set** $v_1, \ldots, v_k$ as the children of v_0 in T;
9 $\widehat{\mathcal{H}_h} \leftarrow \widehat{\mathcal{H}_h} \ominus \{\langle p_1, l(v_1), S_1\rangle, \ldots, \langle p_k, l(v_k), S_k\rangle\}$;
10 **return** T;

Algorithm 2: APDL2TREE.

定理 4. アルゴリズム 2 は APDL ヒストグラム $\mathcal{H}_{APDL}(T)$ から木 T を一意に構成することができる. したがって, APDL ヒストグラム距離 δ_{APDL} はメトリックである.

定理 5. 木 T_1 と T_2 に対して以下が成り立つ (図 5 参照).
(1) δ_{AP}, δ_S, δ_{DL} は比較不能である.
(2) δ_{APDL}, δ_{APS}, δ_{CS} は比較不能である.
(3) $\delta_L(T_1, T_2) \leq \delta_{AP}(T_1, T_2) \leq \delta_{APDL}(T_1, T_2)$ かつ $\delta_L(T_1, T_2) \leq \delta_{DL}(T_1, T_2) \leq \delta_{APDL}(T_1, T_2)$ が成り立つ.
(4) $\delta_L(T_1, T_2) \leq \delta_{AP}(T_1, T_2) \leq \delta_{APS}(T_1, T_2)$ かつ $\delta_L(T_1, T_2) \leq \delta_S(T_1, T_2) \leq \delta_{APS}(T_1, T_2)$ が成り立つ.

(5) $\delta_L(T_1,T_2) \le \delta_S(T_1,T_2) \le \delta_{CS}(T_1,T_2)$ かつ
$\delta_L(T_1,T_2) \le \delta_{DL}(T_1,T_2) \le \delta_{CS}(T_1,T_2)$ が成り立つ.

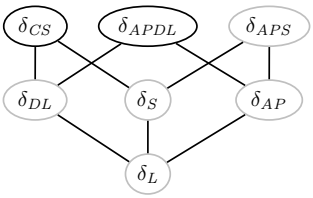


図 3 $Pat \in \{L, S, AP, DL, CS, APS, APDL\}$ に対する δ_{Pat} の階層構造. ここで, 黒線で囲まれた距離はメトリックであり灰色線で囲まれた距離はメトリックではないことを表している.

定理 6. 木 T_1 と T_2 について, 以下が成り立つ.
(1) δ_{AP} と τ_{TAI} は比較不能である.
(2) δ_{DL} と τ_{TAI} は比較不能である.
(3) $\tau_{\mathrm{TAI}}(T_1,T_2) \le \delta_{APS}(T_1,T_2)$ かつ $\tau_{\mathrm{TAI}}(T_1,T_2) \le \delta_{APDL}(T_1,T_2)$ が成り立つ.

定理 7. $n = \max\{|T_1|, |T_2|\}$ のとき, 以下の条件を満たす 2 つの木 T_1 と T_2 が存在する.
(1) $\delta_{CS}(T_1,T_2) = \tau_{\mathrm{TAI}}(T_1,T_2) = O(1)$ だが, $\delta_{APDL}(T_1,T_2) = O(n)$ となる.
(2) $\delta_{APDL}(T_1,T_2) = \tau_{\mathrm{TAI}}(T_1,T_2) = O(1)$ だが, $\delta_{CS}(T_1,T_2) = O(n)$ となる.

定理 7.1 を満たす木としては, 任意の木 T_1 と, T_1 と根のラベルだけが異なり T_1 と同型となる木 T_2 が該当する. また, 定理 7.2 を満たす木としては, 頂点数 n のパス T_1 と頂点数 $n-2$ のパスの葉に二つの葉を接続した木 T_2 が該当する. したがって, δ_{CS} は葉に近いような深い頂点の影響を受けやすく, δ_{APDL} は根に近いような浅い頂点の影響を受けやすい.

4. 実験

本節では, 3 節のアルゴリズムを実装して実データに適用することで APDL ヒストグラム距離を計算し, 編集距離の変種である孤立部分木距離と比較する, また, 実データに対する APDL ヒストグラム距離を含むヒストグラム距離と孤立部分木距離, LCA 保存距離, トップダウン距離の分布について考察する. 実行環境は, OS が Ubuntu 18.04.3, CPU が Intel Xeon E5-1650 v3(3.50GHs), RAM が 3.8GB である.

4.1 データ

まず, 実データとして使用する木構造データについて説明する. #data は木の数, n は頂点の数の平均, d は次数の平均, h は高さの平均, λ は葉数の平均, β はラベル数の平均を表す.

表 3 各データセットの詳細

dataset	#data	n	d	h	λ	β
nglycan	2142	11.0696	2.0724	5.3838	3.2876	5.4253
allglycan	10683	6.3831	1.6524	3.5982	2.1390	3.1458
cslogs	59691	12.9364	4.4879	3.4278	8.1956	11.3636
$\mathrm{dblp}_{0.1\%}$	5154	41.7581	40.7294	1.0107	40.7399	10.6202
$\mathrm{SwissProt}_{10\%}$	5000	136.9320	48.4021	2.9774	112.0690	25.6585
TCP-H	86800	17.0000	16.0000	1.0000	16.0000	17.0000
$\mathrm{Auction}^-$	259	4.2857	3.0000	0.7143	3.1429	4.2857
Nasa	2435	195.7470	21.5310	5.7614	125.5090	38.9782
$\mathrm{Protein}_{1\%}$	2625	284.9210	89.1957	5.0000	223.8960	53.7765
University	6739	22.5299	11.7544	2.3134	18.1192	18.3753

データとして, KEGG[*1]から提供された N-glycans と allglycans, CSLOGS[*2], dblp[*3], UW XML Repository[*4] から提供された SwissProt, TPC-H, Auction, Nasa, Protein, および, University を用いる. ここで, 表 3 の#data, n, d, h, λ, β をそれぞれ, 木構造データの数, 頂点の数の平均, 次数の平均, 高さの平均, 葉の数の平均, ラベルの種類の平均とする. Auction に対して $\mathrm{Auction}^-$ は, Auction の任意の木に対して根となる頂点を削除することによって得られる木の集合を表している. また, $\mathrm{dblp}_{0.1\%}$ は dblp のデータの内, 大きい方から上位 0.1%を取り出したものであり, 同様に, $\mathrm{Protein}_{1\%}$ はデータの上位 1%, $\mathrm{SwissProt}_{10\%}$, は上位 10%を取りだしたものである.

4.2 実験結果

表 3 のすべてのデータセットに対して, それぞれのデータセット内のデータで総当たりで APDL ヒストグラム距離 δ_{APDL} と孤立部分木距離 τ_{ILST} を計算した. 表 4 は, その計算時間 [ms] である.

表 4 により, すべてのデータセットにおいて APDL ヒストグラム距離 δ_{APDL} の計算は孤立部分木距離の計算時間よりも高速であるということが確認できる. ただし, cslogs, TCP-H における δ_{APDL} の計算では, 他のデータにおける δ_{APDL} の計算よりも圧倒的に計算時間がかかっている. その理由として, cslogs はデータ数が他のデータと比べて 5 倍以上, TCP-H は 8 倍以上であるため関数を呼び出す回数が多くなることで時間がかかると考えられ, 一方, $\mathrm{Protein}_{1\%}$

[*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

表 4 APDL ヒストグラム距離 δ_{APDL} と孤立部分木距離 τ_{Ilst} の計算時間 [ms]

dataset	δ_{APDL}	τ_{ILST}
nglycan	20,251	46,904
allglycan	426,127	476,261
cslogs	38,499,418	71,657,645
$\mathrm{dblp}_{0.1\%}$	310,830	18,036,096
$\mathrm{SwissProt}_{10\%}$	1,546,414	116,486,976
TPC-H	42,270,733	225,564,979
$\mathrm{Auction}^{-}$	173	186
Nasa	1,123,476	35,819,294
$\mathrm{Protein}_{1\%}$	1,527,080	191,469,185
University	391,625	2,772,599

や $\mathrm{SwissProt}_{10\%}$ の計算時間がかかる孤立部分木距離の計算では, データ数よりも木の頂点数の方が計算時間に影響していると考えられる.

図 4 は, それぞれのデータセットに対する, 縦軸を孤立部分木距離, 横軸を APDL ヒストグラム距離とした散布図である. 図 4 では, N-glycans, all-glycans, $\mathrm{dblp}_{0.1\%}$ のように広範囲に散布するデータセットと cslogs, SwissProt, $\mathrm{Protein}_{1\%}$, University のようにある程度狭い範囲にまとまるデータセットに分かれており, 後者は孤立部分木距離の近似ができていることが見て取れる.

また, 図 5 は図 4 の散布図がある程度狭い範囲にまとまっていた $\mathrm{Protein}_{1\%}$ における APDL ヒストグラム距離, APS ヒストグラム距離, AP ヒストグラム距離, DL ヒストグラム距離, L ヒストグラム距離, S ヒストグラム距離, 孤立部分木距離, LCA 保存距離, トップダウン距離の分布をヒストグラムで表したものである. ここで, 各距離の計算結果における最大値ですべての計算結果を標準化することでヒストグラムの横軸を 0 から 1 までに統一している.

図 5 より, APDL ヒストグラム距離, APS ヒストグラム距離, AP ヒストグラム距離のピークは 2 つであり, 山が大きく 2 つに分かれていること, DL ヒストグラム距離と S ヒストグラム距離はピークが 3 つあり, また, L ヒストグラム距離はピークが 2 つであるものの, 最初のピークが低くなっている. また, ヒストグラム距離は最大のピークが 0.9 から離れているのに対して, 編集距離の変種はピークが 2 つだが, 最大のピークが 0 に非常に近くなっている.

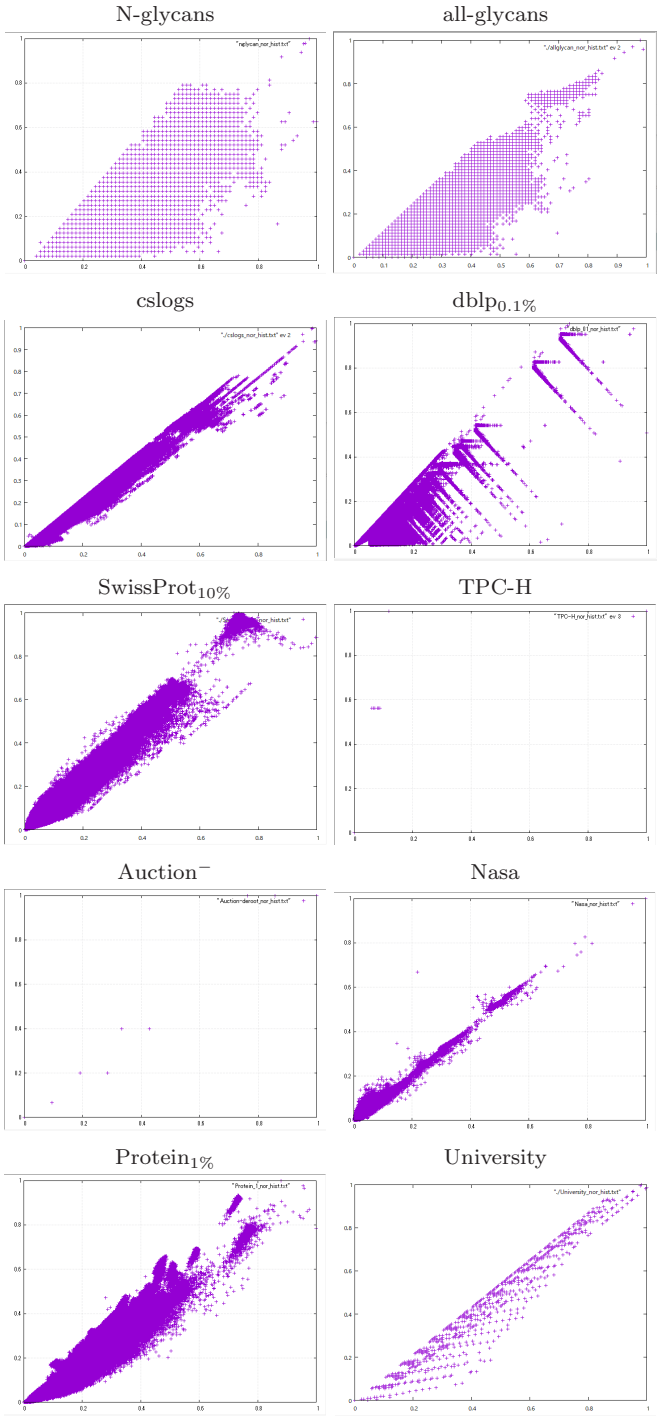


図 4 孤立部分木距離と APDL 距離のデータごとの散布図

5. まとめと今後の課題

本論文では, メトリックであり孤立部分木距離よりも高速で計算が可能である APDL ヒストグラム距離を導入し, APDL ヒストグラム距離を計算するプログラムを作成して実データを用いた計算機実験を行った. その結果として, 編集距離のもっとも一般的な変種である孤立部分木距離の計算時間を大きく下回り, cslogs, SwissProt, $\mathrm{Protein}_{1\%}$, University のようにある程度狭い範囲にまとまるデータセットにおいては孤立部分木距離の近似ができたと考える. また, メトリックであるほかの距離と比較した場合も比較的同様な計算結果が得られたと考える.

今後の課題として, 図 5 のような分布の特徴, 特に, ヒストグラム距離と編集距離の変種それぞれの特徴があるデータそのものの特徴を解明することが挙げられる. また, このような分布となるヒストグラム距離が木の比較として有用となる場合について考察することも今後の課題である.

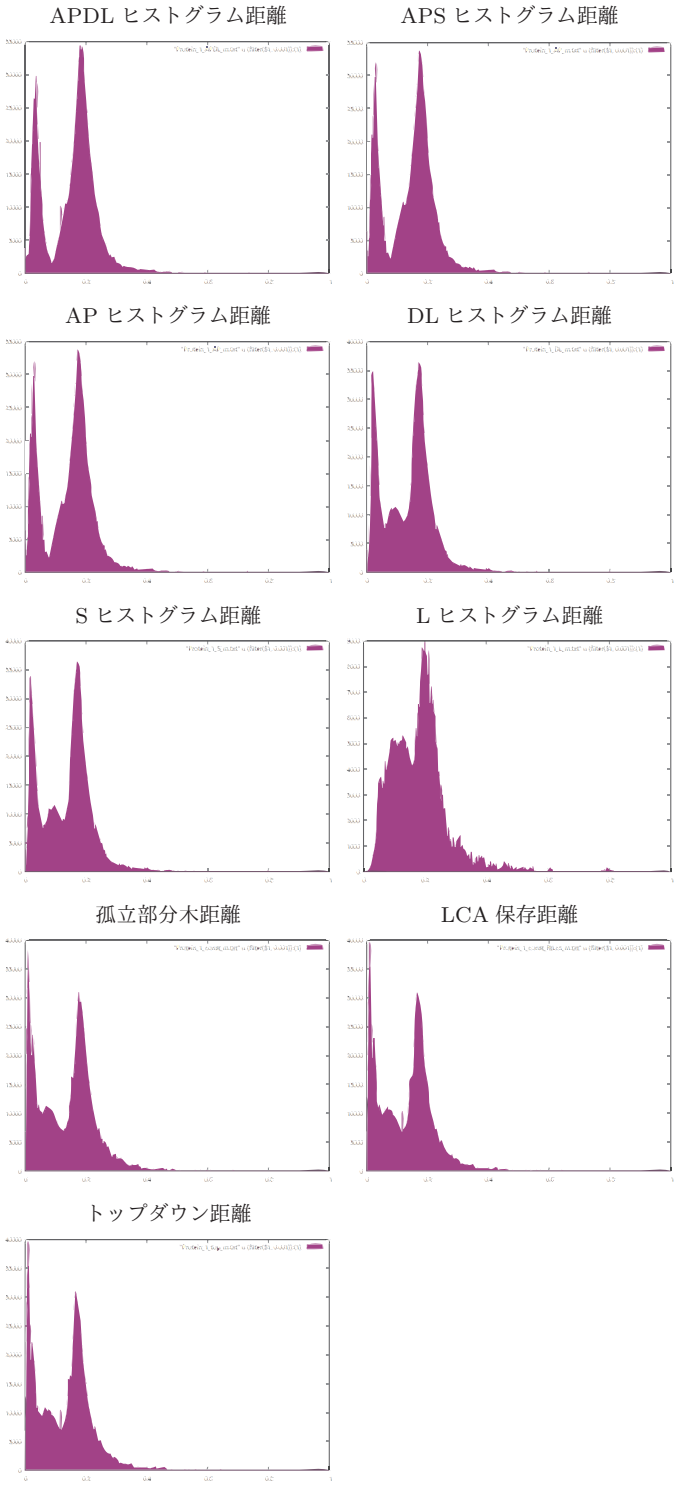


図 5 $\text{Protein}_{1\%}$ における APDL ヒストグラム距離, APS ヒストグラム距離, AP ヒストグラム距離, DL ヒストグラム距離, L ヒストグラム距離, S ヒストグラム距離, 孤立部分木距離, LCA 保存距離, トップダウン距離の分布を表すヒストグラム

参考文献

[1] T. Akutsu, D. Fukagawa, M. M. Halldórsson, A. Takasu, K. Tanaka: *Approximation and parameterized algorithms for common subtrees and edit distance between unordered trees*, Theoret. Comput. Sci. **470**, 10–22, 2013.

[2] T. Aratsu, K. Hirata, T. Kuboyama: *Sibling distance for rooted ordered trees*, JSAI PAKDD 2008 Post-Workshop Proceedings, LNAI **5433**, 99–110, 2009.

[3] S.-S. Chawathe: *Comparing hierarchical data in external memory*, Proc. VLDB'99, 90–101, 1999.

[4] D. Fukagawa, T. Akutsu, A. Takasu: *Constant factor approximation of edit distance of bounded height unordered trees*, Proc. SPIRE'09, LNCS **5721**, 7–17, 2009.

[5] K. Kailing, H.-P. Kriegel, S. Schönauer, T. Seidl: *Efficient similarity search for hierarchical data in large databases*, Proc. EBDT'94, LNCS **2992**, 676–693, 2004.

[6] T. Kawaguchi, T. Yoshino, K, Hirata: *Path histogram distance and complete subtree histogram distance for rooted labeled caterpillars*, J. Inform. Telecommun. **4**, 199–212, 2020.

[7] F. Li, H. Wang, J. Li, H. Gao: *A survey on tree edit distance lower bound estimation techniques for similarity join on XML data* XML data, SIGMOD Recrd 42, 29–39, 2013.

[8] S.-M. Selkow: *The tree-to-tree editing problem*, Inform. Process. Lett. 6,184–186, 1977.

[9] K.-C. Tai: *The tree-to-tree correction problem.*, J. ACM 26, 422–433, 1979.

[10] J.-T.-L. Wang, K. Zhang: *Finding similar consensus between trees: An algorithm and a distance hierarchy*, Pattern recog. 34, 127–137, 2001.

[11] Y. Yamamoto, K. Hirata, T. Kuboyama: *Tractable and intractable variations of unordered tree edit distance*, Int. J.found. Comput. Sci. 25, 307–329, 2014.

[12] W. Yang: *Identifying syntactic differences between two programs*, Software Pract. Exp. 21, 739–755, 1991.

[13] K. Zhang, T. Jiang: *Some MAX SNP-hard results concerning unordered labeled trees*, Inform. Process. Lett. 49, 249–254, 1994.

[14] K. Zhang J.-T.-L. Wang, D. Shasha: *On the editing distance between undirected acyclic graph*, Int. J. Found. Comput. 34, 127–137, 2001.