2005－MPS－53（6）
2005／3／10

木の編集距離尺度の理論的解析

久保山　哲二† 　　申　吉浩†† 　宮原哲浩†††

† 東京大学 国際・産学共同研究センター　†† 東京大学 先端科学技術研究センター
††† 広島市立大学 情報科学部

概要　木の近似照合は広い適用領域をもち、半構造化文書やRNA2次構造の類似性判定、XMLのスキーマ発見・統合、プログラムの差分検出をはじめとする様々な分野で、独立に多様なアルゴリズムが提案されている。木の近似照合アルゴリズムの多くは、編集距離による操作的な記述により特徴づけられてきたが、独立して提案されてきたこれらの様々なアルゴリズムの関連性については、ほとんど研究されていない。本論文では、編集距離に基づく既存の様々な木の近似照合を統一的に記述するための数学的モデルを提案する。文字列においては、アラインメントと編集距離が、その計算において等価であるが、これを木に拡張した場合、両者が等価ではなくなることが知られている。 本提案モデルを用いて、木のアラインメントと等価な編集距離のクラスを同定する。すなわち、従来、別々のアルゴリズムであると考えられていた木のアラインメントと less-constrainded 編集距離が等価であることを示す。

A Theoretical Analysis of Tree Edit Distance Measures

Tetsuji KUBOYAMA†, Kilho SHIN††, and Tetsuhiro MIYAHARA†††

† Center for Collaborative Research, The University of Tokyo
†† Research Center for Advanced Science and Technology, University of Tokyo
†††Faculty of Information Sciences, Hiroshima City University

Abstract　The notion of tree edit distance provides a unifying framework for measuring distance and finding approximate common patterns between two trees. In prior work, the edit distance measures have been not well-formalized. So the essentially equivalent distance measures have been independently proposed as the measures different from each other. In this paper, we present a theoretical framework for tree edit distance. By using our framework, we establish the relationship between alignment of trees, and a tree edit distance measure called less-constrained edit distance.

1. Introduction

Trees, a mathematical abstraction, play a significant role in the efficient organization of information. In particular, the problem of comparing tree structures emerges across a wide range of applications in computational biology [8], image analysis [10], pattern recognition [1], natural language processing, information extraction [7] from Web pages, and many others.

A *tree edit distance* method provides a general framework in comparing trees, measuring similarities, finding common tree patterns, and merging trees.

Zhang and Shasha [14] first gave an efficient algorithm for a tree edit distance measure as a natural generalization of string edit distance [11]. These early works show that the study of tree edit distance has a long history. But this study did not have a firm theoretical foundation.

Many algorithms for calculating tree edit distance are described and characterized by tree edit operations. A lot of those algorithms have been proposed independently in various fields, and the lack of a unifying framework has lead to confusion. That is, the relationship between various algorithms for tree edit distance has hardly been studied, and essentially equivalent distance measures have been independently proposed. The equivalence has remained unnoticed in prior work. So a unifying framework

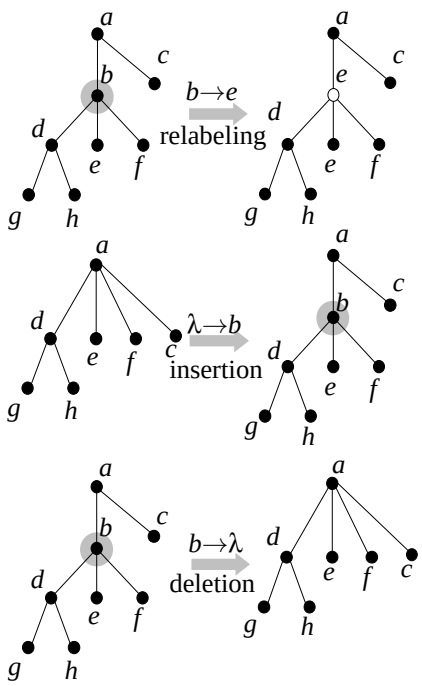

Figure 1 Three elementary edit operations:(1)Relabeling of the node label a to b. (2) Inserting the node label b. (3) Deleting the node label b.

for tree edit distance is needed to describe the semantics of tree edit distance measures.

In this paper, we propose a new mathematical model as a unifying framework for describing tree edit distance measures. This model gives not only operational semantics but also declarative semantics on tree edit distance. As direct results of our model, we point out a misstated statement on tree edit distance in prior work, which have been considered to be true for a few years. Moreover, we elucidate the relationship among existing measures of tree edit distance, which has not been clarified for many years.

2. Tree Edit Distance

In this section, we review the tree edit distace. Trees we consider in this paper are labeled rooted trees, in which each node has a label.

2.1 Operational Definition

The tree edit distance between two trees is defined as the minimum cost of elementary edit operations to transform one tree into the other. In transforming one tree to the other, some elementary edit operations are introduced [9], [14].

Let α be a labeling function which assigns a label from a set $\Sigma = \{a, b, c, \ldots\}$ to each node. Let λ denote the unique null symbol not in Σ.

Definition 1. An *edit operation* on a tree T is any of the following three operations:

- *relabeling* of the label of a node x in T with the label of a new node y in T; denoted by $\alpha(x) \to \alpha(y)$,
- *insertion* of a new node x into T as a child of a node y in T, moving a consecutive subsequence of y's children (and their descendants) right under the new node x; note that this operation is the reverse of deletion; denoted by $\lambda \to \alpha(x)$, and
- *deletion* of a non-root node x from T, moving all children of x right under the parent of x; denoted by $\alpha(x) \to \lambda$.

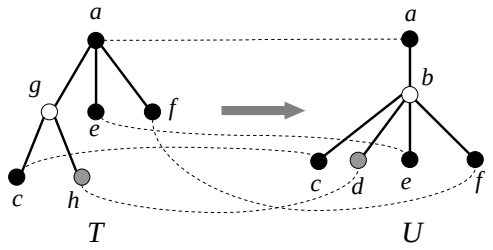

Figure 2 An e-mapping: relabeling the node labeled h with d, deleting the node labeled g, and inserting the node labeled b.

Figure 1 illustrates the edit operations. These operations are used to transform a tree T to another tree U.

Let $\mathcal{S}$ be a sequence of edit operations to transform T to U. Let γ be a cost function of edit operations. γ is defined to be a distance metric as follows: for $a, b, c \in \Sigma \cup \{\lambda\}$, (i) $\gamma(a \to b) \geqq 0$; (ii) $\gamma(a \to b) = \gamma(b \to a)$; and (iii) $\gamma(a \to c) \leqq \gamma(a \to b) + \gamma(b \to c)$. The cost function γ for edit operations is generalized for sequences $\mathcal{S} = \{s_1, \ldots, s_k\}$ $(k \geqq 0)$ of edit operations by letting $\gamma(\mathcal{S}) = \Sigma_{i=1}^{k} \gamma(s_i)$.

The edit distance δ between two trees T and U is defined [9] as

$$\delta(T, U) = \min_{\mathcal{S}} \{\gamma(\mathcal{S})\}.$$

2.2 Edit Mappings

The effect of a sequence of edit operations is reduced to a structure called *edit mapping* [9], which is comparable to *trace* [11] in string edit distance. An *edit mapping* depicts node-to-node correspondences between two trees according to the structural similarity, or shows how nodes in one tree are preserved after transformed to the other(See Fig. 2).

Definition 2. An *edit mapping* from a tree T to a tree U is a set $M \subseteqq V(T) \times V(U)$ such that, for all (x_1, x_2), $(y_1, y_2) \in M$,

(1) $x_1 = y_1$ if and only if $x_2 = y_2$,

(2) x_1 is an ancestor of y_1 if and only if x_2 is an ancestor of y_2.

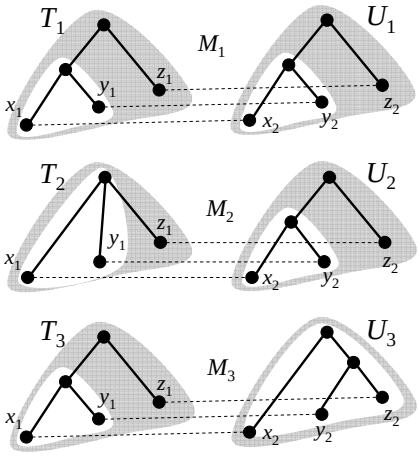

Figure 3 Examples of e-mappings: each white region illustrates how the tree image is mapped by each mapping M_i $(i \in \{1, 2, 3\})$.

For simplicity, we refer to the edit mapping as the *e-mapping*.

3. Distance Measures between Trees

In this section, we give a cursory review of related work, which includes four distance measures between trees. Moreover, we point out a definition not well-stated in the prior work.

We denote by $x \smile y$ the least common ancestor (or the nearest common ancestor) of two nodes x and y. An ancestor of a node is either the node itself, or an ancestor of the parent of the node.

3.1 Standard Edit Distance

The edit distance defined by the e-mapping in Definition 2 is the most general form of e-mapping. We refer this distance measure defined by the e-mapping as the *standard edit distance*, and call this e-mapping the *standard e-mapping*.

For ordred trees, a polynomial-time algorithm was given by Zhang and Shasha [14]. As for unordred trees, this problem is known to be NP-complete [15] (in fact MAX-SNP hard [13]), even for binary trees having a label alphabet of size two.

3.2 Less Constrained Edit Distance

The less-constrained mapping was introduced by Lu. *et al.* [5] to relax the condition of the constrained e-mapping introduced by Zhang [12]. In Fig. 3, both M_1 and M_2 are the less-constrained e-mappings whereas M_1 is the only constarined e-mapping.

Definition 3 (Lu *et al.* 2001 [5])**.** An e-mapping M is *less-constrained* if the following conditions hold: for all $(x_1, x_2), (y_1, y_2), (z_1, z_2) \in M$ such that none of x_1, y_1, and z_1 is an ancestor of the others, $x_1 \smile y_1 = x_1 \smile z_1$ and $x_1 \smile z_1$) is an ancestor of $y_1 \smile z_1$ if and only if $x_2 \smile y_2 = y_2 \smile z_2$ and $y_2 \smile z_2$ is an ancestor of $x_2 \smile z_2$.

The definition of the e-mapping in [5] is not correct since it excludes the case $x_1 \smile y_1 = x_1 \smile z_1 = y_1 \smile z_1$ and $x_2 \smile y_2 = x_2 \smile z_2 > y_2 \smile z_2$. We rectify this e-mapping definition in Section 4. 1.

3.3 Alignment of Trees

The alignment of trees was introduced by Jiang *et al.* [4] as a natural extension of alignment of strings. An efficient algorithm for similar trees were proposed for ordered trees [3], and unordered trees [2]. The definition of the alignment has been given in an operational way as follows.

Definition 4 (Jiang *et al.* 1995 [4])**.** Let T and U be two trees. An alignment of T and U is obtained by first inserting nodes labeled with λ into T and U such that the two resulting trees T' and U' have the same structure, i.e., they are identical if the labels are ignored, and then *overlaying* T' on U'.

An example of alignment is shown in Fig. 4. As for alignment of trees, there has been no definition by an e-mapping condition.

For ordred trees, a polynomial-time algorithm was introduced by Jiang *et al.* [4]. As for the unordred trees, this problem is known to be MAX-SNP hard [4].

4. Formalization of Tree Edit Distance

4.1 Less Constrained E-Mapping Corrected

As mentioned in Section 3. 2, The definition given by Lu *et al.* [5] is incorrect. Thus, the definition should be corrected as follows.

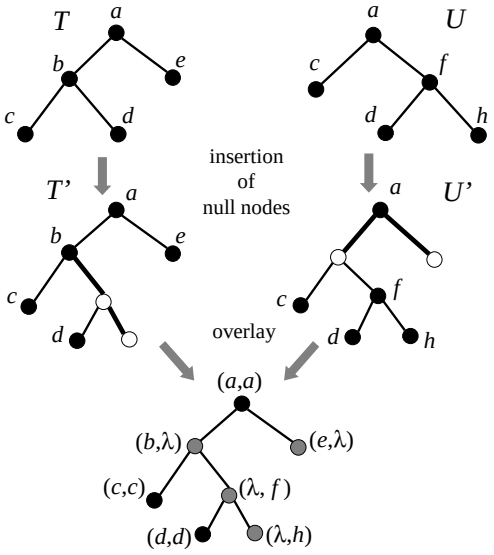

Figure 4 An alignment of trees between T and U

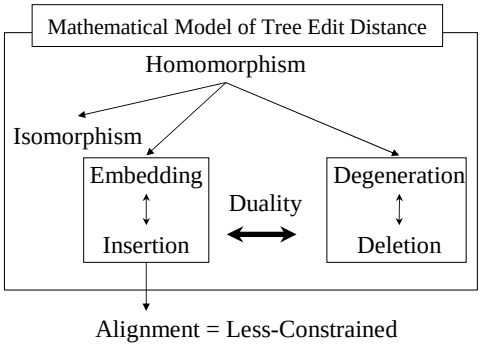

Figure 5 The architecture of our model

Definition 5. A standard e-mapping M is *less-constrained* if the following conditions hold:

(1) for all $(x_1, x_2), (y_1, y_2), (z_1, z_2) \in M$ if $x_1 \smile y_1 < x_1 \smile z_1$, then $y_2 \smile z_2 = x_2 \smile z_2$,

(2) for all $(x_1, x_2), (y_1, y_2), (z_1, z_2) \in M$, if $x_2 \smile y_2 < x_2 \smile z_2$, then $y_1 \smile z_1 = x_1 \smile z_1$.

4.2 Main Result

In our formalization, we first introduce a very general mapping between trees, and call it a homomorphism. Starting with the notion of homomorphism, we tighten the mapping gradually to fit in existing edit operations. Figure 5 shows the architecture of our model.

Through our theoretical model for tree edit distance, we prove an important theorem.

Our main result is the following:

Theorem 1. The edit mapping of the alignment of trees is equivalent to that of the less-constrained edit distance.

This theorem has the following significance:

- We have given the e-mapping condition for alignment of tree, which has been unknown in prior work. This implies that we obtain a declarative definition for alignment of trees.
- We have shown an e-mapping condition which implies that finding the common subtree pattern between two trees is equivalent to finding the common supertree pattern between two trees in terms of minor containment [6].
- Both tree edit distance and alignment of trees have been introduced as natural generalizations of those for strings. Although these two measures are the same originally in strings, these are not the same in trees. We have shown the confluent point between tree edit distance and alignment of trees.

5. Conclusion

In this paper, we have introduced a new theoretical formulation as a unifying framework, which allows us to describe distinct semantics for tree edit distance measures. We have focused on two edit distance measures, the alignment of trees and the less-constrained edit distance which have been independently proposed, but the relationship between them has remained unnoticed in prior work. By using our formulation, we have redefined the semantics of these measures. We then rectified a misstatement in prior work, and established the relationship between these two measures. That is, we have showed that the alignment of trees is essentially equivalent to the less-constrained edit distance.

References

[1] Ferraro, P. and Godin, C.: A Distance Measure between Plant Architectures, *Annals of Forest Science*, Vol. 57, pp. 445–461 (2000).

[2] Fukagawa, D. and Akutsu, T.: Fast algorithms for comparison of similar unordered trees, *Proc. 15th Int. Symp. Algorithms and Computation (ISAAC 2004)* (2004).

[3] Jansson, J. and Lingas, A.: A fast algorithm for optimal alignment between similar ordered trees, *Fundamenta Informaticae*, Vol. 56, pp. 105–120 (2003).

[4] Jiang, T., Wang, L. and Zhang, K.: Alignment of trees — an alternative to tree edit, *Theoretical Computer Science*, Vol. 143, pp. 137–148 (1995).

[5] Lu, C. L., Su, Z.-Y. and Tang, G. Y.: A New Measure of Edit Distance between Labeled Trees, *Lecture Notes in Computer Science*, Vol. 2108, Springer-Verlag Heidelberg, pp. pp. 338–348 (2001).

[6] Nishimura, N., Ragde, P. and Thilikos, D. M.: Finding Smallest Supertrees under Minor Containment, *Lecture Notes in Computer Science*, Vol. 1665, pp. 303–312 (1999). WG'99.

[7] Reis, D. C., Golgher, P. B., Silva, A. S. and Laender, A. H. F.: Automatic Web News Extraction Using Tree Edit Distance, *WWW2004*, pp. 502–511 (2004).

[8] Sakakibara, Y.: Pair hidden Markov models on tree structures, *Bioinformatics*, Vol. 19, pp. 232–240 (2003).

[9] Taï, K.-C.: The Tree-to-Tree Correction Problem, *Journal of the ACM*, Vol. 26, No. 3, pp. 422–433 (1979).

[10] Torsello, A. and Hancock, E. R.: Graph Clustering with Tree-Unions, *LNCS*, Vol. 2756, Springer-Verlag Heidelberg, pp. pp. 451 – 459 (2003). ISBN: 3-540-40730-8.

[11] Wagner, R. and Fischer, M.: The string-to-string correction problem, *Journal of the ACM*, Vol. 21, No. 1, pp. 168–173 (1974).

[12] Zhang, K.: Algorithms for the constrained editing distance between ordered labeled trees and related problems, *Pattern Recognition*, Vol. 28, No. 3, pp. 463–474 (1995).

[13] Zhang, K. and Jiang, T.: Some MAX SNP-hard results concerning unordered labeled trees, *Information Processing Letters*, Vol. 49, pp. 249–254 (1994).

[14] Zhang, K. and Shasha, D.: Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems, *SIAM Journal on Computing*, Vol. 18, No. 6, pp. 1245–1262 (1989).

[15] Zhang, K., Statman, R. and Shasha, D.: On the editing distance between unordered labeled trees, *Information Processing Letters*, Vol. 42, No. 3, pp. 133–139 (1992).