Complexity of computing distances between geometric trees

Aasa Feragen

Department of Computer Science, University of Copenhagen, Universitetsparken 1, 2100 Copenhagen, Denmark
aasa@diku.dk

Abstract. Geometric trees can be formalized as unordered combinatorial trees whose edges are endowed with geometric information. Examples are skeleta of shapes from images; anatomical tree-structures such as blood vessels; or phylogenetic trees. An inter-tree distance measure is a basic prerequisite for many pattern recognition and machine learning methods to work on anatomical, phylogenetic or skeletal trees. Standard distance measures between trees, such as tree edit distance, can be readily translated to the geometric tree setting. It is well-known that the tree edit distance for unordered trees is generally NP complete to compute. However, the classical proof of NP completeness depends on a particular case of edit distance with integer edit costs for trees with discrete labels, and does not obviously carry over to the class of geometric trees. The reason is that edge geometry is encoded in continuous scalar or vector attributes, allowing for continuous edit paths from one tree to another, rather than finite, discrete edit sequences with discrete costs for discrete label sets. In this paper, we explain why the proof does not carry over directly to the continuous setting, and why it does not work for the important class of trees with scalar-valued edge attributes, such as edge length. We prove the NP completeness of tree edit distance and another natural distance measure, QED, for geometric trees with vector valued edge attributes.

1 Introduction

Trees are basic structures in mathematics and computer science, as well as in nature. Tree-structures appear, for instance, as airway trees in the lungs [20,21], as blood vessel trees [13], or as skeleta of more general shapes [4,9,10,15,17,19]. Anatomical and biological trees carry information about the organ or organism that contains them, and many pattern recognition algorithms, e.g., in computer vision and medical image analysis, require a distance measure between tree-structures as input [5,10,15]. Tree edit distance (TED) is a classical distance measure between trees, which has been used in many applications [9,10,14,15, 17]. Anatomical trees are geometric trees, in the sense that they carry useful geometric information about their branches' shape, size and position. TED is readily translated to handle geometric properties, but anatomical trees are often

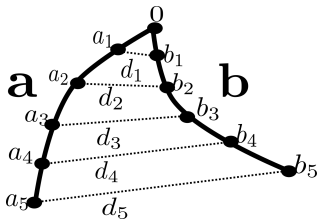

Fig. 1. For many applications, each edge e is represented by an edge attribute consisting of a set of n equidistant landmark points $a_i \in \mathbb{R}^m$, where $m = 2, 3$, giving a point $a = (a_i)_{i=1}^n \in \mathbb{R}^{mn}$. We typically assume that the first landmark point a_1 is translated to the origin. The cost of deforming one edge attribute, or shape, a into another edge attribute, or shape, b is the Euclidean norm $\|a - b\|_2 = \sqrt{d_1^2 + \ldots + d_n^2}$.

not adorned with a natural branch labeling or order. This means that we need to be able to compare unordered trees.

Tree edit distance for unordered trees is generally NP complete to compute [1, 22]. However, the classical proof of NP completeness is made for a particular case of edit distance with integer edit costs for trees with discrete labels, and it does not obviously carry over to the class of geometric trees. This is because the geometric trees have branch descriptors that are vectors or scalars, which thus form a path-connected set of branch attributes, with continuous edit costs.

1.1 Geometric trees

By a *tree* we shall mean a rooted combinatorial tree $\mathscr{T} = \{V, E, r\}$ where V is a set of vertices, $E \subset V \times V$ is a set of edges, and $r \in V$ is a designated root vertex. By *geometric tree* we shall mean a pair $(\mathscr{T}, x)$ where $\mathscr{T}$ is a combinatorial tree and $x\colon E \to A$ is a map from the edge set of $\mathscr{T}$ into a space A of geometric attributes, which attaches an edge attribute $x_e \in A$ to every edge $e \in E$. The space A of geometric attributes could, for instance, be a space of edge lengths, $(\mathbb{R}_{\geq 0})$, a space of edge embeddings into plane or space $(\{f\colon [0,1] \to \mathbb{R}^m\}$, $m = 2, 3)$, or, as a discretization of the latter, a space of landmark point sets that describe the shape of the edge in plane or space $((\mathbb{R}^m)^n$, where n is the number of landmark points per edge, and $m = 2, 3)$, see fig. 1. In this paper, we shall consider situations where the attribute space is $\mathbb{R}_{\geq 0}$ or $\mathbb{R}^N$ for some $N \in \mathbb{N}$.

1.2 Related work

Tree edit distance, or TED [1, 11, 16, 22], is defined as the minimal total sum of costs of edit operations needed in order to turn the first tree into the second. In its most general form, TED is formulated for combinatorial trees $\mathscr{T} = (V, E, r)$ endowed with edge (or vertex) *labels* given by a mapping $x\colon E \to \mathscr{L}$, where $\mathscr{L}$ is a space of labels. The set of labels could be a vector space, as in the case of geometric trees, but in many applications previously studied, the set of labels

is a finite dictionary. The set of edit operations typically consists of *deletion* of edges, *insertion* of edges, and *relabeling* of edges[1] (although extra edit operations have been introduced in some cases [9]). Note that insertion and deletion can also happen to edges which are not leaves. A sequence of edit operations that turn one tree into another is called an *edit path* between the two trees. When restricted to classes of trees with additional assumptions, such as *edge order* or *bounded size*, there exist a number of polynomial time algorithms [3, 11, 22] for computing the tree edit distance, and with further restrictions on the allowed complexity of the edit paths, there are linear time algorithms available [18]. However, for general, unordered trees, the edit distance computation problem has been shown to be NP complete by Zhang, Statman and Shasha [22]. Their proof can be simplified to the trick explained in section 2 below, originally used by Matousek and Thomas to prove NP completeness of the subtree problem [12]. However, as we shall see below, this proof does not automatically transfer to geometric trees with continuous edge labels, and in fact it fails for trees with scalar valued labels. The same is true for the original, slightly more complicated proof in [22]. Using a construction similar to the NP completeness proof of [12], we prove that the computation of tree edit distance is NP complete for the space of geometric trees with vector valued edge labels.

From a statistical point of view, TED is not an optimal distance between geometric trees, as it does not define unique geodesics [6]. Feragen et al. [7] have defined a metric on geodesic trees called *the QED metric* and showed that it has better statistical properties [6]. In section 3 we give a brief account of this metric and prove that it, too, is NP complete to compute for geometric trees with vector edge attributes.

2 Tree Edit Distance

The original proof of NP completeness for edit distance between unordered, rooted trees, is formulated for the class of rooted trees $\mathscr{T} = (V, E, r)$ with edge labels $x\colon E \to \mathscr{L}$ where $\mathscr{L}$ is a discrete set of labels. The available edit operations are *edge deletion*, *edge insertion* and *edge relabeling*, which have cost 1 each. This measure is called *integer TED*.

The exact 3-cover problem. The NP completeness proof for integer TED [12, 22] is based on the exact 3-cover problem. Let $L = \{l_1, \ldots, l_{3q}\}$ be a set, and let $\mathscr{S} = \{C_i | i = 1 \ldots N\}$ be a cover[2] of L by sets $C_i \subset L$, all with 3 elements.

[1] In some papers, e.g. [22] the edit operations (delete, add, edit) are performed on vertices rather than edges. This is equivalent to the approach taken here: Represent the branches of an anatomical tree as attributed nodes, joined together in the obvious tree structure. By defining edit operations on nodes, we would get exactly the same definition as the one used here. We, however, prefer to represent branches in geometric trees as edges, as this is more intuitive, and also quite standard [9, 15].

[2] A *cover* of L is a family of subsets $C_i \subset L$ such that $L \subset \bigcup_{i=1}^{N} C_i$.

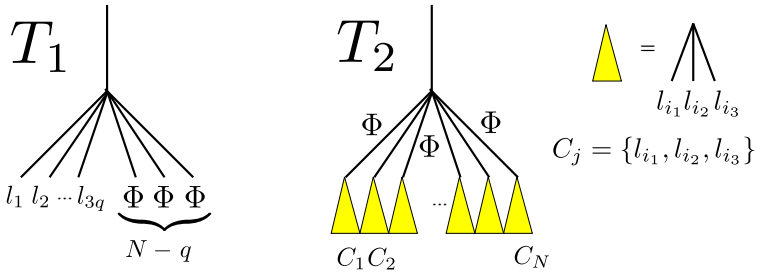

Fig. 2. Any instance of the exact 3-cover problem can be solved by computing the edit distance between these two trees.

The exact 3-cover problem is the problem of deciding whether there is an exact subcover[3] of $\mathscr{S}$ (and identifying such a subcover). The exact 3-cover problem is a classical NP-complete problem [8].

TED and the exact 3-cover problem. We first review the original proof of NP completeness for integer TED [12, 22]. Assume given an instance of the exact 3-cover problem, i.e. assume given a finite set $L = \{l_1, \ldots, l_{3q}\}$ with a cover $\mathscr{S} = \{C_i | i = 1 \ldots N\}$ by sets C_i that have 3 elements each. Build the edge-labeled trees T_1 and T_2 with labels from $L \cup \{\Phi\}$, as in fig. 2, where Φ is some label not in L. We shall see that

i) by computing the TED distance between T_1 and T_2, we can determine whether there exists an exact subcover of $\mathscr{S}$, and
ii) if there is an exact subcover, we can retrieve it from the optimal edit path from T_1 to T_2.

Let us ignore the tree-structure of T_1 and T_2 for a second and only consider the two sets of attributed edges. To find the minimal total cost of editing one set to become the other, note that the set of edge attributes $L_1 = x_1(E_1)$ in T_1 is contained in the set of edge attributes $L_2 = x_2(E_2)$ in T_2. There are $N + 2q + 1$ edge attributes in L_1 and $4N + 1$ edge attributes in L_2, so in order to transform $L_1 \subset L_2$ into L_2, we only need to insert $3N - 2q$ edges, at a total cost of

$$b_l = (4N + 1) - (N + 2q + 1) = 3N - 2q.$$

This number b_l is a lower bound for the edit distance between T_1 and T_2.

If there is a solution to the exact 3-cover problem on S, consisting of a set $\mathscr{S}' = \{C_i | i = 1 \ldots q\}$ of 3-sets, then the following edit path from T_1 to T_2 actually *has* length b_l:

– insert an edge with attribute Φ above each triple of elements in some $C_i \in \mathscr{S}'$, $i = 1 \ldots q$.

[3] An *exact subcover* of $\mathscr{S}$ is a sub-family $\mathscr{S}' = \{C_{i_j}\}_{j=1}^{M}$ of $\mathscr{S}$ such that $\mathscr{S}'$ is a cover of L and $C_{i_{j_1}} \cap C_{i_{j_2}} = \emptyset$ for all $j_1 \neq j_2$.

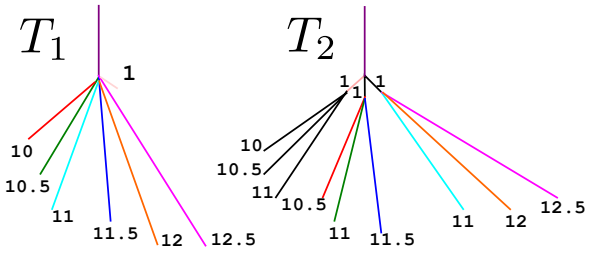

Fig. 3. The edit path indicated by the colored edge matches has minimal length, even though the corresponding exact 3-cover problem does not have a solution. Thus, the proof from integer TED does not carry over to the case of continuous edge attributes.

– insert the remaining $3(N-q)$ edges with attributes belonging to the remaining $N-q$ yellow subtrees below each of the Φ branches in T_1.

Since this edit path has length b_l, which is also the lower bound for the length, the TED distance between T_1 and T_2 is b_l. Thus, a solution to the exact 3-cover problem yields a) a solution to the TED problem and b) a total distance b_l between T_1 and T_2. If we can show that edit paths that do *not* yield solutions to the exact 3-cover problem are *longer* than b_l, then we have proven our claim. But this is easy, since any mapping from T_1 to T_2 which does not correspond to a solution to the exact 3-cover problem must involve either changing some label l_i to another label l_j, or deleting edges, or both. This has to cost more than b_l.

It follows that the computation of integer TED is NP complete.

2.1 Example: Geometric trees with scalar branch attributes

To see that the same idea of proof does not carry directly over to geometric trees, consider the following set $L = \{10, 10.5, 11, 11.5, 12, 12.5\}$ and the following cover of L by 3-sets: $\mathscr{S} = \{\{10, 10.5, 11\}, \{10.5, 11, 11.5\}, \{11, 12, 12.5\}\}$. Clearly, $\mathscr{S}$ does not have an exact subcover. As above we form trees T_1 and T_2 as in fig. 3, where the lengths of edges labeled by elements in L are the corresponding real numbers, and the lengths of edges labeled with Φ are, say, 1.

A lower bound b_l for the edit distance between T_1 and T_2 is, just like above, found by just considering sets of edge attributes, forgetting about tree topology for a second, matching the sets of edge attributes up, and adding the costs of the entire matching process. Again, all edge attributes from T_1 can be matched to an identical edge attribute from T_2, so the only nonzero matching costs come from the additional edges in T_2, namely $2 * \|\Phi\| + 10.5 + 11 + 11 = 34.5$.

We already know that there is no exact 3-cover of $\mathscr{S}$; nevertheless, we can, in fact, find an edit path from T_1 to T_2 of length b_l, where the branches indicated by colors in fig. 3 are matched (deformed to match) and all branches appearing in black in T_2 are inserted. The total cost of deformation edits is 1 and the total cost of insertion edits is 33.5, giving an edit distance of $34.5 = b_l$ between T_1 and T_2, although the edit path does not correspond to a solution of the exact

3-cover problem. It follows that the proof from the integer edit distance does not carry over to TED for geometric trees.

2.2 Tree edit distance for geometric trees

Building on the original proof described in the previous section, we consider the class $\mathscr{X}$ of all geometric trees $(\mathscr{T}, x)$ with edge attributes $x\colon E \to \mathbb{R}^N$, $N \in \mathbb{N}$. These edge attributes could, e.g., be edge length, or shape descriptors as in fig. 1.

Define the tree edit distance (TED) between two geometric trees T_1 and T_2 in $\mathscr{X}$ as the smallest possible total cost of transforming T_1 into T_2 through a finite sequence of edit operations, which belong to the following three categories:

i) *Delete an edge* $e \in E$ (and correspondingly a vertex from V), which costs $\|x(e)\|$, where $\|\cdot\|$ is the Euclidean norm,
ii) *Insert an edge* e to E (and correspondingly a vertex from V), which costs $\|x(e)\|$, and
iii) *Deform an edge* $e \in E$ by changing its attribute from $x(e)$ to a new value a; this costs $\|a - x(e)\|$.

Theorem 1 *If $N \geq 2$, then computing tree edit distance in $\mathscr{X}$ is NP-complete.*

Proof. As for the combinatorial edit distance, this is proven by reducing an arbitrary instance of the exact 3-cover problem to an instance of the edit distance problem. We prove the theorem for $N = 2$; the proof trivially generalizes to $N \geq 2$. Denote by T_1, T_2 the trees in fig. 2, labeled with elements from L and an additional label Φ, where the l_i and Φ represent *distinct* vector edge attributes of length 1.

As before, we can forget about the tree structure and only consider sets of edges. The set of edge attributes in T_1 is, again, contained in the set of edge attributes in T_2, and the minimal total edit cost of transforming the set of $N + 2q + 1$ edge attributes in T_1 to the set of $4N + 1$ edge attributes in T_2 is the cost of inserting the rest of the edge attributes from T_2, which all cost 1 each. This gives us total cost

$$b_l = (4N + 1) - (N + 2q + 1) = 3N - 2q.$$

Again, we need to prove that any edit path that does not correspond to a solution to the exact 3-cover problem must have length $> b_l$. An edit path that does not correspond to a solution to the exact 3-cover problem will have to either:

a) map some edge with (nonzero) attribute l_i to an edge with (nonzero) attribute l_j which is *not* l_i, or
b) delete some edge, or
c) map the edges from T_1 into edges in more than q subtrees C_i in T_2.

Note that

a) the cost of mapping l_i to some $l_j \neq l_i$ has cost $\|l_i - l_j\|$, which is > 0 since the l_i are distinct. This cost comes in addition to inserting at least $3N - 2q$ branches, which gives total cost $> 3N - 2q = b_l$.

b) this means we have to insert more than $3N - 2q$ branches, giving total cost $> 3N - 2q = b_l$.
c) this means we will have to delete some of the branches with attribute Φ from T_1, and thus we have to grow out more than $3N - 2q$ branches, giving total cost $> 3N - 2q = b_l$.

Thus, any edit path which does not correspond to a solution to the exact 3-cover problem has length $> b_l$. This concludes the proof of theorem 1. □

Remark 2 In a), the crucial part is, in fact, that the vectors l_i are not parallel. This is to avoid examples like the scalar attribute case in section 2.1.

3 NP completeness for Quotient Euclidean Distance

In order to use geometric tools for statistical analysis of geometric trees, e.g., use of geodesics in the spirit of manifold statistics, it is useful to construct a space of geometric trees, and endow it with a geodesic metric. The non-uniqueness of TED geodesics disqualifies TED as a metric of choice in such a framework. A more suitable metric is the QED metric on the space of tree-like shapes as defined by Feragen et al. [6,7], which has been used to study the shapes of airway trees from human lungs.

By a *tree-shape*, we shall mean a tree which is embedded in $\mathbb{R}^d$, where d is typically 2 or 3. In this paper, we are mainly concerned with the case $d = 3$, since planar trees ($d = 2$) typically induce a canonical edge ordering. The space of tree-like shapes is constructed as follows: Consider a combinatorial rooted, binary tree $T = (V, E, r, <)$ which is sufficiently large to span all the tree-like shapes of interest (T could be infinitely large). The space

$$X = \prod_{e \in E} (\mathbb{R}^m)^n, \ m = 2, 3, \tag{3}$$

contains representatives of all tree-shapes spanned by T, whose edges are represented by landmark point shape descriptors as in fig. 1. That is, a point $x \in X$ corresponds to a map $x \colon E \to (\mathbb{R}^m)^n$. Trees with fewer edges are represented by collapsing (contracting) redundant branches, and higher-order vertices are represented in a similar fashion, also using collapsed branches, as in fig. 4. Some tree-shapes will have more than one representative in X, also shown in fig. 4. In the space of tree-like shapes, these representations are all identified through an equivalence relation. That is, whenever two points $x_1, x_2 \in X$ represent the same tree-shape, they are said to be equivalent: $x_1 \sim x_2$. The space of tree-like shapes $\bar{\bar{X}}$ is defined as the quotient space of X by the equivalence $\sim$:

$$\bar{\bar{X}} = X/\sim .$$

The induced tree-shape space $\bar{\bar{X}}$ is highly nonlinear, and has self-intersections that stem from the identifications made by the equivalence. From the Euclidean

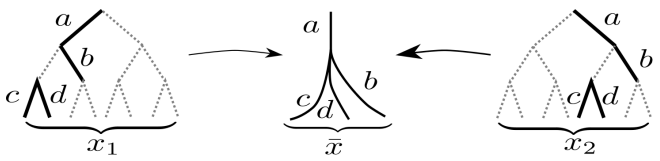

Fig. 4. Higher-order vertices can be represented by the binary tree by collapsing internal branches, shown as dotted lines.

metric on X, Feragen et al. work with the *quotient metric* on $\bar{\bar{X}}$, which in this case is called the *QED metric*. The quotient metric is a standard mathematical construction [2], which here creates a piecewise Euclidean metric on $\bar{\bar{X}}$. Note that $\bar{\bar{X}}$ geometrically corresponds to a folded Euclidean space. This construction is actually closely related to the TED metric: If the Euclidean metric on X is replaced with an l_1 product of Euclidean metrics on $(\mathbb{R}^m)^n$ in the product in (3), the geometric TED metric studied in section 2 above is retrieved as quotient metric on $\bar{\bar{X}}$ [7].

It turns out that computing the QED distance is generally also NP complete:

Theorem 4 *Computing QED distances in $\bar{\bar{X}}$ is NP-complete.*

Proof. Just as for TED, the QED shortest paths consist of deleting, inserting and deforming edges. Using the same two trees T_1 and T_2 shown in fig. 2, we see that again, if we disregard the tree structure, the lower bound b_l for the QED distance from T_1 to T_2 is given by $b_l = \sqrt{3N - 2q}$, which can be obtained as a shortest QED path length if and only if there exists a solution to the exact 3-cover problem, using the same matchings as in the TED case. Again, the non-parallel property as noted in Rem. 2 is essential. □

Remark 5 As in section 2.1 the proof would not hold if we replaced the edge shape space $(\mathbb{R}^m)^n$ by scalar edge descriptors $\mathbb{R}$, because the proof depends on the non-parallel assumption on attributes.

4 Discussion and conclusion

In this paper we see that the most common distances between unlabeled, unordered geometric trees with vector edge attributes are generally NP complete to compute, just like the edit distance between purely combinatorial unordered, unlabeled trees. NP completeness is a result of the exponential search space which arises when there is no or little formal limitation to the possible mappings between the trees. For trees with scalar edge attributes, such as edge length, the proofs of NP completeness do not hold, and we conjecture that computing these distances is, in fact, also NP complete.

5 Acknowledgements

The author would like to thank Sean Skwerer and Scott Provan for valuable discussions on complexity of tree algorithms.

References

1. P. Bille. A survey on tree edit distance and related problems. *Theor. Comput. Sci.*, 337(1-3):217–239, 2005.
2. M. R. Bridson and A. Haefliger. *Metric spaces of non-positive curvature*. Springer, 1999.
3. E.D. Demaine, S. Mozes, B. Rossman, and O. Weimann. An optimal decomposition algorithm for tree edit distance. *ACM Trans. Algorithms*, 6:2:1–2:19, 2009.
4. F. Demirci, A. Shokoufandeh, and S.J. Dickinson. Skeletal shape abstraction from examples. *TPAMI*, 31(5):944–952, 2009.
5. M. Demirci, B. Platel, A. Shokoufandeh, L. Florack, and S. Dickinson. The representation and matching of images using top points. *JMIV*, 35:103–116, 2009.
6. A. Feragen, S. Hauberg, M. Nielsen, and F. Lauze. Means in spaces of tree-like shapes. In *ICCV*, 2011.
7. A. Feragen, F. Lauze, P. Lo, M. de Bruijne, and M. Nielsen. Geometries on spaces of treelike shapes. In *ACCV 2010*, volume 6493 of *LNCS*, pages 160–173. 2011.
8. M.J. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, 1979.
9. P. Klein, S. Tirthapura, D. Sharvit, and B. Kimia. A tree-edit-distance algorithm for comparing simple, closed shapes. In *In SODA*, pages 696–704, 2000.
10. P. N. Klein, T. B. Sebastian, and B. B. Kimia. Shape matching using edit-distance: an implementation. SODA, pages 781–790, 2001.
11. P.N. Klein. Computing the edit-distance between unrooted ordered trees. In *Proc. 6th Annual European Symposium on Algorithms*, pages 91–102, 1998.
12. J. Matoušek and R. Thomas. On the complexity of finding iso- and other morphisms for partial k-trees. *Discrete Mathematics*, 108(1-3):343–364, 1992.
13. J. H. Metzen, T. Kröger, A. Schenk, S. Zidowitz, H-O. Peitgen, and X. Jiang. Matching of anatomical tree structures for registration of medical images. *Im. Vis. Comp.*, 27:923–933, 2009.
14. K. Riesen and H. Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Im. Vis. Comp.*, 27(7):950–959, 2009.
15. T. B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing their shock graphs. *TPAMI*, 26(5):550–571, 2004.
16. K.-C. Tai. The tree-to-tree correction problem. *J. ACM*, 26:422–433, 1979.
17. A. Torsello, A. Robles-Kelly, and E.R. Hancock. Discovering shape classes using tree edit-distance and pairwise clustering. *IJCV*, 72(3):259–285, 2007.
18. H. Touzet. A linear tree edit distance algorithm for similar ordered trees. In *Combinatorial Pattern Matching*, volume 3537 of *LNCS*, pages 334–345. 2005.
19. N. Trinh and B. Kimia. Skeleton search: Category-specific object recognition and segmentation using a skeletal shape model. *IJCV*, pages 1–26, 2011.
20. J. Tschirren, G. McLennan, K. Palágyi, E. A. Hoffman, and M. Sonka. Matching and anatomical labeling of human airway tree. *TMI*, 24(12):1540–1547, 2005.
21. E. R. Weibel. What makes a good lung? *Swiss Med. Weekly*, 139(27-28):375–386, 2009.
22. K. Zhang, R. Statman, and D. Shasha. On the editing distance between unordered labeled trees. *Inf. Process. Lett.*, 42(3):133–139, 1992.