A linear-time algorithm for comparing similar ordered trees

Hélène Touzet

LIFL – University of Lille 1 – France
Comparison with $k$ errors

- **Problem:**
  - **Input:** two ordered trees (that are assumed to be similar) a natural number $k$
  - **Output:** the best mapping $M$ containing less than $k$ errors, if it exists

- **Error:** insertion of a node, deletion of a node

- **Edit operations:** substitution, deletion, insertion

- **Comparison model:** edit distance vs alignment
How to compare trees: edit operations

Substitution

Deletion

Insertion
How to compare trees: comparison model

  - all mappings are valid
  - largest common subtree

- **Alignment** [Jiang et al. 1995]
  - insertions should precede deletions
  - smallest common supertree
## Previous results

<table>
<thead>
<tr>
<th>Strings</th>
<th>Tree distance</th>
<th>Tree alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>full mapping</td>
<td>$O(n^2)$</td>
<td>$O(n^4)$ Zhang-Shasha $O(n^3 \log(n))$ Klein</td>
</tr>
<tr>
<td>$k$-errors</td>
<td>$O(kn)$</td>
<td></td>
</tr>
</tbody>
</table>

$n$ : size of the tree  
$d$ : maximal degree of the tree  
$k$ : bound on the number of errors - known in advance
## Previous results

<table>
<thead>
<tr>
<th></th>
<th>Strings</th>
<th>Tree distance</th>
<th>Tree alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>full mapping</td>
<td>$O(n^2)$</td>
<td>$O(n^4)$ Zhang-Shasha $O(n^3 \log(n))$ Klein</td>
<td>$O(n^2 d^2)$ Jiang <em>et al.</em></td>
</tr>
<tr>
<td>$k$-errors</td>
<td>$O(kn)$</td>
<td>$O(k^3 n)$</td>
<td>$O(n \log(n) d^3 k^2)$ Jansson-Lingas</td>
</tr>
</tbody>
</table>

- $n$: size of the tree
- $d$: maximal degree of the tree
- $k$: bound on the number of errors - known in advance
Edit graph for the string alignment problem

- Two-dimensional grid
- Three kinds of arcs: deletion, insertion and substitution

Time complexity: $O(n^2)$
Edit graph for the string alignment problem

- Two-dimensional grid
- Three kinds of arcs: deletion, insertion and substitution

Time complexity: $O(n^2)$

With $k$-errors: $O(kn)$
Tree edit graph

- Trees as strings: enumerate the nodes in postorder traversal
- Supplementary constraints imposed by the tree structure
Tree edit graph

- Trees as strings: enumerate the nodes in postorder traversal
- Supplementary constraints imposed by the tree structure

Legal path
Tree edit graph

- Trees as strings: enumerate the nodes in postorder traversal
- Supplementary constraints imposed by the tree structure
Tree edit graph

- **Trees as strings**: enumerate the nodes in postorder traversal
- **Supplementary constraints imposed by the tree structure**
Tree edit graph

- Trees as strings: enumerate the nodes in postorder traversal
- Supplementary constraints imposed by the tree structure
Edit graph for trees

- **Deletion arcs** (horizontal arcs):
  \[(x, y) \rightsquigarrow (x - 1, y)\] labeled by \(\text{del}\)

- **Insertion arcs** (vertical arcs):
  \[(x, y) \rightsquigarrow (x, y - 1)\] labeled by \(\text{ins}\)

- **Substitution arcs**:
  \[(x, y) \rightsquigarrow (x - \text{size}(x), y - \text{size}(y))\]
  labeled by the distance between \(A(x)\) and \(B(y)\)

- **Size of the graph** : \(O(mn)\)
and so on ...
**Usage of the tree edit graph**

**How to compute the valuations of the arcs?**
- The label of the substitution arc starting from \((x, y)\) is the weight of an optimal path in the subgraph delimited by \(A(x) \times B(y)\)

**Time complexity :** \(O(n^4)\)
**Space complexity :** \(O(n^2)\)

**How to recover the mapping from the tree edit graph?**
Multi-level tracing back:
- Construction of an optimal path for \(A \times B\)
- Iteration for subgraphs induced by matching pairs of nodes

**Time complexity :** \(O(n^3)\)
**Space complexity :** \(O(n^2)\)
Optimal paths for $td(x, y)$

\[ h = x - \text{size}(x), \; l = y - \text{size}(y) \]

\[
fd(h, l, h, l) = 0 \\
fd(i, l, h, l) = fd(i - 1, l, h, l) + \text{del} \\
fd(h, j, h, l) = fd(h, j - 1, h, l) + \text{ins}
\]

\[
fd(i, j, h, l) = \min \left\{ \begin{array}{l}
fd(i - 1, j, h, l) + \text{del} \\
fd(i, j - 1, h, l) + \text{ins} \\
fd(i - \text{size}(i), j - \text{size}(j), h, l) + td(i, j)
\end{array} \right. 
\]

For the subtrees

\[
\text{if } fd(x - 1, y - 1, h, l) + \text{sub}(x, y) < \\
\min\{fd(x - 1, y, h, l) + \text{del}, fd(x, y - 1, h, l) + \text{ins}\}
\text{ then } td(x, y) \leftarrow fd(x - 1, y - 1, h, l) + \text{sub}(x, y) \\
\text{else } td(x, y) \leftarrow +\infty
\]

This is Zhang&Shasha algorithm

Klein and Dulucq&Touzet algorithms build the same edit graph, but they use alternative strategies to compute the valuations of the arcs.
Edit distance with \( k \) errors

- **Error**: insertion of a node, deletion of a node

- **Problem**:
  - **Input**: two ordered trees, a natural number \( k \)
  - **Output**: the best mapping containing less than \( k \) errors, (if it exists)

- **Method**: pruning the tree edit graph
Edit distance with $k$ errors

Idea 1: the best mappings have their path near the main diagonal
**Edit distance with $k$ errors**

**Idea 1:** the best mappings have their path near the main diagonal

$k$-strip$=\{(x, y); \ |x - y| \leq k\}$
Edit distance with $k$ errors

**Idea 1**: the best mappings have their path near the main diagonal

$k$-strip = \{(x, y); \ |x - y| \leq k\}

Size of the graph: $O(nk)$

Computation time for each node: $O(\text{size}(A, x)k)$

$O(k^2 \sum \text{size}(A, x))$
Edit distance with $k$ errors

**Idea 2**: when inspecting the subtree rooted at $x$, there is no need to visit the nodes of depth $> k + 1$
Edit distance with $k$ errors

**Idea 2**: when inspecting the subtree rooted at $x$, there is no need to visit the nodes of depth $\geq k + 1$
Idea 2: when inspecting the subtree rooted at \( x \), there is no need to visit the nodes of depth \( > k + 1 \)

\[
A(x, k) = \{ i \in A(x); \text{depth}(i) - \text{depth}(x) \leq k + 1 \}
\]
Edit distance with $k$ errors

**Idea 2**: when inspecting the subtree rooted at $x$, there is no need to visit the nodes of depth $> k + 1$

\[
A(x, k) = \{ i \in A(x); \quad \text{depth}(i) - \text{depth}(x) \leq k + 1 \}
\]

Size of the graph: $O(nk)$

Computation time for each node: $O(\text{size}(A, x, k)k)$

\[
O(k^2 \sum \text{size}(A, x, k)) = O(k^3 n)
\]
Tree edit graph for $k$ errors: $O(k^3 n)$

**Input:** two trees $A$ and $B$, positive integer $k$

**Output:** tree edit graph

```plaintext
for $(x, y) \in k\text{-strip}(A, B)$ do $O(k^2 \sum \text{size}(A, x, k)) = O(k^3 n)$
    if not $k$-relevant($x, y$)
        then $td(x, y) \leftarrow +\infty$
    else for $i \in A(x, k)$ do $O(k \text{size}(A, x, k))$
        for $j \in B$ such that $(i, j) \in k\text{-strip}(A, B)$ do $O(k)$
            compute $fd(i, j)$ $O(1)$
        end do
    end do
    compute $td(x, y)$ $O(1)$
end if
end do
```

Recovering the optimal mapping: $O(k^3 n)$