Good Tree Edit Similarity Learning by Loss Minimization

José F. Bernabeu, Aurélien Bellet, Amaury Habrard, Marc Sebban

July 13, 2012

GRFIA 2012
Common approach in supervised classification: learn to classify objects using a pairwise similarity (or distance) function.

Best way to get a "good" similarity function for a specific task: learn it from data!

GESL (Good Edit Similarity Learning) is a novel, efficient way to learn string similarities.

We adapt GESL to tree edit cost learning.
The string edit distance

Standard (Levenshtein) edit distance $e_L$ between two strings $x$ and $y$: minimum number of operations to transform $x$ into $y$. Allowable operations are insertion, deletion and substitution of symbols.

**Example 1**

$e_L(abb, aa) = C(b, a) + C(b, $) = 1 + 1 = 2$

Generalized version $e_C$: use a cost for each operation.

**Example 2**

<table>
<thead>
<tr>
<th></th>
<th>$</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>-</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
<td>3</td>
<td>-</td>
</tr>
</tbody>
</table>

$\Rightarrow e_C(abb, aa) = C(b, $) + C(b, $) + C($, a) = 1$

$: empty symbol
There exists a decent amount of literature on learning edit costs (or probabilities) from data. See Ristad & Yianilos (1998), Bilenko & Mooney (2003), Oncina & Sebban (2006), Takasu (2009)...

**Drawbacks of the state-of-the-art**

- most of them use an iterative procedure, which can be costly.
- they often make use of positive pairs only (i.e., moving examples of the same class "closer" together). What about negative pairs?
- above all, they are not guaranteed to perform well for the task at hand.
An iterative approach is usually needed because the optimal edit script (= best sequence of operations) depends on the edit costs. → Solution: define a different type of edit function!

**Definition of** $e_G$

$$e_G(x, x') = \sum_{0 \leq i, j \leq A} C_{i,j} \times \#_{i,j}(x, x')$$

where $A$ is the size of the alphabet, $C$ the edit cost matrix and $\#_{i,j}(x, x')$ the number of times the operation $(i, j)$ appears in the Levenshtein script.

We will optimize:

**Definition of** $K_G$

$$K_G(x, x') = 2e^{-e_G(x, x')} - 1 \in [-1, 1]$$


- Make use of the theory of learning with \((\epsilon, \gamma, \tau)\)-good similarity functions (Balcan et al.).
- Is not based on a costly iterative procedure but on solving an efficient convex program.
- Make use of positive and negative pairs.
- The resulting similarities provably generalize well to new examples,
- Induce low-error classifiers for the task at hand.

GESL phases:
- Learn edit cost matrix.
- Learn linear classifier.
Tree adaptation

\[ e_G(x, x') = \sum_{0 \leq i, j \leq A} C_{i,j} \times \#_{i,j}(x, x') \]

- is a linear combination of the edit costs.
- \( \#_{i,j}(x, x') \) the number of times the operation \((i, j)\) appears in the Levenshtein script.
- we can use other tree edit scripts:
  - Selkow 1977.
Set of 420 monophonic variations played by musicians.
Themes (8-12 bars) of 20 worldwide well-known tunes of different musical genres.
Musicians played the songs using MIDI controllers, real-time sequencing them.
21 different variations were obtained for each song.
The goal is to select the correct song given a query taken from the variations.
Results (*Pascal Corpus*)

- 3-fold cross-validation scheme.
- Selkow 90.5 ± 0.9 using 1-NN.

<table>
<thead>
<tr>
<th>Cost Matrix</th>
<th>Learning costs (1-NN)</th>
<th>Learning classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symetric</td>
<td>91.9 ± 0.5</td>
<td>93.8 ± 0.9</td>
</tr>
<tr>
<td>No Symetric</td>
<td>88.8 ± 0.2</td>
<td>93.8 ± 0.5</td>
</tr>
</tbody>
</table>
3-fold cross-validation scheme.

Shasha 93.6 ± 0.5 using 1-NN.

<table>
<thead>
<tr>
<th>Cost Matrix</th>
<th>Learning costs (1-NN)</th>
<th>Learning classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symetric</td>
<td>92 ± 2</td>
<td>95.5 ± 0.7</td>
</tr>
<tr>
<td>No Symetric</td>
<td>88.8 ± 0.5</td>
<td>94.8 ± 0.9</td>
</tr>
</tbody>
</table>
Conclusions

- In this work, we adapt GESL to use tree-structured data.
- GELS improve the results using Selkow and Shasha edit scripts.
- It is not always appropriate to use the learned cost matrix to classify with the 1-NN rule.
Learning Good Tree Edit Similarities with Generalization Guarantees 12 / 12

José F. Bernabeu, Aurélien Bellet, Amaury Habrard, Marc Sebban

July 13, 2012

GRFIA 2012