Learning an Unbiased Stochastic Edit Distance in the form of a Memoryless Finite-State Transducer

Jose Oncina\textsuperscript{1} and Marc Sebban\textsuperscript{2}

\textsuperscript{1}University of Alicante, Department of Software and Computing Systems
\textsuperscript{2}University Jean Monnet of Saint-Etienne, EURISE
Outline

- Research context and motivations
- Related work and preliminary experiments
- Learning parameters of a conditional memoryless transducer
- Series of experiments in GI dealing with noisy data
- Perspectives
Research Context and Motivations
Research Context and Motivations

- **Original goal**: dealing with noisy data in GI to avoid
  - overfitting phenomena,
  - high structural complexity of the inferred automata.

- 

Research Context and Motivations

- **Original goal:** dealing with noisy data in GI
  - overfitting phenomena,
  - high structural complexity of the inferred automata.

- **3 possible approaches consisting in:**
  1. *a priori* removing noisy and irrelevant strings/trees [Habrard et al. 2005, Fundamenta Informaticae]
Research Context and Motivations

• **Original goal**: dealing with noisy data in GI
  – overfitting phenomena,
  – high structural complexity of the inferred automata.

• **3 possible approaches consisting in:**
  1. *a priori* removing noisy and irrelevant strings/trees [Habrard et al. 2005, Fundamenta Informaticae]
  2. modifying merging rules of inference algorithms [Habrard et al. 2004, ECML’04]
Research Context and Motivations

- **Original goal**: dealing with noisy data in GI
  - overfitting phenomena,
  - high structural complexity of the inferred automata.

- **3 possible approaches consisting in**:
  1. *a priori* removing noisy and irrelevant strings/trees [Habrard et al. 2005, Fundamenta Informaticae]
  2. modifying merging rules of inference algorithms [Habrard et al. 2004, ECML’04]
  3. correcting noisy data
Research Context and Motivations

- **Original goal**: dealing with noisy data in GI
  - overfitting phenomena,
  - high structural complexity of the inferred automata.

- **3 possible approaches consisting in**:
  1. *a priori* removing noisy and irrelevant strings/trees [Habrard et al. 2005, Fundamenta Informaticae]
  2. modifying merging rules of inference algorithms [Habrard et al. 2004, ECML’04]
  3. correcting noisy data
      - by estimating the level of noise [Habrard et al. 2004, FLAIRS’05]
Research Context and Motivations

- **Original goal**: dealing with noisy data in GI
  - overfitting phenomena,
  - high structural complexity of the inferred automata.

- **3 possible approaches consisting in**:
  1. *a priori* removing noisy and irrelevant strings/trees [Habrard et al. 2005, Fundamenta Informaticae]
  2. modifying merging rules of inference algorithms [Habrard et al. 2004, ECML’04]
  3. correcting noisy data
    - by estimating the level of noise [Habrard et al. 2004, FLAIRS’05]
    - **by learning a given noise model**
Learning an Edit Noise Model

“An Edit Noise is the result of the corruption of an input string by purely random errors, called edit operation errors. [Sakakibara and Siromoney, 1992]
Learning an Edit Noise Model

“An Edit Noise is the result of the corruption of an input string by purely random errors, called edit operation errors. [Sakakibara and Siromoney, 1992]

Edit Operations $\rightarrow$ Underlying Probability Distribution $\rightarrow$ Stochastic Edit Distance (ED)
Learning an Edit Noise Model

“An Edit Noise is the result of the corruption of an input string by purely random errors, called edit operation errors. [Sakakibara and Siromoney, 1992]

Edit Operations → Underlying Probability Distribution
→ Stochastic Edit Distance (ED)

Learning a Stochastic ED → Learning a Probabilistic Model
Related Work and Preliminary Experiments
Related Work

- Stochastic model (memoryless transducer) for string edit distance [Ristad and Yanilos, 1996, 1998]
- Pair HMM [Durbin et al. 1998]
- Error correcting technique for smoothing PDFA [Dupont and Amengual, 2000].
Related Work

The previous approaches share the following properties:
→ use the EM algorithm
→ learn a joint distribution over the pairs \((x, y)\) of strings
→ deduce the stochastic edit distance as follows:
\[
d_s(x, y) = -\log p(x, y), \forall x \in X^*, \forall y \in Y^*
\]
Problems due to a joint distribution learning

- Statistical dependence between primitive edit operations

\[ \sum_{a \in X \cup \{\lambda\}, b \in Y \cup \{\lambda\}} c(a, b) = 1, \]

- Statistical dependence on the input string distribution \( p(x) \)

\[ p(x) = \sum_{y \in Y^*} p(x, y), \]

- Biased estimates of \( p(y|x) = \frac{p(x,y)}{p(x)} \), useful for applications.
Experiments with Ristad and Yanilos’s approach

<table>
<thead>
<tr>
<th>$c^*(a, b)$</th>
<th>$\lambda$</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$c^*(a)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.00</td>
<td>0.05</td>
<td>0.08</td>
<td>0.02</td>
<td>0.02</td>
<td>0.17</td>
</tr>
<tr>
<td>$a$</td>
<td>0.01</td>
<td>0.04</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.08</td>
</tr>
<tr>
<td>$b$</td>
<td>0.02</td>
<td>0.01</td>
<td>0.16</td>
<td>0.04</td>
<td>0.01</td>
<td>0.24</td>
</tr>
<tr>
<td>$c$</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
<td>0.15</td>
<td>0.00</td>
<td>0.19</td>
</tr>
<tr>
<td>$d$</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.28</td>
<td>0.32</td>
</tr>
</tbody>
</table>

Table 1: Target distribution

Figure 1: Input Distribution
Experiments with Ristad and Yanilos’s approach

We computed an average difference between the target and the learned distributions, defined as follows:

\[
d(c, c^*) = \frac{\sum_{a \in X \cup \{\lambda\}} \sum_{b \in Y \cup \{\lambda\}} |c(a, b) - c^*(a, b)|}{2}
\]
Experiments with Ristad and Yanilos’s approach

Distance between theoretical and learnt transducers

Number of Strings

- a (0.08) b (0.24) c (0.19) d (0.32) # (0.17)
- a (0.06) b (0.28) c (0.03) d (0.32) # (0.31)
- a (0.07) b (0.28) c (0.01) d (0.40) # (0.24)
- a (0.12) b (0.15) c (0.26) d (0.14) # (0.32)
- a (0.28) b (0.23) c (0.21) d (0.02) # (0.32)
Learning parameters of a conditional memoryless transducer (CMT)
Learning parameters of a CMT

**Definition 1** A conditional memoryless transducer is denoted by a tuple \((X, Y, c, \gamma)\) where:

- **\(X\)** is the input alphabet,
- **\(Y\)** is the output alphabet,
- **\(c\)** is the primitive conditional probability function \(c : E \rightarrow [0, 1]\), where \(E = E_s \cup E_d \cup E_i\) is the alphabet of primitive edit operations, \(E_s = X \times Y\), is the set of substitutions, \(E_d = X \times \{\lambda\}\) is the set of deletions, \(E_i = \{\lambda\} \times Y\) is the set of insertions.
- **\(\gamma\)** is the probability of the termination symbol of a string (also noted \(c(\lambda|\lambda)\)).
Learning parameters of a CMT

The probability $p(y|x)$ can be recursively computed by means:

- of a **forward function** $\alpha : X^* \times Y^* \to \mathbb{R}^+$ as:

  $$\alpha(y|x) = [1]_{x=\lambda \land y=\lambda} + [c(b|a) \cdot \alpha(y'|x')]_{x=x'\land y=y'}$$
  $$+ [c(\lambda|a) \cdot \alpha(y|x')]_{x=x'\land y=\lambda} + [c(b|\lambda) \cdot \alpha(y'|x')]_{y=y'}. $$

where $[f(x)]_{\pi(x,\ldots)} = f(x)$ if the predicate $\pi(x,\ldots)$ holds and 0 otherwise. Then,

$$\rightarrow p(y|x) = \alpha(y|x)\gamma$$
Learning parameters of a CMT

- of a backward function $\beta : X^* \times Y^* \rightarrow \mathbb{R}^+$ as:

$$\beta(y|x) = [1]_{x=\lambda \land y=\lambda} + [c(b|a) \cdot \beta(y'|x')]_{x=ax' \land y=by'}$$

$$+ [c(\lambda|a) \cdot \beta(y|x')]_{x=ax'} + [c(b|\lambda) \cdot \beta(y'|x)]_{y=by'}.$$

Then,

$$\rightarrow p(y|x) = \beta(y|x)\gamma.$$

In both cases (forward or backward), we have to learn the costs $c(b|a) \ \forall b \in Y \cup \{\lambda\}, \ a \in Y \cup \{\lambda\}.$
Learning parameters of a CMT with the EM algorithm

EM works on the principle that the corpus likelihood can be maximized subject to some maximization constraints on the parameters. In our case:

1. $\gamma > 0$, $c(b|a)$, $c(b|\lambda)$, $c(\lambda|a) \geq 0$ $\forall a \in X, b \in Y$

2. $\sum_{b \in Y} c(b|\lambda) + \sum_{b \in Y} c(b|a) + c(\lambda|a) = 1$ $\forall a \in X$

3. $\sum_{b \in Y} c(b|\lambda) + \gamma = 1$
Learning parameters of a CMT with the EM algorithm

• The **Expectation** step:

\[
\delta(b|a) = \sum_{(xax', yby') \in S} \frac{\alpha(y|x)c(b|a)\beta(y'|x')\gamma}{p(yby'|xax')}
\]

\[
\delta(b|\lambda) = \sum_{(xx', yby') \in S} \frac{\alpha(y|x)c(b|\lambda)\beta(y'|x')\gamma}{p(yby'|xx')}
\]

\[
\delta(\lambda|a) = \sum_{(xax', yy') \in S} \frac{\alpha(y|x)c(\lambda|a)\beta(y'|x')\gamma}{p(yy'|xax')}
\]

\[
\delta(\lambda|\lambda) = \sum_{(x,y) \in S} \frac{\alpha(y|x)\gamma}{p(y|x)} = |S|.
\]
Learning parameters of a CMT with the EM algorithm

- **The Maximization step:**

  (we begin by normalizing the insertion $c(b|\lambda)$ because it appears in constraints 2 and 3).

  $$c(b|\lambda) = \frac{\delta(b|\lambda)}{N}$$

  where

  $$N = \sum_{a \in X \cup \{\lambda\}} \delta(b|a)$$

  $$\sum_{b \in Y \cup \{\lambda\}}$$
Learning parameters of a CMT with the EM algorithm

- **The Maximization step:**
  From constraint 3, we deduce then:

  \[
  \gamma = \frac{N - N(\lambda)}{N}
  \]

  where

  \[
  N(\lambda) = \sum_{b \in Y} \delta(b|\lambda)
  \]
Learning parameters of a CMT with the EM algorithm

- The **Maximization** step:

\[ c(b|a) = \frac{\delta(b|a)}{N(a)} \frac{N - N(\lambda)}{N} \quad c(\lambda|a) = \frac{\delta(\lambda|a)}{N(a)} \frac{N - N(\lambda)}{N} \]

where

\[ N(a) = \sum_{b \in \mathcal{Y} \cup \{\lambda\}} \delta(b|a). \]
Experiments
Experiments (I): Convergence Results

Distance between theoretical and learnt transducers vs Number of Strings

- a (0.10) b (0.17) c (0.26) d (0.20) # (0.28)
- a (0.09) b (0.29) c (0.04) d (0.27) # (0.31)
- a (0.15) b (0.18) c (0.32) d (0.15) # (0.20)
- a (0.23) b (0.32) c (0.13) d (0.05) # (0.27)
- a (0.18) b (0.23) c (0.16) d (0.15) # (0.28)
Experiments in Grammatical Inference (II)

- Learning Sample
- Noisy Input Data
- Oracle
- Noise free pairs of strings
- Stochastic Transducer

**PDFA 1**
- "noisy"
- Test Set with Noisy Inputs
- Correction
- Perplexity

**PDFA 2**
- "corrected"
- Test Set with Corrected Inputs
- Perplexity

**PDFA 3**
- "Oracle"
- Unnoisy Inputs
- ALERGIA

**Perplexity**
Experiments in Grammatical Inference (II)

\[ X = Y = \{a, b, c, e, f, g\}, |LS| = 1000, |TS| = 1000 \]
Conclusions and Perspectives

- We overcame the drawback of state of the art methods by directly learning a conditional distribution over the edit operations.

- Current series of experiments on handwritten digits.

- We plan to extend this model to:
  1. non memoryless transducers (but breaking the link with the edit distance?)
  2. the learning of a stochastic edit distance over trees (PASCAL pump-priming proposal)