Information Extraction by Grammatical Inference

Gunter Grieser
FG Intellektik, FB Informatik
TU Darmstadt
Overview

- Information extraction
- wrappers
  - island wrappers
- representation language
  - EFS, AEFS
  - representability
- learning
  - learning models LIM and PAC
  - learning of AEFS, of island wrappers, and of the subtasks
Computers: from toolboxes to assistants

Computer as tool

- artificial communication
- machine logic
- no world knowledge, no context

does what I say

Computer as assistant

- my communication style
- thinking amplifier
- context, world knowledge

does what I mean
Consider information in a web page

Information Extraction by Grammatical Inference
G. Grieser
How to extract information from such documents?

there is some growing interest in powerful information extraction procedures, e.g.

  to allow for an explicit access to information that is hidden in various documents (knowledge management)

as a result thereof, there is some growing need for techniques that allow for an 'interactive' creation of powerful information extraction procedures
Information Extraction by Grammatical Inference

G. Grieser

Information Extraction Wrappers

Example of a more adequate communication
<table>
<thead>
<tr>
<th>Company Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>L' Air Liquide GmbH</td>
<td>Düsseldorf</td>
</tr>
<tr>
<td>Messer Griesheim GmbH</td>
<td>Krefeld</td>
</tr>
<tr>
<td>Tyczka Industrie-Gase GmbH</td>
<td></td>
</tr>
</tbody>
</table>
Information Extraction by Grammatical Inference
G. Grieser

IE and formal languages

- documents are strings over a certain alphabet
- information is contained in the documents
- can view documents as well as contained information as formal languages
Often, information can be identified by its context.
Information Extraction by Grammatical Inference
G. Grieser

**IE and formal languages**

- documents are strings over a certain alphabet
- information is contained in the documents

- can view documents as well as contained information as well as context as formal languages
Information Extraction by Grammatical Inference
G. Grieser

**Island Wrappers**

In general: delimiters not unique
⇒ delimiter languages

n: arity of the island wrapper
⇒ 2n delimiter languages: \( L_1, R_1, \ldots, L_n, R_n \)

*island wrapper*: 2n-tuple of formal languages
\[
(L_1, R_1, \ldots, L_n, R_n)
\]
an island wrapper \((L_1,R_1,\ldots,L_n,R_n)\) extracts a tuple \((v_1,v_2,\ldots,v_n)\) from document \(d\) iff:

- \(d = x_1l_1v_1r_1x_2l_2v_2r_2x_3\ldots x_nl_nv_nr_nx_{n+1}\)
- \(x_1 \in \Sigma^*\) \(x_{n+1} \in \Sigma^*\)
- \(l_1 \in L_1\) \(r_1 \in R_1\) \(l_2 \in L_2\) \(r_2 \in R_2\) \ldots \(l_n \in L_n\) \(r_n \in R_n\)
- \(v_1 \in \Sigma^+/(\Sigma^*R_1\Sigma^*)\) \ldots \(v_n \in \Sigma^+/(\Sigma^*R_n\Sigma^*)\)
an island wrapper \((L_1, R_1, \ldots, L_n, R_n)\) extracts a tuple 
\((v_1, v_2, \ldots, v_n)\) from document \(d\) iff:

- \(d = x_1l_1v_1r_1x_2l_2v_2r_2x_3\ldots x_nl_nv_nr_nx_{n+1}\)
- \(x_1 \in \Sigma^* \quad x_{n+1} \in \Sigma^*\)
- \(l_1 \in L_1 \quad r_1 \in R_1 \quad l_2 \in L_2 \quad r_2 \in R_2 \quad \ldots \quad l_n \in L_n \quad r_n \in R_n\)
- \(v_1 \in \Sigma^+\backslash(\Sigma^*R_1\Sigma^*) \quad \ldots \quad v_n \in \Sigma^+\backslash(\Sigma^*R_n\Sigma^*)\)
- \(x_2 \in \Sigma^\backslash(\Sigma^*L_2\Sigma^*) \quad \ldots \quad x_n \in \Sigma^\backslash(\Sigma^*L_n\Sigma^*)\)
How to represent such wrappers?
Elementary formal systems (EFS)

- $\Sigma = \{a, b\}$ ... characters
- $\Pi = \{p\}$ ... predicate symbols
- $\Xi = \{X\}$ ... variables
- patterns like $baX, aX, a$
- atoms like $p(baX), p(aX), p(a)$
- rules like $p(baX) :- p(aX), p(a)$.

- EFS $S = (\Sigma, \Pi, \Gamma)$, where $\Gamma$ is a set of rules
Information Extraction by Grammatical Inference
G. Grieser

**EFS Semantics**

- relies on a well-known idea from logic programming; i.e., we focus our attention on ground atoms (g.a.)
  - for an EFS $S = (\Sigma, \Pi, \Gamma)$, we let
    $$\text{Sem}(S) = \{ \text{g.a.} | \text{g.a. holds in all Herbrand models for } S \}$$

- characterizations of $\text{Sem}(S)$
  - $\text{Sem}(S) = \{ \text{g.a.} | \text{g.a. holds in the least Herbrand model for } S \}$
  - thus, it suffices to enumerate the g.a. that hold in a distinguished model (using a simple operator, starting with the empty set)
Advanced elementary formal systems (AEFS)

- characters, variables, patterns, atoms ... as for EFS
- rules as for EFS and, additionally, rules like \( q(X) :- \text{not } p(X) \).
- AEFS \( S = (\Sigma, \Pi, \Gamma) \), where \( \Gamma \) is a set of rules that meet particular syntactical constraints

Why syntactical constraints at all?
if negation is allowed for, there is generally no least Herbrand model and, thus, the idea to enumerate the ground facts that hold in a distinguished model doesn't work
similarly as before, for an AEFS $S = (\Sigma, \Pi, \Gamma)$, we let
\[ \text{Sem}(S) = \{ \text{g.a.} \mid \text{g.a. holds in all Herbrand models for } S \} \]

the introduced syntactical constraints on the rules in $\Gamma$
guarantee that we obtain the same characterizations of
\[ \text{Sem}(S) = \{ \text{g.a.} \mid \text{g.a. holds in the least Herbrand model} \} \]
Let an AEFS $S = (\Sigma, \Pi, \Gamma)$ and some distinguished predicate symbol $p$ from $\Pi$ be fixed, then

$$L(S,p) = \{ w \in \Sigma^+ \mid (w) \in \text{Sem}(S) \}$$
Variable-bounded EFS/AEFS

examples:

\[
\begin{align*}
q(X) & : - \text{not } p(X) . \\
p(XY) & : - p(X) . \\
p(aa) . \\
\end{align*}
\]

counterexamples:

\[
\begin{align*}
p(XY) & : - p(X) , q(Y,Z) .
\end{align*}
\]

- every variable in the body of a rule has to appear in the head, as well

Theorem:

\[ L \in L(\text{vb-EFS}) \iff^* L \text{ is a r.e. language.} \]

Theorem:

There are \( L \in L(\text{vb-AEFS}) \) that are not r.e.
Length-bounded EFS/AES

examples:

\[
q(X) :- \text{not } p(X).
p(XY) :- p(X).
p(aa).
\]

counterexamples:

\[
p(XY) :- p(X), q(Y,Y).
\]

- variable-bounded
- if some X appears k times in the body of a rule, it must occur at least k times in its head

Theorem:

\[
L \in L(\text{lb-EFS}) \text{ iff } L \text{ is context-sensitive.}
\]

Theorem:

\[
L \in L(\text{lb-AEFS}) \text{ iff } L \text{ is context-sensitive.}
\]
Regular EFS/AEFS

examples:
\[
q(X) :- \text{not} \ p(X).
p(XY) :- p(X).
p(aa).
\]

counterexamples:
\[
p(XYY) :- p(X).
p(XY) :- q(X,Y).
\]

- length-bounded
- only unary predicate symbols
- only regular patterns in the head of a rule

Theorem:
\[ L \in L(\text{reg-EFS}) \text{ iff } L \text{ is context-free.} \]

Theorem:
There are \( L \in L(\text{reg-AEFS}) \text{ that are not context-free.} \)
Closedness properties

Theorem: The AEFS definable language classes $L(\text{reg-AEFS})$, $L(\text{lb-AEFS})$, and $L(\text{vb-AEFS})$ are closed under the operation union, intersection, and complement.
Representing island wrappers as AEFS

\[
\text{extract} (V_1, V_2, X_1 L_1 V_1 R_1 X_2 L_2 V_2 R_2 X_3) :\)
\[\begin{align*}
l_1 (L_1), r_1 (R_1), l_2 (L_2), r_2 (R_2), \\
\text{nc-} r_1 (V_1), \text{nc-} r_2 (V_2), \text{nc-} l_2 (X_2). \\
\end{align*}\]

\text{nc-} r_1 (X) :\) not \(c- r_1 (X).\)
\[\begin{align*}
c- r_1 (X) & :\) r_1 (X). \\
c- r_1 (XY) & :\) c- r_1 (X). \\
c- r_1 (XY) & :\) c- r_1 (Y). \\
\text{nc-} r_2 (X) & :\) analogously \\
\text{nc-} l_2 (X) & :\) analogously \\
\end{align*}\]

\[\begin{align*}
l_1 (X), r_1 (X), l_2 (X), r_2 (X) & \text{ freely definable} \\
\end{align*}\]

Information Extraction  Wrappers  AEFS  Learning
Information Extraction by Grammatical Inference
G. Grieser

Learning
Information Extraction by Grammatical Inference
G. Grieser

Interaction in LExIKON

select a document d

mark missing tuple(s) in d and/or reject extracted tuple(s)

select other document d

stop learning process save wrapper w

Induction
generate wrapper w

Extraction
apply wrapper w to d

Query User
show all extracted tuples (ask whether these are correct)

User

Information Extraction Wrappers AEFS Learning
• When is this interaction cycle successful?
  → Learning

• 2 different models
  • learning in the limit
  • PAC learning

• learnability results for
  • representation language (AEFS)
  • island wrappers
  • composite learning tasks
Learning in the limit

- **learning goal**
  - a finite description of a target language $L$
- **information available about a target language $L$**
  - learning from positive data (text)
    - sequence of words exhausting $L$
  - learning from positive and negative data (informant)
    - sequence of labelled words that exhausts $\Sigma^+$; the words are labelled by `+´ and `-´ according to their membership in $L$
- **IIM**
  - receives as input finite segments of a text (an informant) and outputs a hypothesis about the target language
  - learns $L$ in the limit iff, on every text/informant, the sequence of hypotheses stabilizes on a correct description of the target language $L$
### Results

LimInf/LimTxt: set of all languages learnable from Informant/Text

<table>
<thead>
<tr>
<th>Theorem:</th>
<th>L(lb-EFS) ∈ LimInf</th>
<th>Theorem:</th>
<th>L(lb-AEFS) ∈ LimInf</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>L(lb-EFS) ∉ LimTxt</td>
<td>(i)</td>
<td>L(lb-AEFS) ∉ LimTxt</td>
</tr>
<tr>
<td>(ii)</td>
<td>L(lb-EFS(k)) ∈ LimTxt for k ∈ N</td>
<td>(ii)</td>
<td>L(lb-AEFS(1)) ∈ LimTxt</td>
</tr>
<tr>
<td>(iii)</td>
<td>L(lb-AEFS(k)) ∉ LimTxt for all k &gt; 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Information Extraction Wrappers AEFS Learning
Learning island wrappers

- remember:

- available information / examples:

- task: learn delimiter languages $L_1, R_1, ..., L_n, R_n$ from examples of form

  \[ \Sigma^* L_1 \{\#\} \Sigma R_1 \{\#\} R_1 \Sigma L_2 \Sigma L_2 ... L_n \{\#\} \Sigma R_n \{\#\} R_n \Sigma^* \]

  where $\Sigma_L = \Sigma^* \setminus (\Sigma^* \Sigma \Sigma^*)$
Results

IW(L): set of all island wrappers with delimiter languages from L

Theorem:
IW(∅(Σ*)) ∈ LimInf

Theorem:
IW(∅(Σ*)) ∉ LimTxt

Theorem:
IW(∅(Σ^k)) ∈ LimTxt for k ∈ N
Subtasks when learning island wrappers

- problem A: learn $L_1$ from $\Sigma^*L_1$
- problem B: learn $R_n$ from $\Sigma_{R_n}\{\#\}R_n\Sigma^*$
- problem C: learn $R_m$ and $L_{m+1}$ from $\Sigma_{R_m}\{\#\}R_m\Sigma_{L_{m+1}}L_{m+1}$
- problem D: learn delimiter languages from standard information (reference problem)
Information Extraction by Grammatical Inference
G. Grieser

Results

Theorem:
The learning problems A, B, C, and D are incomparable.
Information Extraction by Grammatical Inference
G. Grieser

\[ \Sigma = \{a, b, c\} \]
\[ L_0 = \{a^m b \mid m \geq 1\} \cup \{c\} \]
\[ L_{n+1} = \{a^m b \mid 1 \leq m \leq n+1\} \cup \{c, ca\} \]

**problem A (learn \( L \) from \( \Sigma^*L \)) solvable**

\[ M: \text{on input } w_0, \ldots, w_m \text{ check whether some string ends with } a. \text{ If no such string occurs, output a description for } \Sigma^*L_0, \text{ otherwise for } \Sigma^*L_1 \]

**problem B (learn \( R \) from \( \Sigma_R \{\#\}R\Sigma^* \)) not solvable**
PAC learning

- learning goal
  - finite description that approximates L sufficiently well
- learning algorithm
  - receives a finite set of positive and negative examples and computes a hypothesis about the target language L
- C is polynomial-time PAC-learnable iff there exists a learning algorithm A such that given $\epsilon, \delta \in [0,1]$, $n \in \mathbb{N}$, and any probability distribution $\Pr$ over $\Sigma^n$
  - A takes $q(1/\epsilon, 1/\delta, n, s)$ examples randomly generated with respect to $\Pr$ and outputs, in polynomial time, a hypothesis $h$ such that, with probability $1 - \delta$, $\Pr(w \in L \Delta h) < \epsilon$
  - here, $s$ denotes the size of the smallest description of L
Hereditary EFS/AEFS

Examples:

\[
\begin{align*}
q(X) & : - \text{not} \ p(X). \\
p(abXaY) & : - \ p(bX), \ q(Y).
\end{align*}
\]

Counterexamples:

\[
\begin{align*}
p(aXbY) & : - \ p(aaX).
\end{align*}
\]

- Every pattern in the body of a rule is a subword of a pattern in its head.

- \(h-(A)\text{EFS}(m,k,t,r)\) - set of all hereditary \((A)\text{EFS}\) with
  - at most \(m\) rules
  - at most \(k\) variables occurrences in head of every rule
  - at most \(t\) atoms in the body of every rule
  - arity of each predicate symbol at most \(r\)

Information Extraction Wrappers AEFS Learning
Theorem:
For all $m,k,t,r \in \mathbb{N}$, $L(h\text{-EFS}(m,k,t,r))$ is polynomial time PAC learnable.

Theorem:
For all $m,k,t,r \in \mathbb{N}$, $L(h\text{-AEFS}(m,k,t,r))$ is polynomial time PAC learnable.

Note, that already $L(h\text{-AEFS}(2,1,1,1)) \setminus L(h\text{-EFS}) \neq \emptyset$.

Corollary:
If $L$ is polynomial time PAC learnable then also $I\text{W}(L)$ is polynomial time PAC learnable.
Overview

- Information extraction
- wrappers
  - island wrappers
- representation language
  - EFS, AEFS
  - representability
- learning
  - learning models LIM and PAC
  - learning of AEFS, of island wrappers, and of the subtasks