String Extension Learning

Jon Rawski
LIN 962: Learnability

October 27, 2017

1 General Argument of the Paper

- So far we have looked at a couple subregular languages and seen that they are all *Gold-Learnable*, e.g. learnable in the limit from positive data
- These learning results can be unified by defining the grammars as *String Extension Grammars*, and the learning as *String Extension Learning*
- SEL has the properties of being *incremental, globally consistent, locally conservative, and set-driven*
- Unified accounts help us to formally decscribe processes in language and cognition

2 String Extension

2.1 Some New Learning Properties

We want our learner ϕ of language class $\mathcal{L}$ to have certain properties:

1. **globally consistent**: for all *i* and for all texts *t* for some $L \in \mathcal{L}$, $content(t[i]) \subseteq L(\phi(t[i]))$
2. **locally conservative**: when $\phi(t[i]) \neq \phi(t[i-1])$, we know $t(i) \notin L(\phi(t[i-1]))$
3. **set-driven**: doesn't depend on text order

2.2 String Extension Grammars

- **String extension functions** map strings to a finite subset of some set A (a partition!)
- **String Extension Grammars** are finite subsets of a set A.
- The strings generated by a String Extension Grammar form a set called a String Extension Language
- An element of a SEG is **useful** if it helps define the language. if not, it is useless).

2.3 String Extension Learning

The formal notions, theorems, and proofs are in the paper, but given the definitions above, the learning is quite simple, and we have seen it before. Initially, the learner hypothesizes the empty grammar. Given an observation, the learner applies the function f to it and combines the set with the previous one via set union.

Because this learner has the properties from section 2.1, each individual string in the text reveals an aspect of the grammar G. The idea of **usefulness** also comes in handy here. The learner is guaranteed to see a word in L and apply the function f to it, and get parts of the grammar G. Since there are only finitely many **useful** elements of G, there's some point where each element of the grammar is seen and added. So SEL learns in the limit! Neat!

2.4 Formal Language Examples

With lots of new definitions and a jungle of theorems and proofs, visualizing can be difficult. Let's see how SEL works on some languages that we already know!

Many Subregular language classes (**Strictly k-Piecewise, K-Piecewise testable, strictly k-Local, and locally k-testable**) fit the description of SEL, and have corresponding string extension learners. Let's look at some examples.

2.4.1 Strictly 2-piecewise

i	$t(i)$	$SP_2(t(i))$	Grammar G	Language of G
-1			$\emptyset$	$\emptyset$
0	$aaaa$	$\{\lambda, a, aa\}$	{λ, **a, aa**}	a^*
1	aab	$\{\lambda, a, b, aa, ab\}$	{λ, a, aa, **b, ab**}	$a^* \cup a^*b$
2	baa	$\{\lambda, a, b, aa, ba\}$	{λ, a, b, aa, ab, **ba**}	$\Sigma^*\backslash(\Sigma^*b\Sigma^*b\Sigma^*)$
3	aba	$\{\lambda, a, b, ab, ba\}$	{λ, a, b, aa, ab, ba}	$\Sigma^*\backslash(\Sigma^*b\Sigma^*b\Sigma^*)$
...				

2.4.2 2-factor

i	$t(i)$	$fac_2(t(i))$	Grammar G	$L(G)$
-1			$\emptyset$	$\emptyset$
0	$aaaa$	$\{aa\}$	{**aa**}	aaa^*
1	aab	$\{aa,\ ab\}$	$\{aa,$ **ab**$\}$	$aaa^* \cup aaa^*b$
2	a	$\{a\}$	{**a**, $aa,\ ab$}	$aa^* \cup aa^*b$
...				

2.4.3 Parikh Mapping

Def: A word is well-formed iff the number of a's in the word belongs to some finite set of numbers

i	$t(i)$	$f_a(t(i))$	Grammar G	$L(G)$
0	$aaaa$	$\{4\}$	{**4**}	B_4
1	$bbabbba$	$\{2\}$	$\{4,$ **2**$\}$	$B_4 \cup B_2$
2	$bbbaa$	$\{2\}$	$\{4,\ 2\}$	$B_4 \cup B_2$
3	$aaab^{100}$	$\{3\}$	$\{4,\ 2,$ **3**$\}$	$B_4 \cup B_2 \cup B_3$
...				

3 Relevance to Language and Cognition

- We have seen many patterns in human languages can be classified as patterns which exhibit string extension properties
  - Phonotactics over strings
  - syntactic dependencies over trees
- If we can characterize their grammars and languages as String extensions, then we might have a unified grammar-independent learning algorithm