Symbolic learning of regular transformations

MPRI internship proposed by Peter Habermehl
University Paris Diderot, IRIF, CNRS

January 25, 2019

Designing algorithms which are capable of learning classifiers from examples is a fundamental problem in computer science. A regular language given by a finite-state automaton can be seen as a classifier of words over a finite alphabet. Different scenarios for learning regular languages have been considered.

In the **passive** learning scenario a learner is provided with examples from a regular language $L$ and she has to infer this language from them. The examples can be either only positive (words in $L$) or positive and negative (words not in $L$). One well-known algorithm in this scenario is RPNI [OG92] which works by constructing an automaton accepting exactly the provided positive examples and then merging states to generalise. Along the way, possible generalisations are rejected if they make the automaton accept negative examples.

In the **active** learning scenario introduced by Angluin [Ang87] the learner can actively ask questions to an all-knowing oracle (also called teacher). The queries allowed are membership queries (is a word in the language ?) and equivalence queries (is a given learning hypothesis equivalent to the language to be learnt ?). Both the passive and active scenarios are useful in practice. In a practical setting, an oracle with the mentioned capabilities is typically not available. However, approximations of the oracle might be (for example via testing of a system). Angluin has proposed an active learning algorithm called $L^*$ which can learn with a polynomial number of queries a regular language. A crucial property of regular languages is that each language has a canonical representation as a minimal deterministic finite-state automaton. Therefore, the learner is provided with a unique (up to isomorphism) learning target which simplifies the problem.

Regular transformations [EH01] are a generalisation of regular languages to transformations from strings to strings. We consider here (partial) functions which map a string to one string. Regular transformations can be defined via deterministic two-way finite-state transducers which can read the input word in both directions and produce output (or not) at each step. There are two equivalent models (streaming string transducers [AC11] and transformations defined via MSO Logic on strings).

We are interested in extending the passive and active learning algorithms to this setting. However, no nice canonical representation of regular transformations exists. For example the transformation which transforms $ab$ into $bbc$
could start by producing $bbc$ while reading $a$ or produce it while reading $b$, etc. Recently, it has been shown in [Boj14] that if one equips transducers with origin information, that is for each output symbol one knows at which point in the reading of the input it was produced, one obtains a canonical characterisation of the regular transformation (based on regular languages). This, theoretically, allows to learn it by using $L^*$. However, this characterisation is machine-independent and therefore one does not learn a transducer, but only its characterisation.

1 Proposed work

In this internship, based on the canonical characterisation of regular transformations of [Boj14], the student will

- devise an active learning algorithm for deterministic two-way finite-state transducers with origin information (and/or streaming string transducers)
- devise a passive learning algorithm for deterministic two-way finite-state transducers with origin information (and/or streaming string transducers)

References


