Notes on Learning Automata for COMP 330

Ariella Smofsky

October 6, 2021

1 Introduction

The word automata contains "auto", which refers to the concept of self. Automata theory is the theoretical framework used to study the behaviour of self-running processes and systems. Computer scientists use automata as theoretical constructs to analyze how machines run programs and ultimately solve problems. In the field of formal verification automata are used to model processes and guarantee desired behaviour. In machine learning many models are based on automata enriched with probabilities, for example Markov decision processes.

1.1 Learning Automata

Thus far in the course, automata have been used to model languages and prove their regularity. Problems have been of the form given a language, create the DFA (or NFA or ϵ-NFA) that represents this language. The learning automata problem reverses this problem: Given a set of words labelled as belonging or not belonging to the unknown language $\mathcal{L}$, build an automaton representing this unknown language $\mathcal{L}$.

This learning automata problem can be thought of generally as finding the underlying structure from a given set of observations. Learning automata sounds like an awfully popular area of Computer Science: Machine Learning!

1.2 Passive Learning vs. Active Learning

Within Machine Learning, passive learning refers to learning algorithms where all information is provided as an initial batch of data. An example is supervised learning, where a learning algorithm is trained on a fixed and labelled dataset. Interestingly, the task of learning automata was proven to be NP-hard in the passive learning setting.

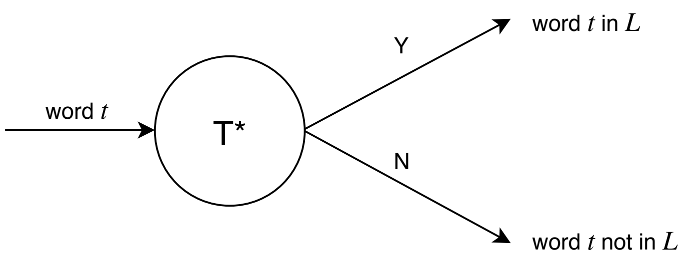
Active learning refers to learning algorithms which have the ability to query their environment during runtime. In the learning automata context, the learning algorithm can query a teacher for information about the unknown language $\mathcal{L}$. The crux of active learning is how to optimally query the environment.

2 Learning DFAs from Queries and Counterexamples

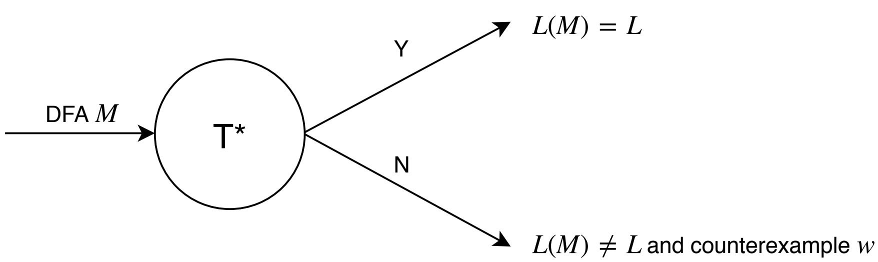
2.1 The Learning Automata Problem

Given access to a minimally adequate teacher T^*, learn a DFA representing an unknown regular language $\mathcal{L}$ from the words that belong, or don't belong, to $\mathcal{L}$. What does the teacher T^* know about $\mathcal{L}$?

Membership Query



Conjecture



2.2 L^* Algorithm

The key to active learning is asking optimal questions. Dana Angluin proposed a learning automata algorithm L^* in her seminal 1987 paper "Learning regular sets from queries and counterexamples". By strategically asking the teacher T^* questions about the unknown language $\mathcal{L}$, the learner L^* efficiently learns a DFA from any T^* in polynomial time. The time complexity is polynomial in m, the max length of any counterexample provided by T^*, and in n, the number of states of the minimum DFA representing $\mathcal{L}$.

3 Observation Tables

An observation table is a structure used by the learning algorithm L^* to store answers received from previous queries and to choose the next optimal query. When L^* terminates, the learning algorithm returns a DFA constructed from the observation table. Below is an example of an observation table over the finite alphabet $\Sigma = \{a, b\}$.

$S \cup S \cdot \Sigma$ \ E	ϵ	a	aa
ϵ			
b			T(baa)=1
a			
ba			
bb	T(bbϵ) = 0		

Initially, $S = E = \{\epsilon\}$. The membership query function maps each cell in the observation table to a 0 or 1 where $T : ((S \cup S \cdot \Sigma) \cdot E) \to \{0, 1\}$ and $T(se) = 1 \iff se \in \mathcal{L}$.

3.1 Construct a DFA from an Observation Table

Again, let $\Sigma = \{a, b\}$. Given a particular unknown language $\mathcal{L}$ assume below is the observation table at a particular moment in the execution of L^*.

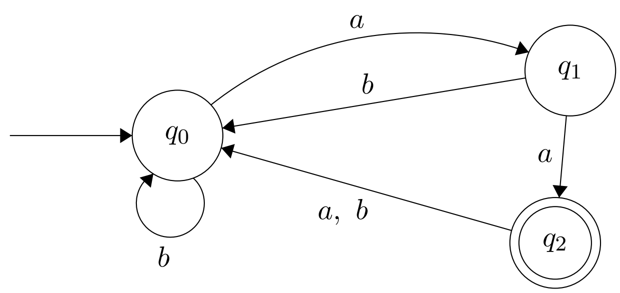
$S \cup S \cdot \Sigma$ \ E	ϵ	a
ϵ	0	0
a	0	1
aa	1	0
b	0	0
ab	0	0
aaa	0	0
aab	0	0

Let $row : (S \cup \; S \cdot \Sigma) \to (E \to \{0, 1\})$ and $row(s) := T(se) \; \forall e \in E$. For example, this observation table yields $row(\epsilon) = 00$ and $row(a) = 01$. Now, let us define a DFA from a given observation table.

$$\begin{aligned} q_0 &= row(\epsilon) \\ Q &= \{row(s) \mid s \in S\} \\ F &= \{row(s) \mid s \in S \text{ and } T(s\epsilon) = 1\} \\ \delta(row(s), x) &= row(sx) \text{ where } x \in \Sigma \end{aligned}$$

We now construct the DFA corresponding to the above observation table.

$row(s)$ \ x	a	b
row(ϵ) = 00 row(a)=01 row(aa)=10	row(ϵa) = 01 row(aa)=10 row(aaa)=00	row(ϵb) = 00 row(ab)=00 row(aab)=00



3.2 Special Properties of the Observation Table

An observation table is closed when

$$\forall t \in S \cdot \Sigma \ \exists s \in S \text{ such that } row(t) = row(s)$$

An observation table is consistent when

$$s_1, s_2 \in S.row(s_1) = row(s_2) \implies \forall x \in \Sigma \ row(s_1 x) = row(s_2 x)$$

4 L^* Algorithm: an Example

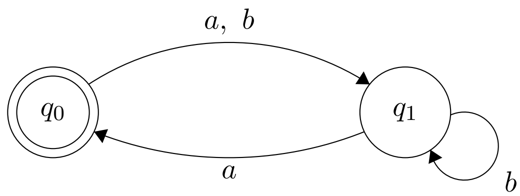
Let $\Sigma = \{a, b\}$ and let the unknown language be $\mathcal{L}$, the set of all words containing an even number of a's and b's. The observation table initialization is

S $\cup S \cdot \Sigma$ \ E	ϵ
ϵ	1
a	0
b	0

Is this observation table closed? No. $row(a) = 0 \neq 1 = row(\epsilon)$. Add a to set S and extend the observation table.

E / $S \cup S \cdot \Sigma$	ϵ
ϵ	1
a	0
b	0
aa	1
ab	0

Is this observation table closed? Yes. Consistent? Yes. L^* is ready to make its first conjecture M_1:



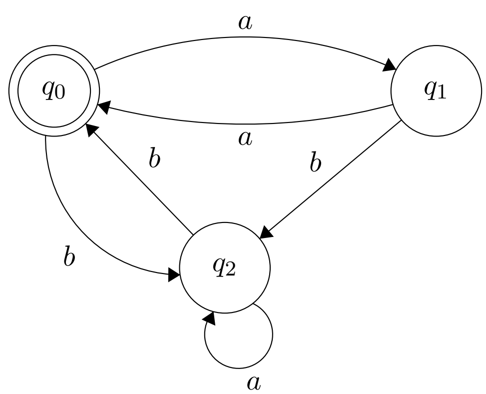
The teacher T^* responds to this conjecture with NO. Assume the counterexample returned is bb. L^* adds the returned counterexample and all of its prefixes to the observation table.

E / $S \cup S \cdot \Sigma$	ϵ
ϵ	1
a	0
b	0
bb	1
aa	1
ab	0
ba	0
bba	0
bbb	0

Is this observation table closed? Yes. Consistent? No. $row(a) = row(b)$ but $row(aa) \neq row(ba)$. Add a to the set E.

E / $S \cup S \cdot \Sigma$	ϵ	a
ϵ	1	0
a	0	1
b	0	0
bb	1	0
aa	1	0
ab	0	0
ba	0	0
bba	0	1
bbb	0	0

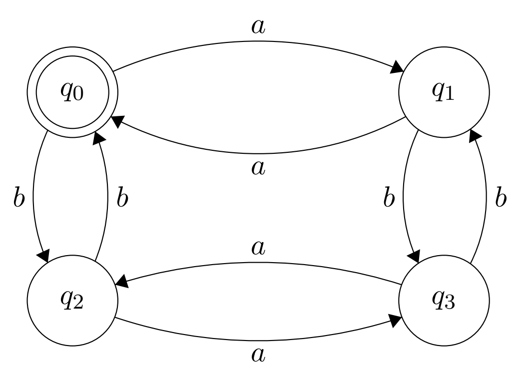
Is this observation table closed? Yes. Consistent? Yes. L^* now makes its second conjecture M_2:



T^* responds to this conjecture with NO. Assume the counterexample returned is abb. L^* adds the returned counterexample and all of its prefixes to the observation table. After noticing this updated observation table is closed but not consistent, update the observation table to yield the following.

$S \cup S \cdot \Sigma$ \ E	ϵ	a	b
ϵ	1	0	0
a	0	1	0
b	0	0	1
bb	1	0	0
ab	0	0	0
abb	0	1	0
aa	1	0	0
ba	0	0	0
bba	0	1	0
bbb	0	0	1
abba	1	0	0
abbb	0	0	0
aba	0	0	1

L^* proposes a third conjecture M_3:



Finally, T^* returns YES and L^* terminates with M_3 as the DFA expressing the regular language $\mathcal{L}$, the set of all words containing an even number of a's and b's.

5 L^* Algorithm

Algorithm 1 L* Algorithm

$S, E \leftarrow \{\epsilon\}$ and extend observation table
repeat
  while (S, E) not closed or not consistent **do**
    if (S, E) not closed **then**
      find $s_1 \in S$ and $x \in \Sigma$ such that $row(s_1 x) \neq row(s) \forall s \in S$
      $S \leftarrow S \cup \{s_1 x\}$ and extend observation table
    if (S, E) not consistent **then**
      find $s_1, s_2 \in S, x \in \Sigma, e \in E$ such that $row(s_1) = row(s_2)$
      and $T(s_1 xe) \neq T(s_2 xe)$
      $E \leftarrow E \cup \{xe\}$ and extend observation table
  make conjecture $M = M(S, E)$
  if Teacher replies NO with counterexample t **then**
    $S \leftarrow S \cup prefixes(t)$ and extend observation table
until Teacher replies YES to conjecture $M = M(S, E)$
return M