An L* Algorithm for Deterministic Weighted Regular Languages

Clemente Pasti **Talu Karagöz** **Anej Svete**
Franz Nowak **Reda Boumasmoud** **Ryan Cotterell**
{clemente.pasti, anej.svete, franz.nowak, ryan.cotterell}@inf.ethz.ch
reda.boumasmoud@math.ethz.ch, talukaragoz@gmail.com

ETH zürich

Abstract

Extracting finite state automata (FSAs) from black-box models offers a powerful approach to gaining interpretable insights into complex model behaviors. To support this pursuit, we present a weighted variant of Angluin's (1987) $\mathbf{L}^*$ algorithm for learning FSAs. We stay faithful to the original algorithm, devising a way to exactly learn deterministic weighted FSAs whose weights support division. Furthermore, we formulate the learning process in a manner that highlights the connection with FSA minimization, showing how $\mathbf{L}^*$ directly learns a minimal automaton for the target language.

github.com/rycolab/weighted-angluin

1 Introduction

Learning formal languages from data is a classic problem in computer science. Unfortunately, learning only from positive examples is impossible (Gold, 1978). By granting the learner access to more than just positive examples, Angluin (1987) introduced the *active* learning scheme $\mathbf{L}^*$, where the learner interacts with an oracle by asking it queries. Concretely, Angluin's (1987) $\mathbf{L}^*$ algorithm learns regular languages in the form of deterministic finite-state automata (DFSAs) from *membership* queries (analogous to asking for a ground truth label of a string in the training dataset) and *equivalence* queries (analogous to asking whether a hypothesis is correct).

Weighted formal languages, where strings are assigned weights such as probabilities or costs, naturally generalize membership-based (boolean) formal languages. Weighted languages, especially probabilistic languages, serve as a cornerstone in the conceptual framework of many NLP problems (Mohri, 1997). Their significance is twofold: First, in practical applications, where they underpin algorithms for tasks such as parsing (Goodman, 1996) and machine translation (Mohri, 1997), and second, as an analytical framework for better understanding modern language models (Weiss et al., 2018; Jumelet and Zuidema, 2023; Nowak et al., 2024, *inter alia*). This has motivated the development of various weighted extensions of Angluin's (1987) $\mathbf{L}^*$ algorithm. For instance, Weiss et al. (2019) describes a generalization that (approximately) learns a probabilistic DFSA by querying a neural language model to interpret it. Less faithfully to the original $\mathbf{L}^*$ algorithm, multiple algorithms for learning *non-deterministic* weighted FSAs have been proposed (Bergadano and Varricchio, 1996; Beimel et al., 2000; Balle and Mohri, 2012; Balle et al., 2014; Daviaud and Johnson, 2024; Balle and Mohri, 2015). These algorithms involve the solution of a linear system of equations, and therefore they cannot be used when the underlying algebraic structure lacks subtraction.

We present a novel weighted generalization of the $\mathbf{L}^*$ algorithm that learns *semifield-* weighted *deterministic* FSAs. In contrast to other algorithms inspired by $\mathbf{L}^*$, ours is a more faithful generalization of the original learning scheme. We generalize Angluin's (1987) original algorithm, resulting in a familiar procedure that, just like the original, learns a DFSA exactly in a finite number of steps if the automaton can be determinized.[1] Additionally, we loosen the requirement for field-weighted FSAs; our algorithm works for semifield-weighted FSA. Our exposition further illuminates the connection between weighted minimization (Hopcroft and Ullman, 1979; Mohri, 1997) and $\mathbf{L}^*$.

2 Weighted Regular Languages

Semirings and Semifields. Throughout this paper, we fix a **semifield** $K = (\mathbb{K}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, where $\mathbb{K}$ is a set equipped with two associative laws, $\oplus$ and $\otimes$, along with distinguished elements $\mathbf{0}$ and $\mathbf{1}$, satisfying the following conditions:

1. $(\mathbb{K}, \oplus, \mathbf{0})$ is a **commutative monoid**,
2. $(\mathbb{K}\backslash\{\mathbf{0}\}, \otimes, \mathbf{1})$ is a **group**,

[1] All boolean-weighted FSA can be determinized, which is why Angluin's (1987) $\mathbf{L}^*$ always halts.

3. The law $\otimes$ distributes over $\oplus$, so for all $w_1, w_2, w_3 \in \mathbb{K}$, $(w_1 \oplus w_2) \otimes w_3 = (w_1 \otimes w_3) \oplus (w_2 \otimes w_3)$ and $w_1 \otimes (w_2 \oplus w_3) = (w_1 \otimes w_2) \oplus (w_1 \otimes w_3)$; and
4. $\mathbf{0}$ acts as an annihilator for $\otimes$, meaning $w \otimes \mathbf{0} = \mathbf{0} \otimes w = \mathbf{0}$ for all $w \in \mathbb{K}$.

Strings and Languages. An **alphabet** Σ is a non-empty, finite set of symbols. A **string** is a finite sequence of symbols from an alphabet. We write $\mathbf{xy}$ to denote the concatenation of the strings $\mathbf{x}$ and $\mathbf{y}$. Let $\Sigma^{n+1} \stackrel{\text{def}}{=} \{\mathbf{y}a \mid \mathbf{y} \in \Sigma^n, a \in \Sigma\}$ and $\Sigma^0 \stackrel{\text{def}}{=} \{\varepsilon\}$, where ε is the empty string. The **Kleene closure** $\Sigma^* \stackrel{\text{def}}{=} \bigcup_{n=0}^{\infty} \Sigma^n$ of Σ is the set containing all strings made with symbols of Σ. We further introduce the set $\Sigma^{\leqslant k} = \bigcup_{n=0}^{k} \Sigma^n$. Given an alphabet Σ and a semiring $\langle \mathbb{K}, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$, a **weighted formal language** is a function $\mathbf{L}\colon \Sigma^* \to \mathbb{K}$ that assigns weights $w \in \mathbb{K}$ to strings $\mathbf{y} \in \Sigma^*$. Unless differently specified, in this paper we will assume that all weighted languages are *semifield*-weighted.

Weighted Finite-state Automata. A **weighted finite-state automaton** (WFSA) $\mathcal{A}$ over a semifield $\langle \mathbb{K}, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$ is a 5-tuple $(\Sigma, Q, \delta, \lambda, \rho)$ where Σ is an alphabet, Q is a finite set of states, δ is a set of weighted arcs rendered as $p \xrightarrow{a/w} q$ with $p, q \in Q$, $a \in \Sigma$, and $w \in \mathbb{K}$,[2] and $\lambda\colon Q \to \mathbb{K}$ and $\rho\colon Q \to \mathbb{K}$ are the initial and final weight function, respectively. A **path** $\boldsymbol{\pi}$ in $\mathcal{A}$ is a finite sequence of contiguous arcs, denoted as

$$q_0 \xrightarrow{a_1/w_1} q_1, \cdots, q_{N-1} \xrightarrow{a_N/w_N} q_N. \tag{1}$$

We call $i(\boldsymbol{\pi}) = q_0$ the initial state of the path, and $f(\boldsymbol{\pi}) = q_N$ the final state of the path. The **weight** of $\boldsymbol{\pi}$ is $\boldsymbol{w}(\boldsymbol{\pi}) = w_1 \otimes \cdots \otimes w_N$ and its **yield** is $\sigma(\boldsymbol{\pi}) = a_1 \cdots a_N$. With $\Pi_{\mathcal{A}}$, we denote the set of all paths in $\mathcal{A}$, and with $\Pi_{\mathcal{A}}(\boldsymbol{p})$ the subset of all paths in $\mathcal{A}$ with yield $\boldsymbol{p}$. A state $\boldsymbol{q}$ is **accessible** if there exists a path $\boldsymbol{\pi}$ with $w(\boldsymbol{\pi}) \neq \mathbf{0}$, $\lambda(i(\boldsymbol{\pi})) \neq \mathbf{0}$ and $f(\boldsymbol{\pi}) = \boldsymbol{q}$. It is **coaccessible** if there exists a path $\boldsymbol{\pi}$ with $w(\boldsymbol{\pi}) \neq \mathbf{0}$, $\rho(f(\boldsymbol{\pi})) \neq \mathbf{0}$ and $i(\boldsymbol{\pi}) = \boldsymbol{q}$. A WFSA is **trimmed** if all its states are simultaneously accessible and coaccessible. We say that a WFSA $\mathcal{A} = (\Sigma, Q, \delta, \lambda, \rho)$ is **deterministic** (a WDFSA) if, for every $\boldsymbol{p} \in Q, a \in \Sigma$, there is at most one $\boldsymbol{q} \in Q$ such that $\boldsymbol{p} \xrightarrow{a/w} \boldsymbol{q} \in \delta$ with $w > 0$, and there is a single state q_I with $\lambda(q_I) \neq 0$. In such case, we refer to q_I as the **initial state**. A WDFSA can have at most one path yielding a string $\mathbf{y} \in \Sigma^*$ from the initial state q_I. A WDFSA $\mathcal{A}$ is said to be **minimal** if no equivalent WDFSA with fewer states exists.

Weighted Regular Languages. Every WFSA $\mathcal{A}$ generates the weighted language

$$\mathbf{L}_{\mathcal{A}}(\boldsymbol{p}) \stackrel{\text{def}}{=} \bigoplus_{\boldsymbol{\pi} \in \Pi_{\mathcal{A}}(\boldsymbol{p})} \lambda(i(\boldsymbol{\pi})) \otimes \boldsymbol{w}(\boldsymbol{\pi}) \otimes \rho(f(\boldsymbol{\pi})) \tag{2}$$

for $\boldsymbol{p} \in \Sigma^*$. We define the set $\text{supp}(\mathbf{L}) = \{\boldsymbol{p} \in \Sigma^* \mid \mathbf{L}(\boldsymbol{p}) \neq \mathbf{0}\}$ to be the **support** of $\mathbf{L}$. A weighted language is said to be **regular** if there exists a WFSA that generates it. If two WFSAs generate the same language, they are said to be **equivalent**. Finally, a weighted regular language is said to be **deterministic** if there exists a WDFSA that generates it. In contrast to the boolean case, not every weighted regular language can be generated by a deterministic WFSA (Allauzen and Mohri, 2003). Weighted deterministic regular languages are thus a strict subset of weighted regular languages. This distinction plays a critical role in our exposition—we develop a generalization of Angluin's (1987) algorithm that learns weighted *deterministic* regular languages.

Homothetic equivalence. Let X be a subset of Σ^* and $\langle \mathbb{K}, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$ a semifield. Let us denote with $\mathcal{L}(X)$ the set of functions $X^{\mathbb{K}}$. We introduce the equivalence relation

$$\begin{aligned} \mathbf{L}_1 \equiv_X \mathbf{L}_2 \iff \exists k \in \mathbb{K} \backslash \{\mathbf{0}\} : \\ \mathbf{L}_1(\mathbf{x}) = k \otimes \mathbf{L}_2(\mathbf{x}),\ \forall \mathbf{x} \in X \end{aligned} \tag{3}$$

between any two functions L_1, L_2 in $\mathcal{L}(X)$. We call this relation **homothetic equivalence**.

For every string $\mathbf{x} \in \Sigma^*$, we introduce the **right language** $\mathbf{x}^{-1}\mathbf{L}\colon \mathbf{y} \mapsto \mathbf{L}(\mathbf{xy})$, and define the following equivalence relation on Σ^*:

$$\mathbf{x} \sim_{\mathbf{L}} \mathbf{z} \iff \mathbf{x}^{-1}\mathbf{L} \equiv_{\Sigma^*} \mathbf{z}^{-1}\mathbf{L} \tag{4}$$

3 Empirical Hankel Systems

Hankel matrices. Let $\mathrm{P} \subseteq \Sigma^*$ be a prefix-closed set of prefixes and let $\mathrm{S} \subseteq \Sigma^*$ be a suffix-closed set of suffixes.[3] An **empirical Hankel matrix** is a map $\mathbf{H}\colon \mathrm{P} \circ \Sigma^{\leqslant 1} \times \Sigma^{\leqslant 1} \circ \mathrm{S} \to \mathbb{K}$. For every $\boldsymbol{p} \in \mathrm{P} \circ \Sigma^{\leqslant 1}$ we define the right language map

[2] We do not consider ε-transitions. This is without loss of generality; any regular language can be represented by an ε-free automaton (Mohri, 2009, Theorem 7.1).

[3] A set of strings is prefix-closed (suffix-closed) if it contains all prefixes (suffixes) of each of its elements. In particular $\varepsilon \in \mathrm{P} \cap \mathrm{S}$.

$\mathbf{H}_{\boldsymbol{p}}\colon \Sigma^{\leqslant 1}\circ \mathrm{S} \to \mathbb{K}, \boldsymbol{s} \mapsto \mathbf{H}(\boldsymbol{p}, \boldsymbol{s})$. Using homothetic equivalence (Eq. (4)), we introduce the equivalence relation $\sim_{\mathbf{H}}$ on $\mathrm{P}\circ\Sigma^{\leqslant 1}$ for $\boldsymbol{p}, \boldsymbol{q} \in \mathrm{P}\circ\Sigma^{\leqslant 1}$ as

$$\boldsymbol{p} \sim_{\mathbf{H}} \boldsymbol{q} \iff \mathbf{H}_{\boldsymbol{p}} \equiv_{\Sigma^{\leqslant 1}\circ \mathrm{S}} \mathbf{H}_{\boldsymbol{q}} \tag{5}$$

We denote $\boldsymbol{p}$'s equivalence class by $[\boldsymbol{p}] = \{\boldsymbol{q} \in \mathrm{P}\circ\Sigma^{\leqslant 1} \mid \boldsymbol{q} \sim_{\mathbf{H}} \boldsymbol{p}\}$.

The naïve Hankel automaton. Given an empirical Hankel matrix $\mathbf{H}$, consider the map

$$d_{\mathbf{H}}\colon \mathrm{P}\circ\Sigma^{\leqslant 1} \to \mathbb{K},\ \boldsymbol{p} \mapsto \oplus_{\boldsymbol{s}\in\mathrm{S}}\mathbf{H}(\boldsymbol{p}, \boldsymbol{s}) \tag{6}$$

We introduce it here to streamline the construction of **the naïve Hankel automaton** associated with $\mathbf{H}$; the WFSA[4] $\mathcal{A}_{\mathbf{H}} = (\Sigma, Q_{\mathbf{H}}, \delta_{\mathbf{H}}, \lambda_{\mathbf{H}}, \rho_{\mathbf{H}})$ with:
(1) **States.** We define the states $Q_{\mathbf{H}} \stackrel{\text{def}}{=} \mathrm{P}$.
(2) **Transitions.** For every state $\boldsymbol{p} \in \mathrm{P}$ and every symbol $a \in \Sigma$, let the transition $\boldsymbol{p} \xrightarrow{a/w} \boldsymbol{p}'$ be in $\delta_{\mathbf{H}}$ whenever $\boldsymbol{p}\,a \sim_{\mathbf{H}} \boldsymbol{p}'$ and where

$$w \stackrel{\text{def}}{=} \begin{cases} \frac{d_{\mathbf{H}}(\boldsymbol{p}a)}{d_{\mathbf{H}}(\boldsymbol{p})} & \textbf{if } d_{\mathbf{H}}(\boldsymbol{p}) \neq 0, \\ \mathbf{0} & \textbf{otherwise} \end{cases} \tag{7}$$

(3) **Initial weight.** For every state $\boldsymbol{p} \in \mathrm{P}$, we define its initial weight as

$$\lambda_{\mathbf{H}}(\boldsymbol{p}) \stackrel{\text{def}}{=} \begin{cases} d_{\mathbf{H}}(\varepsilon) & \textbf{if } \boldsymbol{p} = \varepsilon, \\ \mathbf{0} & \textbf{otherwise} \end{cases} \tag{8}$$

(4) **Final weights.** For every state $\boldsymbol{p} \in \mathrm{P}$, we define its final weight

$$\rho_{\mathbf{H}}(\boldsymbol{p}) \stackrel{\text{def}}{=} \begin{cases} \frac{\mathbf{H}(\boldsymbol{p},\varepsilon)}{d_{\mathbf{H}}(\boldsymbol{p})} & \textbf{if } d_{\mathbf{H}}(\boldsymbol{p}) \neq \mathbf{0}, \\ \mathbf{0} & \textbf{otherwise} \end{cases} \tag{9}$$

Empirical Hankel systems.

Definition 1. *An **empirical Hankel system** is a triplet $\overline{\overline{\mathbf{H}}} = (\mathrm{P}, \mathrm{S}, \mathbf{H})$, where $\mathrm{P} \subseteq \Sigma^*$ prefix closed, $\mathrm{S} \subseteq \Sigma^*$ a suffix closed, and $\mathbf{H}\colon \mathrm{P}\circ\Sigma^{\leqslant 1} \times \Sigma^{\leqslant 1}\circ \mathrm{S} \to \mathbb{K}$ is an empirical Hankel matrix that is:*
*(1.) **non-trivial**: $\mathbf{H}_{\boldsymbol{p}} \neq \mathbf{0}$, for all $\boldsymbol{p} \in \mathrm{P}$;*
*(2.) **closed**: for every $\boldsymbol{p} \in \mathrm{P}$ and $a \in \Sigma$ such that $\mathbf{H}_{\boldsymbol{p}a} \neq \mathbf{0}$, there exists $\boldsymbol{q} \in \mathrm{P}$ such that $\boldsymbol{p}\,a \sim_{\mathbf{H}} \boldsymbol{q}$—in particular, $\mathrm{P}\circ\Sigma^{\leqslant 1}/\sim_{\mathbf{H}} = \mathrm{P}/\sim_{\mathbf{H}}$; and*
*(3.) **consistent**: for every $\boldsymbol{p}, \boldsymbol{q} \in \mathrm{P}$*

$$\boldsymbol{p} \sim_{\mathbf{H}} \boldsymbol{q} \Rightarrow \boldsymbol{p}\,a \sim_{\mathbf{H}} \boldsymbol{q}\,a, \quad \forall a \in \Sigma. \tag{10}$$

We define the dimension of an empirical Hankel system to be $\dim(\overline{\overline{\mathbf{H}}}) \stackrel{\text{def}}{=} |\mathrm{P}/\sim_{\mathbf{H}}|$. Given an empirical Hankel matrix $\mathbf{H}\colon \mathrm{P}\circ\Sigma^{\leqslant 1} \times \Sigma^{\leqslant 1}\circ\mathrm{S} \to \mathbb{K}$ and a weighted language $\mathbf{L}$, we say that $\mathbf{L}$ **contains** $\mathbf{H}$ if $\mathbf{H}(\boldsymbol{p}, \boldsymbol{s}) = \mathbf{L}(\boldsymbol{ps})$ for every $\boldsymbol{p} \in \mathrm{P}\circ\Sigma^{\leqslant 1}$ and $\boldsymbol{s} \in \Sigma^{\leqslant 1}\circ\mathrm{S}$. Likewise, we say that a WFSA $\mathcal{A}$ contains $\mathbf{H}$ if the language $\mathbf{L}_{\mathcal{A}}$ contains $\mathbf{H}$.

We define a **partial order** on the set of empirical Hankel systems as follows: given two empirical Hankel systems $\overline{\overline{\mathbf{H}}}_1 = (\mathrm{P}_1, \mathrm{S}_1, \mathbf{H}_1)$ and $\overline{\overline{\mathbf{H}}}_2 = (\mathrm{P}_2, \mathrm{S}_2, \mathbf{H}_2)$, we define $\overline{\overline{\mathbf{H}}}_1 \preceq \overline{\overline{\mathbf{H}}}_2$, if $\mathrm{P}_1 \subset \mathrm{P}_2$, $\mathrm{S}_1 \subset \mathrm{S}_2$ and $\mathbf{H}_1(\boldsymbol{p}, \boldsymbol{s}) = \mathbf{H}_2(\boldsymbol{p}, \boldsymbol{s})$ for any $(\boldsymbol{p}, \boldsymbol{s}) \in \mathrm{P}_1\circ\Sigma^{\leqslant 1} \times \mathrm{S}_1\circ\Sigma^{\leqslant 1}$.

3.1 Minimal Hankel Automaton

Theorem 1 ($\mathcal{A}_{\mathbf{H}}$ is transition-regular). *Let $\overline{\overline{\mathbf{H}}} = (\mathrm{P}, \mathrm{S}, \mathbf{H})$ be an empirical Hankel system. The equivalence relation $\sim_{\mathbf{H}}$ on $\mathcal{A}_{\mathbf{H}}$ is transition-regular (see Def. 2), which means that for every $\boldsymbol{p} \in \mathrm{P}$ and every $a \in \Sigma$:*
1. There exists $\boldsymbol{r} \in \mathrm{P}$ such that $\boldsymbol{p} \xrightarrow{a/w_1} \boldsymbol{r} \in \delta_{\mathbf{H}}$ for some $w_1 \in \mathbb{K}\backslash\{\mathbf{0}\}$.
2. If $\boldsymbol{q} \in \mathrm{P}$ is another prefix with $\boldsymbol{p} \sim_{\mathbf{H}} \boldsymbol{q}$, then:
(a) for all $\boldsymbol{r} \in \mathrm{P}$:

$$\boldsymbol{p} \xrightarrow{a/w_1} \boldsymbol{r} \in \delta_{\mathbf{H}} \iff \boldsymbol{q} \xrightarrow{a/w_2} \boldsymbol{r} \in \delta_{\mathbf{H}}$$

and: (i) $\boldsymbol{r} \sim_{\mathbf{H}} \boldsymbol{p}a \sim_{\mathbf{H}} \boldsymbol{q}a$, (ii) $w_1 = w_2$.
(b) $\lambda_{\mathbf{H}}(\boldsymbol{p}) = \lambda_{\mathbf{H}}(\boldsymbol{q})$ and $\rho_{\mathbf{H}}(\boldsymbol{p}) = \rho_{\mathbf{H}}(\boldsymbol{q})$.

Proof. Fix a prefix $\boldsymbol{p} \in \mathrm{P}$ and a symbol $a \in \Sigma$.

1. Since $\mathbf{H}$ is closed, there exists $\boldsymbol{r} \in \mathrm{P}$ and $k \in \mathbb{K}\backslash\{\mathbf{0}\}$ such that $\mathbf{H}_{\boldsymbol{r}} = k \otimes \mathbf{H}_{\boldsymbol{p}a}$, which implies, by definition of $\mathcal{A}_{\mathbf{H}}$, that $\boldsymbol{p} \xrightarrow{a/w_1} \boldsymbol{r} \in \delta_{\mathbf{H}}$ for some $w_1 \in \mathbb{K}$. Since $\overline{\overline{\mathbf{H}}}$ is non-trivial, $d_{\mathbf{H}}(\boldsymbol{r}) = k \otimes d_{\mathbf{H}}(\boldsymbol{p}a) \neq \mathbf{0}$, and so $w_1 = \frac{d_{\mathbf{H}}(\boldsymbol{p}a)}{d_{\mathbf{H}}(\boldsymbol{p})} \neq \mathbf{0}$.

2. (a.i) By definition of $\mathcal{A}_{\mathbf{H}}$, if $\boldsymbol{p} \xrightarrow{a/w_1} \boldsymbol{r}$ is in $\delta_{\mathbf{H}}$, then $\boldsymbol{p}\,a \sim_{\mathbf{H}} \boldsymbol{r}$ for some $\boldsymbol{r} \in \mathrm{P}$. Now, let $\boldsymbol{q} \in \mathrm{P}$ such that $\boldsymbol{p} \sim_{\mathbf{H}} \boldsymbol{q}$. By consistency of $\mathbf{H}$, we have $\boldsymbol{q}\,a \sim_{\mathbf{H}} \boldsymbol{p}\,a \sim_{\mathbf{H}} \boldsymbol{r}$ and so $\boldsymbol{q} \xrightarrow{a/w_2} \boldsymbol{r} \in \delta_{\mathbf{H}}$ for some $w_2 \in \mathbb{K}$. The reverse follows similarly.

2. (a.ii) Let us show $w_2 = w_1$. By assumption, we know that there exists $k \in \mathbb{K}\backslash\{\mathbf{0}\}$ such that $\mathbf{H}_{\boldsymbol{p}}(\boldsymbol{s}) = k \otimes \mathbf{H}_{\boldsymbol{q}}(\boldsymbol{s})\ \forall \boldsymbol{s} \in \Sigma^{\leqslant 1}\circ\mathrm{S}$. Hence $d_{\mathbf{H}}(\boldsymbol{p}a) = \oplus_{\boldsymbol{s}\in\mathrm{S}}\mathbf{H}_{\boldsymbol{p}}(a\boldsymbol{s}) = k \otimes d_{\mathbf{H}}(\boldsymbol{q}a)$ and $d_{\mathbf{H}}(\boldsymbol{p}) = \oplus_{\boldsymbol{s}\in\mathrm{S}}\mathbf{H}_{\boldsymbol{p}}(\boldsymbol{s}) = k \otimes d_{\mathbf{H}}(\boldsymbol{q})$. Accordingly, $d_{\mathbf{H}}(\boldsymbol{p}) \neq \mathbf{0} \iff d_{\mathbf{H}}(\boldsymbol{q}) \neq \mathbf{0}$ and in which case

$$w_1 = \frac{d_{\mathbf{H}}(\boldsymbol{p}a)}{d_{\mathbf{H}}(\boldsymbol{p})} = \frac{d_{\mathbf{H}}(\boldsymbol{q}a)}{d_{\mathbf{H}}(\boldsymbol{q})} = w_2 \tag{11}$$

2. (b) If $\boldsymbol{p} \sim_{\mathbf{H}} \boldsymbol{q}$, then we have

$$\rho_{\mathbf{H}}(\boldsymbol{p}) = \frac{\mathbf{H}_{\boldsymbol{p}}(\varepsilon)}{d_{\mathbf{H}}(\boldsymbol{p})} = \frac{\cancel{k} \otimes \mathbf{H}_{\boldsymbol{q}}(\varepsilon)}{\cancel{k} \otimes d_{\mathbf{H}}(\boldsymbol{q})} = \rho_{\mathbf{H}}(\boldsymbol{q}) \tag{12}$$

The computation for $\lambda_{\mathbf{H}}(\boldsymbol{p})$ follows similarly. ■

[4]This automaton is *not* necessarily determinisitc.

Theorem 2 (The empirical Hankel Automaton $\widetilde{\mathcal{A}}_{\mathbf{H}}$)**.** *Let* $\overline{\mathbf{H}} = (\mathrm{P}, \mathrm{S}, \mathbf{H})$ *be an empirical Hankel system and let* $\widetilde{\mathcal{A}}_{\mathbf{H}}$ *be the quotient of* $\mathcal{A}_{\mathbf{H}}$ *modulo the transition-regular equivalence relation* $\sim_{\mathbf{H}}$ *as defined in Def. 3. Then:*
(1) The weighted automaton $\widetilde{\mathcal{A}}_{\mathbf{H}}$ *is trimmed and deterministic.*
(2) $\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}(\boldsymbol{ps}) = \mathbf{H}(\boldsymbol{p}, \boldsymbol{s})$ *for all* $\boldsymbol{p} \in \mathrm{P}$ *and* $\boldsymbol{s} \in \mathrm{S}$, *meaning that* $\widetilde{\mathcal{A}}_{\mathbf{H}}$ *contains* $\mathbf{H}$.

Proof. *(1)* Since the automaton is built on an empirical Hankel system, by definition, every $\boldsymbol{p}$ is the prefix of a string $\mathbf{x} = \boldsymbol{ps}$, such that $\mathbf{H}(\boldsymbol{p}, \boldsymbol{s}) \neq \mathbf{0}$ for at least one $\boldsymbol{s} \in \mathrm{S}$, hence $\boldsymbol{p}$ is accessible and coaccessible. This shows that $\mathcal{A}_{\mathbf{H}}$ is trimmed, and so is $\widetilde{\mathcal{A}}_{\mathbf{H}}$. Determinism and *(2)* follow from Lem. 1. ∎

Theorem 3 (Minimality of $\widetilde{\mathcal{A}}_{\mathbf{H}}$)**.**
(i) For any $\boldsymbol{p}, \boldsymbol{q} \in \mathrm{P}$, *we have*

$$\boldsymbol{p} \sim_{\mathbf{H}} \boldsymbol{q} \iff \boldsymbol{p} \sim_{\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}} \boldsymbol{q} \tag{13}$$

(ii) $\mathrm{P}/\sim_{\mathbf{H}} = \mathit{supp}(\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}})/\sim_{\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}}$.
(iii) Any automaton that contains $\overline{\mathbf{H}}$ *must have at least* $|\mathrm{P}/\sim_{\mathbf{H}}| = |Q_{\widetilde{\mathcal{A}}_{\mathbf{H}}}|$ *states.*
(iv) Let $\mathcal{A}'$ *be a WDFSA that contains* $\overline{\mathbf{H}}$. *Then,* $\mathbf{L}_{\mathcal{A}'}(\mathbf{x}) = \mathbf{L}_{\mathcal{A}_{\mathbf{H}}}(\mathbf{x})$, $\forall \mathbf{x} \in \mathit{supp}(\mathbf{L}_{\mathcal{A}_{\mathbf{H}}})$. *If* $\mathcal{A}'$ *is not equivalent to* $\mathcal{A}_{\mathbf{H}}$, *then*

$$|Q_{\mathcal{A}'}| \geqslant |\mathrm{P}/\sim_{\mathbf{H}}| + 1.$$

(v) In particular, $\widetilde{\mathcal{A}}_{\mathbf{H}}$ *is minimal.*

Proof.
(i) ($\Leftarrow$). If $\boldsymbol{p}^{-1}\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}} \equiv_{\Sigma^*} \boldsymbol{q}^{-1}\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}$, then restricting the two maps to $\Sigma^{\leqslant 1} \circ \mathrm{S}$ shows that we also have $\mathbf{H}_{\boldsymbol{p}} \equiv_{\Sigma^{\leqslant 1}\circ \mathrm{S}} \mathbf{H}_{\boldsymbol{q}}$. ($\Rightarrow$). Clearly, $\boldsymbol{p} \sim_{\mathbf{H}} \boldsymbol{q}$ implies $[\boldsymbol{p}] = [\boldsymbol{q}]$.
(ii) From *(i)*, we have that the restriction

$$\begin{aligned}\{\boldsymbol{p}^{-1}\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}} : \boldsymbol{p} \in \mathrm{P}\} \twoheadrightarrow \\ \{\boldsymbol{p}^{-1}\mathbf{H} = \boldsymbol{p}^{-1}\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}|_{\mathrm{S}} : \boldsymbol{p} \in \mathrm{P}\}\end{aligned} \tag{14}$$

provides a natural surjection. Let $\mathbf{x} \in \mathrm{supp}(\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}})$ and $\boldsymbol{\pi}_{\mathbf{x}}$ its path in $\widetilde{\mathcal{A}}_{\mathbf{H}}$. Let $\boldsymbol{p}_{\mathbf{x}} = f(\boldsymbol{\pi}_{\mathbf{x}}) \in \mathrm{P}$ be the final state of $\boldsymbol{\pi}_{\mathbf{x}}$ in $\mathcal{A}_{\mathbf{H}}$. Thus, by definition, we have $\boldsymbol{p}_{\mathbf{x}}^{-1}\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}} \equiv_{\Sigma^*} \mathbf{x}^{-1}\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}$. Accordingly, the natural projection map $\mathrm{P} \to \mathrm{supp}(\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}})/\equiv_{\Sigma^*}$ is surjective, and hence we have a bijection $\mathrm{P}/\sim_{\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}} \simeq \mathrm{supp}(\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}})/\equiv_{\Sigma^*}$. Since in *(i)* we showed $\mathrm{P}/\sim_{\mathbf{H}} \simeq \mathrm{P}/\sim_{\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}}$, we conclude

$$\mathrm{P}/\sim_{\mathbf{H}} = \mathrm{supp}(\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}})/\sim_{\mathbf{L}_{\widetilde{\mathcal{A}}_{\mathbf{H}}}} \tag{15}$$

(iii) Let $\mathcal{A}'$ be any WDFSA that contains $\overline{\mathbf{H}}$. Clearly, we have $\boldsymbol{p} \sim_{\mathbf{L}_{\mathcal{A}'}} \boldsymbol{q} \Rightarrow \boldsymbol{p} \sim_{\mathbf{H}} \boldsymbol{q}$, hence we have a surjective map $\mathrm{P}/\sim_{\mathbf{L}_{\mathcal{A}'}} \twoheadrightarrow \mathrm{P}/\sim_{\mathbf{H}}$, which shows

$$|Q_{\mathcal{A}'}| \geqslant |\mathrm{P}/\sim_{\mathbf{L}_{\mathcal{A}'}}| \geqslant |\mathrm{P}/\sim_{\mathbf{H}}| = |Q_{\widetilde{\mathcal{A}}_{\mathbf{H}}}| \tag{16}$$

(iv) Consider the sub-WDFSA $\mathcal{A}'_{\mathrm{P}} \subset \mathcal{A}'$ with states

$$Q'_{\mathrm{P}} = \{\boldsymbol{q} \in Q_{\mathcal{A}'} \mid \boldsymbol{q} = f(\boldsymbol{\pi}_{\boldsymbol{p}}) \text{ for } \boldsymbol{p} \in \mathrm{P}\} \tag{17}$$

Clearly $\mathcal{A}'_{\mathrm{P}}$ contains $\overline{\mathbf{H}}$. In addition, because $\overline{\mathbf{H}}$ is closed and consistent, $\mathcal{A}'_{\mathrm{P}}$ and $\mathcal{A}_{\mathbf{H}}$ have the same transitions (not necessarily same weights). We hence have $\mathbf{L}_{\mathcal{A}'_{\mathrm{P}}} = \mathbf{L}_{\mathcal{A}_{\mathbf{H}}}$. In other words, $\mathbf{L}_{\mathcal{A}'}(\mathbf{x}) = \mathbf{L}_{\mathcal{A}_{\mathbf{H}}}(\mathbf{x})$ for all $\mathbf{x} \in \mathrm{supp}(\widetilde{\mathcal{A}}_{\mathbf{H}})$. Accordingly, if $\mathbf{L}_{\mathcal{A}'} \neq \mathbf{L}_{\mathcal{A}_{\mathbf{H}}}$, then there exists $\mathbf{x} \in \mathrm{supp}(\mathbf{L}_{\mathcal{A}'}) \backslash \mathrm{supp}(\mathbf{L}_{\mathcal{A}_{\mathbf{H}}})$ and $[\mathbf{x}] \in (\mathrm{supp}(\mathbf{L}_{\mathcal{A}'})/\sim_{\mathbf{L}_{\mathcal{A}'}}) \backslash (\mathrm{supp}(\mathbf{L}_{\mathcal{A}_{\mathbf{H}}})/\sim_{\mathbf{L}_{\mathcal{A}'}})$. Thus:

$$\begin{aligned}|Q_{\mathcal{A}'}| &\geqslant |\mathrm{supp}(\mathbf{L}_{\mathcal{A}'})/\sim_{\mathbf{L}_{\mathcal{A}'}}| \\ &\geqslant |\mathrm{supp}(\mathbf{L}_{\mathcal{A}_{\mathbf{H}}})/\sim_{\mathbf{L}_{\mathcal{A}'}}| + 1 \\ &\geqslant |\mathrm{supp}(\mathbf{L}_{\mathcal{A}_{\mathbf{H}}})/\sim_{\mathbf{L}_{\mathcal{A}_{\mathbf{H}}}}| + 1 \\ &= |\mathrm{P}/\sim_{\mathbf{H}}| + 1\end{aligned}$$

(v) If in particular, $\mathcal{A}'$ is any WDFSA equivalent to $\widetilde{\mathcal{A}}_{\mathbf{H}}$, then by *(iii)* $|Q_{\mathcal{A}'}| \geqslant |Q_{\widetilde{\mathcal{A}}_{\mathbf{H}}}|$. ∎

Corollary 1 (Termination)**.** *Let* $\overline{\mathbf{H}}$ *be an empirical Hankel system and* $\mathcal{A}'$ *any automaton that contains it. If* $|Q_{\mathcal{A}'}| = |\mathrm{P}/\sim_{\mathbf{H}}|$, *then* $\mathbf{L}_{\mathcal{A}_{\mathbf{H}}} = \mathbf{L}_{\mathcal{A}'}$.

4 A Weighted L* Algorithm

Like Angluin (1987), we assume we have access to an **oracle** that answers the following queries about a deterministic regular language $L^\star : \Sigma^* \to \mathbb{K}$:
(1) **Membership query**: What is the weight $L^\star(\boldsymbol{p})$ of the string $\boldsymbol{p} \in \Sigma^*$?
(2) **Equivalence query**: Does a hypothesis automaton $\widetilde{\mathcal{A}}_{\mathbf{H}}$ generate $L^\star$? If it does *not*, the oracle provides a **counterexample**, which is a string $\boldsymbol{t}$ such that $L_{\widetilde{\mathcal{A}}_{\mathbf{H}}}(\boldsymbol{t}) \neq L^\star(\boldsymbol{t})$.
At a high level, the algorithm iteratively constructs empirical Hankel systems of increasing dimensions that capture observed patterns of the target language $L^\star$. Once sufficient observations are accumulated, the automaton derived from these Hankel systems will generate exactly $L^\star$.

4.1 The Learning Algorithm

Our weighted **L*** algorithm, with its main loop detailed in Alg. 1, employs the subroutines outlined in Alg. 2.

Algorithm 1 The Weighted $\mathbf{L}^*$ algorithm. Initially, the empirical Hankel matrix $\mathbf{H}$ is set to the zero matrix and the sets P, S to $\{\varepsilon\}$.

1. **def** $\mathbf{L}^*$ $(\mathbb{O})$:
2.   **while true** :
3.   **while true** :
4.   **if** $\mathbf{H}$ is **not** consistent :
5.   MAKECONSISTENT$(\mathbb{O}, \mathbf{H})$
6.   **else if** $\mathbf{H}$ is **not** closed :
7.   MAKECLOSED$(\mathbb{O}, \mathbf{H})$
8.   **else** : **break**
9. $\overline{\mathbf{H}} \leftarrow$ REMOVENULLROWS$(\mathbf{H})$
10. $\widetilde{\mathcal{A}}_{\mathbf{H}} \leftarrow$ MAKEAUTOMATON$(\overline{\mathbf{H}})$
11.   **if** EQUIVALENT$(\mathbb{O}, \widetilde{\mathcal{A}}_{\mathbf{H}})$: **return** $\widetilde{\mathcal{A}}_{\mathbf{H}}$
12.   **else** :
13. $\boldsymbol{p} \leftarrow$ COUNTEREXAMPLE$(\mathbb{O}, \widetilde{\mathcal{A}}_{\mathbf{H}})$
14.   **for** $t = 1$ **to** $|\boldsymbol{p}| + 1$:
15. $\mathrm{P} \leftarrow \mathrm{P} \cup \{\boldsymbol{p}_{<t}\}$
16.   COMPLETE$(\mathbb{O}, \mathbf{H})$

Algorithm 2 Subroutines of Alg. 1.

1. **def** MAKECONSISTENT$(\mathbb{O}, \mathbf{H})$:
2.   **for** $\langle \boldsymbol{p}, \boldsymbol{p}' \rangle \in \mathrm{P} \times \mathrm{P}$:
3.   **if** $\mathbf{H}_{\boldsymbol{p}} \equiv_{\Sigma^{\leqslant 1} \circ \mathrm{S}} \mathbf{H}_{\boldsymbol{p}'}$:
4.   **for** $\langle a, \boldsymbol{s} \rangle \in \Sigma \times \mathrm{S}$:
5.   **if** $\frac{\mathbf{H}_{\boldsymbol{p}a}(\boldsymbol{s})}{d_{\mathbf{H}}(\boldsymbol{p}a)} \neq \frac{\mathbf{H}_{\boldsymbol{p}'a}(\boldsymbol{s})}{d_{\mathbf{H}}(\boldsymbol{p}'a)}$:
6. $\mathrm{S} \leftarrow \mathrm{S} \cup \{a\boldsymbol{s}\}$
7.   COMPLETE$(\mathbb{O}, \mathbf{H})$
8. **def** MAKECLOSED$(\mathbb{O}, \mathbf{H})$:
9.   **for** $\langle \boldsymbol{p}, a \rangle \in \mathrm{P} \times \Sigma$:
10.   **if** $\nexists \boldsymbol{p}' \in \mathrm{P}$ s.t. $\mathbf{H}_{\boldsymbol{p}a} \equiv_{\Sigma^{\leqslant 1} \circ \mathrm{S}} \mathbf{H}_{\boldsymbol{p}'}$:
11. $\mathrm{P} \leftarrow \mathrm{P} \cup \{\boldsymbol{p}a\}$
12.   COMPLETE$(\mathbb{O}, \mathbf{H})$
13. **def** COMPLETE$(\mathbb{O}, \mathbf{H})$:
14.   **for** $\boldsymbol{p} \in \mathrm{P} \circ \Sigma^{\leqslant 1}$:
15.   **for** $\boldsymbol{s} \in \Sigma^{\leqslant 1} \circ \mathrm{S}$:
16. $\mathbf{H}(\boldsymbol{p}, \boldsymbol{s}) \leftarrow$ MEMBERSHIP$(\mathbb{O}, \boldsymbol{ps})$

Initialization. P and S are initialized as $\{\varepsilon\}$ and the $\mathbf{H}$ to the zero matrix.

Handling inconsistencies. MAKECONSISTENT in Line 7 of Alg. 1 looks for rows $\boldsymbol{p}, \boldsymbol{p}' \in \mathrm{P}$ that make $\mathbf{H}$ non-consistent, i.e., $\mathbf{H}_{\boldsymbol{p}a} \not\equiv_{\Sigma^{\leqslant 1} \circ \mathrm{S}} \mathbf{H}_{\boldsymbol{p}'a}$: It normalizes a row $\mathbf{H}_{\boldsymbol{p}a}$ as $\frac{\mathbf{H}_{\boldsymbol{p}a}}{d_{\mathbf{H}}(\boldsymbol{p}a)}$ (Alg. 2, Line 5), which allows testing homothetic equivalence with equality.[5] For every $\boldsymbol{s} \in \mathrm{S}$ that makes $\mathbf{H}$ inconsistent, $a\boldsymbol{s}$ is added to S. This results in the new equivalence classes $[\boldsymbol{p}]$ and $[\boldsymbol{p}']$ because $\mathbf{H}_{\boldsymbol{p}}$ and $\mathbf{H}_{\boldsymbol{p}'}$ do not match anymore on the column indexed by $a\boldsymbol{s}$. See Lem. 2 for more details.

Closing H. MAKECLOSED (Alg. 1, Line 7; Alg. 2) adds to P the missing prefixes required to make $\mathbf{H}$ closed. This results in the new equivalence class $[\boldsymbol{p}a]$. See Lem. 2 for more details.

Filling out H. COMPLETE fills the empty entries of $\mathbf{H}$ by asking membership queries to the oracle.

Handling inconsistencies, closing $\mathbf{H}$, and filling $\mathbf{H}$ is carried out by the inner while loop (Lines 3 to 8) of Alg. 1 until $\mathbf{H}$ is both closed and consistent.

Generating $\widetilde{\mathcal{A}}_{\mathbf{H}}$. When $\mathbf{H}$ is closed and consistent, Alg. 1 first removes $\mathbf{0}$-rows from the matrix to obtain an empirical Hankel system, then it generates the empirical Hankel automaton $\widetilde{\mathcal{A}}_{\mathbf{H}}$, and lastly submits an equivalence query to the oracle (Line 11). If the oracle answers positively, Alg. 1 halts and returns $\widetilde{\mathcal{A}}_{\mathbf{H}}$. Otherwise, the oracle provides a counterexample, which is added to P along with its prefixes. $\mathbf{H}$ is then updated through membership queries (Lines 13 to 16). The algorithm continues until $\mathbf{H}$ is closed and consistent again.

Theorem 4. *Let $\mathbb{K}$ be a semifield and Σ an alphabet. Let $\mathbb{O}$ be an oracle for a deterministic regular language $L^\star : \Sigma^* \to \mathbb{K}$, whose minimal WDFSA has N states. Then, Alg. 1 returns a minimal WDFSA generating $L^\star$ in time $\mathcal{O}\left(N^5 M^2 |\Sigma|^2\right)$, where M is the length of the longest counterexample that $\mathbb{O}$ can provide.*

Proof. See App. B. ∎

5 Conclusion

We introduce a weighted $\mathbf{L}^*$ algorithm, an oracle-based algorithm for learning weighted regular languages, building upon the paradigm pioneered by Angluin (1987). While similar methods have been proposed before, our method is novel in that it learns an *exact* deterministic WFSA, akin to the original Angluin's (1987) unweighted version.

Limitations

One of the limitations of weighted $\mathbf{L}^*$ is that it requires an oracle capable of answering membership and equivalence queries. However, in the case we want to use $\mathbf{L}^*$ to study a language model, this is

[5] When the entire row $\mathbf{H}_{\boldsymbol{p}a}$ is zero, we do not normalize; this is omitted in the pseudocode for brevity.

the ideal setting, as we can use the language model itself as the oracle (Weiss et al., 2018; Okudono et al., 2019; Weiss et al., 2019). Another limitation to the applications of our work is that not every language model is efficiently representable as a finite-state machine. For instance, Merrill (2019) shows that LSTMs are strictly more powerful than FSAs. Therefore, in practice, one may have to use a simplified abstraction of the model one aims to learn (Weiss et al., 2019), inevitably reducing the model's expressivity.

Acknowledgements

Ryan Cotterell acknowledges support from the Swiss National Science Foundation (SNSF) as part of the "The Nuts and Bolts of Language Models" project. Anej Svete is supported by the ETH AI Center Doctoral Fellowship.

References

Cyril Allauzen and Mehryar Mohri. 2003. Efficient algorithms for testing the twins property. *Journal of Automata, Languages and Combinatorics*, 8(2):117–144.

Dana Angluin. 1987. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106.

Borja Balle, Xavier Carreras, Franco M. Luque, and Ariadna Quattoni. 2014. Spectral learning of weighted automata. *Machine Learning*, 96(1):33–63.

Borja Balle and Mehryar Mohri. 2012. Spectral learning of general weighted automata via constrained matrix completion. In *Advances in Neural Information Processing Systems*, volume 25.

Borja Balle and Mehryar Mohri. 2015. Learning weighted automata. In *Algebraic Informatics*, pages 1–21. Springer International Publishing.

Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. 2000. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530.

Francesco Bergadano and Stefano Varricchio. 1996. Learning behaviors of automata from multiplicity and equivalence queries. *SIAM Journal on Computing*, 25(6):1268–1280.

Laure Daviaud and Marianne Johnson. 2024. Feasability of learning weighted automata on a semiring. *Preprint*, arXiv:2309.07806.

E Mark Gold. 1978. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320.

Joshua Goodman. 1996. Parsing algorithms and metrics. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 177–183, Santa Cruz, California, USA. Association for Computational Linguistics.

John E. Hopcroft and Jeffrey D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Publishing Company, Inc.

Jaap Jumelet and Willem Zuidema. 2023. Transparency at the source: Evaluating and interpreting language models with access to the true distribution. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4354–4369, Singapore. Association for Computational Linguistics.

William Merrill. 2019. Sequential neural networks as automata. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 1–13, Florence. Association for Computational Linguistics.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Mehryar Mohri. 2009. *Weighted Automata Algorithms*, pages 213–254. Springer Berlin Heidelberg, Berlin, Heidelberg.

Franz Nowak, Anej Svete, Alexandra Butoi, and Ryan Cotterell. 2024. On the representational capacity of neural language models with chain-of-thought reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12510–12548, Bangkok, Thailand. Association for Computational Linguistics.

Takamasa Okudono, Masaki Waga, Taro Sekiyama, and Ichiro Hasuo. 2019. Weighted automata extraction from recurrent neural networks via regression on state spaces. *Preprint*, arXiv:1904.02931.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2018. Extracting automata from recurrent neural networks using queries and counterexamples. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5247–5256. PMLR.

Gail Weiss, Yoav Goldberg, and Eran Yahav. 2019. Learning deterministic weighted automata with queries and counterexamples. In *Advances in Neural Information Processing Systems*, volume 32.

A Transition-regular Equivalence Relations on Automata

Definition 2. *Let $\mathcal{A} = (\Sigma, Q, \delta, \lambda, \rho)$ be a WFSA. An equivalence relation $\sim$ on Q is **transition-regular** if, for any states $\boldsymbol{p}, \boldsymbol{q} \in Q$, whenever $\boldsymbol{p} \sim \boldsymbol{q}$, we have:*

- ***Outgoing Transition Consistency**: For every symbol $a \in \Sigma$, if there exists a state $\boldsymbol{r} \in Q$ such that $\boldsymbol{p} \xrightarrow{a/w_1} \boldsymbol{r} \in \delta$ with weight $w_1 \neq \mathbf{0}$, then*
  - *there must exist a transition $\boldsymbol{q} \xrightarrow{a/w_2} \boldsymbol{r} \in \delta$ with $w_2 = w_1$.*
  - *for any other state $\boldsymbol{r}' \in Q$ such that $\boldsymbol{p} \xrightarrow{a/w_1'} \boldsymbol{r}' \in \delta$ we must have $\boldsymbol{r} \sim \boldsymbol{r}'$.*
- ***Initial and Final Weight Consistency**: The initial and final weights of $\mathbf{p}$ and $\mathbf{q}$ are identical:*

$$\lambda(\boldsymbol{p}) = \lambda(\boldsymbol{q}) \tag{18a}$$
$$\rho(\boldsymbol{p}) = \rho(\boldsymbol{q}). \tag{18b}$$

Definition 3. *Let $\mathcal{A} = (\Sigma, Q, \delta, \lambda, \rho)$ be a WFSA. Given a transition-regular equivalence relation $\sim$ on Q, we define the **quotient automaton** $\widetilde{\mathcal{A}} = (\widetilde{Q}, \Sigma, \widetilde{\delta}, \widetilde{\lambda}, \widetilde{\rho})$ as follows:*

- ***States**: The state set of the quotient automaton is $\widetilde{Q} = Q/\sim$.*
- ***Transitions**: Define the transition set $\widetilde{\delta}$ as follows. For each equivalence class $[\boldsymbol{p}] \in \widetilde{Q}$ and each symbol $a \in \Sigma$, if there exists a state $\boldsymbol{q} \in Q$ such that $\boldsymbol{p} \xrightarrow{a/w} \boldsymbol{q} \in \delta$, then there is a corresponding transition*

$$[\boldsymbol{p}] \xrightarrow{a/w} [\boldsymbol{q}] \in \widetilde{\delta},$$

  where $[\boldsymbol{q}]$ denotes the equivalence class of $\boldsymbol{q}$.
- ***Initial and Final Weights**: Define the initial and final weight functions for each equivalence class $[\boldsymbol{p}] \in \widetilde{Q}$ as:*

$$\widetilde{\lambda}([\boldsymbol{p}]) = \lambda(\boldsymbol{p}),$$
$$\widetilde{\rho}([\boldsymbol{p}]) = \rho(\boldsymbol{p}).$$

Lemma 1. *The quotient automaton $\widetilde{\mathcal{A}}$ is deterministic and it generates the following weighted language:*

$$\mathbf{L}_{\mathcal{A}}(\boldsymbol{p}) = |\Pi_{\mathcal{A}}(\boldsymbol{p})| \mathbf{L}_{\widetilde{\mathcal{A}}}(\boldsymbol{p}), \qquad \forall \boldsymbol{p} \in \Sigma^*. \tag{19}$$

Proof. The proof is straightforward.

■

B Proof of Thm. 4

Theorem 4. *Let $\mathbb{K}$ be a semifield and Σ an alphabet. Let $\mathbb{O}$ be an oracle for a deterministic regular language $L^\star : \Sigma^* \to \mathbb{K}$, whose minimal WDFSA has N states. Then, Alg. 1 returns a minimal WDFSA generating $L^\star$ in time $\mathcal{O}\left(N^5 M^2 |\Sigma|^2\right)$, where M is the length of the longest counterexample that $\mathbb{O}$ can provide.*

B.1 Termination

We begin by stating the following lemma

Lemma 2 (Number of equivalence classes increases)**.** *When Alg. 1*
(1) adds a suffix to S because $\mathbf{H}$ is not consistent,
(2) adds a prefix to P because the table is not closed,
(3) adds a prefix to P because the oracle replied with a counterexample,

the number of equivalence classes $\mathrm{P}/\sim_{\mathbf{H}}$ *increases.*

Proof.
(1) If the empirical Hankel matrix is not consistent, MAKECONSISTENT (Alg. 1, Line 5) finds two prefixes $\boldsymbol{p}, \boldsymbol{p}' \in \mathrm{P}$ such that $\mathbf{H}_{\boldsymbol{p}} \equiv \mathbf{H}_{\boldsymbol{p}'}$ but $\mathbf{H}_{\boldsymbol{p}a} \not\equiv \mathbf{H}_{\boldsymbol{p}'a}$. Then it searches for a tuple $(a, \boldsymbol{s})$, $a \in \Sigma$, $\boldsymbol{s} \in \Sigma^{\leqslant 1} \circ \mathrm{S}$ that makes the relation $\mathbf{H}_{\boldsymbol{p}a} \equiv \mathbf{H}_{\boldsymbol{p}'a}$ false, and adds $a\boldsymbol{s}$ to S. After adding $a\boldsymbol{s}$ to S, we have that $\mathbf{H}_{\boldsymbol{p}} \not\equiv \mathbf{H}_{\boldsymbol{p}'}$, and therefore the equivalence class $[\boldsymbol{p}]$ is divided in two different ones.
(2) If $\mathbf{H}$ is not closed, MAKECLOSED (Alg. 1, Line 7) finds $\boldsymbol{p} \in \mathrm{P}$ and $a \in \Sigma$ such that $\mathbf{H}_{\boldsymbol{p}a} \not\equiv \mathbf{H}_{\boldsymbol{p}'}$ for every $\boldsymbol{p}' \in \mathrm{P}$ and adds $\boldsymbol{p}a$ to P. Since there was no $\boldsymbol{p}'$ in P such that $\boldsymbol{p}' \sim_{\mathbf{H}} \boldsymbol{p}a$, it follows that a new equivalence class $[\boldsymbol{p}a]$ is added to $\mathrm{P}/\sim_{\mathbf{H}}$.
(3) When the Oracle replies negatively to the equivalence query, the counterexample $\boldsymbol{t}$, together with its prefixes, is added to P. We show that even in this case, $\dim(\overline{\mathbf{H}})$ increases. Let $\overline{\mathbf{H}}$ and $\overline{\mathbf{H}}^{\boldsymbol{t}}$ denote the empirical Hankel system before and after adding $\boldsymbol{t}$, and let $\widetilde{\mathcal{A}}_{\mathbf{H}}$ and $\widetilde{\mathcal{A}}_{\mathbf{H}^{t}}$ denote the corresponding empirical Hankel automaton in each case. We note that both $\widetilde{\mathcal{A}}_{\mathbf{H}}$ and $\widetilde{\mathcal{A}}_{\mathbf{H}^{t}}$ contain $\overline{\mathbf{H}}$ and therefore by Thm. 3—since the automata are not equivalent—$\widetilde{\mathcal{A}}^{\boldsymbol{t}}_{\mathbf{H}}$ must have at least one more state than $\widetilde{\mathcal{A}}_{\mathbf{H}}$. By construction of the empirical Hankel automaton, we know that this implies that $\dim(\overline{\mathbf{H}})$ increases. ■

Let $(\overline{\mathbf{H}}_k = (\mathrm{P}_k, \mathrm{S}_k, \mathbf{H}_k))_{k \geqslant 0}$ be the sequence of empirical Hankel systems constructed at each iteration of the main loop of Alg. 1. By Lem. 2, this sequence is increasing; that is, $\overline{\mathbf{H}}_k \preceq \overline{\mathbf{H}}_{k+1}$ for all $k \geqslant 0$. Let $\mathcal{A}^{\star}$ denote any minimal automaton for the target language $L^{\star}$. On the one hand, we know that the automaton $\mathcal{A}^{\star}$ contains $\overline{\mathbf{H}}_k$ for all $k \geqslant 0$. On the other hand, by Lem. 2, there exists $n \in \mathbb{N}$ such that $\dim(\overline{\mathbf{H}}_n) = |\mathrm{P}_n/\sim_{\mathcal{A}_{\mathbf{H}_n}}| = |Q_{\mathcal{A}^{\star}}|$. Therefore, by applying Cor. 1, we conclude that $L^{\star} = \mathbf{L}_{\mathcal{A}_{\mathbf{H}_n}}$.

Consequently, after a finite number of iterations, the oracle will respond positively to the equivalence query, causing the algorithm to halt. Furthermore, we observe that the inner loop of Alg. 1 executes at most $|Q_{\mathcal{A}^{\star}}|$ times, as $\dim(\overline{\mathbf{H}})$ increases at every step by Lem. 2, and at each iteration, $\mathcal{A}^{\star}$ continues to contain $\mathbf{H}$.

B.2 Run-Time

First, we note that since $L^{\star}$ always contains $\mathbf{H}$, by Lem. 2 any of the following events can only occur at most N times in total, where N is the number of states of the minimal automaton accepting $L^{\star}$: *i)* we add a prefix because the table is not closed, *ii)* we add a suffix because the table is not consistent, *iii)* we add a counterexample because the oracle replies negatively to the equivalence query. Then, let us give a bound on the size of the prefix and suffix sets:

- $|\mathrm{P}| \in \mathcal{O}(NM)$, in fact initially $\mathrm{P} = \{\varepsilon\}$, and P can be incremented at most N times because the matrix is not consistent and at most NM times because the oracle replies with a counterexample, where M is the length of the longest counterexample.
- $|\mathrm{S}| \in \mathcal{O}(N)$, in fact initially $\mathrm{S} = \{\varepsilon\}$, and S can be incremented at most N times.

Next, we shall analyze the runtime of the operations executed during the main loop of Alg. 1.

- MAKECONSISTENT $\in \mathcal{O}\left(|\mathrm{P}|^2|\mathrm{S}|^2|\Sigma|^2\right)$
- MAKECLOSED $\in \mathcal{O}\left(|\mathrm{P}|^2|\mathrm{S}||\Sigma|^2\right)$
- COMPLETE $\in \mathcal{O}\left(|\mathrm{P}||\mathrm{S}||\Sigma|^2\right)$
- MAKEAUTOMATON $\in \mathcal{O}(|\mathrm{P}| + |\mathrm{P}||\Sigma|)$,

In the analysis above we used the fact that the map $d_{\mathbf{H}} : \mathrm{P} \circ \Sigma^{\leqslant 1} \to \mathbb{K}$ is fixed for every empirical Hankel matrix and can be precomputed in time $\mathcal{O}\left(|\mathrm{P}||\mathrm{S}||\Sigma|^2\right)$ and reused multiple times.

We note that each of these operations can be executed at most N times before the algorithm halts, and therefore—by substituting in the bounds for P and S—we compute the total runtime of Alg. 1 as:

$$\mathcal{O}\left(N\left(N^4M^2|\Sigma|^2 + N^3M^2|\Sigma|^2 + N^2M|\Sigma|^2 + N^2M|\Sigma| + MN + NM|\Sigma|\right)\right) \tag{20a}$$
$$= \mathcal{O}\left(N^5M^2|\Sigma|^2\right) \tag{20b}$$

We note an important distinction between our weighted version of $\mathbf{L}^{*}$ and Angluin's (1987). In fact, in the weighted case, we need to make sure that the empirical Hankel matrix has a column for every element $\Sigma^{\leqslant 1} \circ \mathrm{S}$ and not only for S. This is a fundamental step to enforce that the relation $\sim_{\mathbf{H}}$ is transition regular

(Thm. 1), and it is related to the fact that in the weighted case, we don't seek language equality, but rather equality modulo multiplication by a constant k.