A Review of Learning Weighted Automata

Comp 599 Mathematical Techniques for Machine Learning
Roland Riachi

1. Introduction

Picture an owner of a shipping company trying to study the logistics of their business; profit generated by a shipment would depend largely on the kind of good being transported, its origin, and its destination. With potentially infinite choices of delivery routes, the owner would likely be interested in a finite, ideally minimal, description of possible profits.

This desire to model quantitative systems is what motivates the study of learning weighted finite automata (WFA). In fact, algorithms to design WFAs mimicking such systems boast applications in many fields; for example, image processing [25, 30], speech recognition [29, 31, 35], speech synthesis [1, 38], phonological and morphological rule compilation [26, 27, 33], parsing [32], bioinformatics [2, 20], sequence modeling and prediction [19], formal verification [3], optical character recognition [16], and more.

In this paper we report some techniques for learning WFAs reviewed by Balle and Mohri [7]. The goal is to illustrate some relatively new approaches to the problem (namely, learning WFAs from queries [13, 14], learning stochastic WFAs from i.i.d. samples [6, 24], and a learning WFAs - not necessarily probabilistic - from finite string-value pair samples [10]), to clarify some proofs and to provide hopefully insightful intuition into the motivations and definition behind each technique.

1.1. Weighted Automata

Let Σ be a finite alphabet. We write $|x|$ to denote the length of a string $x \in \Sigma^*$ and ε the *empty string*, where $|\varepsilon| = 0$.

Definition 1. A *weighted finite automaton* (WFA) over a semiring $(\mathbb{S}, \oplus, \otimes, \bar{0}, \bar{1})$ is a 5-tuple $(Q, \Sigma, \delta, \boldsymbol{\alpha}, \boldsymbol{\beta})$[1], where Q is a finite set of states, $\delta \subset Q \times \Sigma \times \mathbb{S} \times Q$ is a finite set of transitions, $\boldsymbol{\alpha} \in \mathbb{S}^{|Q|}$ is the vector of initial weights, and $\boldsymbol{\beta} \in \mathbb{S}^{|Q|}$ is the vector of final weights.

We remark that one can define a WFA to include a set of initial states, I such that instead $\boldsymbol{\alpha} \in \mathbb{S}^{|I|}$, or a set of accepting states, F, such that instead $\boldsymbol{\beta} \in \mathbb{S}^{|F|}$ but the above definition suffices for the techniques described in this paper.

For any transition $e \in \delta$ we denote by $w[e]$ the weight of e, $w[\pi]$ the weight of the path $\pi = e_1 e_2 \cdots e_n$ where we define $w[\pi] = w[e_1] \otimes w[e_2] \otimes \cdots \otimes w[e_n]$, s_π the source state, and t_π the target state. Hence a WFA maps strings in Σ^* to $\mathbb{S}$, we which abusively denote

$$\mathcal{A}(x) = \bigoplus_{\pi \in P(x)} \bigg(\boldsymbol{\alpha}[s_\pi] \otimes w[\pi] \otimes \boldsymbol{\beta}[t_\pi] \bigg),$$

for $x \in \Sigma^*$, where $P(x)$ denotes the finite set of paths labeled with x.

[1] Some papers define delta as a multiset, allowing multiple transitions between the same two states with the same labels and weights; however in practice such copies are represented as a single transition whose weight is the semiring sum of the weights of the individual transitions.

To motivate this definition, we return to the example of the shipping company. Given that transporting a certain good between various pairs of cities may generate a different profit per route, we'd like the weight of the good to reflect its overall profit without directly depending on the individual routes [2]. In fact, the above definition provides the following useful result. For any $a \in \Sigma$, let $\mathcal{A}_a$ be the matrix $[A_a]_{pq} = \oplus_{e \in P(p,a,q)} w[e]$, where $P(p, a, q)$ is the set of transitions labeled with a from states p to q. Then for $x = x_1 x_2 \cdots x_n$, $\mathcal{A}(x)$ can be rewritten as the matrix multiplication

$$\mathcal{A}(x) = \boldsymbol{\alpha}^T \mathcal{A}_{x_1} \mathcal{A}_{x_2} \cdots \mathcal{A}_{x_n} \boldsymbol{\beta}.$$

In light of this, for some alphabet Σ, a WFA $\mathcal{A}$ in n states can be entirely described by its weight matrices. Therefore if the number of states is clear, we may also write $\mathcal{A} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathcal{A}_a : a \in \Sigma\})$. Note that in application we begin with some function $f : \Sigma^* \to \mathbb{S}$ we wish to learn using WFAs and so the following definition is important.

Definition 2. We say a function $f : \Sigma^* \to \mathbb{S}$ is *rational* if there exists a WFA $\mathcal{A}$ such that for all $x \in \Sigma^*$ $f(x) = \mathcal{A}(x)$.

Finally, we denote by $|\mathcal{A}| = |Q| + |\delta|$ the size of a WFA $\mathcal{A}$.

1.2. Hankel Matrices

By expressing the weight of a word as multiplication of matrices, we have added linear algebra techniques to our arsenal of strategies with which we can attack the problem of learning WFAs. However, in order to describe learning algorithms of WFAs we must first introduce the notion of a Hankel matrix. This tool represents all possible outputs of a WFA, and we will show how to reconstruct a WFA from sufficiently descriptive sub-blocks of its Hankel matrix.

To this end, in the sections that follow we assume that the semiring $\mathbb{S}$ is a field, which makes it possible to discuss the rank of the Hankel matrix of a WFA and thus design efficient algorithms. Note that some of the results can in fact be extended to rings.

Definition 3. Let $\mathbf{H} \in \mathbb{S}^{|\Sigma^*| \times |\Sigma^*|}$ be a bi-infinite matrix whose rows and columns we index by strings in Σ^* and whose entries are denoted by $\mathbf{H}[u, v]$, where $u \in \Sigma^*$ is the row index and $v \in \Sigma^*$ is the column index. Then $\mathbf{H}$ is a *Hankel matrix* if $\mathbf{H}[u, v] = \mathbf{H}[u', v']$ for all $u, v, u', v' \in \Sigma^*$ such that $uv = u'v'$. We write rank($\mathbf{H}$) to denote the rank of $\mathbf{H}$.

Definition 4. Let $f : \Sigma^* \to \mathbb{S}$ be a function. The *Hankel matrix of f* $\mathbf{H}_f$ is the matrix induced by $\mathbf{H}_f(u, v) = f(uv)$, for all $u, v \in \Sigma^*$. We remark that any Hankel matrix $\mathbf{H}$ also induces a rational function $f : \Sigma^* \to \mathbb{S}$ with the mapping $f(u) = \mathbf{H}(u, \varepsilon)$ for all $u \in \Sigma^*$.

From these definitions we obtain the following characterization.

Theorem 1 (Fliess [22]). *Let $\mathbb{S}$ be a field and let $f : \Sigma^* \to \mathbb{S}$. The rank of the Hankel matrix $\mathbf{H}_f$ is finite if and only if f is rational. In particular, there exists a WFA $\mathcal{A}$ with rank($\mathbf{H}_f$) states such that for all $x \in \Sigma^*$ $f(x) = A(x)$ and for any other WFA $\mathcal{A}'$ such that $f(x) = \mathcal{A}'(x)$, $\mathcal{A}'$ has at least as many states as $\mathcal{A}$.*

When rank($\mathbf{H}_f$) is finite, we say the WFA $\mathcal{A}$ representing f with rank($\mathbf{H}_f$) states is *minimal.* This notion of minimality differs from the notion of size we previously defined as it depends solely on the number of states of $\mathcal{A}$. However, WFAs that are minimal (in the number of states) typically have many transitions.

The proof of Fliess' theorem (see also [18]) is excluded because it is a standard result that is fairly easy to show. The forward direction follows by induction on $|w|$ for $w \in \Sigma^*$, after first constructing

[2] For example, if it is highly profitable to transport a good between two cities but unprofitable to transport the same good between two other cities, should the good be considered profitable or not?

a basis for $\mathbf{H}_f$. Meanwhile, the converse direction follows since $f(uv) = \mathcal{A}(uv) = (\boldsymbol{\alpha}^T \mathbf{A}_u)(\mathbf{A}_v \boldsymbol{\beta})$, and for any matrices $\mathbf{A}$ and $\mathbf{B}$ of appropriate dimensions, $\text{rank}(\mathbf{AB}) \leq \min\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\}$.

Fliess' theorem characterizes the conditions which guarantee when algorithms for learning WFAs are applicable. This importance of the rank motivates a natural interest in the bases of a Hankel matrix, dubbed Hankel bases. A Hankel basis for a Hankel matrix can be thought of as a finite sub-block of that matrix that serves as a minimal and complete description of the weights of all possible strings in Σ^*. In order to formally discuss Hankel bases, we provide the following definitions.

Definition 5. Let $\mathcal{P}, \mathcal{S} \subset \Sigma^*$. Then the pair $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ is called a *Hankel mask*, and we call the strings in $\mathcal{P}$ *prefixes* and the strings in $\mathcal{S}$ *suffixes* of the mask.

Definition 6. Let $\mathbf{H} \in \mathbb{S}^{|\Sigma^*| \times |\Sigma^*|}$ be a Hankel matrix and let $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ be a Hankel mask. We denote by $\mathbf{H}_{\mathcal{B}} \in \mathbb{S}^{|\mathcal{P}| \times |\mathcal{S}|}$ the *Hankel sub-block* of $\mathbf{H}$ with rows indexed by strings in $\mathcal{P}$ and suffixes indexed by strings in $\mathcal{S}$.

Therefore for all $u \in \mathcal{P}$ and $v \in \mathcal{S}$ we have $\mathbf{H}_{\mathcal{B}}[u, v] = \mathbf{H}[u, v]$. Thus for all $u, u' \in \mathcal{P}$ and $v, v' \in \mathcal{S}$ such that $uv = u'v'$ we have $\mathbf{H}_{\mathcal{B}}[u, v] = \mathbf{H}_{\mathcal{B}}[u', v']$.

Definition 7. Let $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ be a Hankel mask and $\mathbf{H}_{\mathcal{B}}$ its associated Hankel sub-block. For each $a \in \Sigma$, we define the Hankel mask $\mathcal{B}_a = (\mathcal{P}a, \mathcal{S})$ where $\mathcal{P}a = \{ua : u \in \mathcal{P}\}$ and from here on we write $\mathbf{H}_a$ instead of $\mathbf{H}_{\mathcal{B}_a}$. Furthermore, we define the block matrix $\mathbf{H}_\Sigma$ by $\mathbf{H}_\Sigma^T = [\mathbf{H}_{a_1}^T | \mathbf{H}_{a_2}^T | \cdots | \mathbf{H}_{a_r}^T]$, if $\Sigma = \{a_1, a_2, \ldots, a_r\}$.

Note that for all $u \in \mathcal{P}$ and $v \in \mathcal{S}$ we have $\mathbf{H}_a[u, v] = \mathbf{H}[ua, v]$.

Definition 8. We say a Hankel mask $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ is *complete* if $\varepsilon \in \mathcal{P} \cap \mathcal{S}$ and $\text{rank}([\mathbf{H}_{\mathcal{B}}^T | \mathbf{H}_\Sigma^T]) = \text{rank}(\mathbf{H}_{\mathcal{B}}^T)$. We say a complete Hankel mask is *minimal* if $\text{rank}(\mathbf{H}_{\mathcal{B}}) = |\mathcal{P}|$. This implies $|\mathcal{P}| \leq |\mathcal{S}|$.

Thus, a Hankel mask is complete if the paths $\pi = uav$ for $u \in \mathcal{P}$, $a \in \Sigma$, and $v \in \mathcal{S}$ do not provide any new insight into the system that is being learned. Meanwhile, a Hankel mask is minimal if there is no redundant encoding of information of the system by the strings in $\mathcal{P}$. To better understand these definitions, for a rational function f whose corresponding Hankel matrix is finite dimensional, we can intuitively think of the system as a space in $\mathbb{R}^d$.

Suppose $\text{rank}(\mathbf{H}_f) = 2$, then $\mathbf{H}_f$ represents a two-dimensional plane. To imagine a mask which is not complete, choose $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ such that $|\mathcal{P}| = 1$ and $|\mathcal{S}| = 3$, then $\mathbf{H}_{\mathcal{B}}$ represents a line. Adding a "new dimension" by considering the mask $\mathcal{B}' = (\mathcal{P}\Sigma, \mathcal{S})$ may augment the line to a plane. To imagine a mask which is not minimal, consider $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ where $|\mathcal{P}| = |\mathcal{S}| = 3$, then $\mathbf{H}_{\mathcal{B}}$ represents a two-dimensional plane in $\mathbb{R}^3$. One row in the matrix $\mathbf{H}_{\mathcal{B}}$ will contain a row of zeros and so $\text{rank}(\mathbf{H}_{\mathcal{B}}) \neq |\mathcal{P}|$.

It is important to note that a complete and minimal Hankel mask is not necessarily a Hankel basis. On the other hand, by definition a Hankel basis is complete yet it need not be minimal. Thus we would like to develop theory which provides bounds on the rank of a matrix and determines when such bounds are attainable. To this end, we provide the following propositions and corollaries for Hankel matrices of finite rank, which result from the definition of the rank and the fact that the rank of a matrix is upper bounded by both its dimensions.

Proposition 1. *Let $\mathbf{H}$ be a Hankel matrix with rank$(\mathbf{H}) = n$, then there exists a Hankel basis $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ for $\mathbf{H}$ with $|\mathcal{P}| = |\mathcal{S}| = n$.*

Proposition 2. *Let $\mathbf{H}$ be a Hankel matrix with rank$(\mathbf{H}) = n$, then there exists a Hankel basis $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ for $\mathbf{H}$ with $|\mathcal{P}| = |\mathcal{S}| = n$, where $\mathcal{P}$ is prefix-closed and $\mathcal{S}$ is suffix-closed.*

The proof of Proposition 2 is a consequence of an algorithm for minimizing WFAs that was first presented by Schützenberger [37] (see also [36]). An overview of an improved version is presented by Cardon and Crochmere [17]. Given this result, we mention that a useful consequence is an upper bound on the number of strings that need to be tested in order to find a prefix and suffix-closed Hankel for a Hankel matrix.

Corollary 1. *Let* $\mathbf{H}$ *be a Hankel matrix with rank*$(\mathbf{H}) = n$*. Then* $\mathcal{B} = (\Sigma^{<n}, \Sigma^{<n})$ *is a Hankel basis for* $\mathbf{H}$*.*

This follows from Proposition 2 and the fact that for any string $w \in \Sigma^*$, w has $|w| + 1$ decompositions of the form $w = uv$ for $u, v \in \Sigma^*$. Thus for a given $X \subset \Sigma^*$ which is either prefix or suffix-closed and such that $|X| = n$, then $|x| < n$ for $x \in X$.

2. WFA Reconstruction using Hankel Masks

2.1. WFA Reconstruction using Complete Minimal Masks

In this section, we describe how to obtain an WFA given a complete and minimal Hankel mask. Furthermore, if in addition the mask is a Hankel basis for a Hankel matrix $\mathbf{H}_f$, then we show that the WFA recovered is minimal and computes f. The input to the reconstruction algorithm is a complete minimal Hankel mask $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ for a Hankel matrix $\mathbf{H}$ and the corresponding sub-blocks $\mathbf{H}_\mathcal{B}$ and $\mathbf{H}_\Sigma$. It outputs a WFA $\mathcal{A}$ with $k = |\mathcal{P}|$ states.

Writing $\mathcal{P} = \{u_1, u_2, \ldots, u_k\}$ and $\mathcal{S} = \{v_1, v_2, \ldots, v_{k'}\}$, we reconstruct the WFA $\mathcal{A}$ as follows. First, let $\boldsymbol{\alpha}^T = [\bar{1}, \bar{0}, \ldots, \bar{0}] \in \mathbb{S}^k$ and $\boldsymbol{\beta}^T = [\mathbf{H}_\mathcal{B}[u_1, \varepsilon], \mathbf{H}_\mathcal{B}[u_2, \varepsilon], \ldots, \mathbf{H}_\mathcal{B}[u_k, \varepsilon]] = (\mathbf{H}_\mathcal{B}[\cdot, \varepsilon])^T$. Since $\mathcal{B}$ is complete and minimal, we have that $\operatorname{rank}([\mathbf{H}_\mathcal{B}^T | \mathbf{H}_a^T]) = \operatorname{rank}(\mathbf{H}_\mathcal{B}^T) = k$, for every $a \in \Sigma$. Therefore, by the Rouché-Capelli theorem, for each $a \in \Sigma$ there exists a unique matrix $\mathcal{A}_a \in \mathbb{S}^{k \times k}$ such that $\mathcal{A}_a \mathbf{H}_\mathcal{B} = \mathbf{H}_a$, where we can solve each $\mathcal{A}_a$ using Gaussian elimination. Thus, we obtain the matrices $\boldsymbol{\alpha}^T$, $\boldsymbol{\beta}$, and $\mathcal{A}_a$ for every $a \in \Sigma$ necessary for describing the transitions between states in a WFA. That is, we have the initial and final weight vectors, as well as the transition matrices for every letter in the alphabet, which given a state space $Q = \{q_1, q_2, \ldots, q_k\}$, does indeed define a WFA $\mathcal{A} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \{\mathcal{A}_a\})$. Since each $\mathcal{A}_a$ is computed via Gaussian elimination, each system of equations is solved in $O(k^2(k + k'))$ arithmetic operations. Therefore, since $|\mathcal{P}| \leq |\mathcal{S}|$, the entire algorithm requires $O(|\Sigma||\mathcal{P}|^2|\mathcal{S}|)$ arithmetic operations to reconstruct a WFA $\mathcal{A}$.

Theorem 2. *If* $\mathcal{B}$ *is a complete minimal Hankel basis for* $\mathbf{H}_f$*, then the reconstructed WFA* $\mathcal{A}$ *computes* f *and is minimal.*

Proof. Let $\mathcal{A}$ be the WFA generated using the reconstruction algorithm and let $\mathcal{A}' = (\boldsymbol{\alpha}', \boldsymbol{\beta}', \{\mathcal{A}'_a\})$ be a minimal WFA computing f where $\operatorname{rank}(\mathbf{H}_f) = n$. Let $\mathbf{P}_{\mathcal{A}'} \in \mathbb{S}^{|\Sigma^*| \times n}$ such that $\mathbf{P}_{\mathcal{A}'}[u, \cdot] = \boldsymbol{\alpha}'^T \mathcal{A}'_u$ and $\mathbf{S}^T_{\mathcal{A}'} \in \mathbb{S}^{n \times |\Sigma^*|}$ such that $\mathbf{S}^T_{\mathcal{A}'}[\cdot, v] = \mathcal{A}'_v \boldsymbol{\beta}'$. Then $\mathbf{H}_f = \mathbf{P}_{\mathcal{A}'} \mathbf{S}^T_{\mathcal{A}'}$ is rank factorization of $\mathbf{H}_f$ since $\operatorname{rank}(\mathbf{P}'_\mathcal{A}) = \operatorname{rank}(\mathbf{S}_{\mathcal{A}'}) = n$ by minimality of $\mathcal{A}'$. Restricting to $\mathcal{B}$, we get associated rank factorizations $\mathbf{H}_\mathcal{B} = \mathbf{P}'\mathbf{S}'^T$, where $\mathbf{P}', \mathbf{S}'^T \in \mathbb{S}^{n \times n}$. Then for every $a \in \Sigma$, $u \in \mathcal{P}$, and $v \in \mathcal{S}$, we have $\mathbf{H}_a[u, v] = \mathbf{H}_\mathcal{B}[ua, v] = \boldsymbol{\alpha}'^T \mathcal{A}'_u \mathcal{A}'_a \mathcal{A}'_v \boldsymbol{\beta}'$, and so in general $\mathbf{H}_a = \mathbf{P}' \mathcal{A}'_a \mathbf{S}'^T$.

Since the transition matrices of $\mathcal{A}$ satisfy $\mathcal{A}_a \mathbf{H}_\mathcal{B} = \mathbf{H}_a$, together with the above rank factorizations, we get $\mathcal{A}_a \mathbf{P}'\mathbf{S}'^T = \mathbf{P}' \mathcal{A}'_a \mathbf{S}'^T$. Since $\mathbf{P}'$ is a square matrix with full rank, it is invertible. Hence by the invertibility of $\mathbf{P}'$ and full column rank of $\mathbf{S}'$, it follows that $\mathcal{A}_a = \mathbf{P}' \mathcal{A}'_a \mathbf{P}'^{-1}$. Similarly, we can show $\boldsymbol{\alpha}^T = \boldsymbol{\alpha}'^T \mathbf{P}'^{-1}$ and $\boldsymbol{\beta} = \mathbf{P}' \boldsymbol{\beta}'$. Therefore we have

$$\mathcal{A}(x) = \boldsymbol{\alpha}^T \mathcal{A}_{x_1} \mathcal{A}_{x_2} \cdots \mathcal{A}_{x_n} \boldsymbol{\beta} = \boldsymbol{\alpha}'^T \mathbf{P}'^{-1} \mathbf{P}' \mathcal{A}'_{x_1} \mathbf{P}'^{-1} \mathbf{P}' \mathcal{A}'_{x_2} \mathbf{P}'^{-1} \cdots \mathbf{P} \mathcal{A}'_{x_n} \mathbf{P}'^{-1} \mathbf{P}' \boldsymbol{\beta}' = \mathcal{A}'(x).$$

That is, $\mathcal{A}$ and $\mathcal{A}'$ compute the same function f. Furthermore we have that $\mathcal{A}$ is minimal since $|Q_\mathcal{A}| = \operatorname{rank}(\mathbf{H}_\mathcal{B}) = \operatorname{rank}(\mathbf{H}_f)$. □

2.2. Reconstruction via Rank Factorizations

Previously, we reconstructed a WFA $\mathcal{A}$ from a complete minimal Hankel mask $\mathcal{B}$. If, in addition, $\mathcal{B}$ were a Hankel basis, we showed that $\mathcal{A}$ is minimal. In fact, it is possible to relax these requirements to allow for non-minimality of the Hankel mask, albeit at the expense of the runtime. The main difference between the WFA reconstructed in the previous section and the WFA resulting from the new algorithm is that the number of states of the latter is not equal to the number of prefixes $|\mathcal{P}|$, but instead to $\mathrm{rank}(\mathbf{H}_\mathcal{B})$, which may be much smaller.

Let $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ be a complete Hankel mask in Σ^* with $\varepsilon \in \mathcal{P} \cap \mathcal{S}$. We define $\mathbf{h}_\mathcal{P} \in \mathbb{S}^{|\mathcal{P}|}$ such that $\mathbf{h}_\mathcal{P}[u] = \mathbf{H}[u, \varepsilon]$ and $\mathbf{h}_\mathcal{S} \in \mathbb{S}^{|\mathcal{S}|}$ such that $\mathbf{h}_\mathcal{S}[v] = \mathbf{H}[\varepsilon, v]$. We remark that $\mathbf{h}_\mathcal{P}$ and $\mathbf{h}_\mathcal{S}$ are well-defined since $\varepsilon \in \mathcal{P} \cap \mathcal{S}$. Let $\mathrm{rank}(\mathbf{H}_\mathcal{B}) = k$, then $\mathbf{H}_\mathcal{B} = \mathbf{P}_\mathcal{B}\mathbf{S}_\mathcal{B}^T$ is a rank factorization found using Gaussian elimination [23], with $\mathbf{P}_\mathcal{B} \in \mathbb{S}^{|\mathcal{P}| \times k}$ and $\mathbf{S}_\mathcal{B} \in \mathbb{S}^{|\mathcal{S}| \times k}$.

The algorithm is as follows. For the initial and final weight vectors we solve for the unique solutions $\mathbf{S}_\mathcal{B}\boldsymbol{\alpha} = \mathbf{h}_\mathcal{P}$ and $\mathbf{P}_\mathcal{B}\boldsymbol{\beta} = \mathbf{h}_\mathcal{S}$, respectively. Existence and uniqueness of these solutions comes from the fact that $\mathbf{S}_\mathcal{B}$ contains a basis of vectors for the column space of $\mathbf{H}_\mathcal{B}$ and $\mathbf{h}_\mathcal{P}$ is a column of $\mathbf{H}_\mathcal{B}$, and a similar argument holds for $\boldsymbol{\beta}$.

For the transition matrix of each $a \in \Sigma$, we solve for the unique solution to $\mathbf{H}_a = \mathbf{P}_\mathcal{B}\mathcal{A}_a\mathbf{S}_\mathcal{B}^T$. To this end, we equivalently solve for a unique solution to $(\mathbf{S}_\mathcal{B} \otimes_K \mathbf{P}_\mathcal{B})\mathrm{vec}(\mathcal{A}_a) = \mathrm{vec}(\mathbf{H}_a)$, where $\otimes_K$ is the Kronecker product and $\mathrm{vec}(M)$ is the vector obtained by combining the columns of M into a single column vector. Since $\mathcal{S}_\mathcal{B}$ and $\mathcal{P}_\mathcal{B}$ are $|\mathcal{S}|$ by k and $|\mathcal{P}|$ by k matrices, respectively, $\mathbf{S}_\mathcal{B} \otimes_K \mathbf{P}_\mathcal{B} \in \mathbb{S}^{|\mathcal{S}||\mathcal{P}| \times k^2}$. Therefore, this new linear system of equations admits k^2 unknowns, whose coefficients satisfy $\mathrm{rank}(\mathbf{S}_\mathcal{B} \otimes_K \mathbf{P}_\mathcal{B}) = \mathrm{rank}(\mathbf{S}_\mathcal{B})\mathrm{rank}(\mathbf{P}_\mathcal{B}) = k^2$, since the number of singular values in a Kronecker product is the product of the number of singular values in its factors, and $\mathrm{rank}([\mathbf{S}_\mathcal{B} \otimes_K \mathbf{P}_\mathcal{B} | \mathrm{vec}(\mathbf{H}_a)]) = \mathrm{rank}([\mathbf{S}_\mathcal{B} \otimes_K \mathbf{P}_\mathcal{B}])$, since the columns of $\mathbf{H}_a$ are in the column space of $\mathbf{P}_\mathcal{B}$ by the completeness of the mask $\mathcal{B}$. Thus, the Rouché-Capelli theorem gives that there exists a unique solution for $\mathcal{A}_a$ which can be determined using Gaussian elimination.

Since we solve for $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $|\Sigma|$ transition matrices $\mathcal{A}_a$ for $a \in \Sigma$, we have that the algorithm usees $O(|\mathcal{S}|k + \mathcal{P}l + |\Sigma||\mathcal{P}||\mathcal{S}|k^2) = O(|\Sigma||\mathcal{P}||\mathcal{S}|k^2)$ arithmetic operations. Once again, if the mask $\mathcal{B}$ is also a Hankel basis then the reconstructed WFA $\mathcal{A}$ is minimal and computes f.

Theorem 3. *If $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ is a complete Hankel basis for $\mathbf{H}_f$, then the reconstructed WFA $\mathcal{A}$ computes f and is minimal.*

We omit the proof of this result, since it is nearly identical to the proof of Theorem 2.

2.3. Reconstruction from Noisy Hankel Matrices

In this section, we illustrate a WFA reconstruction algorithm for systems where there is some noise. While in the previous two sections it was assumed that we had access to the true transition weights of some Hankel matrix $\mathbf{H}$, in reality we would have acquired this data through observing the system, which would introduce measurement errors to the values used in our algorithms.

In light of this, in the following reconstruction algorithm we assume the inputs are approximations of Hankel sub-blocks, specified by Hankel masks. This assumption is reasonable since it resembles practical learning scenarios, where one is interested in representing a system via a WFA, given only some observed data. The algorithm is a variation of the one given in the Section 2.2; however, it heavily relies on a singular value decomposition (SVD).

Note that the computation of a SVD requires that $\mathbb{S}$ be a field obtained as the intersection of real closed fields [34]. Furthermore, since most problems involve $\mathbb{S} = \mathbb{R}$, we the give the algorithm for this case. The steps can nevertheless be generalized to fields wherein a SVD is computable, such as $\mathbb{C}$.

Let $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ be an arbitrary Hankel mask. Also, suppose that $\widehat{\mathbf{H}}_\mathcal{B}$, $\widehat{\mathbf{H}}_a$, $\widehat{\mathbf{h}}_\mathcal{P}$, and $\widehat{\mathbf{h}}_\mathcal{S}$ are approximate versions of $\mathbf{H}_\mathcal{B}$, $\mathbf{H}_a$, $\mathbf{h}_\mathcal{P}$, and $\mathbf{h}_\mathcal{S}$. That is, let $\widehat{\mathbf{H}}_\mathcal{B} = \mathbf{H}_\mathcal{B} + \mathbf{E}_\mathcal{B}$, $\widehat{\mathbf{H}}_a = \mathbf{H}_a + \mathbf{E}_a$,

$\widehat{\mathbf{h}}_{\mathcal{P}} = \mathbf{h}_{\mathcal{P}} + \mathbf{e}_{\mathcal{P}}$, and $\widehat{\mathbf{h}}_{\mathcal{S}} = \mathbf{h}_{\mathcal{S}} + \mathbf{e}_{\mathcal{S}}$, where $\mathbf{E}_{\mathcal{B}}$, $\mathbf{E}_a$, $\mathbf{e}_{\mathcal{P}}$, and $\mathbf{e}_{\mathcal{S}}$ are noise matrices which represent the measurement error associated to observing a given word in Σ^*.

A naive idea would be to try to reconstruct a WFA from the approximations using the rank factorization technique described above. The flaw in such an approach is that rank$(\widehat{\mathbf{H}}_{\mathcal{B}})$ may be much larger than rank$(\mathbf{H}_{\mathcal{B}})$, thereby making the algorithm slow and the result certainly not minimal. Not to mention the outputted WFA may be unacceptably inaccurate.

A better idea would be to try to reconstruct a WFA with fewer than rank$(\widehat{\mathbf{H}}_{\mathcal{B}})$ states and if the level of noise is small, agrees, up to some tolerable margin of error, with a reconstruction that used $\mathbf{H}_{\mathcal{B}}$. To this end, we use a SVD to replace the rank factorization in the previous algorithm with a low rank approximation of $\widehat{\mathbf{H}}_{\mathcal{B}}$. The algorithm is as follows.

Let the Hankel mask $\mathcal{B}$, the number of states k' in the reconstructed WFA, and the approximated Hankel sub-blocks described above be the inputs. First, we compute the SVD of $\widehat{\mathbf{H}}_{\mathcal{B}}$ and obtain a k' rank approximation $\widehat{\mathbf{H}}_{\mathcal{B}} \approx \widehat{\mathbf{U}}\widehat{\mathbf{D}}\widehat{\mathbf{V}}^T$, where $\widehat{\mathbf{D}} = \text{diag}(\widehat{\mathfrak{s}}_1, \widehat{\mathfrak{s}}_2, \ldots, \widehat{\mathfrak{s}}_{k'})$ is the diagonal matrix consisting of the k' largest singular values of $\widehat{\mathbf{H}}_{\mathcal{B}}$, and $\widehat{\mathbf{U}} \in \mathbb{R}^{|\mathcal{P}| \times k'}$ and $\widehat{\mathbf{V}} \in \mathbb{R}^{|\mathcal{S}| \times k'}$ contain the left and right singular vectors, respectively. Note that since the singular values are an indication of the amount of information contained in each dimension of the column space, $\widehat{\mathbf{U}}\widehat{\mathbf{D}}\widehat{\mathbf{V}}^T$ is the best approximation of $\widehat{\mathbf{H}}_{\mathcal{B}}$ with rank k'.

Defining $\widehat{\mathbf{P}}_{\mathcal{B}} = \widehat{\mathbf{U}}\widehat{\mathbf{D}}$ and $\widehat{\mathcal{S}}_{\mathcal{B}} = \widehat{\mathbf{V}}$ we have that $\widehat{\mathbf{P}}\widehat{\mathbf{S}}$ is a rank factorization of $\widehat{\mathbf{U}}\widehat{\mathbf{D}}\widehat{\mathbf{V}}^T$. From here, we cannot simply continue as in the algorithm in Section 2.2, since the linear system of equations $\widehat{\mathcal{S}}_{\mathcal{B}}\widehat{\boldsymbol{\alpha}} = \widehat{\mathbf{h}}_{\mathcal{P}}$, $\widehat{\mathbf{P}}_{\mathcal{B}}\widehat{\boldsymbol{\beta}} = \widehat{\mathbf{h}}_{\mathcal{S}}$, and $(\widehat{\mathbf{S}}_{\mathcal{B}} \otimes \widehat{\mathbf{P}}_{\mathcal{B}})\text{vec}(\widehat{\mathcal{A}}_a) = \text{vec}(\widehat{\mathbf{H}}_a)$ that need to be solved may not have unique solutions, but instead may have none or infinitely many. Consequently, we calculate the Moore-Penrose pseudo-inverses $\widehat{\mathbf{P}}_{\mathcal{B}}^+$ of $\widehat{\mathbf{P}}_{\mathcal{B}}$ and $\widehat{\mathbf{S}}_{\mathcal{B}}^+$ of $\widehat{\mathbf{S}}_{\mathcal{B}}$, which for any linear system of equations $\mathbf{Mx} = \mathbf{b}$, provides a solution $x = \mathbf{M}^+ b$ if one exists, and otherwise minimizes $\|\mathbf{Mx} - \mathbf{b}\|$.

Since $\widehat{\mathbf{P}}_{\mathcal{B}}\widehat{\mathbf{S}}_{\mathcal{B}}$ is a rank factorization, we can easily calculate $\widehat{\mathbf{P}}_{\mathcal{B}}^+ = (\widehat{\mathbf{U}}\widehat{\mathbf{D}})^+ = \widehat{\mathbf{D}}^{-1}\widehat{\mathbf{U}}^T$, $\widehat{\mathbf{S}}_{\mathcal{B}}^+ = \widehat{\mathbf{V}}^+ = \widehat{\mathbf{V}}^T$, and $(\widehat{\mathbf{P}}_{\mathcal{B}} \otimes_K \widehat{\mathbf{S}}_{\mathcal{B}})^+ = (\widehat{\mathbf{P}}_{\mathcal{B}}^+ \otimes_K \widehat{\mathbf{S}}_{\mathcal{B}}^+) = (\widehat{\mathbf{V}}^T \otimes_K \widehat{\mathbf{D}}^{-1}\widehat{\mathbf{U}}^T)$. Then we get, at worst, good approximations $\widehat{\boldsymbol{\alpha}} = \widehat{\mathcal{S}}_{\mathcal{B}}^+\widehat{\mathbf{h}}_{\mathcal{P}}$, $\widehat{\beta} = \widehat{\mathbf{P}}_{\mathcal{B}}^+\widehat{\mathbf{h}}_{\mathcal{S}}$, and $\widehat{\mathcal{A}}_a = \widehat{\mathbf{D}}^{-1}\widehat{\mathbf{U}}^T\widehat{\mathbf{H}}_a\widehat{\mathbf{V}}^T$.

Since the bottleneck of this algorithm is the SVD calculation, which requires $O(|\mathcal{P}||\mathcal{S}|k')$ arithmetic operations, and we need to calculate $\widehat{\mathbf{A}}_a$ for each $a \in \Sigma$, we have that the entire algorithm has the computational complexity $O(|\Sigma||\mathcal{P}||\mathcal{S}|k')$.

In the remainder of this section, we provide an error bound for the approximate WFA produced and we show that in the absence of any noise, the above reconstruction algorithm returns a minimal WFA $\mathcal{A}$ which computes f. Note that different choices for norms of vectors and matrices produce slightly different analyses; however, for the purpose of this paper we choose the Euclidean norm for vectors and the Frobenius norm for matrices. We define $\varepsilon_{\mathcal{B}} = \|\mathbf{E}_{\mathcal{B}}\|_F$, $\varepsilon_a = \|\mathbf{E}_a\|_F$ for every $a \in \Sigma$, $\varepsilon_{\mathcal{P}} = \|\mathbf{e}_{\mathcal{P}}\|_2$, $\varepsilon_{\mathcal{S}} = \|\mathbf{e}_{\mathcal{S}}\|_2$, and, for convenience, $\varepsilon = \max\{\varepsilon_{\mathcal{B}}, \varepsilon_{a_1}, \varepsilon_{a_2}, \ldots, \varepsilon_{a_r}\varepsilon_{\mathcal{P}}, \varepsilon_{\mathcal{S}}\}$ for $\Sigma = \{a_1, a_2, \ldots, a_r\}$.

We now show that when there is no noise, the reconstruction algorithm returns a minimal WFA $\mathcal{A}$ which computes f. Suppose $k' = \text{rank}(\mathbf{H}_{\mathcal{B}}) = k$ and $\varepsilon = 0$, then $\widehat{\mathbf{H}}_{\mathcal{B}} = \mathbf{H}_{\mathcal{B}}$ and the k rank SVD gives $\mathbf{H}_{\mathcal{B}} = \mathbf{P}\mathbf{S}^T = (\mathbf{U}\mathbf{D})(\mathbf{V})^T$. Hence we get a WFA with k states characterized by $\boldsymbol{\alpha} = \mathbf{V}^T\mathbf{h}_{\mathcal{P}}$, $\boldsymbol{\beta} = \mathbf{D}^{-1}\mathbf{U}^T\mathbf{h}_{\mathcal{S}}$, and $\mathcal{A}_a = \mathbf{D}^{-1}\mathbf{U}^T\mathbf{H}_a\mathbf{V}^T$ for each $a \in \Sigma$. By invoking Theorem 4, we get the following corollary.

Corollary 2. *Suppose $k' = rank(\mathbf{H}_{\mathcal{B}})$ and $\varepsilon = 0$. If $\mathcal{B}$ is a complete basis for $\mathbf{H}_f$, then the reconstruct WFA $\mathcal{A}$ computes f and is minimal.*

Furthermore, we have the following error bound guarantee.

Theorem 4. *Suppose $k' = rank(\mathbf{H}_{\mathcal{B}})$, and let $\mathcal{A}$ and $\widehat{\mathcal{A}}$ be the WFAs obtained in the $\varepsilon = 0$ and noisy cases, respectively. Then as $\varepsilon \to 0$, we have the following guarantee on the maximum error of $\widehat{\mathcal{A}}$*

$$\Delta = \max\left\{\|\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}}\|_2, \|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\|_2, \|\mathcal{A}_{a_1} - \widehat{\mathcal{A}}_{a_1}\|_F, \|\mathcal{A}_{a_2} - \widehat{\mathcal{A}}_{a_2}\|_F, \ldots, \|\mathcal{A}_{a_r} - \widehat{\mathcal{A}}_{a_r}\|_F\right\} = O(\varepsilon).$$

The proof of this result is very technical and requires several pages of a perturbation theory-based analysis of the singular values and vectors - it is therefore omitted (see [9], Chapter 5).

3. Algorithms for Learning WFAs

Now that we have seen many different techniques for reconstructing WFAs given a variety of assumptions, in this section we illustrate applications of these reconstruction algorithms to various algorithms for learning WFAs. Of the three presented, the first algorithm assumes $\mathbb{S}$ is an arbitrary field, while the remaining two assume $\mathbb{S} = \mathbb{R}$. We note that in the spirit of keeping this paper relatively concise, informative, and not overly technical, we omit the proofs of the error bound guarantees for the second and third algorithms presented, and humbly apologize for disappointing the reader.

3.1. Learning WFAs From Queries

The first algorithm presented is a generalization of Angluin's L^* algorithm for learning DFAs of regular languages via membership and equivalence queries [4]. As in the case of the L^* algorithm, we have a *learner* and a *minimally adequate teacher*. Given a rational function $f : \Sigma^* \to \mathbb{S}$, the learner is allowed to make two kinds of queries. The first is a membership query MQ_f, consisting of a string $w \in \Sigma^*$; the answer is the value $f(w)$. The second is an equivalence query EQ_f, consisting of a description of a WFA $\mathcal{A}$; the answer is yes if $\mathcal{A}$ computes f, and a counter-example $w \in \Sigma^*$ with $f(w) \neq \mathcal{A}(w)$ otherwise.

Instead of building a closed and consistent observation table, the algorithm builds a complete minimal Hankel basis $\mathcal{B}$ for $\mathbf{H}_f$. Furthermore, many complete minimal Hankel masks may be constructed before ultimately constructing one which is a Hankel basis. Given a Hankel mask $\mathcal{B}'$, the algorithm uses MQ_f to augment $\mathcal{B}'$ until it is complete and minimal. At this point, the algorithm uses EQ_f to determine whether $\mathbf{H}_{\mathcal{B}'}$ computes f. If yes, it returns the associated WFA and terminates; otherwise, it augments $\mathcal{B}'$ using EQ_f according to the counter-example provided.

For convenience, for $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ and $\mathcal{B}' = (\mathcal{P}', \mathcal{S}')$ we write $\mathcal{B} \subset \mathcal{B}'$ if $\mathcal{P} \subset \mathcal{P}'$ and $\mathcal{S} \subset \mathcal{S}'$. Furthermore, we see (and shall prove at the end) that given a sequence of complete minimal Hankel masks $\mathcal{B}_0 \subset \mathcal{B}_1 \subset \cdots \subset \mathcal{B}_d$ constructed by the algorithm, we have $\text{rank}(\mathbf{H})_{\mathcal{B}_{i+1}} > \text{rank}(\mathbf{H})_{\mathcal{B}_i}$. Therefore the algorithm is guaranteed to terminate in at most $d < \text{rank}(\mathbf{H}_f)$ steps.

Letting $\mathcal{B}_0 = (\{\varepsilon\}, \{\varepsilon\})$, we now describe the main subroutine of the algorithm. Given $\mathcal{B}_i = (\mathcal{P}_i, \mathcal{S}_i)$, the algorithm first uses MQ_f to request values of the words $w = uv$ for all $u \in \mathcal{P}_i$ and $v \in \mathcal{S}_i$ in order to fill the Hankel sub-block $\mathbf{H}_{\mathcal{B}_i}$, then reconstructs a WFA $\mathcal{A}_i$ using the reconstruction algorithm in Section 2.1. Once done, the algorithm queries EQ_f with $\mathcal{A}_i$. If it answers yes, the algorithm terminates and returns $\mathcal{A}_i$. Otherwise, it answers with a counter-example $w \in \Sigma^*$ such that $\mathcal{A}_i \neq f(w)$.

To process the counter-example, the algorithm first finds a decomposition $w = uav$ where u is the longest prefix of w in $\mathcal{P}_i$. Next, it sets $\mathcal{S}_{i+1} = \mathcal{S}_i \cup \{y : \exists x \in \Sigma^* \ v = xy\}$. That is, it adds all suffixes of v to $\mathcal{S}_i$. Finally, starting from $\mathcal{P}_{i+1} = \mathcal{P}$, while $\mathcal{B}_{i+1} = (\mathcal{P}_{i+1}, \mathcal{S}_{i+1})$ is not complete, it adds to $\mathcal{P}_{i+1}$ prefixes $ua \in \mathcal{P}_{i+1}\Sigma$ such that $\text{rank}([\mathbf{H}_{\mathcal{B}_{i+1}}^T | \mathbf{H}_a[u, :]^T]) = \text{rank}(\mathbf{H}_{\mathcal{B}_{i+1}}) + 1$. That is, it adds prefixes of the form ua which increment the rank of the Hankel sub-block by one.

By construction, we have that $\mathcal{B}_{i+1}$ is complete and minimal because the rank of its corresponding Hankel sub-block is increased by exactly one for each prefix in $\mathcal{P}_{i+1}$. Also by construction we have that $\mathcal{P}_{i+1}$ is prefix-closed and $\mathcal{S}_{i+1}$ is suffix-closed. It remains to show that the algorithm eventually terminates. To this end, we first prove the following lemma.

Lemma 1. *Let $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ be a complete minimal Hankel mask for $\mathbf{H}_f$, where $\mathcal{P}$ is prefix-closed and $\mathcal{S}$ is suffix-closed. Then the WFA $\mathcal{A}$ reconstructed from $\mathbf{H}_{\mathcal{B}}$ and $\mathbf{H}_\Sigma$ satifies $f(uv) = \mathcal{A}(uv)$ and $f(uav) = \mathcal{A}(uav)$ for all $u \in \mathcal{P}$, $v \in \mathcal{S}$, and $a \in \Sigma$.*

Proof. Let $k = \text{rank}(\mathbf{H}_{\mathcal{B}}) = |\mathcal{P}|$ and $\mathcal{P} = \{u_1, u_2, \ldots, u_k\}$ with $u_1 = \varepsilon$ and $|u_i| \leq |u_{i+1}|$. Similarly, let $k' = |\mathcal{S}|$, $\mathcal{S} = \{v_1, v_2, \ldots, v_{k'}\}$ with $v_1 = \varepsilon$ and $|v_i| \leq |v_{i+1}|$. Let $\mathbf{H}_{\mathcal{A}} = \mathbf{P}\mathcal{S}^T$ be the induced factorization. We claim that $\mathbf{P}_{\mathcal{P}}$ is the identity matrix, where $\mathbf{P}_{\mathcal{P}} \in \mathbb{S}^{|\mathcal{P}| \times k}$ is the sub-block of $\mathbf{P}$ indexed by prefixes in $\mathcal{P}$. We shall prove by induction on i that for $1 \leq i \leq k$ we have $\mathbf{P}_{\mathcal{P}}[u_i, \cdot] = \mathbf{e}_i^T$, where $\mathbf{e}_i$ is the ith indicator vector.

By construction of $\mathcal{A}$, the base case holds since $\mathbf{P}_{\mathcal{P}}[u_1, \cdot] = \mathbf{P}[\varepsilon, \cdot] = \boldsymbol{\alpha}^T = \mathbf{e}_1^T$. Now suppose the claim is true for all $1 \leq j \leq i$. Since for every $1 \leq j \leq i$, $|u_{i+1}| \geq |u_j|$ and $\mathcal{P}$ is prefix-closed, we have that there exists $a \in \Sigma$ and $1 \leq j \leq i$ such that $u_{i+1} = u_j a$. Hence $\mathbf{P}_{\mathcal{P}}[u_{i+1}, \cdot] = \mathbf{P}[u_j a, \cdot] = \mathbf{P}[u_j, \cdot]\mathcal{A}_a = \mathbf{e}_j^T \mathcal{A}_a = \mathcal{A}_a[j, \cdot]$ and by $\mathbf{H}_a[u_j, \cdot] = \mathbf{H}_{\mathcal{B}}[u_j a, \cdot] = \mathbf{H}_{\mathcal{B}}[u_{i+1}, \cdot]$, solving the linear system of equations $\mathcal{A}$ gives $\mathcal{A}_a[j, \cdot] = \mathbf{e}_{i+1}^T$.

Next, we similarly define $\mathbf{S}_{\mathcal{S}} \in \mathbb{S}^{|\mathcal{S}| \times k}$ to be the sub-block of $\mathbf{S}$ indexed by suffixes in $\mathcal{S}$. Since $\mathbf{P}_{\mathcal{P}} = \mathbf{I}$, to show the first statement of the lemma, it suffices to show that $\mathbf{S}_{\mathcal{S}}^T = \mathbf{H}_{\mathcal{B}}$. Again we proceed by induction on i. By construction of $\mathcal{A}$, the base case holds since $\mathbf{S}_{\mathcal{S}}[v_1, \cdot] = \mathbf{S}[\varepsilon, \cdot] = \boldsymbol{\beta}^T = \mathbf{H}_{\mathcal{B}}[\cdot, \varepsilon]^T$. Now suppose the claim is true for all $1 \leq j \leq i$. Since for every $1 \leq j \leq i$, $|v_{i+1}| \geq |v_i|$ and $\mathcal{S}$ is suffix-closed, we have that there exists $a \in \Sigma$ and $1 \leq j \leq i$ such that $v_{i+1} = av_j$. Hence $\mathbf{S}_{\mathcal{S}}[v_{i+1}, \cdot] = \mathbf{S}[av_j, \cdot] = \mathbf{S}[v_j, \cdot]\mathcal{A}_a^T = \mathbf{H}_{\mathcal{B}}[\cdot, v_j]^T \mathcal{A}_a^T = \mathbf{H}_a[\cdot, v_j]^T = \mathbf{H}_{\mathcal{B}}[\cdot, av_j]^T = \mathbf{H}_{\mathcal{B}}[\cdot, v_{i+1}]^T$.

Therefore we indeed have $f(uv) = \mathcal{A}(uv)$ for every $u \in \mathcal{P}$ and $v \in \mathcal{S}$. To conclude the proof, note that we have $\mathcal{A}[u_i av] = \mathbf{P}_{\mathcal{P}}[u_i, \cdot]\mathcal{A}_a \mathbf{S}_s[v, \cdot]^T = \mathbf{e}_i^T \mathcal{A}_a \mathbf{H}_{\mathcal{B}}[\cdot, v] = \mathbf{e}_i^T \mathbf{H}_a[\cdot, v] = \mathbf{H}_a[u_i, v]$ for $u_i \in \mathcal{P}$, $v \in \mathcal{S}$, and $a \in \Sigma$. □

Given this lemma, the next result follows.

Lemma 2. *Let $\mathcal{B}_i' = (\mathcal{P}_i, \mathcal{S}_i)$, where S_{i+1} is the set of suffixes obtained after processing the counter-example w received from the $(i+1)$th call to EQ_f, then rank$([\mathbf{H}_{\mathcal{B}_i'}^T | \mathbf{H}_{\Sigma}^T]) >$ rank$(\mathbf{H}_{\mathcal{B}_i'}^T)$.*

Proof. We proceed by contradiction. Suppose $\text{rank}([\mathbf{H}_{\mathcal{B}_i'}^T | \mathbf{H}_{\Sigma}^T]) = \text{rank}(\mathbf{H}_{\mathcal{B}_i'}^T)$ and let $\mathcal{A}_i'$ be the WFA outputted by the reconstruction algorithm in Section 2.1. Since $\mathcal{P}_i = \mathcal{P}_i'$, both $\mathcal{B}_i$ and $\mathcal{B}_i'$ are complete and minimal, while $\mathcal{S}_i \subset \mathcal{S}_i'$. Therefore $\mathcal{A}_i$ and $\mathcal{A}_i'$ necessarily compute the same function. Choose $w = uav$ such that $u \in \mathcal{P}_i$ and $v \in \mathcal{S}_{i+1}$. Then in the matrix $\mathbf{H}_a$ used in the reconstruction of $\mathcal{A}_i'$, we have $\mathbf{H}_a[u, v] = f(w)$ and by Lemma 1 we have $\mathcal{A}_i'(w) = f(w)$. However this is a contradiction to the fact that $\mathcal{A}(w) \neq f(w)$ and so we conclude that $\text{rank}([\mathbf{H}_{\mathcal{B}_i'}^T | \mathbf{H}_{\Sigma}^T]) > \text{rank}(\mathbf{H}_{\mathcal{B}_i'}^T)$. □

To bound the number of queries made by the algorithm, we first note that the number of calls to EQ_f is $O(n)$ since one call is made per Hankel mask built. Also, note that since $\mathcal{B}_i \subset \mathcal{B}_{i+1}$, each value in the sub-block $\mathbf{H}_{\mathcal{B}_d}$ of the outputted WFA corresponds to exactly one call to MQ_f. Finally, let L be the length of the longest counter-example returned any call to EQ_f, then $|\mathcal{S}_d| \leq 1 + dL$. Since $|\mathcal{P}_d| = \text{rank}(\mathbf{H}_f) = n$, the total number of calls to EQ_f is therefore $O(|\Sigma|n^2 L)$.

For further applications, see [11, 12]. For details of an improved bound $O(|\Sigma|n^2 \log(L))$ to the number of calls to MQ_f, see [15].

3.2. Learning Stochastic WFAs from I.I.D. Samples

Definition 9. We say that a WFA $\mathcal{A}$ is a *stochastic* WFA over Σ if it computes a probability distribution over Σ^*.

In this section, we illustrate an algorithm for learning stochastic WFAs in the common scenario where the learner receives a finite set of strings sampled i.i.d. from the stochastic WFA we wish to learn. The key idea of the algorithm is to use the reconstruction algorithm described in Section 2.3 on a sub-block of the Hankel matrix of the target WFA estimated using the sample.

The algorithm is as follows. Let $\mathcal{A}$ be a fixed unknown stochastic WFA. Let the input be a sample $S = (w_1, w_2, \ldots, w_m) \in (\Sigma^*)^m$ of m strings sampled i.i.d. from the probability distribution computed by $\mathcal{A}$, the alphabet Σ, a number of states n in the reconstructed WFA, and a finite Hankel mask $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ where $n \leq \min\{|\mathcal{P}|, |\mathcal{S}|\}$. To build estimates of sub-blocks of the Hankel matrix of $\mathcal{A}$, for each $u \in \mathcal{P}$ and $v \in \mathcal{S}$, we assign

$$\widehat{\mathbf{H}}_{\mathcal{B}}[u, v] = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}[w_i = uv],$$

where $\mathbb{I}[w_i = uv] = 1$ if $w_i = uv$ and $\mathbb{I}[w_i = uv] = 0$ otherwise. That is, we assign to each uv the relative frequency of the decomposition in S. Similar assignments are done for $\widehat{\mathbf{H}}_a$, $\widehat{\mathbf{h}}_{\mathcal{P}}$, and $\widehat{\mathbf{h}}_{\mathcal{S}}$.

The following theorem provides a *probably approximately correct* (PAC) learning guarantee. Let ε be the desired precision, δ the desired confidence, $\mathfrak{s}_n(\mathbf{H}_{\mathcal{B}})$ the nth singular value of $\mathbf{H}_{\mathcal{B}}$, and L a string length.

Theorem 5. *Let $\varepsilon > 0$, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ over the draw of a sample S of size $m \geq p(|\Sigma|, n, |\mathcal{P}|, |\mathcal{S}|, 1/\mathfrak{S}_n(\mathbf{H}_{\mathcal{B}}), L, 1/\varepsilon, \log(1/\delta))$ from the target probability distribution f, where p is a polynomial, we have*

$$\sum_{w \in \Sigma^{\leq L}} |f(w) - \widehat{\mathcal{A}}(w)| \leq \varepsilon,$$

where $\widehat{\mathcal{A}}$ is the reconstructed WFA.

See [5, 9, 24] for detailed proofs.

3.3. Learning WFAs from String-Value Pairs

In this last section, we demonstrate a generalization of the learning algorithm depicted in the previous section. The idea of the algorithm is to construct a WFA which minimizes the expected value of a chosen loss function applied to a sample of string-value pairs drawn from some distribution. We use the loss function to determine how close the value of a label is to the value of word computed by the WFA. We remark that this is an agnostic setting as we do not assume the sample is drawn from a distribution necessarily computed by a WFA.

Let $S = ((w_1, y_1), (w_2, y_2), \ldots, (w_m, y_m)) \in (\Sigma^* \times \mathbb{S})^m$ be a a sample of m string-value pairs sampled i.i.d. from some distribution $\mathcal{D}$, let $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ be a loss function (some usual choices are quadratic or absolute loss). We want to reconstruct a WFA $\mathcal{A}$ such that $\mathbb{E}_{(w,y)\sim\mathcal{D}}[l(\mathcal{A}(w), y)]$ is minimized. Note that while it is possible to find a WFA which agrees with all labels, such a reconstruction is an overfitting; in fact such a WFA may be very large and [28] has shown that the problem of minimizing such a WFA is NP-hard. To avoid overfitting in general, the algorithm restricts the number of states of the output WFA and the norm of the Hankel matrices involved in the calculations.

The algorithm is as follows. We assume the learner receives as input a sample S as defined above, the alphabet Σ, a number of states n in the reconstructed WFA, and a finite Hankel mask $\mathcal{B} = (\mathcal{P}, \mathcal{S})$ with $\varepsilon \in \mathcal{P} \cap \mathcal{S}$, a convex loss function l, and a regularization parameter $\lambda > 0$. First, the algorithm builds a modified basis $\mathcal{B}' = (\mathcal{P}', \mathcal{S})$, where $\mathcal{P}' = \mathcal{P} \cup \mathcal{P}\Sigma$, and sample $S' = \{(w_i, s_i) \in S : w_i \in \mathcal{P}\mathcal{S}\}$, which only consists of samples which can be described by the mask. Then by solving the convex optimization problem, it calculates

$$\widehat{\mathbf{H}}_{\mathcal{B}'} \in \operatorname*{arg\,min}_{\mathbf{H} \in \mathbb{H}_{\mathcal{B}'}} \frac{1}{|S'|} \sum_{(w,y) \in S'} l(\mathbf{H}(w), y) + \lambda \|\mathbf{H}\|_*,$$

where $\mathbb{H}_{\mathcal{B}'} = \left\{\mathbf{H} \in \mathbb{R}^{|\mathcal{P}'| \times |\mathcal{S}|}\right\}$ is a set of Hankel matrices, $\mathbf{H}(w) = \mathbf{H}[u, v]$ for an arbitrary decomposition $w = uv$ where $u \in \mathcal{P}$ and $v \in \mathcal{S}$, and $\|\mathbf{H}\|_*$ is the nuclear norm of $\mathbf{H}$ (the sum of its singular values).

Next, the algorithm uses $\widehat{\mathcal{B}}'$ to work backwards in order to obtain the Hankel sub-blocks $\widehat{\mathbf{H}}_{\mathcal{B}}$, $\widehat{\mathbf{H}}_a$ for $a \in \Sigma$, $\widehat{\mathbf{h}}_{\mathcal{P}}$, and $\widehat{\mathbf{h}}_{\mathcal{S}}$ associated with the Hankel mask $\mathcal{B}$. From here, it reconstructs a WFA $\mathcal{A}$ according to the algorithm described in Section 2.3.

The nuclear norm is used as a regularization term for finding the Hankel matrix $\widehat{\mathbf{H}}_{\mathcal{B}'}$ for several reasons. First the nuclear norm is a convex surrogate for the rank function commonly used in other machine learning algorithms [21]. By Theorem 1, low-rank Hankel matrices induce WFA with few states, thus choosing the nuclear norm helps in minimizes the reconstructed WFA. Second, is the following theorem.

Let $M > 0$ and define $\tau_M : \mathbb{R} \to \mathbb{R}$ where $\tau_M(y) = \text{sign}(y)M$ if $|y| < M$ and $\tau_M(y) = y$ otherwise. This function is used to bound the tails of the distribution from which we sample the labels. Let S be a sample of m string-value pairs. Given a decomposition $w_i = u_i v_i$ for any $1 \leq i \leq m$, define $U_S = \max_u \in \Sigma^* | \{i : u_i = u\} |$ and $V_S = \max_{v \in \Sigma^*} | \{i : v_i = v\} |$. Then $W_s = \min \max \{U_S, V_S\}$ is an indication of the complexity of S, where the minimum is taken over all possible decomposition of the strings $w_i \in S$. Finally for some constant $R > 0$ we define the following class of functions

$$\mathcal{F}_R = \{f(w) = \tau_M(\mathcal{A}(w)) : \mathcal{A}\ WFA, \|\mathbf{H}_f\|_* \leq R\}.$$

This is a set of functions whose tails resemble that of the distribution of a stochastic WFA. Despite that these distributions are not necessarily computed by WFAs, they behave as such. We have the following result.

Theorem 6. *Let l_1 be the absolute loss function. Suppose there exists $M > 0$ such that $\mathbb{P}_{(w,y)\sim\mathcal{D}}[|y| \leq M] = 1$, then for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of an i.i.d. sample S of size m from $\mathcal{D}$, for all $f \in \mathcal{F}_R$ we have*

$$\mathbb{E}_{(w,y)\sim D}[l_1] \leq \frac{1}{m}\sum_{i=1}^{m} l_1(f(w_i), y_i) + 3M\sqrt{\frac{\log(2/\delta)}{2m}} + O\left(\frac{R(\log(m+1) + \sqrt{W_s \log(m+1)})}{m}\right)$$

A similar result is proven in [10] using a Frobenius norm regularizer instead of the nuclear norm. For a detailed proof this result, see [8].

4. Conclusion

We delineated a review, of a review, of multiple key techniques used in the problem of learning WFAs. Hopefully we clearly motivated and explained definitions, algorithms, and proofs for the reader, while allowing for further readings of more involved topics.

References

[1] Allauzen, C., Mohri, M., Riley, M.: Statistical modeling for unit selection in speech synthesis. In: Proceedings of ACL (2004)

[2] Allauzn, C., Mohri, M. Talwalkar, A.: Sequence kernels for predicting protein essentiality. In: Proceedings of ICML (2008)

[3] Aminof, B., Kupferman, O., Lampert, R.: Formal analysis of online algorithms. In: Proceedings of AVA (2011)

[4] Angluin, D.: Learning regular sets from queries and counterexamples. Information and computation 75(2) (1987)

[5] Bailly, R.: Méthodes spectrales pour l'inférence grammaticale probabiliste de langages stochastiques rationnels. Ph.D. thesis, Aix-Marseille Université (2011)

[6] Bailly, R., Denis, F., Ralaivola, L.: Grammatical inference as a principal component analysis problem. In: Proceedings of ICML (2009)

[7] Balle, B., Mohri, M.: Learning Weighted Automata. Springer (2015)ref

[8] Balle, B. Mohri, M.: On the Rademacher complexity of weighted automata. In: Proceedings of ALT (2015)

[9] Balle, B.: Learning Finite-State Machines: Statistical and Algorithmic Aspects. Ph.D. thesis, Universitat Politecnica de Catalunya (2013)

[10] Balle, B., Mohri, M.: Spectral learning of general weighted automata via constrained matrix completion. In: Proceedings of NIPS (2012)

[11] Beimel, A., Bergadano, F., Bshoutty, N.H., Kushilevitz, E., Varricchio, S.: On the applications of multiplicity automata in learning. In: Proceeding FOCS (1996)

[12] Beimel, A., Bergadano, F., Bshoutty, N.H., Kushilevitz, E., Varricchio, S.: Learning functions represented as multiplicity automata. Journal of the ACM 47(3) (2000)

[13] Bergadano, F., Varricchio, S.: Learning behaviors of automata from multiplicity and equivalence queries. In: Proceedings of CIAC, vol. 778. Springer (1994)

[14] Bergadano, F., Varricchio, S.: Learning behaviors of automata from multiplicity and equivalence queries. SIAM Journal on Computing 25(6) (1996)

[15] Bisht, L., Bshouty, N.H., Mazzawi, H.: On optimal learning algorithms for multiplicity automata. In: Proceedings of COLT (2006)

[16] Breuel, T.M.: The OCRopus open source OCR system. In: Proceedings of IS&T/SPIE (2008)

[17] Cardon, A., Crochemore, M.: Détermination de la représentation standard d'une série reconnaissable. ITA 14(4), 371-379 (1980)

[18] Carlyle, J.W., Paz, A.,: Realizations by stochastic finite automata. J. Comput. Syst. Sci. 5(1) (1971)

[19] Cortes, C., Haffner, P., Mohri, M.: Rational kernels: Theory and algorithms. Journal of Machine Learning Research 5 (2004)

[20] Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.J.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press (1974)

[21] Fazel, M.: Matrix rank minimization with applications. Ph.D. thesis, Standford University (2002)

[22] Fliess, M.: Matrices de Hankel. Journal de Mathématiques Pyres et Appliquées 53 (1974)

[23] Golub, G., Loan, C.V.: Matrix Computations. John Hopkins University Press (1983)

[24] Hsu, D., Kakade, S.M., Zhang, T.: A spectral algorithm for learning hidden markov models. In: Proceedings of COLT (2009)

[25] II, K.C., Kari, J.: Image compression using weighted finite automata. Computers & Graphics 17(3) (1993)

[26] Kaplan, R.M., Kay, M.: Regular models of phonological rule systems. Computational Linguistics 20(3) (1994)

[27] Karttunen, L.: The replace operator. In: Proceedings of ACL (1995)

[28] Kiefer, S., Marusic, I. Worrell, J.: Minimisation of multiplicity tree automata.: In: Proceedings of FOSSACS (2015)

[29] Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics 23(2) (1997)

[30] Mohri, M.: Weighted automata algorithms. In: Handbook of Weighted Automata. Springer (2009)

[31] Mohri, M., Pereira, F., Riley, M.: Speech recognition with weighted finite-state transducers. In: Handbook on Speech Processing and Speech Comm. Springer (2008)

[32] Mohri, M., Pereira, F.C.N.: Dynamic compilation of weighted context0free grammars. In: Proceedings of COLING-ACL (1998)

[33] Mohri, M., Sproat, R.: An efficient compiler for weighted rewrite rules. In: Proceedings of ACL (1996)

[34] Mornhinweg, D., Shapiro, D.B., Valente, K.: The principal axis theorem over arbitrary fields. American Mathematical Monthly (1993)

[35] Pereira, F., Riley, M.: Speech recognition by composition of weighted finite automata. In: Finite-State Language Processing. MIT Press (1997)

[36] Schützenberger, M.P.: On a special class of recurrent events. The Annals of Mathematical Statistics 32(4) (1961)

[37] Schützenberger, M.P.: On the definition of a family of automata. Information and Control 4 (1961)

[38] Sproat, R.: A finite-state architecture for tokenization and grapheme-to-phoneme conversion in multilingual text analysis. In: Proceedings of the ACL SIGDAT Workshop. ACL (1995)