Automata learning: from probabilistic to quantum
Chu, W.

Chapter 1

Introduction

In computer science, the concept of learning stands as a cornerstone, as it forms the very foundations of intelligent systems, algorithms, and applications. In essence, the concept of learning is not confined to the conventional human understanding of the term, often associated with the cognitive processes of acquiring knowledge, skills, or understanding through study, intuition, or experience. Rather, computer science takes a more systematic view based on algorithms designed to automate the process of generalization and is often heavily reliant on data to enable analysis that might be impractical for humans.

Traditional approaches [88] to learning include supervised learning, unsupervised learning, and reinforcement learning. Techniques such as deep learning and neural networks have contributed to the large success of artificial intelligence integrated into many other disciplines.

It is in this context that automata learning emerged as the theoretical lens through which we seek to understand and enhance the learning processes of intelligent systems. At its core, automata theory revolves around the study of abstract mathematical models that capture the behavior and structure of dynamic systems. These models, represented by several variations of the basic automata model, provide a formal framework for not only analyzing the behavior of the system but also for studying algorithms that learn from data with temporal dependencies. The application of automata learning is multifaceted, ranging, for example, on the field analysis and verification of software systems, discerning patterns in biological sequences, or optimizing control strategies in autonomous systems [91, 100, 32, 87].

1.1 Automata learning

The problem of inducing, inferring, or learning automata has been an active subject of research in the last 40 years. In this context, the goal of learning is to find a finite representation of a (probabilistic, weighted) language in the form of an automaton or a grammar given a finite amount of sequential data [58, 34].

Depending on whether the learner can interact with her environment or not, we distinguish between two learning paradigms: active and passive learning [106, 20, 92]. The approaches based on passive learning automata operate in a more observational and receptive manner. The learning algorithms are designed to infer an automaton from a finite set of input sequences from which one needs to deduce the underlying patterns or rules governing the system's behavior represented by the learned automaton. This approach often involves leveraging algorithms that make minimal assumptions about the system at the price of being correct but almost always incomplete, as one cannot learn more than the data it is presented with. The passive learning automata approaches are particularly well-suited for scenarios where the learner lacks the capability to actively query the environment or influence the data generation process. From a theoretical point of view, the interest lies not only in the efficiency and capability of the learner to learn with minimal data but also in the minimal set provided by the teacher to guarantee optimal learning.

In contrast to the passive learning approaches, active learning automata frameworks consist of algorithms that have the ability to interact with their environment by making strategic queries or interventions. This interaction provides the capacity to choose which data instances to query and when, so as to optimally use the data acquisition process and deduce the structure and behavior of the system. This approach is especially advantageous when resources are limited, and the system needs to optimize its learning efficiency by selectively acquiring information that maximizes knowledge gain. An example is Angluin's active learning L^* algorithm [2], which infers an automaton from two types of interactions: membership queries and equivalence queries. Using membership queries, the learner tests whether a certain behavior is allowed by the system to be learned, whereas equivalence queries are used to check if the learned model is correct and complete (i.e. equivalent to) with respect to the target system. In the case the system and the model are different, then a counterexample can be used as additional information.

The success of learning automata is typically measured by assessing its ability to accurately infer a finite automaton from the available data [13, 110]. Experimentally this can be measured using techniques from machine learning: given a set of strings, accuracy is then the ratio of correctly identified strings to the total number of strings. Other metrics and evaluation methods are also employed to determine the effectiveness of the automata learned.

Accuracy is related to the soundness of the learned automaton: all strings belonging to the language of the learned automata should belong to the language of the model. Completeness instead refers to the opposite direction: the learned automata should be able to generate every string in the language of the model. Active learning approaches seek possible interactions that guarantee the soundness and completeness of the learned automaton. In passive learning, instead, one is more interested in how quickly the automaton converges to the correct model when increasing the size of the data available. This viewpoint is referred to as “identification in the limit” [46]: a language is said to be identified when the target is found. Which means the hypothesis is a perfect match with the target. From the teacher’s side, one can be interested in the robustness and effectiveness of the data provided.

In this thesis, we will focus on both active and passive learning algorithms for probabilistic and quantum automata, evaluate the goodness of our algorithms experimentally, and provide a theoretical context using existing results. We will not focus on other evaluation metrics, such as (1) robustness to noise measuring its performance when exposed to data with varying degrees of randomness, and (2) information-theoretic metrics, such as entropy or information gain that assess how well the automaton is capturing the essential information in the data. In particular, we will not consider complexity issues related to query or sample in relation to approximation error and confidence parameters that can be assessed, for example, using the theoretical framework of "Probably Approximately Correct" learning" [118].

1.1.1 Learning probabilistic automata

Probabilistic automata are frameworks for understanding and modeling systems characterized by inherent uncertainty and stochasticity. In essence, they are ordinary non-deterministic automata that incorporate probabilistic transitions between states, as well as assigning an initial and final probability to each state [93] to enable a representation of complex stochastic phenomena. In fact, unlike ordinary automata, probabilistic automata are capable of modeling the inherent uncertainty present in various real-world scenarios, making them well-suited for variable applications, such as natural language processing [68, 7, 60], biological sequence analysis [41, 8], cybersecurity [86], and beyond. The behavior of a probabilistic automaton is given by its associated probabilistic language, that is a discrete probabilistic distribution mapping each string on the alphabet to its probability.

Probabilistic automata are very similar to hidden Markov models, as they both can be used to generate distributions over complete finite prefix-free sets if we do not consider the final state distribution of a probabilistic automaton. On the other hand, hidden Markov models with additional final probabilities and probabilistic automata both generate distributions over strings of finite length. A probabilistic automaton can be converted into a hidden Markov model and vice versa [121, 39].

The core challenges in learning probabilistic automata lie in assigning the right probability to transitions and states given observed sequences of data equipped with associated frequency.

In the context of active learning, spectral methods such as the eigenvalue decomposition of matrices can be used to estimate the parameters of probabilistic automata.

While in general regular languages cannot be passively learned in the limit using only positive examples [46], this is not the case for probabilistic regular languages [3], which can be identified from positive samples with probability 1. Several passive learning algorithms have been proposed for passively learning probabilistic automata, but most of them either assume to know the states of the model or restrict themselves to deterministic probabilistic automata. In the first category belongs the Baum-Welch algorithm [10], a variant of the Expectation-Maximization algorithm specifically designed for hidden Markov models by iteratively updating transition and emission probabilities to maximize the likelihood of the observed sequences. However, this approach is not practical as it has a very large number of parameters [84].

Deterministic probabilistic automata are often learned by state merging algorithms, that initially compactly represent sets of sequences and their probabilities as a tree, and then merge states in an automaton while minimizing the loss of information [24, 25, 101]. In the context of applying these state-merging algorithms is Flexfringe, a tool designed to learn state machine models directly from input data [120]. Contrary to ordinary automata [34], deterministic probabilistic automata are strictly less expressive than probabilistic automata [121, 35, 39]. As such, it is not immediate how to extend state merging methods to learn general probabilistic automata.

In this thesis, we propose to learn separately the structure and the probabilities of an automaton, using, for example, genetic algorithms as an optimization technique to approximate the parameters needed by the automaton for generating the probabilities of the observed sequence of data.

1.1.2 Learning quantum automata

Another model of the systems with uncertainty is given by quantum automata. They differ from probabilistic automata in their underlying computational principles and mechanisms: quantum automata leverage the principles of quantum mechanics, such as superposition and entanglement, while probabilistic automata rely on classical probability theory for modeling uncertainties and random transitions.

Quantum automata have been introduced by Kondacs and Watrous early in 1997 [63]. States are qubits and transitions represent gates (or uniform operators) and are used to represent and process information in quantum states. States can exist in superposition, representing a combination of multiple states simultaneously, and their evolution involves unitary transformations (quantum gates) described by the matrix of all transitions. As dictated by quantum mechanics, measurement causes the system to collapse into one of the possible states with probabilities determined by the coefficients in the superposition. The most basic model is given by one-way quantum finite automata, that allow the input to be read only once.

Measurement can be done either only once at the end of the computation (measure-once one-way quantum automata [83]) or many times in distinct states allowing the computation to continue after the quantum state has been collapsed[63]. Both models are incomparable in terms of expressivity between themselves and with respect to ordinary automata. However, quantum automata can solve certain problems more efficiently than their classical counterparts for some promise problems [52].

The measure once approach follows the conventional model of quantum computing where the final result is obtained through a single measurement at the end of the computation, often applied to problems where a single, precise outcome is enough, such as factoring large numbers in Shor's factoring algorithm [108] or searching an unsorted database in Grover's algorithm [50]. Measure-many quantum automata, instead often employed in quantum simulations, is important for understanding the evolution of the quantum state. This understanding is essential for extracting meaningful information about the evolving quantum state at different stages of the computation.

More flexible types of quantum automata include the possibility to move on the input string back and forth, allowing classical and quantum states, or more general quantum states (in contrast to the pure quantum states) handled by one-way quantum automata. All these extensions offer greater flexibility and expressivity, making them more suitable for a broader range of quantum computing applications. However, the simplicity and efficiency of one-way quantum automata make them interesting in the context of learning.

Quantum learning theory is a field that is still evolving and only in the last 20 years is receiving more attention. It has primarily focused on developing quantum counterparts to classical learning theory paradigms, including quantum exact learning [5], quantum PAC models [23], and quantum agnostic models [6]. Quantum automata learning has seen limited exploration, with only one notable work on active learning [94] and none on passive learning. The work on active learning quantum finite automata allows interaction with the environment by asking for state amplitudes, rather than providing the end probability of a computation. Furthermore, it assumes that the learner possesses prior knowledge about the automaton's structure, including the identity of the accepting state and information about non-halting states [94].

1.2 Research questions

When learning ordinary automata using positive samples, other algorithms can be better suited than state merging methods. For example, learning algorithms based on k-testable languages offer a more concise representation of certain language classes compared to state merging methods [44]. This efficiency can lead to more compact and comprehensible models, especially when dealing with languages that have a significant level of structure and regularity. Also, they can capture the underlying structure of the language with fewer examples, making it advantageous when dealing with sparse or incomplete samples, a scenario where state merging methods may face challenges. Based on this observation, in this thesis, we provide an algorithm that disentangles the deterministic structural components from the probabilistic elements within regular distributions to enhance the efficiency of learning algorithms and gain a better understanding of the interplay between deterministic structure and probabilistic variability. The key research questions and corresponding contributions of this thesis are as follows:

Research Question 1 (RQ 1): *Can we develop an effective passive learning algorithm tailored for deterministic probabilistic regular distributions, achieving a separation between structural information and probabilistic characteristics?*

In Chapter 3, we propose an approach for passively learning probabilistic regular languages using only positive samples and parametric with respect to the length of an observable window on the strings of the sample. This allows us to separate the structural information using a technique for learning ordinary testable language from the probabilistic information that is recovered from the distribution represented by the sample. We show experimentally that our method learns more compact probabilistic automata than those learned by state merging methods. Also, it perfectly learns the model with probability 1 as the sample size tends to infinity.

The above method works well for learning deterministic regular distribution but fails when, for example, abstraction identifies different actions creating inherent uncertainty. The failure is due to the fact deterministic probabilistic automata are less expressive than probabilistic automata as they cannot represent and model probabilistic behaviors with inherent uncertainty. Also, we know that it is not possible to learn at the limit the structure of a regular language from a positive sample only, leading to our second research question:

Research Question 2 (RQ 2): *In the context of passive learning, is it possible to learn regular (not-necessarily deterministic) distributions by learning the structure and the probabilistic information separately?*

We answer this question only partially and by requiring more information than only positive samples. In fact, in Chapter 4, we address the challenge of efficiently learning probabilistic automata from positive and negative samples by learning the non-deterministic structure of the underlying residual language of a distribution. The corresponding residual automaton is non-deterministic and in some cases, it cannot be approximated efficiently by a probabilistic deterministic model. The probabilistic information, in the case of nondeterminism, is distributed fairly among the possible choices. This is not correct, but we show that it behaves better than deterministic methods, including a state merging method and our testable language method.

We improve the learning of the probabilistic information by solving a constraint optimization problem. We learn the parameters of the underlying learned structure of a probabilistic automaton using precise and approximate methods, such as genetic algorithms. The probabilities in the sample enriched with negative information ensure the accurate modeling of the learned distribution. To assess the effectiveness of our approach, we conduct experiments comparing our algorithm with other methods using randomly generated regular distributions as well as a case study on modeling agent behavior in a maze.

Having introduced learning methods for probabilistic automata that separate the structural information from the probabilistic characteristics in the previous chapters, our next step involves investigating the possibility of using a similar separation when learning another model for systems with uncertainty: quantum automata.

Research Question 3 (RQ 3): *Similar to probabilistic automata, quantum automata are models used to describe systems that exhibit uncertainty or randomness. Can we develop an algorithm for learning quantum automata in a realistic setting?*

Having a realistic setting is important as learning quantum automata may aid in simulating the behavior of quantum systems as well may facilitate the design and analysis of quantum computations expressed, for example, as finite sets of observations. In Chapter 6, we explore the applicability of active learning to approximate the parameters of measure-once quantum automata. Our approach combines non-linear optimization techniques and Hankel matrix analysis to learn the number of states and transition weights. The resulting approximation, although not guaranteed to be a quantum automaton, effectively models complex languages. By introducing a new method for measuring the proximity between learned and target automata, our work contributes to quantum automata learning, paving the way for efficient language representation in quantum computing applications.

We conclude the thesis with a novel encoding of measure-once one-way quantum finite automata into quantum optical experiments, something currently possible only with a restriction that we have prior knowledge of the length of the input strings. In Chapter 7, we implement a solution that eliminates the need for this explicit knowledge. By employing a specialized mechanism that dynamically encodes length information through rotations of half-wave plates, we successfully achieve the first implementation of a genuine quantum finite automaton. To close the circle, ideally, we would need to learn quantum automata from the observations of quantum optical experiments, but this would require a passive learning scheme for quantum automata that we leave as future work.

1.3 Underlying Publications

Part of this thesis is based on peer-reviewed publications. The list below shows an overview of these publications (ordered by date). For each publication we mention in which chapter the research material is used and the contribution of each author.

- **Chapter 3** is based on “Learning Probabilistic Languages by k-Testable Machines", *2020 International Symposium on Theoretical Aspects of Software Engineering (TASE)*, 2020 [26], by Wenjing Chu, and Marcello Bonsangue.

  Contribution of authors

  Wenjing Chu: all aspects,
  Marcello Bonsangue: supervision and insight.

- **Chapter 4** is based on “Learning probabilistic automata using residuals", *Theoretical Aspects of Computing–ICTAC 2021: 18th International Colloquium, Virtual Event, Nur-Sultan, Kazakhstan, September 8–10, 2021, Proceedings 18*, 2021 [27], by Wenjing Chu, Shuo Chen, and Marcello Bonsangue.

  Contribution of authors

  Wenjing Chu: all aspects,
  Shuo Chen: technical advice,
  Marcello Bonsangue: supervision and insight.

- **Chapter 4** is also based on “Non-linear optimization methods for learning regular distributions", *International Conference on Formal Engineering Methods*, 2022 [28], by Wenjing Chu, Shuo Chen, and Marcello Bonsangue.

  Contribution of authors

  Wenjing Chu: all aspects,
  Shuo Chen: technical advice,
  Marcello Bonsangue: supervision and insight.

- **Chapter 6** is based on “Approximately Learning Quantum Automata", *International Symposium on Theoretical Aspects of Software Engineering*, 2023 [29], by Wenjing Chu, Shuo Chen, Marcello Bonsangue, and Zenglin Shi.

  Contribution of authors

  Wenjing Chu: all aspects,
  Shuo Chen: technical advice,

Zenglin Shi: technical advice,
Marcello Bonsangue: supervision and insight.