Learning PDFA with Asynchronous Transitions

Borja Balle, Jorge Castro, and Ricard Gavaldà

Universitat Politècnica de Catalunya, Barcelona
{bballe,castro,gavalda}@lsi.upc.edu

Abstract. In this paper we extend the PAC learning algorithm due to Clark and Thollard for learning distributions generated by PDFA to automata whose transitions may take varying time lengths, governed by exponential distributions.

1 Motivation

The problem of learning (distributions generated by) probabilistic automata and related models has been intensely studied by the grammatical inference community; see [4,12,13] and references therein. The problem has also been studied in variants of the PAC model. In particular, it has been observed that polynomial-time learnability of PDFA is feasible if one allows polynomiality not only in the number states but also in other measures of the target automaton complexity. Specifically, Ron et al. [11] showed that acyclic PDFA can be learned w.r.t. the Kullback–Leibler (KL) divergence in time polynomial in alphabet size, $1/\epsilon$, $1/\delta$, number of target states, and $1/\mu$, where μ denotes the *distinguishability* of the target automaton. Clark and Thollard extended the result to general PDFA by considering also as a parameter the expected length of the strings L generated by the automaton [3]. Their algorithm, a state merge-split method, was in turn extended or refined in subsequent work [6,7,5,2].

Here we consider what we call *asynchronous PDFA* (AsPDFA), in which each transition has an associated exponential distribution. We think of this distribution as indicating the 'time' or duration of the transition. Note that there are several models of *timed automata* in the literature with other meanings, for example automata with timing constraints on the transitions. Our model is rather the finite-state and deterministic restriction of so-called *semi-Markov processes*; a widely-studied particular case of the latter are *continuous-time Markov chains*, in which times between transitions are exponentially distributed. We show a general expression for the KL divergence between two given AsPDFA similar to that in [1] for PDFA. Based on this expression and a variant of the Clark–Thollard algorithm from [2], we show that AsPDFA are learnable w.r.t. the KL divergence. Technically, the algorithm requires bounds on the largest and smallest possible values of the parameters of the exponential distributions, which can be thought as defining the 'time-scale' of the target AsPDFA. Full proofs are omitted in this version and will appear elsewhere.

The result above is motivated by the importance of modeling temporal components in many scenarios where probabilistic automata or HMM's are used as

J.M. Sempere and P. García (Eds.): ICGI 2010, LNAI 6339, pp. 271–275, 2010.

modeling tools. We in particular were brought to this problem by the work of one of the authors and other collaborators on modeling users' access patterns to websites [8,9,10]. Models similar to (visible- or hidden- state) Markov Models have been used for this purpose in marketing circles and are called Customer Behavior Model Graphs. After the work in [8,9,10], we noted that the time among successive web clicks, the *user think time*, was extremely informative to discriminate among different user types and predict their future behavior, and this information is not captured by standard PFA.

2 Results

We essentially follow notation and learning model from [3,2]. In particular, the definition of probabilistic deterministic finite automaton (PDFA) and associated notation used here are from [2]. Furthermore, we borrow the KL–PAC model for learning distributions over sequences and the notion of μ-distinguishability of PDFA from [3]. We will denote by $\mathsf{KL}(D_1 \| D_2)$ the relative entropy, or KL divergence, between a pair of distributions over the same set. The distributions are sometimes denoted by their models or parameters. In particular, in the case of two exponential distributions $\mathrm{Exp}(\lambda)$ and $\mathrm{Exp}(\hat{\lambda})$ one has $\mathsf{KL}(\lambda \| \hat{\lambda}) = \ln(\lambda/\hat{\lambda}) + \hat{\lambda}/\lambda - 1$.

An *asynchronous PDFA* (AsPDFA) is a tuple $\langle Q, \Sigma, \tau, \gamma, \xi, q_0, \Lambda \rangle$, where the sub-tuple $\langle Q, \Sigma, \tau, \gamma, \xi, q_0 \rangle$ defines a PDFA and $\Lambda : Q \times \Sigma \to \mathbb{R}$ is a partial function that assigns a *rate parameter* $\Lambda(q, \sigma) = \lambda_{q,\sigma} > 0$ to each transition defined in the PDFA. We will say that an AsPDFA is μ-distinguishable if the underlying PDFA is μ-distinguishable.

When acting as a generator, an AsPDFA works like a PDFA with a minor modification. If q is the current state, after 'deciding' to emit the symbol σ (with probability $\gamma(q, \sigma)$), it also emits a real number t, called the *duration* of the transition, sampled at random from $\mathrm{Exp}(\lambda_{q,\sigma})$, an exponential distribution with parameter $\lambda_{q,\sigma}$. Next state is $\tau(q, \sigma)$. In this process, all durations sampled from exponential distributions are mutually independent. An observation generated by an AsPDFA is a *temporal string* $x = ((\sigma_0, t_0), \ldots, (\sigma_k, t_k), (\xi, t_{k+1}))$ where $\sigma_i \in \Sigma$ and $t_i \in \mathbb{R}$. Thus, an AsPDFA induces a probability measure over the space $X = (\Sigma \times \mathbb{R})^* \times (\{\xi\} \times \mathbb{R})$.

Our first theorem provides an expression for the relative entropy between two AsPDFA that generalizes the formula in [1] for PDFA. Carrasco's formula was used in [3] to bound the KL divergence between a target PDFA and an hypothesis produced by a learning algorithm. By the following result, similar techniques can be use to prove learnability for AsPDFA.

Theorem 1. *Let A and $\hat{A}$ be AsPDFA over the same alphabet Σ with the same terminal symbol ξ. The KL divergence between the probability distributions induced by A and $\hat{A}$ is*

$$\mathsf{KL}(A \| \hat{A}) = \sum_{q \in Q} \sum_{\hat{q} \in \hat{Q}} W(q, \hat{q}) \sum_{\sigma \in \Sigma'} \gamma(q, \sigma) \left[\log \frac{\gamma(q, \sigma)}{\hat{\gamma}(\hat{q}, \sigma)} + \mathsf{KL}(\lambda_{q,\sigma} \| \hat{\lambda}_{\hat{q},\sigma}) \right] , \quad (1)$$

where $\Sigma' = \Sigma \cup \{\xi\}$ *and* $W(q, \hat{q}) = \sum_{s \in P(q,\hat{q})} \gamma(q_0, s)$ *with*

$$P(q, \hat{q}) = \{s \in \Sigma^* \mid \tau(q_0, s) = q \text{ and } \hat{\tau}(\hat{q}_0, s) = \hat{q}\} \quad . \tag{2}$$

Note that (1) yields a decomposition of $\mathsf{KL}(A \| \hat{A})$ as a sum of two terms, one correponding to the KL divergence between the underlying PDFA and another that contains all the terms from Λ and $\hat{\Lambda}$. The proof of Theorem 1 is similar in spirit to that in [1]. However, some measurability issues need to be taken into account in this case. Essentially, this is due to the fact that an AsPDFA defines a probability measure over $(\Sigma \times \mathbb{R})^* \times (\{\xi\} \times \mathbb{R})$, a space which is neither discrete nor continuous.

As already mentioned, the decomposition given by (1) opens the door to algorithms for learning AsPDFA similar to those for PDFA. In particular, a variation of the Clark–Thollard algorithm [3] for learning AsPDFA will be outlined next. The algorithm is called `AsLearner` and is built as an extension, with some improvements, over the `Learner` algorithm from [2].

As input parameters `AsLearner` receives the alphabet size $|\Sigma|$, an upper bound n on the number of states of the target, a confidence parameter δ, and upper and lower bounds, $\lambda_{\max}$ and $\lambda_{\min}$ respectively, on all rate parameters of the target. Furthermore, `AsLearner` is provided with a sample S of examples, in this case temporal strings, drawn independently at random from the target AsPDFA A.

Grosso modo, the algorithm uses S to build a graph which captures all 'essential' parts of A, the so-called frequent states and frequent transitions. Each node and each arc in this graph is assigned a multiset. In the case of nodes, multisets collect suffixes generated from states corresponding to them. These multisets can be used to estimate stopping and transition probabilities associated to that state. For arcs, multisets contain all observed durations of the corresponding transition. From these durations a rate parameter for each transitions can be easily estimated. These estimation steps turn the graph into an hypothesis AsPDFA. Finally, a smoothing step is performed and a ground state is added to the hypothesis. The resulting AsPDFA $\hat{A}$ is returned.

Some little differences between `Learner` and `AsLearner` are to be found on how the graph is constructed. Remarkably, a variation of the distinctness test from [2] requiring less samples is employed. Furthermore, a different stopping condition is used to determine when the graph contains all relevant states and transitions.

The analysis of the algorithm follows a scheme similar to that in [2]. Using Chernoff bounds as the main technical tool, the graph is guaranteed to be correct with high probability. Subsequently, estimations of transition probabilities and rate parameters are shown to be accurate with high probability. Here, a concentration inequality for the sample mean of exponential random variables is invoked. Finally, bounding techniques from [3] are combined with (1) to prove that, with high probability, $\hat{A}$ is close to A w.r.t. the KL divergence. The outcome of this analysis is summarized in the following.

Theorem 2. *Given a sample S from an AsPDFA A, the algorithm* `AsLearner` *outputs, with probability at least $1-\delta$, an hypothesis $\hat{A}$ satisfying $\mathsf{KL}(A\|\hat{A}) < \epsilon$ whenever the number of examples in S is $|S| > N$, where N is a function from*

$$\tilde{O}\left(\frac{n^5 L^9 |\Sigma|^3}{\epsilon^6 \mu^2} \cdot \ln\left(\frac{1}{\delta}\right) \cdot \ln^3\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)\right) \quad . \tag{3}$$

Furthermore, the algorithm runs in time polynomial in $|S|$ and the lengths of examples in S.

3 Discussion

Improving on previous algorithms, `AsLearner` needs less input parameters (about the underlying PDFA) thanks to the new stopping condition. Futhermore, an improved test for comparing states and a sharper analysis yield a dependence on $|\Sigma|$ in the sample bound from Theorem 2 one degree smaller than in the `Learner` algorithm. On the other side, note the dependence on the number of states is one degree larger. That is because some more samples are needed in order to guarantee a good approximation of all relevant rate parameters. Apart from the dependence on $\lambda_{\max}$ and $\lambda_{\min}$, which determine the 'time scale' of the target AsPDFA, the rest of parameters appear with the same degree as in the bounds from [2].

Recall that the distinguishability of an AsPDFA is defined here as the distinguishability of its underlying PDFA. This allows to prove learnability for AsPDFA using almost the same algorithm for learning PDFA. In particular, the same statistical test for distinguishing between different states can be used for learning PDFA and AsPDFA under this definition of distinguishability. However, it is conceivable that a new test using information provided by transition durations in addition to information from suffix distributions can be used to learn AsPDFA. Such a test would require a novel definition of distinguishability and would provide means for learning AsPDFA whose underlying PDFA are not learnable. Thus, we regard our results as a proof of concept on AsPDFA learning which we plan to extend along these lines in future work.

Finally, it is worth remarking that Theorem 1 can be generalized to broader classes of automata where, instead of duration, transitions convey more general types of information. This generalization can be proved under very mild measurability conditions on the distributions that generate such information. Identifying families of distributions, other than exponential, for which learning is feasible, can significantly extend the range of practical applications where these techniques can be used.

Acknowledgements. This work is partially supported by the Ministry of Science and Technology of Spain under contracts TIN-2008-06582-C03-01 (SESAAME) and TIN-2007-66523 (FORMALISM), by the Generalitat de Catalunya 2009-SGR-1428 (LARCA), and by the EU PASCAL2 Network of Excellence (FP7-ICT-216886). B. Balle is supported by an FPU fellowship (AP2008-02064) of the Spanish Ministry of Education.

References

1. Carrasco, R.C.: Accurate computation of the relative entropy between stochastic regular grammars. RAIRO (Theoretical Informatics and Applications) 31(5), 437–444 (1997)
2. Castro, J., Gavaldà, R.: Towards feasible PAC-learning of probabilistic deterministic finite automata. In: Clark, A., Coste, F., Miclet, L. (eds.) ICGI 2008. LNCS (LNAI), vol. 5278, pp. 163–174. Springer, Heidelberg (2008)
3. Clark, A., Thollard, F.: PAC-learnability of probabilistic deterministic finite state automata. Journal of Machine Learning Research (2004)
4. Dupont, P., Denis, F., Esposito, Y.: Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms. Pattern Recognition 38(9), 1349–1371 (2005)
5. Gavaldà, R., Keller, P.W., Pineau, J., Precup, D.: PAC-learning of markov models with hidden state. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 150–161. Springer, Heidelberg (2006)
6. Guttman, O., Vishwanathan, S.V.N., Williamson, R.C.: Learnability of probabilistic automata via oracles. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 171–182. Springer, Heidelberg (2005)
7. Palmer, N., Goldberg, P.W.: PAC-learnability of probabilistic deterministic finite state automata in terms of variation distance. Theor. Comput. Sci. 387(1), 18–31 (2007)
8. Poggi, N., Berral, J.L., Moreno, T., Gavaldà, R., Torres, J.: Automatic detection and banning of content stealing bots for e-commerce. In: NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security (2007), `http://mls-nips07.first.fraunhofer.de/`
9. Poggi, N., Moreno, T., Berral, J.L., Gavaldà, R., Torres, J.: Web customer modeling for automated session prioritization on high traffic sites. In: Conati, C., McCoy, K., Paliouras, G. (eds.) UM 2007. LNCS (LNAI), vol. 4511, pp. 450–454. Springer, Heidelberg (2007)
10. Poggi, N., Moreno, T., Berral, J.L., Gavaldà, R., Torres, J.: Self-adaptive utility-based web session management. Computer Networks 53(10), 1712–1721 (2009)
11. Ron, D., Singer, Y., Tishby, N.: On the learnability and usage of acyclic probabilistic finite automata. J. Comput. Syst. Sci. 56(2), 133–152 (1998)
12. Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., Carrasco, R.C.: Probabilistic finite-state machines - part I. IEEE Trans. Pattern Anal. Mach. Intell. 27(7), 1013–1025 (2005)
13. Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., Carrasco, R.C.: Probabilistic finite-state machines - part II. IEEE Trans. Pattern Anal. Mach. Intell. 27(7), 1026–1039 (2005)