Learning a Random DFA from Uniform Strings and State Information

Dana Angluin and Dongqu Chen

Department of Computer Science
Yale University
New Haven, CT 06520

Abstract. Deterministic finite automata (DFA) have long served as a fundamental computational model in the study of theoretical computer science, and the problem of learning a DFA from given input data is a classic topic in computational learning theory. In this paper we study the learnability of a random DFA and propose a computationally efficient algorithm for learning and recovering a random DFA from uniform input strings and state information in the statistical query model. A random DFA is uniformly generated: for each state-symbol pair $(q \in Q, \sigma \in \Sigma)$, we choose a state $q' \in Q$ with replacement uniformly and independently at random and let $\varphi(q, \sigma) = q'$, where Q is the state space, Σ is the alphabet and φ is the transition function. The given data are string-state pairs (x, q) where x is a string drawn uniformly at random and q is the state of the DFA reached on input x starting from the start state q_0. A theoretical guarantee on the maximum absolute error of the algorithm in the statistical query model is presented. Extensive experiments demonstrate the efficiency and accuracy of the algorithm.

Keywords: Deterministic Finite Automaton, Random DFA, Statistical Queries, Regular Languages, PAC Learning.

1 Introduction

Deterministic finite automata are one of the most elementary computational models in the study of theoretical computer science. The important role of DFA leads to the classic problem in computational learning theory, the learnability of DFA. The applications of this learning problem include formal verification, natural language processing, robotics and control systems, computational biology, data mining and music. Exploring the learnability of DFA is significant to both theoretical and applied realms. In the classic PAC learning model defined by Valiant [21], unfortunately, the concept class of DFAs is known to be inherently unpredictable [14, 15]. In a modified version of Valiant's model which allows the learner to make membership queries, Angluin [1] has shown that the concept class of DFAs is efficiently PAC learnable. Subsequent efforts have searched for nontrivial properly PAC learnable subfamilies of regular languages [16, 2, 6].

Since learning all DFAs is computationally intractable, it is natural to ask whether we can pursue positive results for "almost all" DFAs. This is addressed by studying high-probability properties of uniformly generated random DFAs. The same approach has been used for learning random decision trees and random DNFs from uniform strings [12, 11, 17, 18]. However, the learnability of random DFAs has long been an open problem. Few formal results about random walks on random DFAs are known. Grusho [9] was the first work establishing an interesting fact about this problem. Since then, very little progress was made until a recent subsequent work by Balle [4]. Our work connects these two problems and contributes an algorithm for efficiently learning random DFAs, in addition to positive theoretical results on random walks on random DFAs.

Trakhtenbrot and Barzdin [20] first introduced two random DFA models with different sources of randomness: one with a random automaton graph, one with random output labeling. In this paper we study the former model. A random DFA is uniformly generated: for each state-symbol pair $(q \in Q, \sigma \in \Sigma)$, we choose a state $q' \in Q$ with replacement uniformly and independently at random and let $\varphi(q, \sigma) = q'$, where Q is the state space, Σ is the alphabet and φ is the transition function. Given data are of form (x, q) where x is a string drawn uniformly at random and q is the state of the DFA reached on input x starting from the start state q_0.

Previous work by Freund et al. [8] has studied a different model under different settings. First, the DFAs are generated with arbitrary transition graphs and random output labeling, which is the latter model in [20]. Second, in their work, the learner predicts and observes the exact label sequence of the states along each walk. Such sequential data are crucial to the learner walking on the graph. In our paper, the learner is given noisy statistical data on the ending state, with no information about any intermediate states along the walk.

Like most spectral methods, the theoretical error bound of our algorithm contains a spectral parameter ($|||P_A^\dagger|||_\infty$ in Section 4.1), which reflects the asymmetry of the underlying graph. This leads to a potential future work of eliminating this parameter using random matrix theory techniques. Another direction of subsequent works is to consider the more general case where the learner only observes the accept/reject bits of the final states reached, which under arbitrary distributions has been proved to be hard in the statistical query model by Angluin et al. [3] but remains open under the uniform distribution [4]. Our contribution narrows this gap and pushes forward the study of the learnability of random DFAs.

2 Preliminaries

Deterministic finite automaton (DFA) is a powerful and widely studied computational model in computer science. Formally, a DFA is a quintuple $A = (Q, \varphi, \Sigma, q_0, F)$ where Q is a finite set of states, Σ is the finite alphabet, $q_0 \in Q$ is the start state, $F \subseteq Q$ is the set of accepting states, and φ is the transition function: $Q \times \Sigma \to Q$. Let λ be the empty string. Define the extended transition function $\varphi^* : Q \times \Sigma^* \to Q$ by $\varphi^*(q, \lambda) = q$ and inductively $\varphi^*(q, x\sigma) = \varphi(\varphi^*(q, x), \sigma)$ where $\sigma \in \Sigma$ and $x \in \Sigma^*$. Denote by $s = |\Sigma|$ the size of the alphabet and by

$n = |Q|$ the number of states. In this paper we assume $s \geq 2$. Let $G = (V, E)$ be the underlying directed multi-graph of DFA A (also called an *automaton graph*). We say a vertex set $V_0 \subseteq V$ is *closed* if for any $u \in V_0$ and any v such that $(u, v) \in E$, we must have $v \in V_0$.

A *walk* on an automaton graph G is a sequence of states $(v_0, v_1, \ldots, v_\ell)$ such that $(v_{i-1}, v_i) \in E$ for all $1 \leq i \leq \ell$, where v_0 is the corresponding vertex in G of the start state q_0. A *random walk* on graph G is defined by a transition probability matrix P with $P(u, v) = \#\{(u, v) \in E\} \cdot s^{-1}$ denoting the probability of moving from vertex u to vertex v, where $\#\{(u, v) \in E\}$ is the number of edges from u to v. For an automaton graph, a random walk always starts from the start state q_0. In this paper random walks on a DFA refer to the random walks on the underlying automaton graph. A vertex u is *aperiodic* if $\gcd\{t \geq 1 \mid P^t(u, u) > 0\} = 1$. Graph G (or a random walk on G) is *irreducible* if for every pair of vertices u and v in V there exists a directed cycle in G containing both u and v, and is *aperiodic* if every vertex is aperiodic. A distribution vector ϕ satisfying $\phi P = \phi$ is called a *Perron vector* of the walk. An irreducible and aperiodic random walk has a unique Perron vector ϕ and $\lim_{t \to +\infty} P^t(u, \cdot) = \phi$ (called the *stationary distribution*) for any $u \in V$. In the study of rapidly mixing walks, the *convergence rate* in L_2 distance $\Delta_{L_2}(t) = \max_{u \in V} \|P^t(u, \cdot) - \phi\|_2$ is often used. A stronger notion in L_1 distance is measured by the *total variation distance*, given by $\Delta_{TV}(t) = \frac{1}{2} \max_{u \in V} \sum_{v \in V} |P^t(u, v) - \phi(v)|$. Another notion of distance for measuring convergence rate is the *χ-square distance*:

$$\Delta_{\chi^2}(t) = \max_{u \in V} \left(\sum_{v \in V} \frac{\left(P^t(u, v) - \phi(v)\right)^2}{\phi(v)} \right)^{\frac{1}{2}}$$

As the Cauchy-Schwarz inequality gives $\Delta_{L_2}(t) \leq 2\Delta_{TV}(t) \leq \Delta_{\chi^2}(t)$, a convergence upper bound for $\Delta_{\chi^2}(t)$ implies ones for $\Delta_{L_2}(t)$ and $\Delta_{TV}(t)$.

Trakhtenbrot and Barzdin [20] first introduced the model of random DFA by employing a uniformly generated automaton graph as the underlying graph and labeling the edges uniformly at random. In words, for each state-symbol pair $(q \in Q, \sigma \in \Sigma)$, we choose a state $q' \in Q$ with replacement uniformly and independently at random and let $\varphi(q, \sigma) = q'$.

In a computational learning model, an algorithm is usually given access to an oracle providing information about the target concept. Kearns [13] modified Valiant's model and introduced the *statistical query oracle STAT*. Kearns' oracle takes as input a statistical query of the form (χ, τ). Here χ is any mapping of a labeled example to $\{0, 1\}$ and $\tau \in [0, 1]$ is called the noise *tolerance*. Let c be the target concept and $\mathcal{D}$ be the distribution over the instance space. Oracle $STAT(c, \mathcal{D})$ returns to the learner an estimate for the expectation $\mathbf{E}\chi$, that is, the probability that $\chi = 1$ when the labeled example is drawn according to $\mathcal{D}$. A statistical query can have a condition, in which case $\mathbf{E}\chi$ is a conditional probability. This estimate is accurate within additive error τ. A statistical query χ is *legimate* and *feasible* if and only if:

1. Query χ maps a labeled example $\langle x, c(x) \rangle$ to $\{0, 1\}$;

2. Query χ can be efficiently evaluated in polynomial time;
3. The condition of χ, if any, can be efficiently evaluated in polynomial time;
4. The probability of the condition of χ, if any, should be at least inverse polynomially large.

Kearns [13] proved that the statistical query model is weaker than the classic PAC model. That is, PAC learnability from oracle *STAT* implies PAC learnability from the classic example oracle, but not vice versa.

3 Random walks on a random DFA

Random walks have proven to be a simple, yet powerful mathematical tool for extracting information from well connected graphs. Since automaton graphs are long known to be of strong connectivity with high probability [9], it's interesting to explore the possibilities of applying random walks to DFA learning. In this section we will show that with high probability, a random walk on a random DFA converges to the stationary distribution ϕ polynomially fast in χ-square distance as in Theorem 1.

Theorem 1 *With probability* $1 - o(1)$*, a random walk on a random DFA has* $\Delta_{\chi^2}(t) \leq e^{-k}$ *after* $t \geq 2C(C+1)sn^{1+C}(\log n + k) \cdot \log_s n$*, where constant* $C > 0$ *depends on* s *and approaches unity with increasing* s*.*

A standard proof of fast convergence consists of three parts: irreducibility, aperiodicity and convergence rate. Grusho [9] first proved the irreducibility of a random automaton graph.

Lemma 1 *With probability* $1 - o(1)$*, a random automaton graph* G *has a unique strongly connected component, denoted by* $\tilde{G} = (\tilde{V}, \tilde{E})$*, of size* $\tilde{n}$*, and a)* $\lim_{n \to +\infty} \frac{\tilde{n}}{n} = C$ *for some constant* $C > 0.7968$ *when* $s \geq 2$ *or some* $C > 0.999$ *when* $s > 6$*; b)* $\tilde{V}$ *is closed.*

A subsequent work by Balle [4] proved the aperiodicity.

Lemma 2 *With probability* $1 - o(1)$*, the strongly connected component* $\tilde{G}$ *in Lemma 1 is aperiodic.*

However, the order of the convergence rate of random walks on a random DFA was left as an open question. One canonical technique for bounding the convergence rate of a random walk is to bound the smallest nonzero eigenvalue of the *Laplacian matrix* $\mathcal{L}$ of the graph G, defined by

$$\mathcal{L} = I - \frac{\Phi^{\frac{1}{2}} P \Phi^{-\frac{1}{2}} + \Phi^{-\frac{1}{2}} P^* \Phi^{\frac{1}{2}}}{2}$$

where Φ is an $n \times n$ diagonal matrix with entries $\Phi(u, u) = \phi(u)$ and P^* denotes the transpose of matrix P. For a random walk P, define the *Rayleigh quotient* for any function $f : V \to \mathbb{R}$ as follows.

$$R(f) = \frac{\sum_{u \to v} |f(u) - f(v)|^2 \phi(u) P(u, v)}{\sum_v |f(v)|^2 \phi(v)}$$

Chung [7] proved the connection between the Rayleigh quotient and the Laplacian matrix of a random walk.

Lemma 3

$$R(f) = 2\frac{\langle g\mathcal{L}, g\rangle}{\|g\|_2^2}$$

where $g = f\Phi^{\frac{1}{2}}$ *and* $\langle \cdot, \cdot \rangle$ *means the inner product of two vectors.*

On top of this lemma we can further infer the relation between the Rayleigh quotient and the Laplacian eigenvalues. Suppose the Laplacian matrix $\mathcal{L}$ has eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{n-1}$.

Lemma 4 *For all* $1 \leq i \leq n-1$, *let vector* η_i *be the unit eigenvector of* λ_i *and vector* $f_i = \eta_i \Phi^{-\frac{1}{2}}$. *Then* $\lambda_i = \frac{1}{2}R(f_i)$ *and* f_i *satisfies* $\langle f_i, \phi \rangle = 0$.

Proof By Lemma 3 we know $\frac{1}{2}R(f) = \frac{\langle g\mathcal{L}, g\rangle}{\|g\|^2}$. From the symmetry of Laplacian matrix $\mathcal{L}$, there exists a set of eigenvectors of $\mathcal{L}$ that forms an orthogonal basis. We denote this set of eigenvectors by $\eta_0, \eta_1, \ldots, \eta_{n-1}$ where η_i is the corresponding eigenvector of λ_i. Notice that for all $0 \leq i \leq n-1$ we have

$$\frac{1}{2}R(\eta_i \Phi^{-\frac{1}{2}}) = \frac{\langle \eta_i \mathcal{L}, \eta_i \rangle}{\|\eta_i\|_2^2} = \frac{\lambda_i \|\eta_i\|_2^2}{\|\eta_i\|_2^2} = \lambda_i$$

We let $f_i = \eta_i \Phi^{-\frac{1}{2}}$. According to the definition of $R(f)$, we have $R(f) \geq 0$. We know $\lambda_0 = R(f_0) = 0$. Thus f_0 is the all-one vector and $\eta_0 = \phi^{\frac{1}{2}}$ is the unit eigenvector of eigenvalue 0. For all $1 \leq i \leq n-1$ we have $\langle \eta_i, \eta_0 \rangle = 0$, i.e., $(f_i \Phi^{\frac{1}{2}}) \cdot \phi^{\frac{1}{2}} = \langle f_i, \phi \rangle = 0$. Hence, for all $1 \leq i \leq n-1$, we have $\lambda_i = \frac{1}{2}R(f_i)$ where f_i satisfies $\langle f_i, \phi \rangle = 0$. ■

From this we can see that the Rayleigh quotient serves as an important tool for bounding the Laplacian eigenvalues. A lower bound on $R(f_1)$ is equivalent to one on λ_1. We present a lower bound of λ_1 in terms of the diameter and the maximum out-degree of the vertices in the graph.

Lemma 5 *For a random walk on a strongly connected graph* G, *let* λ_1 *be the smallest nonzero eigenvalue of its Laplacian matrix* $\mathcal{L}$. *Denote by* $Diam$ *the diameter of graph* G *and by* s_0 *the maximum out-degree of the vertices in the graph. Then*

$$\lambda_1 \geq \frac{1}{2n \cdot Diam \cdot s_0^{1+Diam}}$$

Proof Denote $u_0 = \arg\max_{x \in V} \phi(x)$ and $v_0 = \arg\min_{x \in V} \phi(x)$. Let ℓ_0 be the distance from u_0 to v_0. As $\phi P^{\ell_0} = \phi$, we have $\phi(v_0) \geq P^{\ell_0}(u_0, v_0)\phi(u_0) \geq s_0^{-\ell_0}\phi(u_0) \geq s_0^{-Diam}\phi(u_0)$. We then have $1 = \sum_{x \in V} \phi(x) \leq n\phi(u_0) \leq ns_0^{Diam}\phi(v_0)$ and $\phi(v_0) \geq n^{-1}s_0^{-Diam}$.

From Lemma 4 we have $\lambda_1 = \frac{1}{2}R(f_1)$ and $\langle f_1, \phi \rangle = 0$. As $\phi(x) > 0$ for any vertex $x \in V$, there must exist some vertex u with $f_1(u) > 0$ and some vertex v whose $f_1(v) < 0$. Let $y = \arg\max_{x \in V} |f_1(x)|$. Then there must exist some vertex z such that $f_1(y)f_1(z) < 0$. Let $\boldsymbol{r} = (y, x_1, x_2 \ldots, x_{\ell-1}, z)$ be the shortest directed path from y to z, which must exist due to the strong connectivity. Then the length of path $\boldsymbol{r}$ is ℓ. Therefore,

$$\begin{aligned}
\lambda_1 = \frac{1}{2}R(f_1) &= \frac{1}{2}\frac{\sum_{u \to v} |f_1(u) - f_1(v)|^2 \phi(u) P(u,v)}{\sum_v |f_1(v)|^2 \phi(v)} \\
&\left(\text{due to } \min_{x \in V} \phi(x) \geq n^{-1} s_0^{-Diam} \text{ and } \min_{(u,v) \in E} P(u,v) \geq \frac{1}{s_0}\right) \\
&\geq \frac{1}{2ns_0^{1+Diam}} \frac{\sum_{u \to v} |f_1(u) - f_1(v)|^2}{\sum_v |f_1(v)|^2 \phi(v)} \\
&\geq \frac{1}{2ns_0^{1+Diam}} \frac{\sum_{u \to v \in \boldsymbol{r}} |f_1(u) - f_1(v)|^2}{\sum_v |f_1(v)|^2 \phi(v)} \\
&\text{(by letting } x_0 = y \text{ and } x_\ell = z) \\
&= \frac{1}{2ns_0^{1+Diam}} \frac{\sum_{i=0}^{\ell-1} |f_1(x_i) - f_1(x_{i+1})|^2}{\sum_v |f_1(v)|^2 \phi(v)} \\
&\geq \frac{1}{2ns_0^{1+Diam}} \frac{\left[\sum_{i=0}^{\ell-1} (f_1(x_i) - f_1(x_{i+1}))\right]^2}{\ell \cdot \sum_v |f_1(v)|^2 \phi(v)} \\
&= \frac{1}{2ns_0^{1+Diam}} \frac{[f_1(y) - f_1(z)]^2}{\ell \cdot \sum_v |f_1(v)|^2 \phi(v)} \\
&\text{(for } f_1(y)f_1(z) < 0) \\
&\geq \frac{1}{2n \cdot Diam \cdot s_0^{1+Diam}} \frac{|f_1(y)|^2}{\sum_v |f_1(v)|^2 \phi(v)} \\
&\geq \frac{1}{2n \cdot Diam \cdot s_0^{1+Diam}} \frac{|f_1(y)|^2}{|f_1(y)|^2 \sum_v \phi(v)} \\
&= \frac{1}{2n \cdot Diam \cdot s_0^{1+Diam}}
\end{aligned}$$

which completes the proof. ■

As a canonical technique, a lower bound of the smallest nonzero eigenvalue of the Laplacian matrix implies a lower bound of the convergence rate. Chung [7] proved

Theorem 2 *A lazy random walk on a strongly connected graph G has convergence rate of order $2\lambda_1^{-1}(-\log \min_u \phi(u))$. Namely, after at most $t \geq 2\lambda_1^{-1}((-\log \min_u \phi(u)) + 2k)$ steps, we have $\Delta_{\chi^2}(t) \leq e^{-k}$.*

In the paper Chung used lazy walks to avoid periodicity. If the graph is irreducible and aperiodic, we let $\widehat{P} = \frac{1}{2}(I + P)$ be the transition probability matrix of the lazy random walk and vector $\widehat{\phi}$ be its Perron vector, matrix $\widehat{\Phi}$ be the diagonal matrix of $\widehat{\phi}$, matrix $\widehat{\mathcal{L}}$ be its Laplacian matrix.

We know ϕ is the solution of $\phi P = \phi$ or equivalently $\phi(I-P) = 0$ and $\sum_i \phi(i) = 1$. Similarly, $\widehat{\phi}$ is the solution of $\widehat{\phi}(I-\widehat{P}) = 0$ and $\sum_i \widehat{\phi}(i) = 1$. Observe that $I - \widehat{P} = I - \frac{1}{2}(I+P) = \frac{1}{2}(I-P)$ and $\widehat{\phi}(I-\widehat{P}) = \frac{1}{2}\widehat{\phi}(I_P) = 0$, which is equivalently $\widehat{\phi}(I-P) = 0$. Thus $\widehat{\phi} = \phi$ and $\widehat{\Phi} = \Phi$. Then

$$
\begin{aligned}
\widehat{\mathcal{L}} &= I - \frac{1}{2}\left(\widehat{\Phi}^{\frac{1}{2}}\widehat{P}\widehat{\Phi}^{-\frac{1}{2}} + \widehat{\Phi}^{-\frac{1}{2}}\widehat{P}^{*}\widehat{\Phi}^{\frac{1}{2}}\right) \\
&= I - \frac{1}{2}\left(\Phi^{\frac{1}{2}}\cdot\frac{1}{2}(I+P)\cdot\Phi^{-\frac{1}{2}} + \Phi^{-\frac{1}{2}}\cdot\frac{1}{2}(I+P^{*})\cdot\Phi^{\frac{1}{2}}\right) \\
&= I - \frac{1}{2}\left(\frac{1}{2}I + \frac{1}{2}\Phi^{\frac{1}{2}}P\Phi^{-\frac{1}{2}} + \frac{1}{2}I + \frac{1}{2}\Phi^{-\frac{1}{2}}P^{*}\Phi^{\frac{1}{2}}\right) \\
&= I - \frac{1}{2}\left(I + \frac{1}{2}\Phi^{\frac{1}{2}}P\Phi^{-\frac{1}{2}} + \frac{1}{2}\Phi^{-\frac{1}{2}}P^{*}\Phi^{\frac{1}{2}}\right) \\
&= \frac{1}{2}I - \frac{1}{4}\left(\Phi^{\frac{1}{2}}P\Phi^{-\frac{1}{2}} + \Phi^{-\frac{1}{2}}P^{*}\Phi^{\frac{1}{2}}\right) \\
&= \frac{1}{2}\mathcal{L}
\end{aligned}
$$

Let $\widehat{\lambda}_1$ be the smallest positive eigenvalue of $\widehat{\mathcal{L}}$. Then $\lambda_1 = 2\widehat{\lambda}_1$. Therefore, combining this with Lemma 5, we have

Theorem 3 *A random walk on a strongly connected and aperiodic directed graph G has convergence rate of order $2n \cdot Diam \cdot s_0^{1+Diam}(\log(ns_0^{Diam}))$, where $s_0 = \arg\max_{u \in V} d_u$ is the maximum out-degree of a vertex in G. Namely, after at most $t \geq 2n \cdot Diam \cdot s_0^{1+Diam}((\log(ns_0^{Diam}) + 2k))$ steps, we have $\Delta_{\chi^2}(t) \leq e^{-k}$.*

However, the convergence rate is still exponential in s_0 and $Diam$. Fortunately, in our case $s_0 = s$ and Trakhtenbrot and Barzdin [20] proved the diameter of a random DFA is logarithmic.

Theorem 4 *With probability $1 - o(1)$, the diameter of a random DFA is $O(\log_s n)$.*

With the logarithmic diameter we complete the poof of Theorem 1. The constant C in Theorem 1 is the constant used in the proof of Theorem 4 by Trakhtenbrot and Barzdin [20]. It depends on s and approaches unity with increasing s.

Notice that the diameter of an automaton graph won't increase after state-merging operations, thus with high probability, a random DFA has at most logarithmic diameter after DFA minimization. It is also easy to see an irreducible DFA still maintains irreducibility after minimization. Besides, Balle [4] proved DFA minimization preserves aperiodicity. Now we also have Corollary 1.

Corollary 1 *With probability* $1 - o(1)$*, a random walk on a random DFA after minimization has* $\Delta_{\chi^2}(t) \leq e^{-k}$ *after* $t \geq 2C(C+1)sn^{1+C}(\log n + k) \cdot \log_s n$*, where constant* $C > 0$ *depends on* s *and approaches unity with increasing* s*.*

4 Reconstructing a random DFA

In this section we present a computationally efficient algorithm for recovering random DFAs from uniform input strings in the statistical query model with a theoretical guarantee on the maximum absolute error and supporting experimental results.

4.1 The learning algorithm

In our learning model, the given data are string-state pairs (x, q) where x is a string drawn uniformly at random from Σ^t and q is the state of the DFA reached on input x starting from the start state q_0. Here $t = poly(n, s)$ is the length of the example strings. Our goal is to recover the unique irreducible and closed component of the target DFA from the given data in the statistical query model. The primary constraint on our learning model is the need to estimate the distribution of the ending state, while the advantage is that our algorithm reconstructs the underlying graph structure of the automaton. Let quintuple $A = (Q, \varphi, \Sigma, q_0, F)$ be the target DFA we are interested in. We represent the transition function φ as a collection of $n \times n$ binary matrices M_σ indexed by symbols $\sigma \in \Sigma$ as follows. For each pair of states (i, j), the element $M_\sigma(i, j)$ is 1 if $\varphi(i, \sigma) = j$ and 0 otherwise. For a string of m symbols $y = y_1 y_2 \ldots y_m$, define M_y to be the matrix product $M_y = M_{y_1} \cdot M_{y_2} \ldots M_{y_m}$. Then $M_y(i, j)$ is 1 if $\varphi^*(i, y) = j$ and 0 otherwise.

A uniform input string $x \in \Sigma^t$ corresponds to a random walk of length t on the states of the DFA A starting from the start state q_0. By Lemma 1 and 2, we can assume the irreducibility and aperiodicity of the random walk. Due to the uniqueness of the strongly connected component, the walk will finally converge to the stationary distribution ϕ with any start state q_0. For any string $y = y_1 y_2 \ldots y_m$, we define the distribution vector p_y over the state space Q obtained by starting from the stationary distribution ϕ and inputting string y to the automaton. That is, $p_y = \phi M_y$ and $p_\lambda = \phi$. Consequently, each string $y \in \Sigma^*$ and symbol $\sigma \in \Sigma$ contribute a linear equation $p_y M_\sigma = p_{y\sigma}$ where $y\sigma$ is the concatenation of y and σ. Due to Theorem 4, the diameter of a random DFA is $O(\log_s n)$ with high probability. The complete set of $\Theta(\log_s n)$-step walks should have already traversed the whole graph and no new information can be retrieved after $\Theta(\log_s n)$ steps. Hence, we can only consider the equation set $\{p_y M_\sigma = p_{y\sigma} \mid y \in \Sigma^{O(\log_s n)}\}$ for each $\sigma \in \Sigma$. We further observe that the equation system $\{p_y M_\sigma = p_{y\sigma} \mid y \in \Sigma^{\Theta(\log_s n)}\}$ shares the same solution with $\{p_y M_\sigma = p_{y\sigma} \mid y \in \Sigma^{O(\log_s n)}\}$. Let vector z be the i-th column of matrix M_σ, matrix P_A be the $s^{\Theta(\log_s n)} \times n$ coefficient matrix whose rows are $\{p_y \mid y \in \Sigma^{\Theta(\log_s n)}\}$ and vector b be the vector consisting of $\{p_{y\sigma}(i) \mid y \in \Sigma^{\Theta(\log_s n)}\}$.

The task reduces to solving the linear equation system $P_A z = b$ for z. Let ϕ_t be the distribution vector over Q after t steps of random walk. As the random walk always starts from the start state q_0, the initial distribution ϕ_0 is a coordinate vector whose entry of q_0 is 1 and the rest are 0, for which

$$2\|\phi_t - \phi\|_{TV} \le \left(\sum_{v\in V} \frac{(\phi_t(v) - \phi(v))^2}{\phi(v)}\right)^{\frac{1}{2}} \le \max_{u\in V}\left(\sum_{v\in V} \frac{(P^t(u,v) - \phi(v))^2}{\phi(v)}\right)^{\frac{1}{2}} = \Delta_{\chi^2}(t)$$

Theorem 1 claims that a polynomially large $t_0 = 2C(C+1)sn^{1+C}(\log n + \log \frac{2}{\tau}) \cdot \log_s n$ is enough to have the random walk converge to $p_\lambda = \phi$ within any polynomially small χ-square distance $\frac{\tau}{2}$ with high probability where $C > 0$ is the constant in the theorem. Let $t = t_0 + C \log_s n$, which is still polynomially large. We can estimate the stationary distribution for a state i by the fraction of examples (x, q) such that $q = i$. In general, for any string y, we can estimate the value of p_y for a state i as the ratio between the number of pairs (x, q) such that y is a suffix of x and $q = i$ and the number of examples (x, q) where y is a suffix of x.

In the statistical query model we are unable to directly observe the data; instead we are given access to the oracle *STAT*. Define a conditional statistical query $\chi_{y,i}(x, q) = \mathbb{1}\{q = i \mid y \text{ is a suffix of } x\}$ where $\mathbb{1}$ is the boolean indicator function. It's easy to see the legitimacy and feasibility of query $\chi_{y,i}(x, q)$ for any $y \in \Sigma^{\Theta(\log_s n)}$ because: (1) it is a boolean function mapping an example (x, q) to $\{0, 1\}$; (2) the proposition $\mathbb{1}\{q = i\}$ can be tested in $O(1)$ time; (3) the condition $\mathbb{1}\{y \text{ is a suffix of } x\}$ can be tested within $\Theta(\log_s n)$ time; (4) the probability of the condition that y is a suffix of x is inverse polynomially large $s^{-|y|} = s^{-\Theta(\log_s n)} = \Theta(n^{-C})$ for some constant $C > 0$.

Let $\tilde{p}_\lambda$ be the distribution vector over the states after t steps and $\tilde{p}_y = \tilde{p}_\lambda M_y$. Also denote by vector $\widehat{p}_y$ the query result returned by oracle $STAT$ where $\widehat{p}_y(i)$ is the estimate $\mathbf{E}\chi_{y,i}$, and by $\widehat{P}_A$ and $\widehat{b}$ the estimates for P_A and b respectively from oracle *STAT*. We infer the solution z by solving the perturbed linear least squares problem: $\min_z \|\widehat{P}_A z - \widehat{b}\|_2$. Let $\widehat{z}$ be the solution we obtain from this perturbed problem. According to the main theorem, the distance $\|p_\lambda - \tilde{p}_\lambda\|_1 = 2\|\phi_t - \phi\|_{TV} \le \Delta_{\chi^2}(t) \le \frac{\tau}{2}$. Then for any string y, $\|p_y - \tilde{p}_y\|_\infty = \|(p_\lambda - \tilde{p}_\lambda)M_y\|_\infty \le \|p_\lambda - \tilde{p}_\lambda\|_1 \le \frac{\tau}{2}$. If we do the statistical queries with tolerance $\frac{\tau}{2}$, the maximum additive error will be $\|\tilde{p}_y - \widehat{p}_y\|_\infty \le \frac{\tau}{2}$ for any string y. Thus we have $\|p_y - \widehat{p}_y\|_\infty \le \tau$. To conclude a theoretical upper bound on the error, we use the following theorem by Björck [5], which was later refined by Higham [10].

Theorem 5 *Let z be the optimal solution of linear least squares problem* $\min_z \|Mz - b\|_2$ *and $\widehat{z}$ be the optimal solution of* $\min_z \|\widehat{M}z - \widehat{b}\|_2$*. If $|M - \widehat{M}| \precsim \omega E$ and $|b - \widehat{b}| \precsim \omega f$ for some element-wise non-negative matrix E and vector f, where $|\cdot|$ refers to element-wise absolute value and $\precsim$ means element-wise $\le$ comparison, then*

$$\|z - \widehat{z}\|_\infty \le \omega(\||M^\dagger|(E|z| + f)\|_\infty + \||(M^\top M)^{-1}|E^\top|Mz - b|\|_\infty) + O(\omega^2)$$

when M has full column rank, or

$$\|z-\widehat{z}\|_\infty \le \omega(\||\widehat{M}^\dagger|(E|\widehat{z}|+f)\|_\infty + \||(\widehat{M}^\top\widehat{M})^{-1}|E^\top|\widehat{M}\widehat{z}-\widehat{b}|\|_\infty) + O(\omega^2)$$

when $\widehat{M}$ has full column rank, where $M^\dagger$ is the MoorePenrose pseudoinverse of matrix M.

Applying Theorem 5 to our case gives an upper bound on the maximum absolute error.

Corollary 2 *If P_A has full rank with high probability,*

$$\|z-\widehat{z}\|_\infty \le \frac{(1+\varepsilon)\log ns}{\log\log ns}\||P_A^\dagger|\|_\infty \tau + O(\tau^2)$$

with probability $1-o(1)$ for any constant $\varepsilon > 0$.

Proof First in our case the offset $|P_A z - b| = 0$ and $\omega = \tau$. Matrix E is the all-one matrix and vector f is the all-one vector. As a consequence, $\|f\|_\infty = 1$ and $\|E|z|\|_\infty = \|z\|_1$. Now it remains to prove with high probability $\|z\|_1 \le \frac{(1+\varepsilon)\log ns}{\log\log ns}$ for all columns in all $M_\sigma, \sigma \in \Sigma$.

Let θ be the largest 1-norm of the columns in M_σ. According to the properties of a random DFA, the probability of $\theta > n$ is 0 and $\Pr[\theta = n] \le n \cdot n^{-n}$ is exponentially small. For any $k < n$,

$$\begin{aligned}
&\Pr[\theta \ge k] \le n \cdot \Pr[\text{a particular column has 1-norm at least } k] \\
\le& n \cdot \binom{n}{k}\left(\frac{1}{n}\right)^k \\
\le& \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}}{\sqrt{2\pi k}\left(\frac{k}{e}\right)^k e^{\frac{1}{12k+1}} \cdot \sqrt{2\pi(n-k)}\left(\frac{n-k}{e}\right)^{n-k} e^{\frac{1}{12(n-k)+1}}} \cdot n\left(\frac{1}{n}\right)^k \\
\le& \sqrt{\frac{n^3 s^2}{2\pi k(n-k)s^2}} \cdot \frac{e^{\frac{1}{12n}}(n)^n}{(nk)^k(n-k)^{n-k}} \\
\le& \frac{1}{s} \cdot e^{\log ns + n\log n - k\log k - (n-k)\log(n-k) - k\log n + \frac{1}{12n}}
\end{aligned}$$

We only need to choose a k such that the exponent goes to $-\infty$, which is equal to

$$\log ns + k\left(1-\frac{n}{k}\right)\log\left(1-\frac{k}{n}\right) - k\log k + \frac{1}{12n}$$

If $k \ge n$ then $\Pr[\theta \ge k]$ is exponentially small as discussed above. Otherwise we have $\left(1-\frac{n}{k}\right)\log\left(1-\frac{k}{n}\right) \le 1$ in our case. Also notice that $\frac{1}{12n} \le 1$. Let $k = \frac{(1+\varepsilon)\log ns}{\log\log ns}$. The expression is upper bounded by

$$
\begin{aligned}
&\log ns + \frac{(1+\varepsilon)\log ns}{\log\log ns} - \frac{(1+\varepsilon)\log ns}{\log\log ns}\log\frac{(1+\varepsilon)\log ns}{\log\log ns} + 1 \\
=&\log ns + \frac{(1+\varepsilon)\log ns}{\log\log ns} - \frac{(1+\varepsilon)\log ns}{\log\log ns}(\log(1+\varepsilon) + \log\log ns - \log\log\log ns) + 1 \\
=&-\varepsilon\log ns + \left(\frac{1-\log(1+\varepsilon)}{\log\log ns} + \frac{\log\log\log ns}{\log\log ns}\right)(1+\varepsilon)\log ns + 1
\end{aligned}
$$

With respect to n and s, the expression goes to $-\infty$. There are in total s matrices $\{M_\sigma \mid \sigma \in \Sigma\}$. Using a union bound we have $\|z\|_1 \leq \frac{(1+\varepsilon)\log ns}{\log\log ns}$ for all columns in all M_σ with probability $1-o(1)$, and plugging this upper bound into the conclusion of Theorem 5 completes the proof. ■

This further implies that if we set the tolerance $\tau = \frac{\log\log ns}{3|||P_A^\dagger|||_\infty \log ns}$, the solution error $\|z - \widehat{z}\|_\infty < \frac{1}{2}$ with high probability. Based on the prior knowledge we have on z, we could refine $\widehat{z}$ by rounding up $\widehat{z}$ to a binary vector $\tilde{z}$, i.e., for each $1 \leq i \leq n$, $\tilde{z}(i) = 1$ if $\widehat{z}(i) > \frac{1}{2}$ and 0 otherwise, whereby we will have $\tilde{z}(q) = z(q)$ for any state q in the strongly connected component. A toy example is provided in the appendices to demonstrate how the algorithm works.

Our algorithm only recovers the strongly connected component $\tilde{A}$ of a random DFA A because it relies on the convergence of the random walk and any state $q \notin \tilde{A}$ will have zero probability after the convergence. We have no information for reconstructing the disconnected part. In the positive direction, due to Lemma 1, with high probability we are able to recover at least 79.68% of the DFA for any $s \geq 2$ and at least 99.9% of the whole automaton if $s > 6$. Because $\tilde{A}$ is unique and closed, it is also a well defined DFA. In Section 3 we have proved $\min_{q\in Q}\{p_\lambda(q) \mid p_\lambda(q) > 0\} \geq n^{-1}s^{-Diam} = n^{-C}$ for some constant $C > 0$ with high probability. This means we have a polynomially large gap so that we are able to distinguish the recurrent states from the transient ones by making a query to estimate $\tilde{p}_\lambda(q)$ for each state $q \in Q$. In our result $|||P_A^\dagger|||_\infty$ is regarded as a parameter. It might be possible to improve the result by polynomially bounding $|||P_A^\dagger|||_\infty$ with other given parameters n and s using random matrix theory technique. The full-rank assumption is reasonable because a random matrix is usually well conditioned and full-rank. From the empirical results in Section 4.2, the coefficient matrix P_A is almost surely full-rank and $|||P_A^\dagger|||_\infty$ is conjecturally $\leq ns\log s$. Furthermore, according to Corollary 1, our algorithm is also applicable to learning a random DFA after minimization.

4.2 Experiments and empirical results

In this section we present a series of experimental results to study the empirical performance of the learning algorithm, which was run in MATLAB on a workstation built with Intel i5-2500 3.30GHz CPU and 8GB memory. To be more robust against fluctuation from randomness, each test was run for 20 times and the medians were taken. The automata are generated uniformly at random as defined

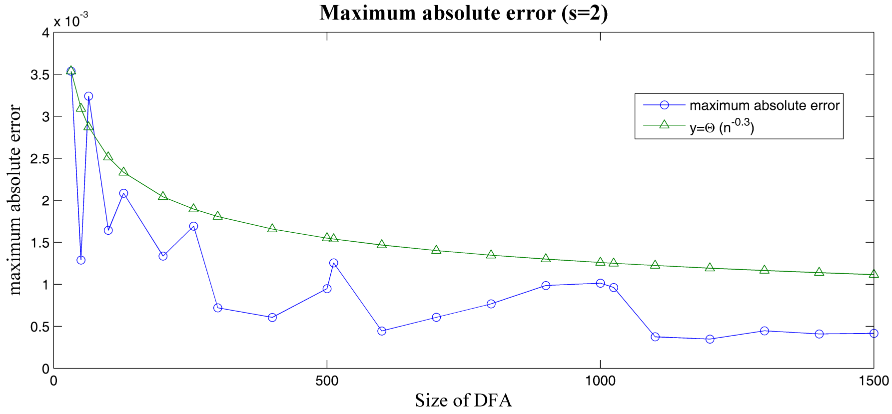

Fig. 1. Maximum absolute error versus n with fixed $s = 2$

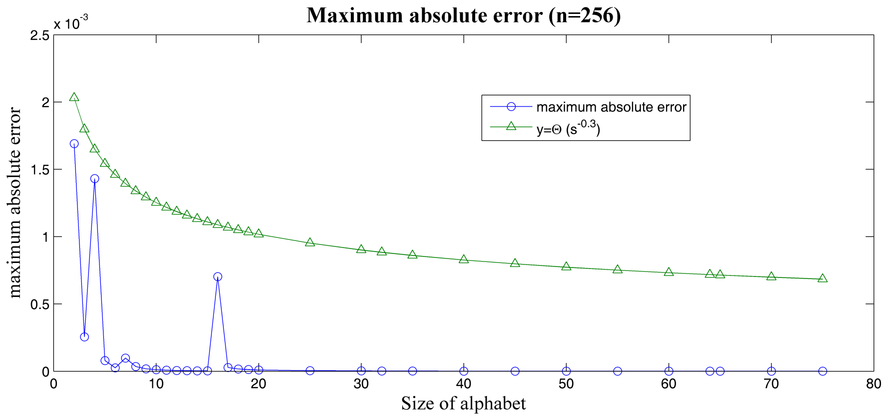

Fig. 2. Maximum absolute error versus s with fixed $n = 256$

and the algorithm solves the equation system $\{p_y M_\sigma = p_{y\sigma} \mid y \in \Sigma^{\leq \lceil \log_s n \rceil}\}$ using the built-in linear least squares function in MATLAB. We simulate the statistical query oracle with uniform additive noise.

The experiments start with an empirical estimate for the norm $|||P_A^\dagger|||_\infty$. We first vary the automaton size n from 32 to 4300 with fixed alphabet size $s = 2$. Figure 3 (in the appendices) shows the curve of $|||P_A^\dagger|||_\infty$ versus n with fixed s. Notice that the threshold phenomenon in the plot comes from the ceiling operation in the algorithm configuration. When n is much smaller than the threshold $s^{\lceil \log_s n \rceil}$, the system is overdetermined with many extra equations. Thus it is robust to perturbation and well-conditioned. When n grows up and approaches the threshold $s^{\lceil \log_s n \rceil}$, the system has fewer extra equations and becomes relatively more sensitive to perturbations, for which the condition number increases until the automaton size reaches $n = s^i$ of the next integer i. One can avoid this threshold phenomenon by making the size of the equation system grow smoothly as n increases. We then fix n to be 256 and vary s from 2 to 75, as shown in Figure 4 (in the appendices). Similarly there is the threshold phenomenon resulting from the ceiling strategy. All peaks where $n = s^i$ are included and plotted. Meanwhile the rank of P_A is measured to support the full-rank assumption. Matrix P_A is almost surely full-rank for large n or s and both figures suggest an upper bound $ns \log s$ for $|||P_A^\dagger|||_\infty$. We set the query tolerance τ as $\frac{\log \log ns}{ns \log ns \log_2 s}$ in the algorithm and measure the maximum absolute error $\|z - \widehat{z}\|_\infty$ at each run. Figures 1 and 2 demonstrate the experimental results. Along with the error curve in each figure a function is plotted to approximate the asymptotic order of the decline rate of the error. An empirical error bound is $O(n^{-0.3})$ with fixed s and $O(s^{-0.3})$ with fixed n.

5 Discussion

In this paper we prove fast convergence of random walks on a random DFA and apply this theoretical result to learning a random DFA in the statistical query model. One potential future work is to validate the full-rank assumption or to polynomially bound $|||P_A^\dagger|||_\infty$ using the power of random matrix theory. Note that $|||P_A^\dagger|||_\infty$ reflects the asymmetry of the automaton graph. The class of permutation automata [19] is one example that has symmetric graph structure and degenerate P_A. Another technical question on the fast convergence result is whether it can be generalized to weighted random walks on random DFAs. An immediate benefit from this generalization is the release from the requirement of uniform input strings in the DFA learning algorithm. However, we conjecture such generalization requires a polynomial lower bound on the edge weights in the graph, to avoid exponentially small nonzero elements in the walk matrix P. A further generalization is applying this algorithm to learning random probabilistic finite automata. In this case we will have a similar linear equation system, but the solution vector z can be continuous, not necessarily being a binary vector.

Acknowledgments. We thank Borja Balle Pigem for helpful discussions and the anonymous reviewers of *ALT* 2015 for their valuable comments.

References

1. D. Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, Nov. 1987.
2. D. Angluin, J. Aspnes, S. Eisenstat, and A. Kontorovich. On the learnability of shuffle ideals. *Journal of Machine Learning Research*, 14:1513–1531, 2013.
3. D. Angluin, D. Eisenstat, L. A. Kontorovich, and L. Reyzin. Lower bounds on learning random structures with statistical queries. In *ALT*, 2010.
4. B. Balle. Ergodicity of random walks on random DFA. *CoRR*, abs/1311.6830, 2013.
5. A. Björck. Component-wise perturbation analysis and error bounds for linear least squares solutions. *BIT Numerical Mathematics*, 31(2):237–244, 1991.

6. D. Chen. Learning shuffle ideals under restricted distributions. In *NIPS*, 2014.
7. F. Chung. Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics*, 9:1–19, 2005.
8. Y. Freund, M. Kearns, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie. Efficient learning of typical finite automata from random walks. In *STOC*, 1993.
9. A. A. Grusho. Limit distributions of certain characteristics of random automaton graphs. *Mathematical notes of the Academy of Sciences of the USSR*, 1973.
10. N. J. Higham. A survey of componentwise perturbation theory in numerical linear algebra. In *Proceedings of symposia in applied mathematics*, 1994.
11. J. C. Jackson, H. K. Lee, R. A. Servedio, and A. Wan. Learning random monotone DNF. In *RANDOM*, pages 483–497. Springer, 2008.
12. J. C. Jackson and R. A. Servedio. Learning random log-depth decision trees under uniform distribution. *SIAM Journal on Computing*, 34(5):1107–1128, 2005.
13. M. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, Nov. 1998.
14. M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, Jan. 1994.
15. L. Pitt and M. K. Warmuth. The minimum consistent DFA problem cannot be approximated within any polynomial. *J. ACM*, 40(1):95–142, Jan. 1993.
16. J. Ruiz and P. Garcia. Learning k-piecewise testable languages from positive data. In *Grammatical Interference: Learning Syntax from Sentences*. 1996.
17. L. Sellie. Learning random monotone DNF under the uniform distribution. In *COLT*, pages 181–192. Citeseer, 2008.
18. L. Sellie. Exact learning of random DNF over the uniform distribution. In *STOC*, pages 45–54. ACM, 2009.
19. G. Thierrin. Permutation automata. *Theory of Computing Systems*, 1968.
20. B. A. Trakhtenbrot and I. M. Barzdin. Finite automata; behavior and synthesis. *Fundamental Studies in Computer Science, V. 1*, 1973.
21. L. G. Valiant. A theory of the learnable. *Commun. ACM*, Nov. 1984.

Appendix: A toy example

Suppose we consider the alphabet $\{0, 1\}$ and a 3-state DFA with the following transition matrices.

$$M_0 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ and } M_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

For this automaton, the stationary distribution p_λ is $(1/3, 4/9, 2/9)$. As $\lceil \log_s n \rceil = \lceil \log_2 3 \rceil = 2$, the algorithm recovers the first column of matrix M_0, denoted by $z = (M_0(1,1), M_0(2,1), M_0(3,1))^\top$, by solving the overdetermined equation system

$$\begin{cases} p_{00} \cdot z = p_{000}(1) \\ p_{01} \cdot z = p_{010}(1) \\ p_{10} \cdot z = p_{100}(1) \\ p_{11} \cdot z = p_{110}(1) \end{cases}, \text{ i.e., } \begin{cases} \frac{1}{3}M_0(1,1) + \frac{2}{3}M_0(2,1) + 0M_0(3,1) = \frac{2}{3} \\ 0M_0(1,1) + \frac{2}{3}M_0(2,1) + \frac{1}{3}M_0(3,1) = 1 \\ 1M_0(1,1) + 0M_0(2,1) + 0M_0(3,1) = 0 \\ 0M_0(1,1) + \frac{4}{9}M_0(2,1) + \frac{5}{9}M_0(3,1) = 1 \end{cases}$$

Similarly the algorithm recovers all columns in M_0 and M_1 and reconstructs the target automaton. Note that in the statistical query model the above equation system is perturbed but we showed the algorithm is robust to statistical query noise.

Appendix: Estimate of $|||P_A^\dagger|||_\infty$

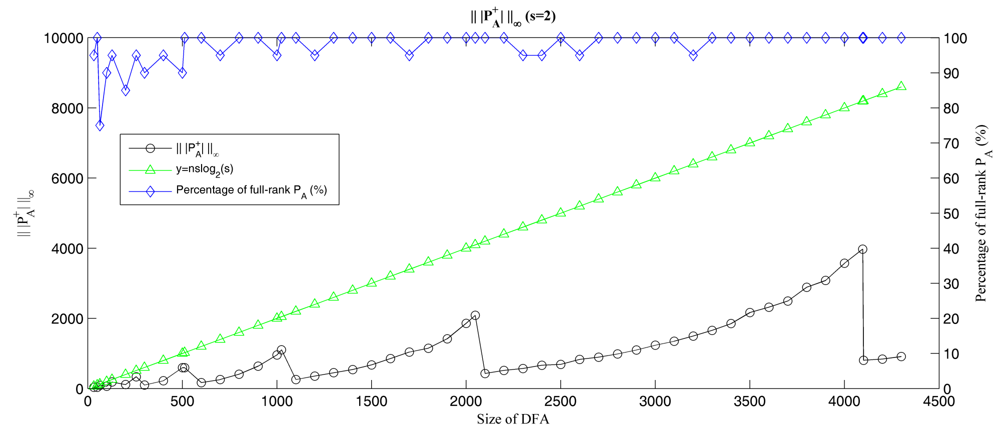

Fig. 3. $|||P_A^\dagger|||_\infty$ versus n with fixed $s = 2$

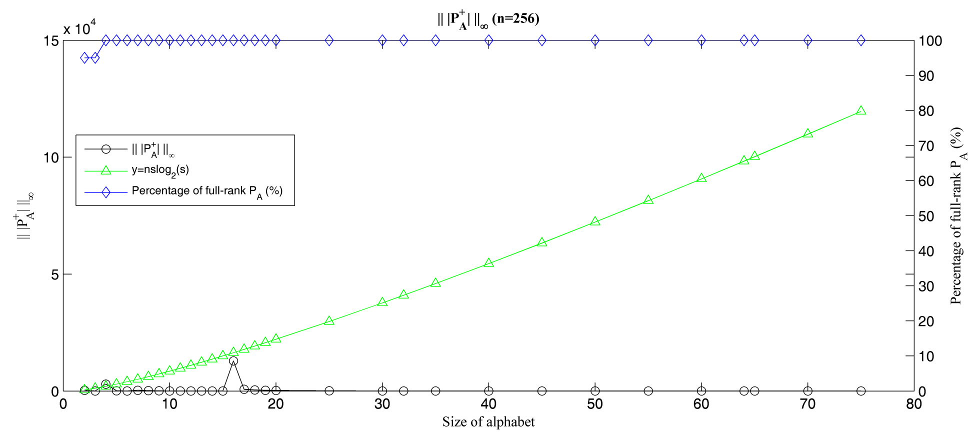

Fig. 4. $|||P_A^\dagger|||_\infty$ versus s with fixed $n = 256$