Abstract

Learning Regular Languages and Automaton Graphs

Dongqu Chen

2016

Learning regular languages has long been a fundamental topic in computational learning theory. In this thesis, we present our contributions to exploring the learnability of regular languages and their representation class, deterministic finite automata (DFAs).

To study the learnability of regular languages in the context of machine learning, we first need to understand how humans learn and acquire a language. We consider a society which consists of n people (or agents), where pairs of individuals are drawn uniformly at random to interact. Each individual has a confidence level for a grammar and a more confident person supports the grammar with higher probability. A person increases her confidence level when interacting with another person supporting the grammar, and decreases her confidence level otherwise. We prove that with high probability the three-state binary signaling process reaches consensus after $\Theta(n \log n)$ interactions in the worst case, regardless of the initial configuration. In the general case, the continuous-time binary signaling process in the limit will converge within $O(r \log nr)$ time (corresponding to $O(nr \log nr)$ interactions in expectation) if the initial configuration is monotone, where r is the number of confidence levels. In the other direction, we also show a convergence lower bound $\Omega(nr + n \log n)$ on the number of interactions when r is large.

The class of shuffle ideals is an important sub-family of regular languages. The

shuffle ideal generated by a string set U is the collection of all strings containing some string $u \in U$ as a (not necessarily contiguous) subsequence. We study the PAC learnability of shuffle ideals and present positive results on this learning problem under element-wise independent and identical distributions and Markovian distributions in the statistical query model. A constrained generalization to learning shuffle ideals under product distributions is also provided. In the empirical direction, we propose a heuristic algorithm for learning shuffle ideals from given labeled strings under general unrestricted distributions.

As a representation class of regular languages, DFAs are one of the most elementary computational models in the study of computer science. 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 absolute error of the algorithm in the statistical query model is presented.

Given that automaton graphs are out-regular, we generalize our DFA learning algorithm to learning random regular graphs in the statistical query model from random paths. In a standard label-guided graph exploration setting, the edges incident from a node in the graph have distinct local labels. The input data to the statistical query oracle are path-vertex pairs (x, v) where x is a random uniform path (a random sequence of edge labels) and v is the vertex of the graph reached on the path x starting from a particular start vertex v_0.

Learning Regular Languages and Automaton Graphs

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Dongqu Chen

Dissertation Director: Dana Angluin

2016

To my family

Acknowledgments

I could never have been more fortunate to have Professor Dana Angluin as my advisor. I came to Yale with little background in learning theory and my research progress seemed hopeless at the beginning. It was Dana who encouraged me and led me to the right track of theoretical research with her tremendous patience and constant guidance throughout my Ph.D. study. Dana has amazingly ingenious insights and a singular vision in learning theory, and I am completely in awe of her approach towards solving problems: a seamless blend of intuition and rigor. Her invaluable advice was always the beacon for my Ph.D. career. From Dana I learned that a real scholar should be wise, dedicated, humble and optimistic, not only for research but also for life. Working with Dana is a pleasure and an honor, both personally and intellectually, and undoubtedly one of the best experiences in my life.

I owe a great amount of gratitude to Professor James Aspnes, who has guided me in the study of the language emergence process with so many stimulating discussions and constructive suggestions. I have benefited a great deal from every meeting with Jim. He has always been open to giving me insightful ideas and bringing me an exceptional sense of the directions that are worth pursuing. Jim's generous support is crucial in the formation of this thesis.

I would like to thank my thesis committee members Professor Amin Karbasi and Professor Rocco Servedio (Columbia). Their valuable questions, insights and com-

ments have greatly improved the content of this work. I also thank Professor Joseph Chang for suggesting supportive references for our shuffle ideal learning algorithms.

I owe special thanks to our department. Without any research grants, I have been supported by the department for my years at Yale. As the heads of our department, Professor Holly Rushmeier and Professor Joan Feigenbaum have been actively seeking for various sources of funding for my study and freed me of any concerns respecting grants.

I am grateful to my undergraduate advisor at USTC, Professor Guangzhong Sun, for enlightening me in doing scientific research and inspiring me to pursue my Ph.D.

I am indebted to my research supervisor at UC Berkeley, Professor Dawn Song, for shaping me as a researcher and leading me to the magnificent academic world.

I enjoyed every minute working alongside my amazing colleagues: Professor Ivan Martinovic (Oxford), Professor Neil Gong (ISU), Dr. Ling Huang (Intel Research), Dr. Cynthia Kuo (Nokia Research) and Zvonimir Pavlinovic (NYU).

I was fortunate to be surrounded by so many brilliant, hard-working and nice fellow students at Yale CS who supported and inspired me everyday. I had the great pleasure of sharing my life at Yale with Ronghui Gu, Jiewen Huang, Jeremie Koenig, Huamin Li, Hongqiang Liu, Yitzchak Lockerman, Xueyuan Su, Huan Wang, Su Xue, Ennan Zhai and all my other friends who made this journey so enjoyable and fruitful.

My deepest appreciation goes to my parents and my brother for their boundless love. I would not be able to achieve anything without the constant encouragement and persistent support from mom and dad. My brother Dongpeng Chen was my first teacher of machine learning and has offered me an immeasurable amount of valuable advice throughout my Ph.D. life. This thesis is dedicated to them.

Contents

1 Introduction **1**

1.1 Language emergence process and population protocols 1

1.2 Learning shuffle ideals 6

1.3 Learning a random DFA 8

1.4 Learning random regular graphs 9

2 A Binary Signaling Model for Language Emergence **12**

2.1 Binary Signaling Consensus Model 13

2.2 Three-State Binary Signaling Consensus 15

2.2.1 The main theorem 15

2.2.2 Bounding $S_\tau^{sc} = O(n \log n)$ 20

2.2.3 Bounding $S_\tau^{c} = O(n \log n)$ 40

2.2.4 Bounding $S_\tau^{g} = O(n \log n)$ 41

2.2.5 Bounding $S_\tau^{b} = O(n \log n)$ and $S_\tau^{w} = O(n \log n)$ 45

2.3 Binary Signaling Consensus with $r > 2$ 48

2.3.1 Continuous-time binary signaling consensus 49

2.3.2 A convergence lower bound 60

2.4 Empirical Results and Conjectures 65

2.5 Conclusion and Future Work 71

3 Learning Shuffle Ideals Under Restricted Distributions 73

3.1 Preliminaries 74

3.2 Learning shuffle ideals from element-wise i.i.d. strings 77

3.2.1 Statistical query algorithm 78

3.2.2 PAC learnability 79

3.2.3 A generalization to instance space $\Sigma^{\leq n}$ 92

3.3 Learning principal shuffle ideals from Markovian strings 93

3.3.1 Statistical query algorithm 94

3.3.2 PAC learnability 94

3.4 A constrained generalization to learning shuffle ideals under product distributions 102

3.5 Learning shuffle ideals under general distributions 114

3.6 Discussion 119

4 Learning a Random DFA from Uniform Strings and State Information 121

4.1 Preliminaries 122

4.2 Random walks on a random DFA 123

4.3 Reconstructing a random DFA 129

4.3.1 The learning algorithm 129

4.3.2 Experiments and empirical results 136

4.4 Discussion 138

5 Learning Random Regular Graphs 140

5.1 Overview 141

5.2 Random walks on random regular graphs 143

5.2.1 Preliminaries 143

5.2.2 The main theorem 145
5.2.3 Fast convergence on RMG^+ and RSG^+ 146
5.2.4 Fast convergence on RMG^-, RSG^-, $RMG^\pm$ and $RSG^\pm$ 159
5.2.5 Fast convergence on RDG and RG 164
5.3 Reconstructing random regular graphs from random paths 183
5.3.1 Preliminaries 183
5.3.2 The learning algorithm 184
5.3.3 Experiments and empirical results 188
5.4 Other applications and discussion 191

List of Figures

2.1 The number of interactions with fixed resistance 2 and varying population 66

2.2 The number of interactions with fixed resistance 50 and varying population 66

2.3 The number of interactions with fixed population 1000 and varying resistance 68

2.4 Convergence time comparison for continuous process 69

3.1 The DFA accepting precisely the shuffle ideal of $U = (a|b|d)a(b|c)$ over $\Sigma = \{a, b, c, d\}$. 75

3.2 Definition of $\theta_{V,a}(x)$ when $V = U[1, \ell]$ 78

3.3 Approximately learning III under product distributions 107

3.4 A greedy algorithm for learning a principal shuffle ideal from example oracle EX 116

3.5 Experiment results with NSF abstracts data set (training 1993; testing 1992) 118

3.6 Experiment results with NSF abstracts data set (training 1999; testing 1998) 119

4.1 $|||P_A^\dagger|||_\infty$ versus n with fixed $s = 2$ 136

4.2 $|||P_A^\dagger|||_\infty$ versus s with fixed $n = 256$ 137
4.3 Maximum absolute error versus n with fixed $s = 2$ 138
4.4 Maximum absolute error versus s with fixed $n = 256$ 139

5.1 A 2-regular digraph with 4 vertices 187
5.2 $|||P_A^\dagger|||_\infty$ of $\mathrm{RSG}^+(s)$, versus n with fixed $s = 2$ 189
5.3 $|||P_A^\dagger|||_\infty$ of $\mathrm{RSG}^+(s)$, versus s with fixed $n = 256$ 190
5.4 $|||P_A^\dagger|||_\infty$ of $\mathrm{RDG}(s)$, versus n with fixed $s = 2$ 191
5.5 $|||P_A^\dagger|||_\infty$ of $\mathrm{RDG}(s)$, versus s with fixed $n = 256$ 192
5.6 $|||P_A^\dagger|||_\infty$ of $\mathrm{RG}(s)$, versus n with fixed $s = 3$ 193
5.7 $|||P_A^\dagger|||_\infty$ of $\mathrm{RG}(s)$, versus s with fixed $n = 242$ 194
5.8 Maximum absolute error for learning a $\mathrm{RSG}^+(s)$, versus n with fixed $s = 2$ 194
5.9 Maximum absolute error for learning a $\mathrm{RSG}^+(s)$, versus s with fixed $n = 256$ 195
5.10 Maximum absolute error for learning a $\mathrm{RDG}(s)$, versus n with fixed $s = 2$ 195
5.11 Maximum absolute error for learning a $\mathrm{RDG}(s)$, versus s with fixed $n = 256$ 196
5.12 Maximum absolute error for learning a $\mathrm{RG}(s)$, versus n with fixed $s = 3$ 196
5.13 Maximum absolute error for learning a $\mathrm{RG}(s)$, versus s with fixed $n = 242$ 197

List of Tables

2.1 Indicators and Counters 17

2.2 Changes in $(g - w)$ by state-changing interactions 30

2.3 Changes in $(3w + g + 1)$ 46

5.1 Random regular graphs with fixed degree s 142

Chapter 1

Introduction

This thesis presents our contributions to learning regular languages and automaton graphs. To study the learnability of regular languages in the context of machine learning, we first need to understand how humans learn and acquire a language. This thesis starts from a binary signaling model in Chapter 2 for the language emergence process in a human society. In Chapter 3, we study the PAC learnability of shuffle ideals, which are a fundamental sub-class of regular languages. In Chapter 4, we propose a computationally efficient algorithm for learning and recovering a random deterministic finite automaton (DFA) from uniform input strings and state information. Chapter 5 generalizes our results in Chapter 4 and applies the DFA learning algorithm to learning random regular graphs from uniform paths.

1.1 Language emergence process and population protocols

"A basic task of science is to build models — simplified and abstracted descriptions — of natural phenomena" [BM96, p. 432]. A central goal of modern linguistic

theory is to explain how people learn and acquire a language, and how languages emerge from communication among people. For decades, language scientists have spent lots of effort on capturing and modeling the language emergence process in human society. Linguists' intuitions about language emergence can be interpreted by dynamical system models, often with strong subjectivity and randomness, from the dynamics of human interactions and the nature of language acquisition. The works of Galantucci [Gal05] and Galantucci et al. [GFR03] develop experimental semiotics, which conduct controlled studies in conventionalization of form-meaning mappings among interacting agents where human develop novel languages, in an experimental way. The works of Coppola and Senghas [CS10] and Meir et al. [MSPA10] study the spontaneous emergence of gestural communication systems in deaf individuals not exposed to spoken or signed language and of natural languages in deaf communities, which offer unique opportunities to study the process of human language emergence. Language scientists have long been occupied with describing phonological, syntactic, and semantic change, often appealing to a relation between language change and evolution, but rarely going beyond analogy. The overall goal of Chapter 2 is to move from this analogy to formal modeling.

Chomsky proposed a model of universal grammar — those aspects of linguistic structure that are presumed innate and thus present in every linguistic system [Cho81, Cho93]. A language is defined by a series of parameters and the learning process for a learner of a language consists of constantly adjusting or fixing a number of parameters. Under this framework, Gibson and Wexler [GW94] formalized the Triggering Learning Model to focus our investigation of parameter learning. Initially, the process starts at some random point in the (finite) space of possible parameter settings, specifying a single hypothesized grammar with its resulting extension as a language. The learner keeps receiving a positive example sentences at each time

stamp from a uniform distribution on the language. If the current grammar doesn't parse the received sentence, the learner selects a single parameter uniformly at random, to flip from its current setting, and changes it if and only if that change allows the current sentence to be analyzed.

In the work of Richie et al. [RYC14], interactions happen randomly between people in the society and each agent updates its state according to the information it receives each time, which an instance of reinforcement learning. Each agent j has probability parameter p_j for a targeted grammar. Following the standard Linear-Reward-Penalty scheme in reinforcement learning [WVO12, p. 453], upon each communication, the listener agent j adjusts p_j to match the speaker agent i's choices: if j receives a message positive to the grammar from i, then $p_j = p_j + \gamma(1 - p_j)$; if j receives a negative message, then $p_j = (1 - \gamma)p_j$, where the learning rate γ is typically a small real number. Kirby et al. [KDG07] use a simple Bayesian method to understand the evolution of language. In this approach, the degree to which a learner should believe in a particular hypothesis (i.e., support or object to a new grammar) is a direct combination of their innate biases and the extent to which the data are consistent with that hypothesis. The agents can then choose their opinions about the grammar based on their degrees of belief. An agent might simply employ the hypothesis that has higher posterior probability, sample from the posterior distribution, or do anything in between.

However, the above language emergence models have very little theoretical analysis and no guarantee on convergence. In fact, the Triggering Learning Model never halts in the usual sense, so does everything built upon it. In Chapter 2, we present a novel and simple model for language emergence process with a neat mathematical framework and formal results on the convergence rate. Lightfoot [Lig91, p. 163] talks about language evolution in this way: "Some general properties of language change

are shared by other dynamic systems in the world". Chapter 2 is good evidence of this statement. Our language emergence model shares some similarities with the population protocols in distributed computing theory. To the best of our knowledge, this is the first work that builds the connection between evolutionary linguistics and population protocols.

A population protocol [AAD$^+$06] is where agents may interact in pairs and each individual agent is extremely limited (in fact, being equipped only with a finite number of possible states). Then the complex behavior of the system emerges from the rules governing the possible pairwise interactions of the agents. The agents in a population protocol are anonymous, i.e., there is only one transition function which is common to all agents and the output of the transition function only depends on the states of the two involved agents, regardless of their identities. Nor does each agent have any knowledge of its identity. Usually it is assumed that interactions between agents happen under some kind of a fairness condition.

Angluin et al. [AAE07] introduced a simple population protocol for majority computation. This protocol assigns only three possible states to every agent, including two opposite states and one intermediate state, and initially every agent starts from one of the two opposite states. There is a 3×3 transition table capturing all possible interactions and the interactions between agents are dictated by a probabilistic scheduler. The essential idea of this protocol is that when two agents with different preferences meet, one drops its preference and enters the intermediate state; an agent at the intermediate state adopts the preference of any biased agent it meet. Nothing happens when two unbiased agents meet. The protocol converges at the point where all the agents have the same preference with no unbiased agents left. They show that with high probability this protocol reaches convergence within $O(n \log n)$ interactions with a complete interaction graph of n vertices, if the process starts from a

biased initial configuration. In addition, if the difference between the initial majority and the initial minority is $\omega(\sqrt{n}\log n)$, their protocol converges to the correct initial majority with high probability.

Becchetti et al. [BCN+15] is the most recent result we know of that generalizes Angluin et al.'s three-state population protocol to computing plurality consensus in the gossip model. Instead of two opposite preferences, an agent could have one of many preferences (or "colors" in the paper). The update rule is the same: when two agents with different preferences meet, one shifts to the intermediate state (blank); a blank agent will be colored by a colored agent with its color. Another major difference concerns timing. They analyze the synchronous version of population protocol in the gossip model, where at each round every agent updates its state simultaneously. Perron et al. [PVV09] analyzed the continuous-time process of Angluin et al.'s three-state population protocol in the limit by studying the corresponding system of differential equations modeling the expected change of the protocol. An additional continuous-time three-state protocol is defined where instead of being passive, a blank agent acts as in the two opposite states uniformly at random in an interaction. The authors gave an elegant upper bound on the time to convergence of a differential equation approximation that converges to the behavior of the discrete process for any fixed time in the limit by Kurtz's theorem [Kur81]. They claim the stronger result that this approximation converges for time $\Theta(\log n)$. While this claim may in fact be true, applying Kurtz's theorem in this case requires an unjustified interchange of limits that gives incorrect results in many cases. To avoid this issue, we employ a potential-function approach similar to that used by Angluin et al. [AAE07]. For more related works and theoretical background about population protocols we refer to the survey of Aspnes and Ruppert [AR09].

Binary signaling consensus in the context of population protocols is where two in-

teracting agents communicate with only one binary bit, without knowing each other's state or identity. The population protocol reaches consensus if all the agents have the same preference and the process stays put forever. For example, the protocol in [AAE07] is not a binary signaling one since the communication between two interacting agents depend on their states, which are ternary, while the second protocol in [PVV09] is, even though the state space of an agent is also ternary. Modeling the human language emergence process in evolutionary linguistics is one scenario of binary signaling consensus. Given the connection between population protocols and biological systems [CCN12], more potential applications of binary signaling consensus may be found in biology and related fields.

1.2 Learning shuffle ideals

The learnablity of regular languages is a classic topic in computational learning theory. The applications of this learning problem include natural language processing (speech recognition, morphological analysis), computational linguistics, robotics and control systems, computational biology (phylogeny, structural pattern recognition), data mining, time series and music [Kos83, DLH05, Moh96, MPR02, Moh97, MMW10, RBB+02, SGSC96]. Exploring the learnability of the family of formal languages is significant to both theoretical and applied realms. In the classic PAC learning model defined by Valiant [Val84], unfortunately, the class of regular languages, or the concept class of deterministic finite automata (DFA), is known to be inherently unpredictable [Ang78, Gol78, PW93, KV94]. In a modified version of Valiant's model which allows the learner to make membership queries, Angluin [Ang87] has shown that the concept class of regular languages is PAC learnable. Subsequent efforts have searched for nontrivial properly PAC learnable subfamilies

of regular languages [AAEK13, Che14, RG96].

Throughout Chapter 3 we study the *PAC learnability* of a fundamental subclass of regular languages, the class of *(extended) shuffle ideals*. The shuffle ideal generated by an augmented string U is the collection of all strings containing some $u \in U$ as a (not necessarily contiguous) subsequence, where an *augmented string* is a finite concatenation of symbol sets (see Figure 3.1 for an illustration). The special class of shuffle ideals generated by a single string is called the *principal* shuffle ideals. In spite of its simplicity, the class of shuffle ideals plays a prominent role in formal language theory. The boolean closure of shuffle ideals is the important language family known as piecewise-testable languages [Sim75]. The rich structure of this language family has made it an object of intensive study in complexity theory and group theory [Lot83, KP08]. In the applied direction, Kontorovich et al. [KRS03] show that shuffle ideals capture some rudimentary phenomena in human language morphology.

Unfortunately, even such a simple class is not PAC learnable, unless RP=NP [AAEK13]. However, in most application scenarios, the strings are drawn from some particular distribution we are interested in. Angluin et al. [AAEK13] prove under the uniform string distribution, principal shuffle ideals are PAC learnable. Nevertheless, the requirement of complete knowledge of the distribution, the dependence on the symmetry of the uniform distribution and the restriction to principal shuffle ideals lead to the lack of generality of the algorithm. Our main contribution in Chapter 3 is to present positive results on learning the class of shuffle ideals under element-wise independent and identical distributions and Markovian distributions. Extensions of our main results include a constrained generalization to learning shuffle ideals under product distributions and a heuristic method for learning principal shuffle ideals under general unrestricted distributions.

1.3 Learning a random DFA

Deterministic finite automata are one of the most elementary computational models in the study of theoretical computer science. The important role of DFAs leads to the classic problem in computational learning theory, the learnability of DFA. Unfortunately, the concept class of DFAs is long known to be not efficiently learnable in the classic PAC learning model define by Valiant [Val84].

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 [JLSW08, JS05, Sel08, Sel09]. However, the learnability of random DFAs has long been an open problem. Few formal results about random walks on random DFAs are known. Grusho [Gru73] 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 [Bal13]. 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 [TB73] first introduced two random DFA models with different sources of randomness: one with a random automaton graph, one with random output labeling. In Chapter 4 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. [FKR+97] 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 [TB73]. 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 Chapter 4, 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 4.3.1), 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. [AEKR10] but remains open under the uniform distribution [Bal13]. Our contribution narrows this gap and pushes forward the study of the learnability of random DFAs.

1.4 Learning random regular graphs

The realm of random graph study was first established by Erdös and Rényi [ER59, ER60, ER61b] after Erdös [Erd47, Erd59, Erd60] had discovered that probabilistic methods that introduce randomness were often useful in tackling extremal problems in graph theory. There were subsequently several works on the study of graph properties of the Erdös-Rényi model such as connectivity [ER64], chromatic number [LW86, Bol88] and cliques [BE76]. These had not at that time gathered much attention until the introduction at the end of the twentieth century of the small world

model [WS98] and the preferential attachment model [BA99] led to an explosion of research.

Random walks on graphs serve as an important tool in the study of Markov chains and graph models. Major previous contributions for random walks on graphs are in the cover time and hitting time [Ald89, Fei95b, Fei95a], mixing rate [LS90] and spectrum [Chu97]. However, the volume of literature studying random walks on random graphs [Jon98, CF08] is much smaller. See Lovász's paper [Lov93] for a survey of random walks on undirected graphs and Cooper's paper [CF09] for an overview of random walks on undirected random graphs.

The study of random regular graphs started with the works of Bender [Ben74], Bollobás [Bol80] and Wormald [Wor81]. Their applications in computer science soon led to a large volume of subsequent works in this area (see the survey by Wormald [Wor99]). Most of these contributions are on the topics of asymptotic enumeration, chromatic number and Hamilton cycles. Nevertheless, research on random walks on random regular graphs is very limited in the literature. Hildebrand [Hil94] showed the fast convergence of random walks on a $\mathrm{RG}(s)$ with the constraint $s = \Theta(\log^C n)$ for some constant $C > 2$ and Cooper and Frieze [CF05] studied the cover time with fixed constant $s = O(1)$ but no convergence result was presented. In the context of DFA learning, Angluin and Chen [AC15] first proved the fast convergence of random walks on a $\mathrm{RMG}^+(s)$ for $s \geq 2$. In Chapter 5, we aim to fill this gap and prove positive convergence results of random walks on a series of random regular graphs. These fast convergence properties together with our results in Chapter 4 inspire us to study the problem of learning random regular graphs in the setting of label-guided graph exploration.

The first known algorithm designed for graph exploration was introduced by Shannon [Sha51]. Since then, many subsequent works have studied the feasibility

of graph exploration in the port numbering setting. Rollik [Rol79] gave a complete proof of that no robot with a finite number of pebbles can explore all graphs. The result holds even when restricted to planar 3-regular graphs. Without pebbles, it was proved [FIP+04] that a robot needs $\Theta(Diam \cdot \log s_0)$ bits of memory to explore all graphs of diameter $Diam$ and maximum degree s_0.

In Chapter 4 we proposed a random-walk based algorithm for learning random DFAs. Observing the connection between DFA learning and label-guided graph exploration, along with the fast convergence results we prove in Chapter 5, we generalize our algorithm in Chapter 4 to learning random regular graphs of fixed out-degree s. The learning model we use is Kearns' statistical query model [Kea98], which is a variant of Valiant's PAC model [Val84] and implies stronger learnability. Learning graphs from exploration is a long studied theoretical learning problem [BS94, BFR+98], where the graphs are usually assumed out-regular. We follow Bender and Slonim's settings but in the passive learning scenario where blind agents passively explore the graph on random paths with no memory of visited vertices. In a regular graph of out-degree s, the s edges incident from a node are associated to s distinct port numbers in $\{1, 2, \ldots, s\}$ in a one-to-one manner, which is a standard label-guided graph exploration setting [FIP+04, Rei05, BS94]. Each edge of a node is labeled with its local port number. The input data to the statistical query oracle are of the form (x, v) where x is a random uniform path (a sequence of edge labels) of a fixed length and v is the vertex of the graph reached on the path x starting from a particular start vertex v_0.

Chapter 2

A Binary Signaling Model for Language Emergence

How do people learn and acquire a language? How do languages emerge in human society? It has become one of the major challenges of modern linguistic theory to understand the language emergence process in human society. In this chapter, we establish the connection between the language emergence process and the study of population protocols, and propose a novel and simple binary signaling consensus model for the language emergence process. We consider a society which consists of n people (or agents), where pairs of individuals are drawn uniformly at random to interact. Each individual has a confidence level for a grammar and a more confident person supports the grammar with higher probability. A person increases her confidence level when interacting with another person supporting the grammar, and decreases her confidence level otherwise. In Section 2.1 we formalize our binary signaling consensus model for human language emergence. A comprehensive analysis of fast convergence for three-state binary signaling consensus then follows in Section 2.2. We show that with high probability, the three-state binary signaling process

converges after $\Theta(n \log n)$ interactions in the worst case, regardless of the initial configuration. In Section 2.3, we study the general binary signaling consensus model with large resistance r. We prove that the continuous-time binary signaling process with large r in the limit will reach consensus within $O(r \log nr)$ time (corresponding to $O(nr \log nr)$ interactions in expectation) if the initial configuration is monotone. We also provide a convergence lower bound of $\Omega(nr + n \log n)$ on the number of interactions in the general case. Experimental results are presented in Section 2.4 to support our theoretical results and to provide evidence for some conjectures.

The content of this chapter appears in [AAC16].

2.1 Binary Signaling Consensus Model

We consider a society of population n. Define an *interaction graph* $G = (V, E)$ over this society to be a directed graph with $|V| = n$ whose edges indicate the possible interactions that may take place. Each agent $i \in V$ in the society has a *confidence level* $cl(i)$ for a grammar, which is an integer between 0 and r. We say r is the *resistance* (or the recalcitrance). At each step, an edge (i, j) is chosen uniformly at random from E. The "source" agent i is the *initiator* (or the speaker), and the "sink" agent j is the *responder* (or the listener). The two agents communicate in a way that the initiator sends a binary bit to the responder. With probability $cl(i)/r$ agent i sends a positive bit to agent j and the latter does the update $cl(j) = \min(cl(j)+1, r)$. Otherwise the initiator sends a negative bit to the responder who updates $cl(j) = \max(cl(j) - 1, 0)$. Starting from an initial configuration, the communication process keeps going until *convergence*, where either all agents are of confidence level r (when the whole society accepts the grammar, i.e., *positive convergence*) or all agents are of confidence level 0 (when the grammar is discarded, i.e., *negative convergence*).

We inherit the terminology "binary signaling" used by [PVV09] and call this model a *binary signaling consensus model* for the language emergence process, in the sense that the signaling between the agents is binary, although the state of an agent is $(r+1)$-ary. In this chapter, we study the cases where the interaction graph G is a complete graph. For algebraic convenience we assume self-loops are allowed in the interaction graph, while all our results can be easily applied to the setting of no self-loops as n goes to infinity.

The parameter r is called the resistance or the recalcitrance as the larger r is, the more difficult to persuade a person of the opposite opinion. A more general model could allow different people to have different resistances, and this is true in real life. In this setting, the range of the confidence level of agent i is from 0 to $r(i)$. Everything remains the same except that the initiator i has probability $cl(i)/r(i)$ of sending a positive bit and the responder updates $cl(j) = \min(cl(j)+1, r(j))$. Although this general setting simulates better the real-world situation, it complicates the model and violates the anonymity condition in population protocols, where the output of the transition function should be independent of the identities of the two involved agents. Hence, in this chapter we assume all agents are of the same resistance r.

Here we model a grammar as a single binary bit in the communication process. It would be more realistic to model a grammar as a binary vector or a set of binary bits which are usually not independent of each other, to model the complexity of a human grammar in linguistics. However, this again complicates the setting and is not theoretically very tractable. This vector-grammar model could be an interesting generalization of our work.

Note that in our setting we consider one particular preference and all agents eventually either accept the preference or reject it. Some readers might prefer an equivalent setting where we consider two opposite preferences (corresponding to being

supportive and being opposed in our setting) and the population eventually agrees with one of them. However, this would make the concrete meaning of confidence level confusing in some contexts. Thus in this chapter we employ the setting we have described above.

To the best of our knowledge, this is the first model that establishes a connection between evolutionary linguistics and population protocols. Unlike the population protocol model proposed in previous works, the convergence of our model is guaranteed with any initial configuration. In addition, our model is a better simulation of real-world language emergence process, as languages and grammars develop gradually from interactions among the society with randomness. In another direction, our model has advantages over the heuristic models in cognitive science (see related works in Section 1.1) that it converges much faster and has a neat mathematical framework and theoretical analysis provided in this chapter.

2.2 Three-State Binary Signaling Consensus

When simulating language emergence in a society, it is common to assume the population n is very large with constant value of resistance r. Since the $r = 0$ case is trivial and the $r = 1$ model doesn't involve probabilistic interactions, the three-state case with $r = 2$ is a reasonable start for us to study this model. In this section, we will prove that starting from any initial configuration, a grammar will eventually survive or be discarded within $\Theta(n \log n)$ interactions with high probability.

2.2.1 The main theorem

Let τ_* be the number of interactions until the three-state binary signaling model reaches consensus. The main result of this section is the following theorem. Note

that the stated convergence bound is a worst-case bound. A best-case bound is trivially $\tau_* = 0$, starting from consensus.

Theorem 2.1 *With probability* $1 - o(1)$, $\tau_* = \Theta(n \log n)$ *in the worst case. In addition, for any constant* $c > 0$ *we have*

$$\mathbb{P}\left(\tau_* \geq 96930(c+1)n \log n\right) \leq \max\left(9n^{-c}, \frac{c \log n}{\sqrt[3]{n}}\right)$$

The convergence lower bound $\tau_* = \Omega(n \log n)$ can be easily obtained from the well-known coupon collector bound. When the initial configuration is $cl(i)$ being 1 for all $i \in V$, in order to achieve consensus, every agent must participate in at least one interaction, leading to the coupon collector lower bound.

Lemma 2.1 *With probability* $1 - o(1)$, $\tau_* = \Omega(n \log n)$ *in the worst case.*

However, the upper bound $\tau_* = O(n \log n)$ requires a substantial amount of work. It may be surprising that fast convergence of such a simple consensus process needs such a lengthy proof. Part of the reason is that we want to obtain exact asymptotic bounds with explicit constants that work for arbitrary configurations.

The core of our proof is to construct a supermartingale for each region in the configuration space. This technique is inspired by the proof used by Angluin et al. [AAE07]. Recall that a *supermartingale* is a discrete stochastic process $\{M_t\}$ where M_t satisfies $\mathbb{E}(|M_t|) < +\infty$ and $\mathbb{E}(M_t \mid M_0, \ldots, M_{t-1}) \leq M_{t-1}$. The expected value of each M_t is bounded by the initial value $\mathbb{E}M_t \leq \mathbb{E}M_0$. Supermartingales are commonly studied with a *stopping time*. A stopping time with respect to a stochastic process $\{M_t\}$ is an almost surely finite random variable τ with positive integer values and the property that the event $\tau = t$ depends only on the values of $M_0, M_1, \ldots, M_t$. A supermatingale with a stopping time is still a supermartingale. In this section,

Indicator	Counter
I_t^{g-}: g decreases by 1	$S_t^{g-} = \sum_{i=1}^{t} I_i^{g-}$
I_t^{g+}: g increases by 1	$S_t^{g+} = \sum_{i=1}^{t} I_i^{g+}$
I_t^{sc}: the configuration is changed	$S_t^{sc} = \sum_{i=1}^{t} I_i^{sc}$
I_t^{c}: $\max(\tilde{b}, \tilde{g}, \tilde{w}) < 3/4$	$S_t^{c} = \sum_{i=1}^{t} I_i^{c}$
I_t^{b}: $\tilde{b} \geq 3/4$	$S_t^{b} = \sum_{i=1}^{t} I_i^{b}$
I_t^{g}: $\tilde{g} \geq 3/4$	$S_t^{g} = \sum_{i=1}^{t} I_i^{g}$
I_t^{w}: $\tilde{w} \geq 3/4$	$S_t^{w} = \sum_{i=1}^{t} I_i^{w}$

Table 2.1: Indicators and Counters

we let $\tau = \min(\tau_*, dn \log n)$ for some fixed constant d. Thus τ is a stopping time. This truncation guarantees that τ and quantities defined in terms of it are finite and well-defined, despite the logical possibility that convergence is not achieved and τ_* is ill-defined.

Now that $r = 2$ and an agent has only three possible states, we denote by w (white), g (gray) and b (black) the states with confidence levels 0 (negative), 1 (neutral) and 2 (positive) respectively. For notational convenience we also overload b, g, y to denote the number of each token in a configuration. Meanwhile, let $\tilde{b} = b/n$, $\tilde{g} = g/n$ and $\tilde{w} = w/n$ be the corresponding proportions. Obviously we always have $\tilde{b}+\tilde{g}+\tilde{w} = 1$. Denote $u = b-w$ and $v = b+w$. Note that $-n \leq u \leq n$ and $0 \leq v \leq n$. The point when $|u| = n$ is equivalent to convergence. The change of basis to u and v allows us to take advantage of the symmetry between b and w tokens. Auxiliary 0-1 indicators and counters for the proof are defined in Table 2.1.

The key to constructing a supermartingale in a region is to design a proper potential function that drops smoothly inside this region and doesn't increase too much elsewhere. Because the behavior of the consensus process is qualitatively different in different regions, we choose a specific potential function for each region of the configuration space. In our proof, we divide the configuration space into four regions:

1. The corner region where at least $3n/4$ agents are of confidence level 0 and $I^w = 1$;

2. The corner region where at least $3n/4$ agents are of confidence level 1 and $I^g = 1$;

3. The corner region where at least $3n/4$ agents are of confidence level 2 and $I^b = 1$;

4. The central region left where the tokens are more evenly balanced and $I^c = 1$.

More concretely, given that the potential function f decreases consistently by $-\Theta(n^{-1})$ in expectation when $I_t^1 = 1$ and increases by a relatively smaller amount in expectation when $I_t^2 = 1$, we are able to construct a stochastic process of the form $\{M_t = \exp((c_1 S_t^1 - c_2 S_t^2)/n) \cdot f\}$ which is a supermartingale, where I_t^1 and I_t^2 are two different binary indicators, $S_t^1 = \sum_{i=1}^t I_t^1$ and $S_t^2 = \sum_{i=1}^t I_t^2$ are their counters, and c_1 and c_2 are two carefully chosen positive constants. The supermartingale property $\mathbb{E}M_\tau \leq \mathbb{E}M_0$ together with Markov's inequality then gives us the desired $O(n \log n)$ upper bound for S_τ^1 (depending on S_τ^2). Here we assume either S_τ^2 is already well bounded (Lemma 2.8, Lemma 2.9, Lemma 2.10 and Lemma 2.11), or there exists some auxiliary inequality relationship between S_τ^1 and S_τ^2 (Lemma 2.4 and Lemma 2.6). A formal statement of this proof technique is presented in Lemma 2.3.

The proof of the upper bound consists of four components. Notice that $t = S_t^c + S_t^b + S_t^g + S_t^w$ for any time t. Thus upper bounds for S_τ^c, S_τ^b, S_τ^g and S_τ^w imply one for τ. We will later find that these four quantities can be bounded using an upper bound on the number of state-changing interactions S_τ^{sc}. Therefore, the proof starts with an $O(n \log n)$ upper bound for S_τ^{sc}.

In three-state binary signaling consensus, every state-changing interaction must increase or decrease the value of g by 1. Hence, we have $S_\tau^{sc} = S_\tau^{g+} + S_\tau^{g-}$. The proof of bounding $S_\tau^{sc} = O(n \log n)$ (Lemma 2.2) is done case by case. First we

show that if the process starts from some point in the region $\{g \leq \min(b,w)/4\}$, then within $O(n \log n)$ state-changing interactions, it will either converge or leave the region (Lemma 2.4). If the former happens then we are happy. Otherwise, we have $g > \min(b,w)/4$ and we prove that within the next $O(n \log n)$ state-changing interactions, either the process will never enter the region $\{g < \min(b,w)/10\}$ again, or it will enter the region $\{\min(b,w) = O(\log n) \bigwedge g = O(\log n)\}$ (Lemma 2.5 and Corollary 2.1). In the first case, we show the population protocol will converge within the next $O(n \log n)$ state-changing interactions (Lemma 2.6). In the latter case, we show the protocol will converge within the next $O(n)$ state-changing interactions (Lemma 2.7).

Based on the upper bound on state-changing interactions, we are able to construct a family of supermartingales for different regions in the configuration space. To bound the number of interactions S_τ^c in the central region, we prove the stochastic process $C_t = \exp((S_t^c - 9S_t^{sc})/n)$ to be a supermartingale. The key observation is that in the central region where $\max(\tilde{b}, \tilde{g}, \tilde{w}) < 3/4$, we should have either $\tilde{b}$ and $\tilde{w}$ are both $\geq 1/8$, or $\tilde{g} \geq 1/8$. We then show that in both cases we have C_t dropping in expectation, which implies an $O(n \log n)$ upper bound for S_τ^c. For the corner region where $\tilde{g} \geq 3/4$, we choose the potential function to be $f = 1/(2v+1)$. We show that this potential function drops consistently by $\Theta(-1/n)$ of its current value in expectation in the large-g region, while its rise when $I_t^g = 0$ can be upper-bounded by $O(I_t^{g+}/n)$. With this we then construct a supermartingale in the form $M = \exp(aS/n)f(b,w)$ as described above and achieve the bound $S_\tau^g = O(n \log n)$. For the corner region where $b \geq 3n/4$, the potential function we use is $f = 3w + g + 1$. Similar to the idea of bounding S_τ^g, we bound S_τ^b by showing the value of the potential function decreases by a factor of $\exp(-\Theta(1/n))$ when b is large, and increases otherwise by an amount we can bound using the previous $O(n \log n)$

bounds on S_t^{g+} and S_t^{g-}. Thus the number of interactions S_τ^b that happen in the large-b region is also $O(n \log n)$. The number of interactions S_τ^w that happen in the large-w region can be bounded in a symmetric way using the potential function $f = 3b + g + 1$. Finally, for $\tau = S_\tau^c + S_\tau^b + S_\tau^g + S_\tau^w$, summing the bounds for all the four regions we will obtain a bound on the total number of interactions. Given a convergence upper bound $O(n \log n)$ with an explicit constant c, we then choose a slighter larger constant $d > c$ to truncate the process and let $\tau = \min(\tau_*, dn \log n)$ to make τ a well defined stopping time. Some readers might think this truncation at $\Theta(n \log n)$ interactions already assumes the correctness of the target statement, but we have proved that the total number of interactions is smaller than $dn \log n$ with high probability so we have the convergence upper bound $\tau_* = O(n \log n)$ as stated in Theorem 2.1.

2.2.2 Bounding $S_\tau^{sc} = O(n \log n)$

In this section we show the number of state-changing interactions S_τ^{sc} is at most $O(n \log n)$ with high probability. In the three-state model, every state-changing interaction must increase or decrease the value of g by 1. Hence, we have $S_\tau^{sc} = S_\tau^{g+} + S_\tau^{g-}$ with the following upper bounds.

Lemma 2.2 *With probability* $1 - o(1)$, $S_\tau^{sc} = O(n \log n)$. *In addition, for any constant* $c > 0$ *we have*

$$\mathbb{P}\left(S_\tau^{sc} \geq 372.72(c+1)n \log n\right) \leq \max\left(5n^{-c}, \frac{c \log n}{\sqrt[3]{n}}\right)$$

$$\mathbb{P}\left(S_\tau^{g+} \geq 186.36(c+1)n \log n\right) \leq \max\left(4n^{-c}, \frac{c \log n}{\sqrt[3]{n}}\right)$$

and

$$\mathbb{P}\left(S_\tau^{g-} \geq 186.36(c+1)n\log n\right) \leq \max\left(4n^{-c}, \frac{c\log n}{\sqrt[3]{n}}\right)$$

The essential idea of our proof is to construct a family of supermartingales for different regions in the configuration space by carefully selecting a series of corresponding potential functions. The following lemma is a general statement of this proof technique.

Lemma 2.3 *Let f be a potential function and A be a region in the configuration space. If in region A, $\mathbb{E}(\Delta f/f \mid I^1) \leq -k_1/n$ and $\mathbb{E}(\Delta f/f \mid I^2) \leq k_2/n$ where k_1 and k_2 are two constants such that $k_1 > k_2 > 0$, and I^1 and I^2 are two binary indicators such that $I_t^1 \cdot I_t^2 \equiv 0$ at any number of interactions t, then the stochastic process $\{M_t\}$ given by*

$$M_t = \exp\left(\frac{c_1 S_t^1 - c_2 S_t^2}{n}\right) \cdot f_t$$

is a supermartingale in region A, where $S_t^1 = \sum_{i=1}^t I_t^1$ and $S_t^2 = \sum_{i=1}^t I_t^2$, and c_1, c_2 are two constants such that $k_1 > c_1 > c_2 > k_2 > 0$.

In addition, given $f_0/f_t \leq n^{c_3}$ for some positive constant $c_3 > 0$ at any number of interactions t, if the process never leaves region A, we have

$$\mathbb{P}\left(c_1 S_\tau^1 \geq c_2 S_\tau^2 + (c_3 + c_4)n\log n\right) \leq n^{-c_4}$$

for any positive constant $c_4 > 0$.

Proof Given

$$\mathbb{E}\left(\frac{\Delta f}{f} \mid I^1\right) = \mathbb{E}\left(\frac{f_{t+1} - f_t}{f_t} \mid I_{t+1}^1\right) \leq -\frac{k_1}{n}$$

and

$$\mathbb{E}\left(\frac{\Delta f}{f} \mid I^2\right) = \mathbb{E}\left(\frac{f_{t+1} - f_t}{f_t} \mid I_{t+1}^2\right) \leq \frac{k_2}{n}$$

we have

$$\mathbb{E}\left(f_{t+1} \mid I_{t+1}^1\right) \leq \left(1 - \frac{k_1}{n}\right) \cdot f_t \leq \exp\left\{-\frac{c_1}{n}\right\} \cdot f_t$$

and

$$\mathbb{E}\left(f_{t+1} \mid I_{t+1}^2\right) \leq \left(1 + \frac{k_2}{n}\right) \cdot f_t \leq \exp\left\{\frac{c_2}{n}\right\} \cdot f_t$$

Boosting the constants from $-k_1$ to $-c_1$ and from k_2 to c_2 is to absorb the second-order and higher terms in the Taylor series expansion of the exponential.

The expected value of M_{t+1} in each case is as follows.

$$\mathbb{E}\left(M_{t+1} \mid I_{t+1}^1 + I_{t+1}^2 = 0\right) = M_t$$

$$\begin{aligned}
\mathbb{E}\left(M_{t+1} \mid I_{t+1}^1\right) =& \mathbb{E}\left(\exp\left(\frac{c_1(S_t^1 + 1) - c_2 S_t^2}{n}\right) \cdot f_{t+1} \mid I_{t+1}^1\right) \\
=& \mathbb{E}\left(\frac{M_t \cdot f_{t+1} \cdot \exp(c_1/n)}{f_t} \mid I_{t+1}^1\right) \\
=& \exp\left\{\frac{c_1}{n}\right\} \cdot \mathbb{E}\left(f_{t+1} \mid I_{t+1}^1\right) \cdot \frac{M_t}{f_t} \\
\leq& M_t
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}\left(M_{t+1} \mid I_{t+1}^2\right) =& \mathbb{E}\left(\exp\left(\frac{c_1 S_t^1 - c_2(S_t^2 + 1)}{n}\right) \cdot f_{t+1} \mid I_{t+1}^2\right) \\
=& \mathbb{E}\left(\frac{M_t \cdot f_{t+1} \cdot \exp(-c_2/n)}{f_t} \mid I_{t+1}^2\right) \\
=& \exp\left\{-\frac{c_2}{n}\right\} \cdot \mathbb{E}\left(f_{t+1} \mid I_{t+1}^2\right) \cdot \frac{M_t}{f_t} \\
\leq& M_t
\end{aligned}$$

In any case we always have $\mathbb{E}\left(M_{t+1}\right) \leq M_t$ so the stochastic process $\{M_t\}$ is a supermartingale in region A. If the process never leaves region A, we have $\mathbb{E}(M_\tau) \leq$

$M_0 = f_0$. Given $f_0/f_t \leq n^{c_3}$ at any number of interactions t (including the stopping time $t = \tau$), we have

$$\mathbb{E}(M_\tau) = \mathbb{E}\left(\exp\left(\frac{c_1 S_\tau^1 - c_2 S_\tau^2}{n}\right) \cdot f_\tau\right) \leq M_0 = f_0$$

and

$$\mathbb{E}\left(\exp\left(\frac{c_1 S_\tau^1 - c_2 S_\tau^2}{n}\right)\right) \leq n^{c_3}$$

From Markov's inequality,

$$\mathbb{P}\left(\exp\left(\frac{c_1 S_\tau^1 - c_2 S_\tau^2}{n}\right) \geq n^{c_3+c_4}\right) \leq n^{-c_4}$$

for any positive constant $c_4 > 0$ and then

$$\mathbb{P}\left(c_1 S_\tau^1 - c_2 S_\tau^2 \geq (c_3 + c_4) n \log n\right) \leq n^{-c_4}$$

which completes the proof. ■

Lemma 2.3 presents the proof technique we use throughout this section. When using this technique, we have either S_τ^2 is already well bounded (Lemma 2.8, Lemma 2.9, Lemma 2.10 and Lemma 2.11), or there exists some auxiliary inequality relationship between S_τ^1 and S_τ^2 (Lemma 2.4 and Lemma 2.6).

Lemma 2.4 *If the binary signaling consensus process starts with $g \leq \min(b, w)/4$, then for any constant $c > 0$, with probability $1 - n^{-c}$ one of the following two events will happen within $O_c(n \log n)$ state-changing interactions:*

1. $g > \min(b, w)/4$.

2. The process converges and

$$\mathbb{P}\left(S_\tau^{g-} \geq \frac{1000}{7}\left(n\log\left(\frac{2}{5}n+1\right)+cn\log n\right)+\frac{392}{7}n\right) \leq n^{-c}$$

and

$$\mathbb{P}\left(S_\tau^{g+} \geq \frac{1000}{7}\left(n\log\left(\frac{2}{5}n+1\right)+cn\log n\right)+\frac{399}{7}n\right) \leq n^{-c}$$

Proof We can prove this fact by showing that if event 1 doesn't happen, then event 2 will surely happen. That is, if we always have $g \leq \min(b,w)/4$ and never have $g > \min(b,w)/4$, then with probability $1-o(1)$ the process converges after $O(n\log n)$ state-changing interactions. For notational convenience, let value $f = u^2 + 5n/2$ so the potential function is $1/f$. We have

$$\begin{aligned}\Delta f =&(u+\Delta u)^2 + 5n/2 - u^2 - 5n/2\\ =&u^2 + 2u\Delta u + (\Delta u)^2 - u^2\\ =&2u(\Delta u) + (\Delta u)^2\end{aligned}$$

Because $|\Delta u| \leq 1$ and $|\Delta f| \leq 2|u|+1$, we have $|\Delta f/f| \leq (2|u|+1)/(u^2+5n/2) = O(\min(1/|u|, 2|u|/5n))$, which is maximized at $u=\Theta(\sqrt{n})$ so that $|\Delta f/f| = O(1/\sqrt{n})$.

Let I^{bw} be the indicator of the event that neither of the two agents in a state-changing interaction is in state gray. Let I^{gv} be the indicator of the event that the speaker in a state-changing interaction is in gray and the listener is in black or white. Denote by $p = \tilde{b} + \tilde{g}/2$ and by $M = 2bw + \frac{1}{2}gv$. The expected change of value f

conditioned on each case of state-changing interactions is as follows.

$$\begin{aligned}\mathbb{E}\left(\Delta f \mid I^{g-}\right) =& p(2u+1)+(1-p)(-2u+1)\\ =& 1+(2p-1)\cdot 2u\\ =& 1+2u\cdot\frac{2b+g-n}{n}\\ =& 1+\frac{2u^2}{n}\end{aligned}$$

$$\mathbb{E}\left(\Delta f \mid I^{bw}\right) = \frac{1}{2}(2u+1)+\frac{1}{2}(-2u+1)=1$$

$$\begin{aligned}\mathbb{E}\left(\Delta f \mid I^{gv}\right) =& (2u+1)\frac{w}{v}+(-2u+1)\frac{b}{v}\\ =& 1+2u\cdot\frac{w-b}{v}\\ =& 1-\frac{2u^2}{v}\end{aligned}$$

$$\mathbb{E}\left(\Delta f \mid I^{g+}\right) = \frac{2bw}{M}+\frac{gv}{2M}\left(1-\frac{2u^2}{v}\right)=1-\frac{gu^2}{M}$$

$$\begin{aligned}\mathbb{E}\left((\Delta f)^2 \mid I^{g-}\right) =& p(2u+1)^2+(1-p)(-2u+1)^2\\ =& p(4u^2+4u+1)+(1-p)(4u^2-4u+1)\\ =& 4u^2+1+(2p-1)\cdot 4u\\ =& 4u^2+1+4u\cdot\frac{2b+g-n}{n}\\ =& 4u^2+1+\frac{4u^2}{n}\end{aligned}$$

$$\mathbb{E}\left((\Delta f)^2 \mid I^{bw}\right) = \frac{1}{2}(2u+1)^2 + \frac{1}{2}(-2u+1)^2 = 4u^2+1$$

$$\begin{aligned}\mathbb{E}\left((\Delta f)^2 \mid I^{gv}\right) =& \frac{w}{v}(2u+1)^2 + \frac{b}{v}(-2u+1)^2 \\ =& \frac{w}{v}(4u^2+4u+1) + \frac{b}{v}(4u^2-4u+1) \\ =& 4u^2+1+\frac{w-b}{v}\cdot 4u \\ =& 4u^2+1-\frac{4u^2}{v}\end{aligned}$$

$$\begin{aligned}\mathbb{E}\left((\Delta f)^2 \mid I^{g+}\right) =& \frac{2bw}{M}(4u^2+1) + \frac{gv}{2M}\left(4u^2+1-\frac{4u^2}{v}\right) \\ =& 4u^2+1-\frac{2gu^2}{M}\end{aligned}$$

When $g \le \min(b,w)/4$, we have

$$\begin{aligned}M =& 2bw + \frac{1}{2}gv \\ =& 2\min(b,w)\cdot\max(b,w) + \frac{1}{2}g\cdot(\min(b,w)+\max(b,w)) \\ \ge& g\cdot\left(\frac{17}{2}\max(b,w) + \frac{1}{2}\min(b,w)\right)\end{aligned}$$

As $4g = 4(n-\min(b,w)-\max(b,w)) \le \min(b,w) \le \max(b,w)$, we know

$$\frac{4}{5}(n-\max(b,w)) \le \min(b,w) \le \max(b,w)$$

Note that function $\frac{17}{2}x + \frac{1}{2}y$ given $\frac{4}{5}(1-x) \le y \le x$ and $y \ge 0$ and $x \le 1$ is at least 4. Thus we have $M \ge 4gn$. Let $z = u^2/n$.

$$
\begin{aligned}
&\mathbb{E}\left(\frac{\Delta(1/f)}{1/f} \mid I^{g-}\right) \\
=&\mathbb{E}\left(-\frac{\Delta f}{f} + \left(\frac{\Delta f}{f}\right)^2 + O(n^{-3/2}) \mid I^{g-}\right) \\
=&-\frac{1+2u^2/n}{u^2+5n/2} + \frac{4u^2+1+4u^2/n}{(u^2+5n/2)^2} + O(n^{-3/2}) \\
=&\left(u^2+5n/2\right)^{-2} \cdot \left(-(1+2u^2/n)(u^2+5n/2) + 4u^2 + 1 + 4u^2/n\right) + O(n^{-3/2}) \\
=&\left(u^2+\frac{5n}{2}\right)^{-2}\left(-u^2 - \frac{5n}{2} - \frac{2u^2}{n}\left(u^2+\frac{5n}{2}\right) + 4u^2 + 1 + \frac{4u^2}{n}\right) + O(n^{-3/2}) \\
=&\left(u^2+\frac{5n}{2}\right)^{-2}\left(3u^2 - \frac{5n}{2} + 1 + \left(-u^2 - \frac{5n}{2} + 2\right)\cdot\frac{2u^2}{n}\right) + O(n^{-3/2}) \\
=&n^{-2}(z+5/2)^{-2}(3zn - 5n/2 + 1 + 2z(-zn - 5n/2 + 2)) + O(n^{-3/2}) \\
=&n^{-2}(z+5/2)^{-2}((3z-5/2)n + 1 - 2z(z+5/2)n + 4z) + O(n^{-3/2}) \\
=&\frac{1}{n}\cdot\frac{-2z^2+(3-5)z-5/2}{(z+5/2)^2} + \frac{1}{n^2}\cdot\frac{1+4z}{(z+5/2)^2} + O(n^{-3/2})
\end{aligned}
$$

where the first equality is due to $\frac{\Delta(1/f)}{1/f} = \sum_{i=1}^{+\infty}(-\Delta f/f)^i$ for $|\Delta f/f| < 1$. Note that function $\frac{-2x^2-2x-5/2}{(x+5/2)^2}$ given $x \ge 0$ is at most $-2/5$ and function $\frac{1+4x}{(x+5/2)^2}$ given $x \ge 0$ is at most $4/9$. Thus we have

$$
\mathbb{E}\left(\frac{\Delta(1/f)}{1/f} \mid I^{g-}\right) \le -\frac{2}{5}n^{-1} + \frac{4}{9}n^{-2} + O(n^{-3/2}) = -\frac{2}{5}n^{-1} + O(n^{-3/2})
$$

In the other case when $I^{g+}=1$, we have

$$\begin{aligned}
&\mathbb{E}\left(\frac{\Delta(1/f)}{1/f} \mid I^{g+}\right)\\
=&\mathbb{E}\left(-\frac{\Delta f}{f}+\left(\frac{\Delta f}{f}\right)^2+O(n^{-3/2}) \mid I^{g+}\right)\\
=&-\frac{1-gu^2/M}{u^2+5n/2}+\frac{4u^2+1-2gu^2/M}{(u^2+5n/2)^2}+O(n^{-3/2})\\
=&\left(u^2+5n/2\right)^{-2}\cdot\left(-(1-gu^2/M)(u^2+5n/2)+4u^2+1-2gu^2/M\right)+O(n^{-3/2})\\
=&\left(u^2+\frac{5n}{2}\right)^{-2}\left(-u^2-\frac{5n}{2}+\frac{gu^2}{M}\left(u^2+\frac{5n}{2}\right)+4u^2+1-\frac{2gu^2}{M}\right)+O(n^{-3/2})\\
=&\left(u^2+\frac{5n}{2}\right)^{-2}\left(3u^2-\frac{5n}{2}+1+\left(u^2+\frac{5n}{2}-2\right)\cdot\frac{gu^2}{M}\right)+O(n^{-3/2})\\
\le&\left(u^2+\frac{5n}{2}\right)^{-2}\left(3u^2-\frac{5n}{2}+1+\left(u^2+\frac{5n}{2}-2\right)\cdot\frac{u^2}{4n}\right)+O(n^{-3/2})\\
=&n^{-2}(z+5/2)^{-2}(3zn-5n/2+1+z(zn+5n/2-2)/4)+O(n^{-3/2})\\
=&n^{-2}(z+5/2)^{-2}((3z-5/2)n+1+z(z+5/2)n/4-z/2)+O(n^{-3/2})\\
=&\frac{1}{n}\cdot\frac{z^2/4+(3+5/8)z-5/2}{(z+5/2)^2}+\frac{1}{n^2}\cdot\frac{1-z/2}{(z+5/2)^2}+O(n^{-3/2})
\end{aligned}$$

Note that function $\frac{x^2/4+29x/8-5/2}{(x+5/2)^2}$ given $x\ge 0$ is at most $1001/2560$ and function $\frac{1-x/2}{(x+5/2)^2}$ given $x\ge 0$ is at most $4/25$. Thus we have

$$\mathbb{E}\left(\frac{\Delta(1/f)}{1/f} \mid I^{g+}\right)\le\frac{1001}{2560}n^{-1}+\frac{4}{25}n^{-2}+O(n^{-3/2})=\frac{1001}{2560}n^{-1}+O(n^{-3/2})$$

According to Lemma 2.3, we can see the stochastic process $\{L_t\}$ given by

$$L_t=\frac{\exp\left((0.399S_t^{g-}-0.392S_t^{g+})/n\right)}{u_t^2+5n/2}$$

is a supermartingale if we always have $g \leq \min(b, w)/4$. Because $u^2 + 5n/2 \leq n^2 + 5n/2$,

$$\mathbb{E}\left(\exp\left(\frac{0.399S_\tau^{g-} - 0.392S_\tau^{g+}}{n}\right)\right) \leq \frac{2(n^2 + 5n/2)}{5n} = \frac{2n}{5} + 1$$

and then

$$\mathbb{P}\left(0.399S_\tau^{g-} - 0.392S_\tau^{g+} \geq n\log(2n/5 + 1) + cn\log n\right) \leq n^{-c}$$

Because at any number of interactions t, the number of gray tokens the process has produced can't be more than the number of gray tokens the process has consumed plus n, we have $S_\tau^{g+} \leq S_\tau^{g-} + n$, giving the bound

$$\mathbb{P}\left(0.399S_\tau^{g-} - 0.392(S_\tau^{g-} + n) \geq n\log(2n/5 + 1) + cn\log n\right) \leq n^{-c}$$

and

$$\mathbb{P}\left(S_\tau^{g-} \geq \frac{1000}{7}\left(n\log\left(\frac{2}{5}n + 1\right) + cn\log n\right) + \frac{392}{7}n\right) \leq n^{-c}$$

which implies

$$\mathbb{P}\left(S_\tau^{g+} \geq \frac{1000}{7}\left(n\log\left(\frac{2}{5}n + 1\right) + cn\log n\right) + \frac{399}{7}n\right) \leq n^{-c}$$

which completes the proof. ■

Now we have shown that if the population starts from the region $\{g \leq \min(b, w)/4\}$, within $O(n\log n)$ state-changing steps, it will either reach consensus or leave the region with high probability. While once the process leaves the region and has $g > \min(b, w)/4$, we prove that within the next $O(n\log n)$ state-changing interac-

interaction	b	w	g	$g-w$
$(b,+,w)$		-1	$+1$	$+2$
$(w,-,b)$	-1		$+1$	$+1$
$(b,+,g)$	$+1$		-1	-1
$(w,-,g)$		$+1$	-1	-2
$(g,+,g)$	$+1$		-1	-1
$(g,-,g)$		$+1$	-1	-2
$(g,+,w)$		-1	$+1$	$+2$
$(g,-,b)$	-1		$+1$	$+1$

Table 2.2: Changes in $(g-w)$ by state-changing interactions

tions, either the population will never enter the region $\{g < \min(b,w)/10\}$, or it will enter the region $\{\min(b,w) = O(\log n) \bigwedge g = O(\log n)\}$ (Lemma 2.5 and Corollary 2.1).

Lemma 2.5 *If the process starts with* $g > \min(b,w)/4$*, then with probability* $1 - n^{-\omega(1)}$*, for any polynomial* $T = poly(n)$*, we have either* $g_t \geq \min(b_t, w_t)/10$ *holds for all* $1 \leq t \leq T$ *or at some stage* $1 \leq t \leq T$*, the process reaches* $\min(b,w) = O(\log n)$*,* $g = O(\log n)$ *and* $\max(b,w) = n - O(\log n)$*.*

Proof Again we can show this fact by showing that if latter event doesn't happen, former event will happen. Let's consider how the value of $(g - \min(b,w))$ changes in different state-changing interactions. Without loss of generality, assume that at the current time step $\max(b,w) = b$ and $\min(b,w) = w$. Let $N = ng + 2bw + \frac{1}{2}gv$. Table 2.2 lists all the cases.

Thus

$$\mathbb{P}(\Delta(g-w) = +1 \mid I^{sc}) = \left(bw + \frac{1}{2}gb\right)/N = \frac{n^2}{N}(1-\tilde{g}-\tilde{w})\left(\tilde{w} + \frac{1}{2}\tilde{g}\right)$$

$$\begin{aligned}\mathbb{P}(\Delta(g-w)=-1 \mid I^{sc}) &= \left(bg+\frac{1}{2}g^2\right)/N \\ &= \frac{n^2}{N}\left((1-\tilde{w}-\tilde{g})\tilde{g}+\frac{1}{2}\tilde{g}^2\right) \\ &= \frac{n^2}{N}\left((1-\tilde{w})\tilde{g}-\frac{1}{2}\tilde{g}^2\right)\end{aligned}$$

$$\begin{aligned}\mathbb{P}(\Delta(g-w)=+2 \mid I^{sc}) &= \left(bw+\frac{1}{2}gw\right)/N \\ &= \frac{n^2}{N}\left(\tilde{w}\left(1-\tilde{g}-\tilde{w}+\frac{1}{2}\tilde{g}\right)\right) \\ &= \frac{n^2}{N}\left(\tilde{w}\left(1-\tilde{w}-\frac{1}{2}\tilde{g}\right)\right)\end{aligned}$$

$$\mathbb{P}(\Delta(g-w)=-2 \mid I^{sc}) = \frac{n^2}{N}\left(\tilde{w}\tilde{g}+\frac{1}{2}\tilde{g}^2\right)$$

Note that if the process enters the region $\{g < \min(b,w)/10\}$ from the initial region $\{g > \min(b,w)/4\}$, it must pass through the region $\{\min(b,w)/10 \le g \le \min(b,w)/4\}$. We show that even passing through this intermediate region already requires strictly more than a polynomial number of state-changing interactions, let alone the whole fleeing path.

Note that function $\frac{(1-x-y)(x+y/2)}{(1-x)y-y^2/2}$ conditioned on $0 \le x \le (1-y)/2$ and $x/10 \le y \le x/4$ is always ≥ 4. Also, function $\frac{x(1-x-y/2)}{xy+y^2/2}$ conditioned on $0 \le x \le (1-y)/2$ and $x/10 \le y \le x/4$ is always ≥ 4. Thus when $w/10 \le g \le w/4$, we always have

$$\frac{\mathbb{P}(\Delta(g-w)=+1 \mid I^{sc})}{\mathbb{P}(\Delta(g-w)=-1 \mid I^{sc})} \ge 4 \text{ and } \frac{\mathbb{P}(\Delta(g-w)=+2 \mid I^{sc})}{\mathbb{P}(\Delta(g-w)=-2 \mid I^{sc})} \ge 4$$

The value of $(g-w)$ never stays put with $I^{sc}=1$.

When $\min(b,w)=\omega(\log n)$, the length of this gap $\min(b,w)/4 - \min(b,w)/10 = 3\min(b,w)/20$ is also $\omega(\log n)$. Let length $\ell = \omega(\log n)$. Consider the following one-dimensional random walk on integers from 0 to ℓ. State 0 is a reflecting barrier

always pushing the walk back to state 1. At any state $1 \leq i \leq \ell - 1$, the forward probability is $1/5$ and the backward probability is $4/5$. The walk starts at state 1 and we are interested in the first hitting time of state ℓ.

The number of steps until this random walk first hits state ℓ provides an upper bound on the number of interactions needed by the process to flee from the region $\{g > \min(b, w)/4\}$ and enter the region $\{g < \min(b, w)/10\}$, conditioned on the event that $\min(b, w) = \omega(\log n)$ always holds. Now we show it needs strictly more than a polynomial number of steps with high probability.

Note that every time that the walk hits state 0, the reflecting barrier "resets" it to state 1. Everything the walk does between two consecutive "resets" can be viewed as a Bernoulli trial. And we shall show with probability $1 - o(1)$, this Bernoulli process needs strictly more than polynomially many trials to succeed. Here for each trial, hitting 0 before hitting ℓ is a failure and otherwise it succeeds.

Denote by $\beta_i = \mathbb{P}(\text{hitting state 0 before hitting state } \ell \mid \text{starting at state } i)$. Then $\beta_0 = 1$ and $\beta_\ell = 0$. The probability of failure is β_1.

For any $1 \leq i \leq \ell - 1$, $\beta_i = \beta_{i+1}/5 + 4\beta_{i-1}/5$. Define

$$\begin{aligned}
\Delta\beta_i =& \beta_i - \beta_{i+1} \\
=& \beta_i - \frac{1}{5}\beta_{i+2} - \frac{4}{5}\beta_i \\
=& \frac{1}{5}(\beta_i - \beta_{i+2}) \\
=& \frac{1}{5}(\beta_i - \beta_{i+1} + \beta_{i+1} - \beta_{i+2}) \\
=& \frac{1}{5}\Delta\beta_i + \frac{1}{5}\Delta\beta_{i+1}
\end{aligned}$$

which implies $\frac{4}{5}\Delta\beta_i = \frac{1}{5}\Delta\beta_{i+1}$ or $\Delta\beta_{i+1} = 4\Delta\beta_i$. Thus $\Delta\beta_i = 4^i\Delta\beta_0 = 4^i(1 - \beta_1)$.

Note that

$$\sum_{i=0}^{\ell-1} \Delta\beta_i = \beta_0 - \beta_1 + \beta_1 - \beta_2 + \ldots + \beta_{\ell-1} - \beta_\ell = \beta_0 - \beta_\ell = 1$$

Then

$$\sum_{i=0}^{\ell-1} \Delta\beta_i = (1-\beta_1) \sum_{i=0}^{\ell-1} 4^i = (1-\beta_1) \cdot \frac{4^\ell - 1}{3} = 1$$

Therefore, $(1-\beta_1)(4^\ell - 1) = 3$ and $\beta_1 = 1 - 3/(4^\ell - 1)$.

Let c be any arbitrarily large constant. The probability that all the first n^c trials fail is

$$\beta_1^{n^c} = \left(1 - \frac{3}{4^\ell - 1}\right)^{n^c} = \left(1 - \frac{1}{n^{\omega(1)}}\right)^{n^c} = \exp\left(-n^{c-\omega(1)}\right) \sim 1 - n^{c-\omega(1)}$$

The probability goes to 1 in order $n^{\omega(1)-c}$. Thus with probability $1 - O(n^{-\omega(1)})$ the random walk won't hit state ℓ within a polynomial number of steps.

Therefore, we can have $g < \min(b,w)/10$ only when $\min(b,w) = O(\log n)$ happens. In this case $g < \min(b,w)/10 = O(\log n)$ too so the other event happens. ■

Because in this problem we are only interested in the next $O(n \log n)$ state-changing interactions, we have

Corollary 2.1 *If the process starts with $g > \min(b,w)/4$, then with probability $1 - n^{-\omega(1)}$, for any $T = O(n \log n)$, we have either $g_t \geq \min(b_t, w_t)/10$ holds for all $1 \leq t \leq T$ or at some stage $1 \leq t \leq T$, the process reaches $\min(b,w) = O(\log n)$, $g = O(\log n)$ and $\max(b,w) = n - O(\log n)$.*

Next we will show the process also converges fast within the region $\{g \geq \min(b,w) \cdot 1/10\}$.

Lemma 2.6 *If $g \geq \min(b,w)/10$ holds for a polynomial number of state-changing interactions, then for any constant $c > 0$, with probability $1 - n^{-c}$, after $O_c(n \log n)$ state-changing interactions, the process will converge and we have*

$$\mathbb{P}\left(S_\tau^{sc} \geq 87n \log\left(\frac{1}{64}n + 1\right) + 87cn \log n\right) \leq n^{-c}$$

$$\mathbb{P}\left(S_\tau^{g+} \geq \frac{87}{2}n \log\left(\frac{1}{64}n + 1\right) + \frac{87}{2}cn \log n + \frac{1}{2}n\right) \leq n^{-c}$$

and

$$\mathbb{P}\left(S_\tau^{g-} \geq \frac{87}{2}n \log\left(\frac{1}{64}n + 1\right) + \frac{87}{2}cn \log n + \frac{1}{2}n\right) \leq n^{-c}$$

Proof In this proof we use the potential function $1/(u^2 + 64n)$ and denote by $f = u^2 + 64n$. Similarly we have $\Delta f = 2u(\Delta u) + (\Delta u)^2$ and $|\Delta f/f| = O(1/\sqrt{n})$. Recall that $N = ng + 2bw + \frac{1}{2}gv$. We have

$$\begin{aligned}
\mathbb{E}\left(\Delta f \mid I^{sc}\right) =& \frac{ng}{N}\left(1 + \frac{2u^2}{n}\right) + \frac{2bw}{N} + \frac{gv}{2N}\left(1 - \frac{2u^2}{v}\right) \\
=& 1 + 2u^2 \cdot \left(\frac{ng}{N} \cdot \frac{1}{n} - \frac{gv}{2N} \cdot \frac{1}{v}\right) \\
=& 1 + \frac{gu^2}{N}
\end{aligned}$$

$$\begin{aligned}
&\mathbb{E}\left((\Delta f)^2 \mid I^{sc}\right) \\
=& \frac{ng}{N}\left(4u^2 + 1 + \frac{4u^2}{n}\right) + \frac{2bw}{N}(4u^2 + 1) + \frac{gv}{2N}\left(4u^2 + 1 - \frac{4u^2}{v}\right) \\
=& 4u^2 + 1 + 4u^2\left(\frac{ng}{N} \cdot \frac{1}{n} - \frac{gv}{2N} \cdot \frac{1}{v}\right) \\
=& 4u^2 + 1 + \frac{2gu^2}{N}
\end{aligned}$$

When $g \geq \min(b,w)/10$, we have $bw = \min(b,w) \cdot \max(b,w) \leq \min(b,w) \cdot n \leq 10bn$ and $N = ng + 2bw + gv/2 \leq gn + 20gn + gn/2 = 43gn/2$. Again let $z = u^2/n$. We

have

$$\begin{aligned}
&\mathbb{E}\left(\frac{\Delta(1/f)}{1/f} \mid I^{sc}\right)\\
=&\mathbb{E}\left(-\frac{\Delta f}{f}+\left(\frac{\Delta f}{f}\right)^2+O(n^{-3/2}) \mid I^{sc}\right)\\
=&-\frac{1+gu^2/N}{u^2+64n}+\frac{4u^2+1+2gu^2/N}{(u^2+64n)^2}+O(n^{-3/2})\\
=&\left(u^2+64n\right)^{-2}\cdot\left(-(1+gu^2/N)(u^2+64n)+4u^2+1+2gu^2/N\right)+O(n^{-3/2})\\
=&\left(u^2+64n\right)^{-2}\left(-u^2-64n-\frac{gu^2}{N}\left(u^2+64n\right)+4u^2+1+\frac{2gu^2}{N}\right)+O(n^{-3/2})\\
=&\left(u^2+64n\right)^{-2}\left(3u^2-64n+1+\left(-u^2-64n+2\right)\cdot\frac{gu^2}{N}\right)+O(n^{-3/2})\\
\le&\left(u^2+64n\right)^{-2}\left(3u^2-64n+1+\left(-u^2-64n+2\right)\cdot\frac{2u^2}{43n}\right)+O(n^{-3/2})\\
=&n^{-2}(z+64)^{-2}(3zn-64n+1+2z(-zn-64n+2)/43)+O(n^{-3/2})\\
=&n^{-2}(z+64)^{-2}((3z-64)n+1-2z(z+64)n/43+4z/43)+O(n^{-3/2})\\
=&\frac{1}{n}\cdot\frac{-2z^2/43+(3-128/43)z-64}{(z+64)^2}+\frac{1}{n^2}\cdot\frac{1+4z/43}{(z+64)^2}+O(n^{-3/2})
\end{aligned}$$

Note that function $\frac{-2x^2/43+x/43-64}{(x+64)^2}$ given $x \ge 0$ is at most $-22015/1893376 < -1/87$ and function $\frac{1+4x/43}{(x+64)^2}$ given $x \ge 0$ is at most $4/9159$. Thus from Lemma 2.3 we have the stochastic process $\{K_t\}$ given by

$$K_t = \frac{\exp(S_t^{sc}/(87n))}{u_t^2+64n}$$

is a supermartingale if we always have $g \ge \min(b,w)/10$. This gives us

$$\mathbb{E}\left(\exp(S_\tau^{sc}/(87n))\right) \le (n^2+64n)/(64n) = n/64+1$$

For Markov's inequality

$$\mathbb{P}\left(\exp(S_\tau^{sc}/(87n)) \geq n^c(n/64+1)\right) \leq n^{-c}$$

and

$$\mathbb{P}\left(S_\tau^{sc}/87 \geq n\log(n/64+1) + cn\log n\right) \leq n^{-c}$$

$$\mathbb{P}\left(S_\tau^{sc} \geq 87n\log\left(\frac{1}{64}n+1\right) + 87cn\log n\right) \leq n^{-c}$$

Since $S_\tau^{sc} = S_\tau^{g+} + S_\tau^{g-}$, $S_\tau^{g+} \leq S_\tau^{g-} + n$ and $S_\tau^{g-} \leq S_\tau^{g+} + n$, we have

$$\mathbb{P}\left(S_\tau^{g+} \geq \frac{87}{2}n\log\left(\frac{1}{64}n+1\right) + \frac{87}{2}cn\log n + \frac{1}{2}n\right) \leq n^{-c}$$

and

$$\mathbb{P}\left(S_\tau^{g-} \geq \frac{87}{2}n\log\left(\frac{1}{64}n+1\right) + \frac{87}{2}cn\log n + \frac{1}{2}n\right) \leq n^{-c}$$

which completes the proof. ■

The only case left is when the protocol enters the region $\{\min(b,w) = O(\log n) \bigwedge g = O(\log n)\}$. Recall that $p = \tilde{b} + \tilde{g}/2$. Once it enters this region, we will have $p = O(\log n/n)$ or $1-p = O(\log n/n)$.

Lemma 2.7 *If the process starts with $p = O(\log n/n)$ or $1-p = O(\log n/n)$, then with probability $1 - O\left(\frac{\log n}{\sqrt[3]{n}}\right)$ the population will reach consensus within $O(n)$ state-changing interactions.*

Proof The proof is completed by worst-case analyses. Without loss of generality, assume $1-p = O(\log n/n)$ and we will show with high probability p will converge

to 1 within $O(n)$ state-changing interactions. In this case, we have

$$\mathbb{P}\left(p_{t+1} = p_t + \frac{1}{2n} \mid I^{sc}\right) = \frac{p_t(1-\tilde{x}_t)}{p_t(1-\tilde{x}_t) + (1-p_t)(1-\tilde{y}_t)}$$

and

$$\mathbb{P}\left(p_{t+1} = p_t - \frac{1}{2n} \mid I^{sc}\right) = \frac{(1-p_t)(1-\tilde{y}_t)}{p_t(1-\tilde{x}_t) + (1-p_t)(1-\tilde{y}_t)}$$

Note that $x_t \leq p_t$. We have

$$\frac{\mathbb{P}\left(p_{t+1} = p_t + 1/2n \mid I^{sc}\right)}{\mathbb{P}\left(p_{t+1} = p_t - 1/2n \mid I^{sc}\right)} \geq \frac{p_t(1-p_t)}{1-p_t} = p_t$$

To provide an upper bound on the moves of p in the region $\{1 - 2\sqrt[3]{n}/(2n) \leq p \leq 1\}$, consider the following one-dimensional random walk on integers from 0 to $2\sqrt[3]{n}$. (Obviously each state i corresponds to the configuration $p = 1 - (2\sqrt[3]{n} - i)/(2n)$.) State 0 is a reflecting barrier always pushing the walk back to state 1. At any state $1 \leq i \leq 2\sqrt[3]{n} - 1$, the forward probability is $p/(p+1) = \frac{1-(2\sqrt[3]{n}-i)/(2n)}{2-(2\sqrt[3]{n}-i)/(2n)}$ and the backward probability is $1/(p+1) = \frac{1}{2-(2\sqrt[3]{n}-i)/(2n)}$. The walk starts at some state $k = 2\sqrt[3]{n} - O(\log n)$. Denote by t_i the number of steps needed to first hit state $2\sqrt[3]{n}$, starting at state i. And let h_i be the number times hitting state 0 before reaching state $2\sqrt[3]{n}$, starting at state i. Then the total number of state-changing interactions for the process starting from this region to entirely converge is at most $h_k \cdot O(n \log n) + t_k$.

Though this walk is already simple, we can further simplify it to the same walk with fixed forward probability $q_+ = \frac{1-n^{-2/3}}{2-n^{-2/3}}$ and backward probability $q_- = \frac{1}{2-n^{-2/3}}$, which also provides an upper bound, because in the region $\{1 - 2\sqrt[3]{n}/(2n) \leq p \leq 1\}$ we always have $p \geq 1 - n^{-2/3}$. We overload the notation t_i and h_i for this simpler walk. Denote by $\bar{t}_i = \mathbb{E}t_i$ and let $\Delta\bar{t}_i$ be the expected number of steps the walk takes

from state $i-1$ to state i. Then $\Delta\bar{t}_1 = 1$ due to the reflecting barrier at state 0. For $i \geq 2$, we have

$$\begin{aligned}\Delta\bar{t}_i =& 1 + q_+ \cdot 0 + q_- \cdot \mathbb{E}(\text{number of steps from state } i-2 \text{ to state } i)\\ =& 1 + q_-(\Delta\bar{t}_{i-1} + \Delta\bar{t}_i)\end{aligned}$$

which implies

$$\begin{aligned}\Delta\bar{t}_i =& \frac{1}{q_+} + \frac{q_-}{q_+}\Delta\bar{t}_{i-1}\\ =& \frac{1}{q_+} + \frac{q_-}{q_+}\left(\frac{1}{q_+} + \frac{q_-}{q_+}\Delta\bar{t}_{i-2}\right)\\ =& \frac{1}{q_+} + \frac{q_-}{q_+}\left(\frac{1}{q_+} + \frac{q_-}{q_+}\left(\frac{1}{q_+} + \frac{q_-}{q_+}\Delta\bar{t}_{i-3}\right)\right)\\ =& \ldots\\ =& \frac{1}{q_+} + \frac{q_-}{q_+^2} + \frac{q_-^2}{q_+^3} + \ldots + \frac{q_-^{i-2}}{q_+^{i-1}} + \frac{q_-^{i-1}}{q_+^{i-1}}\\ =& \frac{1}{q_+}\left(1 + \frac{q_-}{q_+} + \ldots + \frac{q_-^{i-2}}{q_+^{i-2}}\right) + \left(\frac{q_-}{q_+}\right)^{i-1}\end{aligned}$$

Note that for any $0 \leq i \leq 2\sqrt[3]{n}$, we have

$$\left(1 - n^{-\frac{2}{3}}\right)^i \geq \left(1 - n^{-\frac{2}{3}}\right)^{2\sqrt[3]{n}} = \left(\left(1 - n^{-\frac{2}{3}}\right)^{n^{\frac{2}{3}}}\right)^{2n^{-\frac{1}{3}}} = \exp\left(-2/\sqrt[3]{n}\right) \to 1$$

Thus all $\left(\frac{q_-}{q_+}\right)^i \to 1$ for large n. Then we have $\Delta\bar{t}_i = 2(i-1)+1 = 2i-1$. Hence, $\bar{t}_k = \sum_{i=k+1}^{\sqrt[3]{n}} \Delta\bar{t}_i = \Theta(\sqrt[3]{n}\log n)$. Markov's inequality gives

$$\mathbb{P}(t_k \geq n) \leq \frac{\bar{t}_k}{n} = \Theta\left(\frac{\log n}{n^{2/3}}\right)$$

Now if we can show with high probability $h_k = O(1)$ then we are done. But in fact we can do much better: with high probability $h_k = 0$. Denote by $\gamma_i = \mathbb{P}(h_i = 0)$. Then for $1 \leq i \leq 2\sqrt[3]{n} - 1$, $\gamma_i = q_+\gamma_{i+1} + q_-\gamma_{i-1}$. Define

$$\begin{aligned}\Delta\gamma_i =& \gamma_{i+1} - \gamma_i \\ =& q_+\gamma_{i+2} + q_-\gamma_i - \gamma_i \\ =& q_+(\gamma_{i+2} - \gamma_i) \\ =& q_+(\gamma_{i+2} - \gamma_{i+1} + \gamma_{i+1} - \gamma_i) \\ =& q_+\Delta\gamma_{i+1} + q_+\Delta\gamma_i\end{aligned}$$

which implies $\Delta\gamma_{i+1} = \frac{q_-}{q_+}\Delta\gamma_i$ and $\Delta\gamma_i = \left(\frac{q_-}{q_+}\right)^i \Delta\gamma_0$. Note that $\gamma_{2\sqrt[3]{n}} = 1$ and $\gamma_0 = 0$.

$$\sum_{i=0}^{2\sqrt[3]{n}-1} \Delta\gamma_i = \gamma_1 - \gamma_0 + \gamma_2 - \gamma_1 + \ldots + \gamma_{2\sqrt[3]{n}} - \gamma_{2\sqrt[3]{n}-1} = \gamma_{2\sqrt[3]{n}} - \gamma_0 = 1$$

Then $\sum_{i=0}^{2\sqrt[3]{n}-1} \Delta\gamma_i = 2\sqrt[3]{n}\Delta\gamma_0 = 1$ so $\Delta\gamma_0 = 1/(2\sqrt[3]{n})$. And we have

$$\gamma_k = \sum_{i=0}^{k-1} \Delta\gamma_i = \frac{k}{2\sqrt[3]{n}} = \frac{2\sqrt[3]{n} - O(\log n)}{2\sqrt[3]{n}} = 1 - O\left(\frac{\log n}{\sqrt[3]{n}}\right)$$

Thus with probability $1 - O\left(\frac{\log n}{\sqrt[3]{n}}\right)$, we have $h_k = 0$, which means with high probability, a population starting from region $\{\min(b, w) = O(\log n) \bigwedge g = O(\log n)\}$ will reach consensus after $O(n)$ state-changing interactions without leaving this region. ■

Combining all the lemmas we have so far yields Lemma 2.2. Note that the error bounds in these lemmas are all at most n^{-c} except in Lemma 2.7 where the error bound is $O\left(\frac{\log n}{\sqrt[3]{n}}\right)$, which dominates the other error terms (with a tiny increase in the constant) when c is large. Also note that the constants in the O's in Lemma

2.5 ("$\min(b, w) = O(\log n)$ and $g = O(\log n)$") and Lemma 2.7 ("$\min(p, 1-p) = O(\log n/n)$") can be chosen arbitrarily. We simply make them consistent, and choose a proper value to have the error bound $\frac{c \log n}{\sqrt[3]{n}}$ as claimed in Lemma 2.2. Eventually, we have Lemma 2.2. ■

2.2.3 Bounding $S_\tau^c = O(n \log n)$

In this section we bound the number of interactions S_τ^c in the central region where $\max(\tilde{b}, \tilde{g}, \tilde{w}) < 3/4$, using the total number of state-changing interactions S_τ^{sc}.

Lemma 2.8 *With probability* $1-o(1)$, $S_\tau^c = O(n \log n)$. *In addition, for any constant* $c > 0$, *we have*

$$\mathbb{P}(S_\tau^c \geq 9S_\tau^{sc} + cn \log n) \leq n^{-c}$$

Proof We show that the stochastic process $\{C_t\}$ given by

$$C_t = \exp((S_t^c - 9S_t^{sc})/n)$$

is a supermartingale. When $I_t^c = 0$, the value of C_t cannot increase so obviously $\mathbb{E}(C_t \mid \mathcal{F}_{t-1} \bigwedge I_t^c = 0) \leq C_{t-1}$. When $I_t^c = 1$, i.e., $\max(\tilde{b}, \tilde{g}, \tilde{w}) < 3/4$, at least two of $\tilde{w}, \tilde{b}$ and $\tilde{g}$ must be at least $1/8$:

- If $\tilde{b}$ and $\tilde{w}$ are both $\geq 1/8$, then $\mathbb{P}(I^{g+} = 1) = (2bw + gv/2)/n^2 \geq 1/8$, because function $2xy + (x+y)(1-x-y)/2$ given $1/8 \leq x, y \leq 3/4$ and $x + y \leq 1$ is at least $1/8$. Then the probability of the event that the current interaction increases S_t^c but not S_t^{g-} or S_t^{g+} and multiplies C_t by $\exp(1/n)$ is at most $7/8$. The probability of the event that the current interaction increases both S_t^c and S_t^{g+} but not S_t^{g-} and multiplies C_t by $\exp(-8/n)$ is at least $1/8$.

- If $\tilde{g} \geq 1/8$, we have $\mathbb{P}(I^{g-} = 1) = ng/n^2 \geq 1/8$. Then the probability of the event that the current interaction increases S_t^c but not S_t^{g-} or S_t^{g+} and multiplies C_t by $\exp(1/n)$ is at most $7/8$. The probability of the event that the current interaction increases both S_t^c and S_t^{g-} but not S_t^{g+} and multiplies C_t by $\exp(-8/n)$ is at least $1/8$.

This gives the bound

$$\begin{aligned}
\mathbb{E}(C_t \mid \mathcal{F}_{t-1} \bigwedge I_t^c) \leq & C_{t-1}\left(\frac{7}{8}\exp\left(\frac{1}{n}\right) + \frac{1}{8}\exp\left(-\frac{8}{n}\right)\right) \\
= & \left(\frac{7}{8}\left(1+\frac{1}{n}\right) + \frac{1}{8}\left(1-\frac{8}{n}\right) + O(n^{-2})\right) \\
= & C_{t-1}\left(1 - \frac{1}{8}n^{-1} + O(n^{-2})\right) \\
< & C_{t-1}
\end{aligned}$$

where the first equality is due to the Taylor expansion of the exponential function. Thus from Lemma 2.3 $\{C_t\}$ is a supermartingale and

$$\mathbb{P}(S_\tau^c \geq 9S_\tau^{sc} + cn\log n) \leq n^{-c}$$

which completes the proof. ■

2.2.4 Bounding $S_\tau^g = O(n\log n)$

In this section we bound the number of interactions S_τ^g in the corner region where $\tilde{g} \geq 3/4$, using the total number of g-decreasing interactions S_τ^{g-}.

Lemma 2.9 *With probability* $1 - o(1)$, $S_\tau^g = O(n\log n)$. *In addition, for any con-*

stant $c > 0$, we have

$$\mathbb{P}\left(S_\tau^g \geq 26S_\tau^{g^-} + 6cn\log n + 6n\log(2n+1) + \frac{45}{2}n\right) \leq n^{-c}$$

Proof In the large-g region, we choose the potential function $1/(2v+1)$ and let $f = 2v+1$. When $v = 0$, the whole population is at state g and the next interaction is surely (g, g). Then

$$\frac{\Delta(1/f)}{1/f} = 1 \cdot \left(\frac{1}{2\times 1+1} - \frac{1}{2\times 0+1}\right) = -\frac{2}{3}$$

When $v \geq 1$, we have expectation

$$
\begin{aligned}
&\mathbb{E}\left(\frac{\Delta(1/f)}{1/f}\right)\\
=&\mathbb{E}\left((2v+1)\left(I^{g-}\left(\frac{1}{2v+3}-\frac{1}{2v+1}\right)+I^{g+}\left(\frac{1}{2v-1}-\frac{1}{2v+1}\right)\right)\right)\\
=&\mathbb{E}\left(-\frac{2I^{g-}}{2v+3}+\frac{2I^{g+}}{2v-1}\right)\\
=&n^{-2}\left(-\frac{2ng}{2v+3}+\frac{2(2bw+gv/2)}{2v-1}\right)\\
=&(n^2(2v+3)(2v-1))^{-1}\cdot(-2ng(2v-1)+(4bw+gv)(2v+3))\\
=&(n^2(4v^2+4v-3))^{-1}\cdot(-4ngv+2ng+8bwv+12bw+2gv^2+3gv)\\
\leq&(n^2(4v^2+4v-3))^{-1}\cdot(-4ngv+2ng+8v\cdot v^2/4+12v^2/4+2gv^2+3gv)\\
=&(n^2(4v^2+4v-3))^{-1}\cdot(-4ngv+2ng+2v^3+3v^2+2gv^2+3gv)\\
=&\frac{-4n(n-v)v+2n(n-v)+2v^3+3v^2+2(n-v)v^2+3(n-v)v}{n^2(4v^2+4v-3)}\\
=&\frac{-4n^2v+4nv^2+2n^2-2nv+2v^3+3v^2+2nv^2-2v^3+3nv-3v^2}{n^2(4v^2+4v-3)}\\
=&\frac{-4n^2v+6nv^2+2n^2+nv}{n^2(4v^2+4v-3)}\\
=&\frac{-4nv+6v^2+2n+v}{n(4v^2+4v-3)}\\
=&\frac{(2-4v)n+6v^2+v}{n(4v^2+4v-3)}
\end{aligned}
$$

When $v \geq 1$, we have $2-4v<0$. Since $v=n-g\leq n/4$, we have $n\geq 4v$ and

$$
\mathbb{E}\left(\frac{\Delta(1/f)}{1/f}\right)\leq\frac{(2-4v)\cdot 4v+6v^2+v}{n(4v^2+4v-3)}\leq-\frac{1}{5}n^{-1}
$$

This is because function $(-10x^2+9x)/(4x^2+4x-3)$ given $x\geq 1$ is at most $-1/5$.

When the population is not in the large-g region, i.e., $\tilde{g} < 3/4$, we have

$$\frac{\Delta(1/f)}{1/f} = -\frac{2I^{g-}}{2v+3} + \frac{2I^{g+}}{2v-1} \leq -\frac{2I^{g-}}{2n+3} + \frac{2I^{g+}}{2n/4-1}$$

Hence, from Lemma 2.3 the stochastic process $\{G_t\}$ given by

$$G_t = \frac{\exp\left(\left(\frac{1}{6}S_t^g + \sum_{i=1}^t \left(\frac{2}{3}I_i^{g-} - 5I_i^{g+}\right)(1-I_i^g)\right)/n\right)}{2v_t+1}$$

is a supermartingale process. This gives us the bound

$$\mathbb{E}\left(\frac{\exp\left(\left(\frac{1}{6}S_\tau^g + \sum_{i=1}^\tau \left(\frac{2}{3}I_i^{g-} - 5I_i^{g+}\right)(1-I_i^g)\right)/n\right)}{2n+1}\right) \leq \mathbb{E}G_\tau \leq G_0 \leq 1$$

Again for Markov's inequality,

$$\mathbb{P}\left(\frac{1}{6}S_\tau^g + \sum_{i=1}^\tau \left(\frac{2}{3}I_i^{g-} - 5I_i^{g+}\right)(1-I_i^g) \geq cn\log n + n\log(2n+1)\right) \leq n^{-c}$$

Note that $\sum_{i=1}^t I_i^{g-}(1-I_i^g)$ is the number of I^{g-} interactions that occur in the region $\{g < 3n/4\}$ and $\sum_{i=1}^t I_i^{g+}(1-I_i^g)$ is the number of I^{g+} interactions that occur in the region $\{g < 3n/4\}$. If the process never leaves the region after entering it, we have $\sum_{i=1}^t I_i^{g+}(1-I_i^g) \leq \sum_{i=1}^t I_i^{g-}(1-I_i^g) + 3n/4$. If it passes the boundary of the region more than once, because every time that the process leaves the region it must have $g = 3n/4-1$ and every time that it returns to the region it must have $g = 3n/4-1$ too, we still have $\sum_{i=1}^t I_i^{g+}(1-I_i^g) \leq \sum_{i=1}^t I_i^{g-}(1-I_i^g) + 3n/4$. In addition, $\sum_{i=1}^t I_i^{g-}(1-I_i^g) \leq S_t^{g-}$ so we have

$$\mathbb{P}\left(\frac{1}{6}S_\tau^g + \sum_{i=1}^\tau \left(-\frac{13}{3}I_i^{g-}\right)(1-I_i^g) - \frac{15}{4}n \geq cn\log n + n\log(2n+1)\right) \leq n^{-c}$$

$$\mathbb{P}\left(\frac{1}{6}S_\tau^g - \frac{13}{3}S_\tau^{g-} \geq cn\log n + n\log(2n+1) + \frac{15}{4}n\right) \leq n^{-c}$$

and

$$\mathbb{P}\left(S_\tau^g \geq 26S_\tau^{g-} + 6cn\log n + 6n\log(2n+1) + \frac{45}{2}n\right) \leq n^{-c}$$

which completes the proof. ■

2.2.5 Bounding $S_\tau^b = O(n\log n)$ and $S_\tau^w = O(n\log n)$

We first bound the number of interactions S_τ^b in the corner region where $\tilde{b} \geq 3/4$. Then the upper bound for the number of interactions S_τ^w in the other corner region where $\tilde{w} \geq 3/4$ follows in a symmetric way.

Lemma 2.10 *With probability* $1 - o(1)$, $S_\tau^b = O(n\log n)$. *In addition, for any constant* $c > 0$, *we have*

$$\mathbb{P}\left(S_\tau^b \geq 153S_\tau^{g-} + 85S_\tau^{g+} + 17cn\log n + 17n\log(3n+1)\right) \leq n^{-c}$$

Proof In the large-b region, we choose the potential function $f = 3w + g + 1$. Table 2.3 lists the changes in f by different types of interactions. Suppose $3/4 \leq \tilde{b} < 1$ so $\max(g, w) \geq 1$. (The case when $\tilde{b} = 1$ is convergence.) Again we need to bound the

interaction	b	w	g	$3w+g+1$
$(b,+,w)$		-1	$+1$	-2
$(w,-,b)$	-1		$+1$	$+1$
$(b,+,g)$	$+1$		-1	-1
$(w,-,g)$		$+1$	-1	$+2$
$(g,+,g)$	$+1$		-1	-1
$(g,-,g)$		$+1$	-1	$+2$
$(g,+,w)$		-1	$+1$	-2
$(g,-,b)$	-1		$+1$	$+1$
Others				0

Table 2.3: Changes in $(3w+g+1)$

expectation $\mathbb{E}\left(\Delta f/f\right)$:

$$\begin{aligned}
\mathbb{E}\left(\frac{\Delta f}{f}\right) =&(n^2(3w+g+1))^{-1}\cdot\left(-2bw+bw-bg+2gw-\frac{1}{2}g^2+g^2-gw+\frac{1}{2}bg\right)\\
=&(n^2(3w+g+1))^{-1}\cdot\left(-bw-\frac{1}{2}bg+gw+\frac{1}{2}g^2\right)\\
=&(n^2(3w+g+1))^{-1}\cdot\left(-\frac{1}{2}g(2w+g)+\frac{1}{2}g(2w+g)\right)\\
\leq&-\frac{b}{2n^2}\cdot\frac{2w+g}{3w+g+1}+\frac{g(2w+g)}{2n^2(3w+g)}\\
=&\frac{1}{2n}\left(-\tilde{b}\cdot\frac{2w+g}{3w+g+1}+\frac{\tilde{g}(2\tilde{w}+\tilde{g})}{3\tilde{w}+\tilde{g}}\right)\\
\leq&\frac{1}{2n}\left(-\frac{3}{4}\cdot\frac{1}{2}+\frac{1}{4}\right)\\
=&-\frac{1}{16n}
\end{aligned}$$

where the last inequality comes from the facts that function $(2y+x)/(3y+x+1)$ given $x,y\geq 0$ and $\max(x,y)\geq 1$ is at least $1/2$, and that function $(x(2y+x))/(3y+x)$ given $x,y\geq 0$ and $x+y\leq 1/4$ is at most $1/4$.

When the population is not in the large-b region, i.e., $\tilde{b} < 3/4$, we have $w+g \geq n/4$ and

$$\frac{\Delta f}{f} \leq \frac{2I^{g-} + I^{g+}}{3w + g + 1} \leq \frac{2I^{g-} + I^{g+}}{n/4} = (8I^{g-} + 4I^{g+})n^{-1}$$

Hence, from Lemma 2.3 the stochastic process $\{B_t\}$ given by

$$B_t = (3w_t + g_t + 1) \cdot \exp\left(\left(\frac{1}{17}S_t^b - \sum_{i=1}^{t}\left(9I_i^{g-} + 5I_i^{g+}\right)(1 - I_i^b)\right)/n\right)$$

is a supermartingale. This gives us the bound

$$\mathbb{E}\left(\exp\left(\left(\frac{1}{17}S_\tau^b - \sum_{i=1}^{\tau}\left(9I_i^{g-} + 5I_i^{g+}\right)(1 - I_i^b)\right)/n\right)\right) \leq \mathbb{E}B_\tau \leq B_0 \leq 3n + 1$$

Again for Markov's inequality and $\sum_{i=1}^{t} I_i^{g-}(1-I_i^b) \leq S_t^{g-}$ and $\sum_{i=1}^{t} I_i^{g+}(1-I_i^b) \leq S_t^{g+}$, we have

$$\mathbb{P}\left(\frac{1}{17}S_\tau^b - 9S_\tau^{g-} - 5S_\tau^{g+} \geq cn\log n + n\log(3n+1)\right) \leq n^{-c}$$

and

$$\mathbb{P}\left(S_\tau^b \geq 153S_\tau^{g-} + 85S_\tau^{g+} + 17cn\log n + 17n\log(3n+1)\right) \leq n^{-c}$$

which completes the proof. ■

Then the number of interactions S_τ^w in the other region where $\tilde{w} \geq 3/4$ can be bounded in a symmetric way using the potential function $f = 3b + g + 1$.

Lemma 2.11 *With probability* $1 - o(1)$, $S_\tau^w = O(n\log n)$. *In addition, for any constant* $c > 0$, *we have*

$$\mathbb{P}\left(S_\tau^w \geq 153S_\tau^{g-} + 85S_\tau^{g+} + 17cn\log n + 17n\log(3n+1)\right) \leq n^{-c}$$

Finally, combining all the above lemmas implies a bound on $\tau = S_\tau^c + S_\tau^b + S_\tau^g + S_\tau^w$ that

$$\mathbb{P}\left(\tau \geq 96930(c+1)n\log n\right) \leq \max\left(9n^{-c}, \frac{c\log n}{\sqrt[3]{n}}\right)$$

As we explained in Section 2.1, let $\tau = \min(\tau_*, 10^5(c+1)n\log n)$. This makes τ a well-defined stopping time and eventually, we have Theorem 2.1. Again the $O\left(\frac{\log n}{\sqrt[3]{n}}\right)$ error term is because all the lemmas give error bounds at most $O(n^{-c})$ except Lemma 2.7, which gives an $O\left(\frac{\log n}{\sqrt[3]{n}}\right)$ error bound and dominates the other error terms (with a tiny increase in the constant) when c is large. ■

2.3 Binary Signaling Consensus with $r > 2$

In the previous section we have provided a comprehensive study of the binary signaling consensus process with $r = 2$. This is a reasonable start to study binary signaling consensus, as it is common to assume a large population n with a relatively small resistance. However, to understand the model in depth, we have to investigate the general case with larger r.

In this section we allow resistance r to be arbitrarily large, i.e., not necessarily a small constant. Denote by n_i the population of confidence level i and by $x_i = n_i/n$ the corresponding proportion. Any configuration over the society can be represented as a $(r+1)$-dimensional vector $\vec{x} \in [0,1]^{r+1}$ where $\sum_{i=0}^r x_i = 1$. Denote by $p = \sum_{i=0}^r (i/r)x_i$. We say an interaction is a *positive interaction* if the initiator sends a positive bit, and is a *negative interaction* otherwise. Then p is the probability of occurrence of a positive interaction. The curve of p serves as a significant indicator of the underlying status of the society. A large p implies the grammar is almost

accepted and $p = 1$ is equivalent to positive convergence. A small p indicates the grammar is close to extinction and $p = 0$ is equivalent to negative convergence. If we expect a positive convergence, then a positive interaction is never harmful while a negative interaction never helps, and vice versa for negative convergence.

Unfortunately, rigorous and comprehensive analysis of large-r case turns out to be rather difficult. This is not surprising given that even the proof for the three-state model is already very lengthy. The increase of degrees of freedom with large r leads to high dimensionality of the configuration space and makes the process more unpredictable. One path of p could correspond to a large number of possible hidden configuration sequences, which does not permit us to generalize the potential functions in Section 2.2 to large-r case. In addition, the fact that the corresponding systems of differential equations do not have closed-form solutions (even for the $r = 2$ case) rules out arguments based on techniques involving reduction to a continuous process in the limit. In fact we will see later an essential difference between the $r = 2$ case and the $r > 2$ case. In the $r = 2$ case p is always increasing or always decreasing in the limit, but the curve of p in the $r > 2$ case doesn't have this nice property and is more unpredictable. This intrinsic difference is one indication of that we should expect more difficulties in analyzing the large-r case. However, this doesn't prevent us from pursuing theoretical results for the general model.

2.3.1 Continuous-time binary signaling consensus

When the gap between the discrete time steps in the model goes to zero in the limit, the communication process becomes continuous-time. To study this continuous process, we use the asynchronous timing defined by Boyd et al. [BGPS06]. Each agent in the society has a clock which ticks at the times of a Poisson process of rate r. The inter-tick times at each agent are exponentials of rate r, independent across agents

and over time. Equivalently, this corresponds to a single clock ticking according to a Poisson process of rate nr at time t_k, $k \geq 1$, where $\{t_{k+1} - t_k\}$ are i.i.d. exponentials of rate nr. At time t_k, an edge (i, j) is chosen uniformly at random from E and the two chosen agents interact as defined in the model.

Note that the continuous process can be arbitrarily close to but never reaches complete consensus where $p = 0$ or 1. A direct reason is that the derivative of p goes to 0 as the process approaches to convergence. Therefore, instead of entire convergence, we redefine *consensus* for the continuous process to be the region where $\min(p, 1 - p) = O(1/(nr))$, which is the closest point the process can achieve to complete convergence. We say a configuration is *monotone* if it has $x_0 \leq x_1 \leq \ldots \leq x_r$ with at least one $<$ in the middle, or $x_0 \geq x_1 \geq \ldots \geq x_r$ with at least one $>$ in the middle. The set of all monotone configuration is called the monotone region. In this subsection we will show the fast convergence to consensus of the continuous process inside the monotone region.

Theorem 2.2 *If the initial configuration is monotone, then the continuous process will reach consensus within $O(r \log nr)$ time.*

The proof starts with derivation of the corresponding ODE system of the process, which can be inferred by taking the limit of the expectation of the configuration vector. This ODE system provides a mathematical formula of the vector field in the configuration space. We show that the vector field anywhere at the boundary of the monotone region always points inwards into the monotone region, which means the process stays in the monotone region and never leaves. We divide the monotone region into two sub-areas A_+, the region where $x_0 \leq x_1 \leq \ldots \leq x_r$ with at least one $<$ in the middle, and A_-, the region where $x_0 \geq x_1 \geq \ldots \geq x_r$ with at least one $>$ in the middle. The ODE system also gives us the differential equation for p, from

which we prove that p is always increasing in A_+ and is always decreasing in A_-. It suffices to show the convergence bound for A_+, as it holds for A_- symmetrically.

The above two facts already tell us that once the process enters A_+, p will keep increasing until convergence. What we need is a positive lower bound for the derivative of p that will lead to logarithmic convergence time. We need to take care of two cases where $\mathrm{d}p/\mathrm{d}t$ is very small. The first case is when the process is almost at convergence and p is very close to 1. The other is when the configuration vector is almost uniform and p is very close to $1/2$. To do so, we divide the path of p from $1/2 + 1/(nr)$ to $1 - 1/(nr)$ into two corresponding stages: from $\frac{2}{3}$ to $1 - \frac{1}{nr}$ and from $\frac{1}{2} + \frac{1}{nr}$ to $\frac{2}{3}$. We show the time for the former stage is $O(\log nr)$ and the time for the latter is $O(r \log nr)$.

Lemma 2.12 *Once the process enters the monotone region, it never leaves.*

Proof The corresponding systems of differential equations of the process can be inferred by taking the limit of the expectation of the configuration vector. For the change of $\{x_i\}$ and p from time tick t_k to the next time tick t_{k+1}, we have

$$\begin{cases} x_0(t_{k+1}) = x_0(t_k) + (1 - p(t_k)) \cdot x_1(t_k) \cdot \frac{1}{n} - p(t_k) \cdot x_0(t_k) \cdot \frac{1}{n} \\ x_i(t_{k+1}) = x_i(t_k) + p(t_k) \cdot x_{i-1}(t_k) \cdot \frac{1}{n} + (1 - p(t_k)) \cdot x_{i+1}(t_k) \cdot \frac{1}{n} - x_i(t_k) \cdot \frac{1}{n} \\ x_r(t_{k+1}) = x_r(t_k) - (1 - p(t_k)) \cdot x_r(t_k) \cdot \frac{1}{n} + p(t_k) \cdot x_{r-1}(t_k) \cdot \frac{1}{n} \end{cases}$$

where the second equation is for $1 \leq i \leq r - 1$. Dividing both sides of the equations

by the infinitesimal $1/(nr)$ gives

$$\begin{cases} \dfrac{x_0(t_{k+1}) - x_0(t_k)}{1/(nr)} = r \cdot ((1 - p(t_k)) \cdot x_1(t_k) - p(t_k) \cdot x_0(t_k)) \\ \dfrac{x_i(t_{k+1}) - x_i(t_k)}{1/(nr)} = r \cdot (p(t_k) \cdot x_{i-1}(t_k) + (1 - p(t_k)) \cdot x_{i+1}(t_k) - x_i(t_k)) \\ \dfrac{x_r(t_{k+1}) - x_r(t_k)}{1/(nr)} = r \cdot (p(t_k) \cdot x_{r-1}(t_k) - (1 - p(t_k)) \cdot x_r(t_k)) \end{cases}$$

Since nr is the rate of the Poisson process and we let $1/(nr)$ go to 0, we have the ODE system

$$\begin{cases} \dfrac{\mathrm{d}x_0}{\mathrm{d}t} = r \cdot ((1 - p) \cdot x_1 - p \cdot x_0) \\ \dfrac{\mathrm{d}x_i}{\mathrm{d}t} = r \cdot (p \cdot x_{i-1} + (1 - p) \cdot x_{i+1} - x_i) \\ \dfrac{\mathrm{d}x_r}{\mathrm{d}t} = r \cdot (p \cdot x_{r-1} - (1 - p) \cdot x_r) \end{cases}$$

This ODE system provides a mathematical formula for the vector field in the configuration space. To complete the proof, we need to show that the vector field anywhere at the boundary of the monotone region always points inwards into the monotone region. Let A_+ be the region where $x_0 \leq x_1 \leq \ldots \leq x_r$ with at least one $<$ in the middle and A_- be the region where $x_0 \geq x_1 \geq \ldots \geq x_r$ with at least one $>$ in the middle. It is sufficient to prove the lemma for A_+ and the proof for A_- will hold in a symmetric way.

For some $1 \leq i \leq r - 2$, when the process is close to a point where $x_{i+1} - x_i = 0$,

given that the process is in region A_+, we have

$$\begin{aligned}\frac{\mathrm{d}(x_{i+1}-x_i)}{\mathrm{d}t} =& r\cdot(px_i+(1-p)x_{i+2}-x_{i+1}-px_{i-1}-(1-p)x_{i+1}+x_i)\\ =& r\cdot(p(x_i-x_{i-1})+(1-p)(x_{i+2}-x_{i+1}))>0\end{aligned}$$

which pushes the system back to the area with $x_{i+1}-x_i>0$.

Notice that the probability of positive interaction $p=\sum_{i=0}^{r}(i/r)x_i$ is always greater than $1/2$ in region A_+. When the process is close to a point where $x_1-x_0=0$ or $x_r-x_{r-1}=0$, we have

$$\begin{aligned}\frac{\mathrm{d}(x_1-x_0)}{\mathrm{d}t} =& r\cdot(px_0+(1-p)x_2-x_1-(1-p)x_1+px_0)\\ =& r\cdot(2px_0-x_1+(1-p)(x_2-x_1))>0\end{aligned}$$

and

$$\begin{aligned}\frac{\mathrm{d}(x_r-x_{r-})}{\mathrm{d}t} =& r\cdot(px_{r-1}-(1-p)x_r-px_{r-2}-(1-p)x_r+x_{r-1})\\ =& r\cdot(p(x_{r-1}-x_{r-2})-2(1-p)x_r+x_{r-1})>0\end{aligned}$$

which pushes the system back to the area with $x_1-x_0>0$ and $x_r-x_{r-1}>0$ respectively. Therefore, the continuous process will never escape from region A_+ once it is inside. We can similarly show symmetric results in the other region A_-. ■

Lemma 2.13 *The derivative of p is always positive in region A_+ and always negative in region A_-.*

Proof From the above ODE system and $p=\sum_{i=0}^{r}(i/r)x_i$ and $\sum_{i=0}^{r}x_i=1$, we can

infer the differential equation for p, which turns out to be very neat.

$$\frac{\mathrm{d}p}{\mathrm{d}t} = (1 - x_r)p - (1 - x_0)(1 - p)$$

We can interpret the differential equation in a very simple way. With probability p a positive interaction occurs. This is a stay-put interaction if and only if the responder is already fully confident. Thus with conditional probability $(1 - x_r)$ it is a state-changing interaction and increases p by $1/(nr)$. Likewise, with probability $(1 - x_0)(1 - p)$, we will have a negative interaction that decreases p by $1/(nr)$.

Again, it suffices to prove the statement for region A_+. We have already shown the process stays in A_+ once it is inside. Denote by $B = \sum_{i=1}^{r-1}(i/r)x_i$, which is the contribution of x_1 to $x_r - 1$ to the probability p. Then $p = B + x_r$ and

$$\begin{aligned}\frac{\mathrm{d}p}{\mathrm{d}t} =& (1 - x_r)p - (1 - x_0)(1 - p)\\ =& (B + x_r)(1 - x_r) - (1 - B - x_r)(1 - x_0)\\ =& (2 - x_r - x_0)B + (1 - x_r)(x_r - 1 + x_0)\end{aligned}$$

Since $2 - x_r - x_0$ and $B \geq \sum_{i=1}^{r-1}(i/r) \cdot (1 - x_0 - x_r)/(r - 1) = (1 - x_0 - x_r)/2$ in region A_+, we have

$$\frac{\mathrm{d}p}{\mathrm{d}t} \geq (2 - x_r - x_0) \cdot \frac{1}{2}(1 - x_0 - x_r) + (1 - x_r)(x_r - 1 + x_0)$$

Region A_+ also gives $1 = \sum_{i=0}^{r} x_i \geq x_0 + (r - 1)x_0 + x_r$ and $0 \leq x_0 \leq (1 - x_r)/r$. Note that function $(2 - x - y)(1 - x - y)/2 + (1 - x)(x + y - 1)$ for $x \geq y$, $x + y < 1$ and $0 \leq y \leq (1 - x)/2$ is always non-negative. The only case where this function is 0 is when $x = y$ but we always have $x_0 < x_r$ in A_+. For any $r \geq 2$,

$x_0 \leq (1 - x_r)/r \leq (1 - x_r)/2$. Thus, we always have $\mathrm{d}p/\mathrm{d}t > 0$ inside region A_+. In a symmetric way, we know $\mathrm{d}p/\mathrm{d}t < 0$ inside region A_-. ■

Proof (of Theorem 2.2) Again without loss of generality, we only study the region A_+. We know the probability of positive interaction p is always greater than $1/2$ inside A_+ and we expect a positive convergence $p \to 1$. The above two lemmas already tell us that once the process enters A_+, p will keep increasing until convergence. What we need is a positive lower bound for $\mathrm{d}p/\mathrm{d}t$ that will lead to the desired convergence time. Let $\varepsilon = p - 1/2$ and $\delta = 1 - p$. There are two cases where $\mathrm{d}p/\mathrm{d}t$ is very small.

1. The process is almost at convergence and p is very close to 1 with a very small δ;

2. The configuration vector is almost uniform and p is very close to $1/2$ with a very small ε.

To do so, we divide the path of p from $1/2 + 1/(nr)$ to $1 - 1/(nr)$ into two corresponding stages:

1. p goes from $\frac{2}{3}$ to $1 - \frac{1}{nr}$;

2. p goes from $\frac{1}{2} + \frac{1}{nr}$ to $\frac{2}{3}$.

For Stage 1, we have $2/3 \leq p = \sum_{i=0}^{r}(i/r)x_i \leq \sum_{i=0}^{r}(i/r)x_r = (1 + r)x_r/2$ and $x_r \geq 4/(3(r + 1))$. Note that function $(2 - x - y)(1 - x - y)/2 + (1 - x)(x + y - 1)$ for $x \geq y$, $x + y < 1$, $0 \leq y \leq (1 - x)/r$ and $4/(3(r + 1)) \leq x \leq 1 - \delta$ is minimized

at $(y = \delta/r, x = 1 - \delta)$. Thus

$$\begin{aligned}
\frac{\mathrm{d}p}{\mathrm{d}t} \geq & (2 - x_r - x_0) \cdot \frac{1}{2}(1 - x_0 - x_r) + (1 - x_r)(x_r - 1 + x_0) \\
\geq & \left(2 - 1 + \delta - \frac{\delta}{r}\right) \cdot \frac{1}{2}\left(1 - 1 + \delta - \frac{\delta}{r}\right) + \delta\left(1 - \delta - 1 + \frac{\delta}{r}\right) \\
= & \left(1 + \left(1 - \frac{1}{r}\right)\delta\right) \cdot \frac{1}{2}\left(1 - \frac{1}{r}\right)\delta - \delta\left(1 - \frac{1}{r}\right)\delta \\
= & \frac{1}{2}\left(1 - \frac{1}{r}\right)\delta \cdot \left(1 + \left(1 - \frac{1}{r}\right)\delta - 2\delta\right) \\
= & \frac{1}{2}\left(1 - \frac{1}{r}\right)\delta \cdot \left(1 - \left(1 + \frac{1}{r}\right)\delta\right)
\end{aligned}$$

As $\delta < \frac{1}{2}$, we have

$$\begin{aligned}
\frac{\mathrm{d}p}{\mathrm{d}t} > & \frac{1}{2}\left(1 - \frac{1}{r}\right)\delta \cdot \left(1 - \left(1 + \frac{1}{r}\right)\frac{1}{2}\right) \\
= & \frac{1}{2}\left(1 - \frac{1}{r}\right)\delta \cdot \left(1 - \frac{1}{2} - \frac{1}{2r}\right) \\
= & \left[\frac{1}{2}\left(1 - \frac{1}{r}\right)\right]^2 \cdot \delta
\end{aligned}$$

We let $c = \left[\frac{1}{2}\left(1 - \frac{1}{r}\right)\right]^2 > 0$, which doesn't change with time. Now the ODE becomes simply $\mathrm{d}p/\mathrm{d}t > c(1-p)$ which is easy to solve. Let $p(t_1) = 1/2 + 1/(nr)$, $p(t_2) = 2/3$ and $p(t_3) = 1 - 1/(nr)$. We have

$$c(t_3 - t_2) < -\log(1 - p(t_3)) + \log(1 - p(t_2)) = \log nr - \log 3$$

Hence, $t_3 - t_2 < (\log nr - \log 3)/c = O(\log nr)$ time since $1/16 \leq c < 1/4$ for $r \geq 2$.

For Stage 2 from t_1 to t_2, we have $\frac{1}{2} + \varepsilon \leq (1 + r)x_r/2$ and $x_r \geq (1 + 2\varepsilon)/(r + 1)$. Note that function $(2 - x - y)(1 - x - y)/2 + (1 - x)(x + y - 1)$ for $x \geq y$, $x + y < 1$, $0 \leq y \leq (1 - x)/r$ and $(1 + 2\varepsilon)/(r + 1) \leq x \leq 1 - \delta$ is minimized at $(y = (1 - x)/r, x =$

$(1+2\varepsilon)/(r+1)$). Thus letting z be $(1+2\varepsilon)/(r+1)$,

$$\begin{aligned}
\frac{\mathrm{d}p}{\mathrm{d}t} \geq & (2-x_r-x_0)\cdot\frac{1}{2}(1-x_0-x_r)+(1-x_r)(x_r-1+x_0) \\
\geq & \left(2-z-\frac{1-z}{r}\right)\cdot\frac{1}{2}\left(1-z-\frac{1-z}{r}\right)+(1-z)\left(z-1+\frac{1-z}{r}\right) \\
= & \frac{1-z}{2}\left(2-z-\frac{1-z}{r}\right)\left(1-\frac{1}{r}\right)+2\left(z-1+\frac{1-z}{r}\right) \\
= & \frac{1-z}{2r^2}\left(2r-rz-1+z\right)(r-1)+2r\left(rz-r+1-z\right) \\
= & \frac{(1-z)(r-1)}{2r^2}\left[2r-1-(r-1)z-2r\left(1-z\right)\right] \\
= & \frac{(1-z)(r-1)}{2r^2}\left[(r+1)z-1\right] \\
= & \frac{r-1}{2r^2}\cdot 2\varepsilon\cdot\frac{r-2\varepsilon}{r+1} \\
= & \frac{(r-1)(r-2\varepsilon)\varepsilon}{(r+1)r^2}
\end{aligned}$$

As $\varepsilon<\frac{1}{2}$, we have

$$\frac{\mathrm{d}p}{\mathrm{d}t}\geq\frac{(r-1)(r-2\varepsilon)\varepsilon}{(r+1)r^2}>\frac{(r-1)^2}{(r+1)r^2}\cdot\varepsilon$$

Again we let $g=(r-1)^2/((r+1)r^2)>0$, which doesn't change with time. Solving the simple ODE $\mathrm{d}p/\mathrm{d}t>c(p-1/2)$ gives

$$g(t_2-t_1)<\log(2p(t_2)-1)-\log(2p(t_1)-1)=\log\frac{1}{3}-\log\frac{2}{nr}$$

and $t_2-t_1<(\log nr-\log 6)/g=O(r\log(nr))$ time. Thus the total time from $p=1/2+1/(nr)$ to $p=1-1/(nr)$ is $t_3-t_1=(t_3-t_2)+(t_2-t_1)=O(r\log nr)$. The same statement can be proved for the other region A_- in a symmetric way. ■

We have bounded the convergence time for the monotone region. To achieve

a complete bound for the whole configuration space, we need either a convergence bound for the non-monotone region separately if the process can stay in the non-monotone region, or to bound the time until the process enters the monotone region and show this always happens. Empirical results presented in Section 2.4 suggest that the process will eventually enter the monotone region regardless of the initial configuration and that the time needed for this to happen is short (see Conjecture 2.4), which indicates bounding the convergence time in the monotone region will be essential to the general bound for the entire configuration space. This is why the monotone case is interesting to us.

When the resistance $r = 2$ or $r = O(1)$, the convergence time is $O(\log n)$ and the rate of the clock is $O(n)$, so the total number of ticks of the clock is $\Theta(n \log n)$, which matches our result for three-state binary signaling consensus in Section 2.2.

The analysis of the continuous process above gives us the following lemma.

Lemma 2.14 *When $r = 2$, p is always increasing when $p > 1/2$ and is always decreasing when $p < 1/2$. This doesn't hold for any $r > 2$.*

Proof When $r = 2$, we have $p = x_2 + (1 - x_2 - x_0)/2 = (1 + x_2 - x_0)/2$ and

$$\begin{aligned}
\frac{\mathrm{d}p}{\mathrm{d}t} =& (1 - x_2)p - (1 - x_0)(1 - p) \\
=& (1 - x_2) \cdot \frac{1 + x_2 - x_0}{2} - (1 - x_0) \cdot \left(1 - \frac{1 + x_2 - x_0}{2}\right) \\
=& \frac{1}{2}\left((1 - x_2)(1 + x_2) - (1 - x_2)x_0 - (1 - x_0)(1 + x_0) + (1 - x_0)x_2\right) \\
=& \frac{1}{2}(1 - x_2^2 - x_0 + x_0x_2 - 1 + x_0^2 + x_2 - x_0x_2) \\
=& \frac{1}{2}((x_2 - x_2^2) - (x_0 - x_0^2))
\end{aligned}$$

Note that $p > 1/2$ is equivalent to $x_0 < x_2$. Since $x_0 + x_2 \leq 1$ and $0 \leq x_0 < x_2$, we have either $0 \leq x_0 < x_2 < 1/2$ or $0 \leq x_0 < 1/2 \leq x_2 < 1$. In the former case we

have $(x_2 - x_2^2) - (x_0 - x_0^2) > 0$ and $\mathrm{d}p/\mathrm{d}t > 0$. In the latter case we have $x_0 \leq 1 - x_2$ and because function $x - x^2$ is symmetric with respect to line $x = 1/2$, we also have $\mathrm{d}p/\mathrm{d}t > 0$. Likewise we have $\mathrm{d}p/\mathrm{d}t < 0$ when $p < 1/2$ when $r = 2$.

When $r \geq 3$, we have $\mathrm{d}p/\mathrm{d}t = (1 - x_r)p - (1 - x_0)(1 - p)$. We consider a case in which $x_0 = 0$ and $x_1 + x_r = 1$. Then we have $p = x_r + (1 - x_r)/r$ and this becomes

$$\begin{aligned}
\frac{\mathrm{d}p}{\mathrm{d}t} =& (1 - x_r)p - (1 - x_0)(1 - p) \\
=& (1 - x_r)p - (1 - p) \\
=& (2 - x_r)p - 1 \\
=& (2 - x_r) \cdot \left(x_r + \frac{1 - x_r}{r} \right) - 1 \\
=& \frac{1}{r} \cdot ((2 - x_r)((r - 1)x_r + 1) - r) \\
=& \frac{1}{r} \cdot ((2r - 3)x_r - (r - 1)x_r^2 + 2 - r)
\end{aligned}$$

Thus $\mathrm{d}p/\mathrm{d}t$ is negative when $x_r < (r - 2)/(r - 1)$. When $r \geq 3$ we know $(r - 2)/(r - 1) \geq 1/2$. We now let $x_r = (r - 2)/(r - 1) - 1/5$ where $\mathrm{d}p/\mathrm{d}t$ is surely negative. Then

$$\begin{aligned}
p =& x_r + \frac{1 - x_r}{r} = \frac{1}{r} + \left(1 - \frac{1}{r} \right) x_r \\
=& \frac{1}{r} + \frac{r - 1}{r} \cdot \left(\frac{r - 2}{r - 1} - \frac{1}{5} \right) \\
=& \frac{1}{r} + 1 - \frac{2}{r} - \frac{1}{5} \left(1 - \frac{1}{r} \right) \\
=& \frac{4}{5} \cdot \left(1 - \frac{1}{r} \right) \geq \frac{4}{5} \cdot \left(1 - \frac{1}{3} \right) = \frac{8}{15} > \frac{1}{2}
\end{aligned}$$

which disproves the statement for any $r > 2$. ■

Therefore, when $r = 2$ the probability of positive interaction p is always pushed towards convergence in the correct direction, but in the $r > 2$ case the change of p is

more unpredictable. This shows an intrinsic difference between the $r = 2$ case and the $r > 2$ case.

2.3.2 A convergence lower bound

Although convergence upper bounds are our primary interest in the population protocol for binary signaling consensus, in this subsection we study the general protocol in another direction and prove a convergence lower bound on the number of interactions. Recall that for the three-state population protocol, the convergence lower bound $\Omega(n \log n)$ is an immediate result from the well-known coupon collector's bound, because when the initial configuration is $cl(i) = 1$ for all $i \in V$, every agent must participate in at least one interaction in order to achieve consensus. Likewise, to bound the number of interactions for the $r > 2$ case, we consider a generalized version of the coupon collector problem. An *r-coupon collector* is where instead of collecting at least one copy for each type of coupon, we need to keep drawing coupons until we have collected at least r copies for each type of coupon. The number of steps an $(r/2)$-coupon collector takes gives a convergence lower bound for the general binary signaling consensus process, as every agent must participate in at least $r/2$ interactions before convergence, when the initial configuration of the population is $x_{r/2}$=1 and $x_i = 0$ for all $i \neq r/2$. Since we are only interested in the magnitude, we will consider an r-coupon collector instead of an $(r/2)$-coupon collector, for algebraic convenience.

Another important reason for us to study the r-coupon collector problem here is the inspiration from the three-state population protocol that the convergence bound of the binary signaling consensus process is exactly the tight bound of coupon collector. This fact leads to our conjecture that this connection also holds for $r > 2$ (see Conjecture 2.2 in Section 2.4).

We note that the r-coupon collector problem defined above is also known as the *double dixie cup problem* in the literature. Most works in the literature are concerning the asymptotic formula in expectation and the limit distribution of the answer to this problem with $r = O(1) < +\infty$ [NS60, ER61a, Fla82]. To the best of our knowledge, there exists no direct result we can use here for the general case with $r = \omega(1)$. Although getting the asymptotic formula for the general case is rather difficult, we are only interested in the magnitude order here, which is proved in the following theorem.

Theorem 2.3 *An r-coupon collector needs $\Theta(nr + n\log n)$ steps with high probability.*

To prove this bound, we consider the equivalent balls-in-bins problem: if we keep throwing balls uniformly at random into n bins, how many balls do we need to throw such that every bin has at least r balls with high probability? Let N be the answer to this question. The proof is easy for $r = O(1)$, by doing at most r rounds of classic coupon collector to fill the bins. For $r = \omega(1)$, the proof is done by using Poisson approximation. Let Y be the minimum load among the n bins, which is the minimum among n i.i.d. Poisson random variables with mean N/n in Poisson approximation. We show case by case, depending on the magnitude of r, that we can always find an $N_0 = \Theta(nr + n\log n)$ such that $\mathbb{P}(Y < r)$ goes to zero after throwing N_0 balls. Because $\mathbb{P}(Y < r)$ is monotonically decreasing in N, all $N \geq N_0$ have $\mathbb{P}(Y < r) \to 0$. Therefore, we have $N \leq N_0 = O(nr + n\log n)$, which completes the proof.

Proof When $r = O(1)$ is a constant, we have $N = \Omega(n\log n)$ from the classic coupon collector's bound. We also have $N = O(n\log n)$ because at most r rounds of coupon collector are enough to fill the bins. Thus $N = \Theta(n\log n) = \Theta(nr+n\log n)$ is a tight bound for $r = O(1)$. We refer to related works on the equivalent double dixie

cup problem for concrete asymptotic formula in expectation [NS60, ER61a, Fla82].

When $r = \omega(1)$, the lower bound $N = \Omega(nr + n \log n)$ is also easy to see. We must throw at least nr balls to fill the bins and the addend $\Omega(n \log n)$ is again from classic coupon collector. To prove the upper bound $N = O(nr + n \log n)$, by using Poisson approximation, we know the joint distribution of the number of balls in all the bins is well approximated by assuming the load at each bin is an independent Poisson random variable with mean $\lambda = N/n$ after we have thrown N balls in total. More concretely, if the probability of an event is either monotonically increasing or monotonically decreasing in the number of balls, then if this event has probability q in Poisson approximation, it has probability at most $2q$ in the exact balls-in-bins case [MU05, p. 103]. As Y is the minimum load among the n bins, the probability of $Y < r$ is monotonically decreasing in the number of balls and satisfies the condition of Poisson approximation. If $\mathbb{P}(Y < r) \to 0$ holds in Poisson approximation, $\mathbb{P}(Y < r)$ also goes to zero in the exact balls-in-bins case (or equivalently the r-coupon collector problem).

In Poisson approximation, Y is the minimum among n i.i.d. Poisson random variables with mean N/n. We have

$$\mathbb{P}(Y < r) = \mathbb{P}(Y \leq r - 1) = 1 - \left(1 - \frac{\Gamma(r, \lambda)}{(r-1)!}\right)^n$$

where $\lambda = N/n$ and $\Gamma(\cdot, \cdot)$ is the incomplete Gamma function. An asymptotic representation for $\Gamma(\cdot, \cdot)$ is $\Gamma(r, \lambda) = \lambda^{r-1} e^{-\lambda} + o(1)$ when $\lambda \to +\infty$. When $N = \Omega(nr + n \log n)$, $\lambda = N/n = \Omega(r + \log n) = \omega(1)$ and this asymptotic representation

is applicable. Letting $s = r - 1$, we have

$$\begin{aligned}\frac{s!}{n \cdot \Gamma(s+1,\lambda)} =& O(1) \cdot \frac{\sqrt{s} \cdot s^s}{n \cdot e^s \lambda^s e^{-\lambda}} \\ =& O(1) \cdot \exp\left(\frac{1}{2}\log s + s\log s + \frac{N}{n} - s\log\frac{N}{n} - s - \log n\right) \\ =& O(1) \cdot \exp\left(s\left(\frac{N}{ns} - \log\frac{N}{ns} - 1 - \frac{1}{s}\log\frac{n}{\sqrt{s}}\right)\right)\end{aligned}$$

Denote by f the exponent in this expression. The sign and magnitude of f are crucial for the convergence bound. When $f \to -\infty$, we have $s!/\Gamma(s+1,\lambda) = o(n)$ and $\mathbb{P}(Y \geq r) \to 0$; When $f = O(1)$ is a constant, we have $s!/\Gamma(s+1,\lambda) = \Theta(n)$ and $\mathbb{P}(Y \geq r)$ is a constant between 0 and 1; When $f \to +\infty$, we have $s!/\Gamma(s+1,\lambda) = \omega(n)$ and $\mathbb{P}(Y < r) \to 0$.

When $r = o(\log n)$, we have $\Theta(nr + n\log n) = \Theta(n\log n)$. Choose $N_1 = 2n\log n$ and then

$$\begin{aligned}&\frac{N_1}{ns} - \log\frac{N_1}{ns} - 1 - \frac{1}{s}\log\frac{n}{\sqrt{s}} \\ =& 2 \cdot \frac{\log n}{s} - \log\frac{2\log n}{s} - 1 - \frac{1}{s}\log\frac{n}{\sqrt{s}} \\ >& \frac{\log n}{s} - \frac{1}{s}\log\frac{n}{\sqrt{s}} = \frac{1}{2s}\log s\end{aligned}$$

Since $r = \omega(1)$, s is also $\omega(1)$. Thus $f > s \cdot \log s/(2s) = \log s/2 = \omega(1)$ and $\mathbb{P}(Y < r) \to 0$. Because $\mathbb{P}(Y < r)$ is monotonically decreasing in N, all $N \geq N_1$ have $\mathbb{P}(Y < r) \to 0$. Therefore, we have $N \leq N_1 = O(nr + n\log n)$.

When $r = \omega(\log n)$, we choose $N_2 = 2ns + n\log n$ and then

$$\begin{aligned}
&\frac{N_2}{ns} - \log\frac{N_2}{ns} - 1 - \frac{1}{s}\log\frac{n}{\sqrt{s}} \\
=&2 + \frac{\log n}{s} - \log\left(2 + \frac{\log n}{s}\right) - 1 - \frac{1}{s}\log n + \frac{1}{2s}\log s \\
=&1 - \log\left(2 + \frac{\log n}{s}\right) + \frac{1}{2s}\log s > \frac{1}{2s}\log s
\end{aligned}$$

Thus $f > s \cdot \log s/(2s) = \log s/2 = \omega(1)$ and $\mathbb{P}(Y < r) \to 0$. Hence, all $N \geq N_2$ have $\mathbb{P}(Y < r) \to 0$. Therefore, we have $N \leq N_2 = O(nr + n\log n)$.

The only case left is when $r = \Theta(\log n)$ and we need to take care of the constant. When $\lim \frac{\log n}{s} \leq 2$, we choose $N_3 = 3ns + n\log n$ and then

$$\begin{aligned}
&\frac{N_3}{ns} - \log\frac{N_3}{ns} - 1 - \frac{1}{s}\log\frac{n}{\sqrt{s}} \\
=&3 + \frac{\log n}{s} - \log\left(3 + \frac{\log n}{s}\right) - 1 - \frac{1}{s}\log n + \frac{1}{2s}\log s \\
=&2 - \log\left(3 + \frac{\log n}{s}\right) + \frac{1}{2s}\log s \\
\geq&2 - \log 5 + \frac{1}{2s}\log s > \frac{1}{2s}\log s
\end{aligned}$$

Thus $f = \omega(1)$ and $\mathbb{P}(Y < r) \to 0$. Hence, all $N \geq N_3$ have $\mathbb{P}(Y < r) \to 0$. Therefore, we have $N \leq N_3 = O(nr + n\log n)$.

When $\lim \frac{\log n}{s} > 2$, we choose $N_4 = ns + 2n\log n$. Notice that function $x -$

$\log(1+2x)$ is always greater than 0.2 for all $x > 2$.

$$\begin{aligned}
&\frac{N_4}{ns} - \log\frac{N_4}{ns} - 1 - \frac{1}{s}\log\frac{n}{\sqrt{s}} \\
=&1 + 2\cdot\frac{\log n}{s} - \log\left(1 + 2\cdot\frac{\log n}{s}\right) - 1 - \frac{1}{s}\log n + \frac{1}{2s}\log s \\
=&\frac{\log n}{s} - \log\left(1 + 2\cdot\frac{\log n}{s}\right) + \frac{1}{2s}\log s \\
>&0.2 + \frac{1}{2s}\log s > \frac{1}{2s}\log s
\end{aligned}$$

Thus $f = \omega(1)$ and $\mathbb{P}(Y < r) \to 0$. Hence, all $N \geq N_4$ have $\mathbb{P}(Y < r) \to 0$. Therefore, we have $N \leq N_4 = O(nr + n\log n)$.

Combining with the lower bound $N = \Omega(nr+n\log n)$ we have $N = \Theta(nr+n\log n)$ and complete the proof. ■

An intuitive interpretation of this bound is that we throw the first $\Theta(nr)$ balls to have all the bins almost full, and after that the last stage is to wait for these almost-full bins to be eventually full, which is a classic coupon collector. The r-coupon collector gives a convergence lower bound for the binary signaling consensus process.

Corollary 2.2 *With high probability, a binary signaling consensus process needs* $\Omega(nr + n\log n)$ *interactions to converge.*

2.4 Empirical Results and Conjectures

To support our theoretical results, in this section we present a series of empirical results, based on which we propose several conjectures for different aspects of binary signaling consensus. All experiments were run in MATLAB on a workstation built with Intel i5-2500 3.30GHz CPU and 8GB memory. To be more robust against

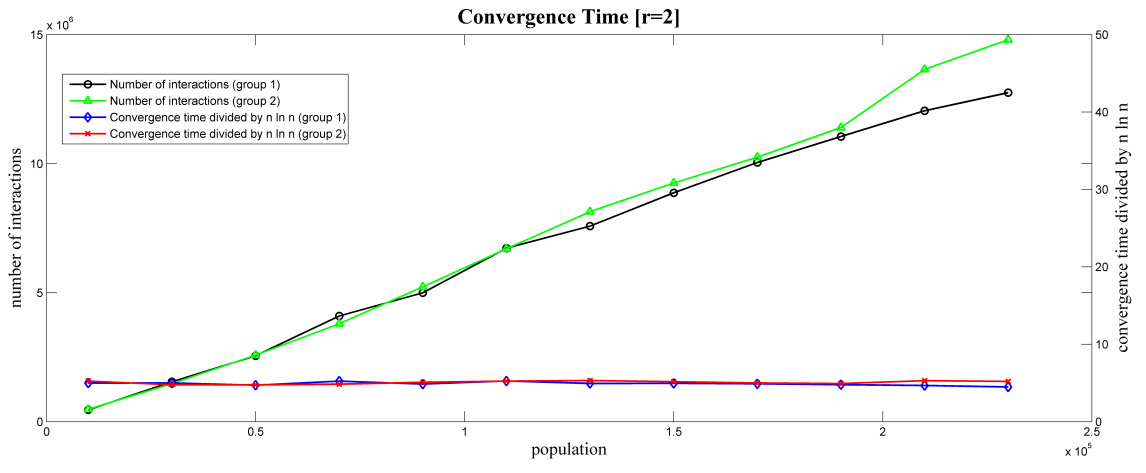

Figure 2.1: The number of interactions with fixed resistance 2 and varying population

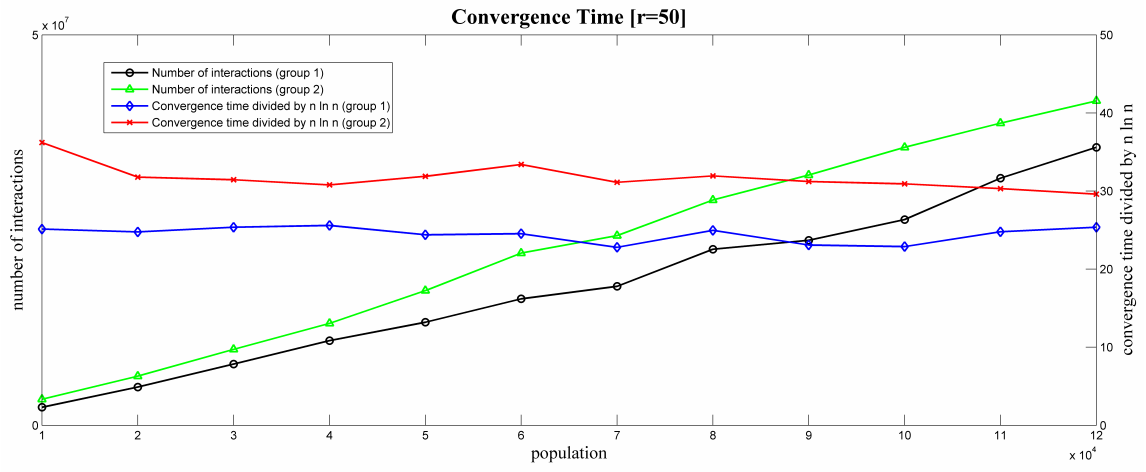

Figure 2.2: The number of interactions with fixed resistance 50 and varying population

fluctuation from randomness, each test was run for ten times and the medians were taken. The simulation of the discrete model strictly follows the description in Section 2.1 using discrete time steps, and the continuous-time process is simulated according to the corresponding system of differential equations derived in Section 2.3.1 using the Runge-Kutta method.

The experiments start from verifying the fast convergence result for three-state binary signaling model. In Section 2.2, we have proved that with high probability

the society with fixed resistance $r = 2$ will reach consensus within $\Theta(n \log n)$ interactions. We study two groups of experiments with different initial configurations. Group 1 is a society starting with everyone in the intermediate state. Group 2 is a society with initial balanced configuration where half of the population supports the grammar with full confidence while the other half is in the opposite state. These are two worst cases that are expected to have the longest convergence time and are ideal for examining convergence upper bound. We fix the resistance r as 2 and vary the population n. The results are plotted in Figure 2.1 with the curves of convergence time (i.e., the number of interactions in discrete model) of the two groups respectively. These two curves indicate the society in group 2 converges slightly slower than the one in group 1. To verify the order of the convergence time and estimate the concrete constant, we divide the number of interactions by $n \log n$ and also show this quotient on the plot. From the results we can see this quotient is stable around 5. This is supportive evidence of our theoretical results on the order of convergence rate. However, the constant we provided in Theorem 2.1 seems too large, as the experiments suggest this constant be 5, or conservatively speaking, smaller than 10, which leads to our first conjecture.

Conjecture 2.1 *With high probability, the number of interactions for a society with resistance* 2 *to reach consensus is at most* $10 \cdot n \log n$ *for all sufficiently large* n*.*

This means there is still space to improve our constants in Theorem 2.1.

What interests us more is the large-r case, for the questions we are not able to answer theoretically. We have noticed that the number of interactions $\Theta(n \log n)$ of the three-state binary signaling model is exactly the tight bound of the coupon collector problem. This inspires us that the tight bound of the r-coupon collector might also indicate (or at least approximate) the convergence time of large-r binary

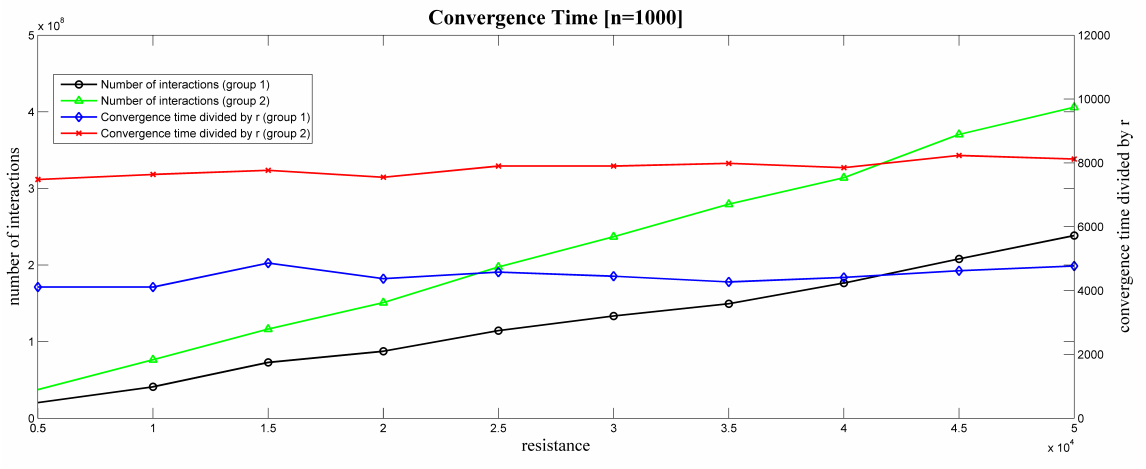

Figure 2.3: The number of interactions with fixed population 1000 and varying resistance

signaling consensus. In Section 2.3.2 we have shown $\Theta(nr + n\log n)$ is a tight bound for the r-coupon collector process. Thus it is reasonable for us to conjecture $\Theta(nr + n\log n)$ as the number of interactions in the large-r case.

Conjecture 2.2 *With high probability, the number of interactions for a society with resistance r to reach consensus in the worst case is $\Theta(nr + n\log n)$.*

We seek empirical evidence to support this conjecture. Since the bound $\Theta(nr + n\log n)$ involves both r and n, we conduct two sets of experiments with fixed r (shown in Figure 2.2) and with fixed n (shown in Figure 2.3) respectively. With fixed r and varying n, we expect the number of interactions to increase in the order of $\Theta(n\log n)$. In fact the experiments we presented above for the three-state model can serve as the supportive fixed-r experiments needed here. Nevertheless, given the essential difference between the $r = 2$ case and the $r > 2$ case discussed in Section 2.3.1, we found it more persuasive to choose a large value of r. In Figure 2.2, we fix the resistance r as 50 and vary the population n. The four curves are plotted as in the previous experiments with $r = 2$ and have similar shapes. The process converges obviously slower with $r = 50$ than with $r = 2$. The constant is

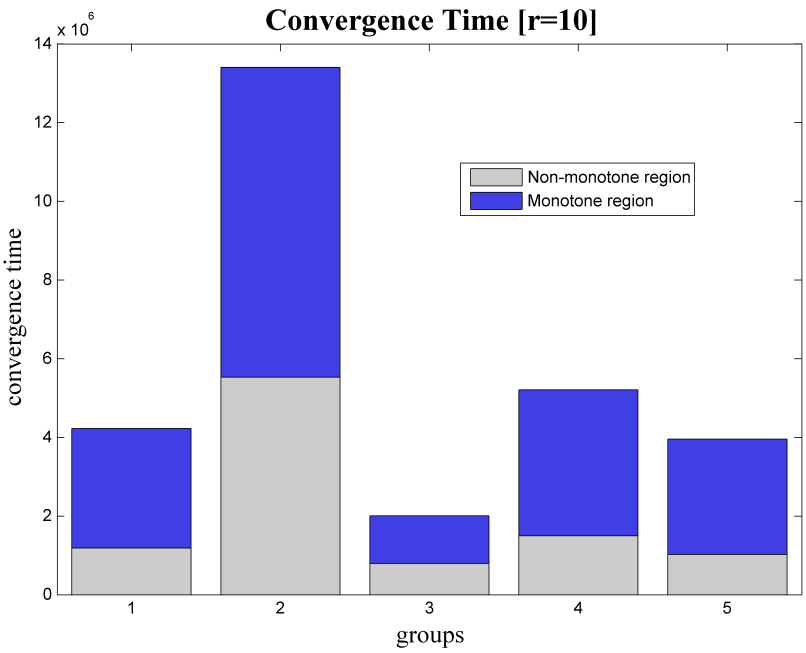

Figure 2.4: Convergence time comparison for continuous process

also larger. For group 1 the quotient is stable around 28 and for group 2 it is around 33. The process in group 2 is still slower to converge than the one in group 1 but the difference is now more apparent. Hence, the behaviors of the curves match what our conjecture predicts. Figure 2.3 shows the curves of convergence time when we fix the population n as 1000 and vary the resistance r. The same four curves are plotted and the only difference is now we divide the number of interactions by r, as we expect the convergence time to be $\Theta(r)$ with fixed n. Group 1 is still faster than group 2 in the sense of convergence and also with smaller constant, which is stable around 5000 while the constant of group 2 is about 8200. These large constants are not surprising since all the values of n and r we choose for this set are quite large. Again these results agree with the prediction of our conjecture.

In Section 2.3.1 we studied the continuous-time process in the limit with infinitesimal time step and showed that the convergence time is $O(r \log nr)$ if it starts from a monotone initial configuration. However, the behavior of the process outside

the monotone region is still uncertain. Fortunately, empirical simulation suggests the process will enter the monotone region fast enough and then go to convergence rapidly. To simulate the continuous-time binary signaling model, we follow the corresponding system of differential equations derived in Section 2.3.1 using the Runge-Kutta method. As this is a numerical method, we are unable to have n equal to infinity with infinitesimal time step. To approximate the process well, we choose a large value of n and let $n = 100000$. In order to show the process will eventually enter the monotone region with any initial configuration, we conduct more groups of simulations with different types of initial configuration. Figure 2.4 demonstrates the experimental results in the form of a bar chart to compare the time in the non-monotone region and the time in the monotone region. The initial setup of each group is as follows.

Group 1: 40% of the population at confidence level 0 and 60% of the population at confidence level r;

Group 2: $1/2 - 1/(nr)$ of the population at confidence level 0 and $1/2 + 1/(nr)$ of the population at confidence level r;

Group 3: 0.1% of the population at confidence level 0 and 99.9% of the population at confidence level r;

Group 4: 50% of the population at confidence level 1 and 50% of the population at confidence level r;

Group 5: 40% of the population at confidence level 1 and 60% of the population at confidence level r.

Group 1 is designed for the majority computation scenario. Group 2 and group 3 are to show the process will enter the monotone region first before convergence regardless of whether the population is almost balanced (group 2) or almost converged (group 3). As expected, group 2 is the slowest to converge while group 3 is

the fastest. Group 4 and group 5 are designed to witness the drop of p in the $p > 1/2$ region, which is an essential difference between the $r = 2$ case and the $r > 2$ case (Lemma 2.14). From these results we propose the following conjecture.

Conjecture 2.3 *A continuous-time binary signaling process will enter the monotone region before convergence starting from any initial configuration.*

The bar chart also suggests the time needed to enter the monotone region doesn't dominate the whole process, although it is still considerable in some special cases such as group 2. Thus it is reasonable to conjecture that the total convergence time is of the same order as the convergence time inside the monotone region we presented in Theorem 2.2.

Conjecture 2.4 *A continuous-time binary signaling process reaches consensus within* $O(r \log nr)$ *time.*

2.5 Conclusion and Future Work

We study here the language emergence process in human society. To capture this process, we describe and analyze a binary signaling consensus model for language emergence, which builds a connection between language emergence process and the study of population protocols. We present a tight convergence bound $\Theta(n \log n)$ with concrete constants for the three-state binary signaling consensus process where the resistance parameter r is 2. Even though this model appears to be quite simple, it turns out to be very hard to analyze. When the resistance r is large, we show the continuous-time binary signaling process in the limit will reach consensus within $O(r \log nr)$ time if the initial configuration is monotone. We show that the binary signaling process needs at least $\Omega(nr + n \log n)$ interactions to converge with high

probability. To support our theoretical results, we have done a series of experiments, based on which we also propose several conjectures for the convergence properties of the process.

One direct open question is to prove or disprove the conjectures we propose in this chapter, especially those for the large-r case. A potential way to study the large-r case is to generalize the proof idea for the three-state model, which divides the configuration space into several regions and constructs a well-bounded supermartingale process for each region using carefully chosen potential functions. The high dimensionality of the configuration space would be one of the trickiest parts in the analysis. Another direction of future work is to study a more general model of binary signaling consensus where, for example, the interaction graph is not necessarily complete, or different people in the society could have different resistance values. We are also interested in multi-valued consensus under this binary signaling setting, where there is more than one grammar spreading among the society which are not independent. With our convergence upper bound for the three-state binary signaling consensus process, additional results can be proved including bounds on approximate majority computation, correctness with epidemic-triggered start and tolerance towards Byzantine agents. Last but not least, we believe this model can be generalized and applied to other real-world problems because, as described in Section 1.1, the language emergence process in human society shares many similarities with other dynamic systems in the world. We are hoping this work will also make a contribution to other related fields such as epidemiology, physics and biology.

Chapter 3

Learning Shuffle Ideals Under Restricted Distributions

The class of shuffle ideals is a fundamental sub-family of regular languages. The shuffle ideal generated by a string set U is the collection of all strings containing some string $u \in U$ as a (not necessarily contiguous) subsequence. In spite of its apparent simplicity, the problem of learning a shuffle ideal from given data is known to be computationally intractable. In this chapter, we study the PAC learnability of shuffle ideals. After introducing the preliminaries in Section 3.1, we present our main result in Section 3.2: the extended class of shuffle ideals is PAC learnable from element-wise i.i.d. strings. That is, the distributions of the symbols in a string are identical and independent of each other. A constrained generalization to learning shuffle ideals under product distributions is also provided. In Section 3.3, we further show the PAC learnability of principal shuffle ideals when the example strings drawn from $\Sigma^{\leq n}$ are generated by a Markov chain with some lower bound assumptions on the transition matrix. In Section 3.5, we propose a greedy algorithm for learning principal shuffle ideals under general unrestricted distributions. Experiments demonstrate the

advantage for both efficiency and accuracy of our heuristic algorithm.

The content of this chapter appears in [Che14].

3.1 Preliminaries

We consider strings over a fixed finite alphabet Σ. The empty string is λ. Let Σ^* be the Kleene star of Σ and $\Sigma^\cup$ be the collection of all subsets of Σ. As strings are concatenations of symbols, we similarly define augmented strings as concatenations of unions of symbols.

Definition 3.1 (Alphabet, simple string and augmented string) *Let Σ be a non-empty finite set of symbols, called the alphabet. A simple string over Σ is any finite sequence of symbols from Σ, and Σ^* is the collection of all simple strings. An augmented string over Σ is any finite sequence of symbol sets from $\Sigma^\cup$, and $(\Sigma^\cup)^*$ is the collection of all augmented strings.*

For example, $(a|b|d)a(b|c)$ is an augmented string. It is the set of the strings which start with an 'a' or a 'b' or a 'd', followed by an 'a', and end with a 'b' or a 'c'.

Denote by s the cardinality of Σ. Because an augmented string only contains strings of the same length, the length of an augmented string U, denoted by $|U|$, is the length of any $u \in U$. We use exponential notation for repeated concatenation of a string with itself, that is, v^k is the concatenation of k copies of string v. Starting from index 1, we denote by v_i the i-th symbol in string v and use notation $v[i, j] = v_i \dots v_j$ for $1 \leq i \leq j \leq |v|$. Define the binary relation $\sqsubseteq$ on $\langle (\Sigma^\cup)^*, \Sigma^* \rangle$ as follows. For a simple string w, $w \sqsubseteq v$ holds if and only if there is a witness $\vec{i} = (i_1 < i_2 < \ldots < i_{|w|})$ such that $v_{i_j} = w_j$ for all integers $1 \leq j \leq |w|$. For an augmented string W, $W \sqsubseteq v$ if and only if there exists some $w \in W$ such that $w \sqsubseteq v$. When there are

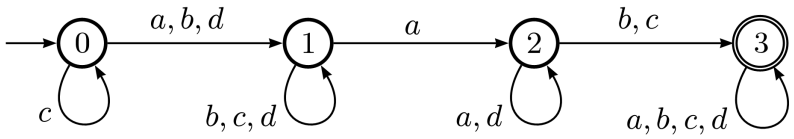

Figure 3.1: The DFA accepting precisely the shuffle ideal of $U = (a|b|d)a(b|c)$ over $\Sigma = \{a, b, c, d\}$.

several witnesses for $W \sqsubseteq v$, we may order them coordinate-wise, referring to the unique minimal element as the leftmost embedding. We will write $I_{W \sqsubseteq v}$ to denote the position of the last symbol of W in its leftmost embedding in v (if the latter exists; otherwise, $I_{W \sqsubseteq v} = \infty$).

Definition 3.2 (Extended/Principal Shuffle Ideal) *The (extended) shuffle ideal of an augmented string $U \in (\Sigma^{\cup})^L$ is a regular language defined as $\text{Ш}(U) = \{v \in \Sigma^* \mid \exists u \in U, u \sqsubseteq v\} = \Sigma^* U_1 \Sigma^* U_2 \Sigma^* \dots \Sigma^* U_L \Sigma^*$. A shuffle ideal is principal if it is generated by a simple string.*

A shuffle ideal is an order ideal on monoid $\langle \Sigma^*, \cdot, \lambda \rangle$ and was originally defined for lattices. Denote by $\sqcup\!\sqcup$ the class of principal shuffle ideals and by Ш the class of extended shuffle ideals. Unless otherwise stated, in this chapter shuffle ideal refers to the extended ideal. An example is given in Figure 3.1. The feasibility of determining whether a string is in the class $\text{Ш}(U)$ is obvious.

Lemma 3.1 *Evaluating relation $U \sqsubseteq x$ and meanwhile determining $I_{U \sqsubseteq x}$ is feasible in time $O(|x|)$.*

Proof The evalution can be done recursively. The base case is $U = \lambda$, where $U \sqsubseteq x$ holds and $I_{U \sqsubseteq x} = 0$. If $U = U_1 U'$ where $U_1 \in \Sigma^{\cup}$, we search for the leftmost occurrence of U_1 in x. If there is no such occurrence, then $U \not\sqsubseteq x$ and $I_{U \sqsubseteq x} = \infty$. Otherwise, $x = y U_1 x'$, where $U_1 \not\sqsubseteq y$. Then $U \sqsubseteq x$ if and only if $U' \sqsubseteq x'$ and $I_{U \sqsubseteq x} = I_{U_1 \sqsubseteq x} + I_{U' \sqsubseteq x'}$. We continue recursively with U' and x'. The total running

time of this procedure is $O(|x|)$. ■

In a computational learning model, an algorithm is usually given access to an oracle providing information about the sample. In Valiant's work [Val84], the example oracle $EX(c, \mathcal{D})$ was defined, where c is the target concept and $\mathcal{D}$ is a distribution over the instance space. On each call, $EX(c, \mathcal{D})$ draws an input x independently at random from the instance space $\mathcal{I}$ under the distribution $\mathcal{D}$, and returns the labeled example $\langle x, c(x) \rangle$.

Definition 3.3 (PAC Learnability: [Val84]) *Let $\mathcal{C}$ be a concept class over the instance space $\mathcal{I}$. We say $\mathcal{C}$ is probably approximately correctly (PAC) learnable if there exists an algorithm $\mathcal{A}$ with the following property: for every concept $c \in \mathcal{C}$, for every distribution $\mathcal{D}$ on $\mathcal{I}$, and for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, if $\mathcal{A}$ is given access to $EX(c, \mathcal{D})$ on $\mathcal{I}$ and inputs ϵ and δ, then with probability at least $1 - \delta$, $\mathcal{A}$ outputs a hypothesis $h \in \mathcal{H}$ satisfying $\mathbb{P}_{x \in \mathcal{D}}(c(x) \neq h(x)) \leq \epsilon$. If $\mathcal{A}$ runs in time polynomial in $1/\epsilon$, $1/\delta$ and the representation size of c, we say that $\mathcal{C}$ is efficiently PAC learnable.*

We refer to ϵ as the error parameter and δ as the confidence parameter. If the error parameter is set to 0, the learning is exact [Bsh97]. Kearns [Kea98] extended Valiant's model and introduced the statistical query oracle $STAT(c, \mathcal{D})$. 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. $STAT(c, \mathcal{D})$ returns an estimate for the expectation $\mathbb{E}\chi$, that is, the probability that $\chi = 1$ when the labeled example is drawn according to $\mathcal{D}$. A statistical query can have a condition so $\mathbb{E}\chi$ can be a conditional probability. This estimate is accurate within additive error τ.

Definition 3.4 (Legitimacy and Feasibility: [Kea98]) *A statistical query χ is legimate and feasible if and only if with respect to $1/\epsilon$, $1/\tau$ and representation size of c:*

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 polynomially large.*

Throughout this chapter, the learnability of shuffle ideals is studied in the statistical query model. Kearns [Kea98] proves that oracle $STAT(c, \mathcal{D})$ is weaker than oracle $EX(c, \mathcal{D})$. In other words, if a concept class is PAC learnable from $STAT(c, \mathcal{D})$, then it is PAC learnable from $EX(c, \mathcal{D})$, but not necessarily vice versa.

Angluin et al. [AAEK13] have proved the class of shuffle ideals is not efficiently PAC learnable unless RP=NP. In the positive direction, they showed that a principal shuffle ideal can be efficiently approximately learned in the statistical query model under the uniform distribution.

3.2 Learning shuffle ideals from element-wise i.i.d. strings

Although learning the class of shuffle ideals has been proved hard, in most scenarios the string distribution is restricted or even known. A very usual situation in practice is that we have some prior knowledge of the unknown distribution. One common

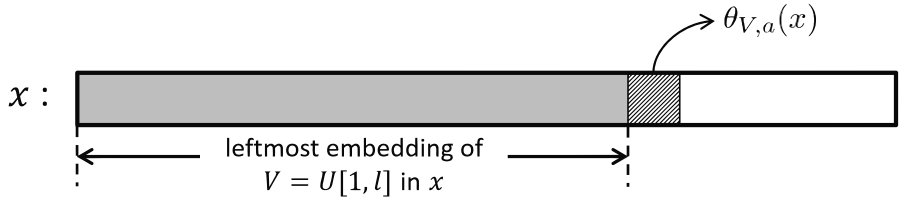

Figure 3.2: Definition of $\theta_{V,a}(x)$ when $V = U[1, \ell]$

example is the string distributions where each symbol in a string is generated independently and identically from an unknown distribution. It is element-wise i.i.d. because we view a string as a vector of symbols. This case is general enough to cover some popular distributions in applications such as the uniform distribution and the multinomial distribution. In this section, we present as our main result a statistical query algorithm for learning the concept class of extended shuffle ideals from element-wise i.i.d. strings and provide theoretical guarantees of its computational efficiency and accuracy in the statistical query model. The instance space is Σ^n. Denote by U the augmented pattern string that generates the target shuffle ideal and by $L = |U|$ the length of U.

3.2.1 Statistical query algorithm

Before presenting the algorithm, we define function $\theta_{V,a}(\cdot)$ and query $\chi_{V,a}(\cdot, \cdot)$ for any augmented string $V \in (\Sigma^{\cup})^{\leq n}$ and any symbol $a \in \Sigma$ as as follows.

$$\theta_{V,a}(x) = \begin{cases} a & \text{if } V \not\sqsubseteq x[1, n-1] \\ x_{I_{V \sqsubseteq x}+1} & \text{if } V \sqsubseteq x[1, n-1] \end{cases}$$

$$\chi_{V,a}(x, y) = \frac{1}{2}(y + 1) \quad \text{given } \theta_{V,a}(x) = a$$

where $y = c(x)$ is the label of example string x. More precisely, $y = +1$ if $x \in \text{Ш}(U)$ and $y = -1$ otherwise. Figure 3.2 explains the definition of $\theta_{V,a}(x)$ when we have

$V = U[1, \ell]$. For any augmented string V, if at least one element in V is a subsequence of $x[1, n-1]$, then $\theta_{V,a}(x)$ is the symbol next to the leftmost embedding of V in x. Otherwise, $\theta_{V,a}(x)$ is simply the symbol a. Conditioned on $\theta_{V,a}(x) = a$, the query $\chi_{V,a}(x, y)$ is 1 if x is a positive string and is 0 otherwise. The expected value of $\chi_{V,a}$ under the distribution over the instance space means the conditional probability of positivity given $\theta_{V,a}(x) = a$.

Our learning algorithm uses statistical queries to recover string $U \in (\Sigma^{\cup})^L$ one element at a time. It starts with the empty string $V = \lambda$. Having recovered $V = U[1, \ell]$ where $0 \leq \ell < L$, we infer $U_{\ell+1}$ as follows. For each $a \in \Sigma$, the statistical query oracle is called with the query $\chi_{V,a}$ at the error tolerance τ claimed in Theorem 3.1. Our key technical observation is that the value of $\mathbb{E}\chi_{V,a}$ effectively selects $U_{\ell+1}$. The query results of $\chi_{V,a}$ will form two separate clusters such that the maximum difference (variance) inside one cluster is smaller than the minimum difference (gap) between the two clusters, making them distinguishable. The set of symbols in the cluster with larger query results is proved to be $U_{\ell+1}$. Notice that this statistical query only works for $0 \leq \ell < L$. To complete the algorithm, we address the trivial case $\ell = L$ with query $\mathbb{P}(y = +1 \mid V \sqsubseteq x)$ and the algorithm halts if the query answer is close to 1.

3.2.2 PAC learnability

We show the algorithm described above learns the class of shuffle ideals from element-wise i.i.d. strings in the statistical query learning model.

Theorem 3.1 *Under element-wise independent and identical distributions over instance space $\mathcal{I} = \Sigma^n$, concept class* III *is approximately identifiable with $O(sn)$ con-*

ditional statistical queries from $STAT(\text{Ш}, \mathcal{D})$ *at tolerance*

$$\tau = \frac{\epsilon^2}{40sn^2 + 4\epsilon}$$

or with $O(sn)$ *statistical queries from* $STAT(\text{Ш}, \mathcal{D})$ *at tolerance*

$$\bar{\tau} = \left(1 - \frac{\epsilon}{20sn^2 + 2\epsilon}\right) \frac{\epsilon^4}{16sn(10sn^2 + \epsilon)}$$

The proof starts from the legitimacy and feasibility of the algorithm. Since $\chi_{V,a}$ computes a binary mapping from labeled examples to $\{0, 1\}$, the legitimacy is trivial. But $\chi_{V,a}$ is not feasible for symbols in Σ of small occurrence probabilities. We avoid the problematic cases by reducing the original learning problem to the same problem with a polynomial lower bound assumption $\mathbb{P}(x_i = a) \geq \epsilon/(2sn) - \epsilon^2/(20sn^2 + 2\epsilon)$ for any $a \in \Sigma$ and achieve feasibility.

The correctness of the algorithm is based on the intuition that the query result $\mathbb{E}\chi_{V,a_+}$ of a symbol $a_+ \in U_{\ell+1}$ should be greater than that of a symbol $a_- \notin U_{\ell+1}$ and the difference is large enough to tolerate the noise from the oracle. To prove this, we first consider the exact learning case. Define an infinite string $U' = U[1, \ell]U[\ell + 2, L]U_{\ell+1}^{\infty}$ and let $x' = x\Sigma^{\infty}$ be the extension of x obtained by padding it on the right with an infinite string generated from the same distribution as x. Let $Q(j, i)$ be the probability that the largest g such that $U'[1, g] \sqsubseteq x'[1, i]$ is j, or formally

$$Q(j, i) = \mathbb{P}(U'[1, j] \sqsubseteq x'[1, i] \wedge U'[1, j + 1] \not\sqsubseteq x'[1, i])$$

By taking the difference between $\mathbb{E}\chi_{V,a_+}$ and $\mathbb{E}\chi_{V,a_-}$ in terms of $Q(j, i)$, we get the query tolerance for exact learning.

Lemma 3.2 *Under element-wise independent and identical distributions over in-*

stance space $\mathcal{I} = \Sigma^n$*, concept class* III *is exactly identifiable with* $O(sn)$ *conditional statistical queries from* $STAT(\text{III}, \mathcal{D})$ *at tolerance*

$$\tau' = \frac{1}{5}Q(L-1, n-1)$$

Lemma 3.2 indicates bounding the quantity $Q(L-1, n-1)$ is the key to the tolerance for PAC learning. Unfortunately, the distribution $\{Q(j,i)\}$ doesn't seem of any strong properties we know of providing a polynomial lower bound. Instead we introduce new quantity

$$R(j,i) = \mathbb{P}(U'[1,j] \sqsubseteq x'[1,i] \wedge U'[1,j] \not\sqsubseteq x'[1,i-1])$$

being the probability that the smallest g such that $U'[1,j] \sqsubseteq x'[1,g]$ is i. An important property of distribution $\{R(j,i)\}$ is its strong unimodality as defined below.

Definition 3.5 (Unimodality: [GK49]) *A distribution* $\{P(i)\}$ *with all support on the lattice of integers is unimodal if and only if there exists at least one integer* K *such that* $P(i) \geq P(i-1)$ *for all* $i \leq K$ *and* $P(i+1) \leq P(i)$ *for all* $i \geq K$*. We say* K *is a mode of distribution* $\{P(i)\}$*.*

Throughout this chapter, when referring to the mode of a distribution, we mean the one with the largest index, if the distribution has multiple modes with equal probabilities.

Definition 3.6 (Strong Unimodality: [Ibr56]) *A distribution* $\{H(i)\}$ *is strongly unimodal if and only if the convolution of* $\{H(i)\}$ *with any unimodal distribution* $\{P(i)\}$ *is unimodal.*

Since a distribution with all mass at zero is unimodal, a strongly unimodal distribution is also unimodal. In this chapter, we only consider distributions with all

support on the lattice of integers. So the convolution of $\{H(i)\}$ and $\{P(i)\}$ is

$$\{H * P\}(i) = \sum_{j=-\infty}^{\infty} H(j)P(i-j) = \sum_{j=-\infty}^{\infty} H(i-j)P(j)$$

We prove the strong unimodality of $\{R(j,i)\}$ with respect to i via showing it is the convolution of two log-concave distributions by induction. We do an initial statistical query to estimate $\mathbb{P}(y=+1)$ to handle two marginal cases $\mathbb{P}(y=+1) \leq \epsilon/2$ and $\mathbb{P}(y=+1) \geq 1-\epsilon/2$. After that an additional query $\mathbb{P}(y=+1 \mid V \sqsubseteq x)$ is made to tell whether $\ell = L$. If the algorithm doesn't halt, it means $\ell < L$ and both $\mathbb{P}(y=+1)$ and $\mathbb{P}(y=-1)$ are at least $\epsilon/2 - 2\tau$. By upper bounding $\mathbb{P}(y=+1)$ and $\mathbb{P}(y=-1)$ using linear sums of $R(j,i)$, the strong unimodality of $\{R(j,i)\}$ gives a lower bound for $R(L,n)$, which further implies one for $Q(L-1,n-1)$ and completes the proof.

Now we present the formal proof. We first provide a quick proof of Lemma 3.1.

Proof (of Lemma 3.1) The evalution can be done recursively. The base case is $U = \lambda$, where $U \sqsubseteq x$ holds and $I_{U \sqsubseteq x} = 0$. If $U = U_1 U'$ where $U_1 \in \Sigma^{\cup}$, we search for the leftmost occurrence of U_1 in x. If there is no such occurrence, then $U \not\sqsubseteq x$ and $I_{U \sqsubseteq x} = \infty$. Otherwise, $x = yU_1x'$, where $U_1 \not\sqsubseteq y$. Then $U \sqsubseteq x$ if and only if $U' \sqsubseteq x'$ and $I_{U \sqsubseteq x} = I_{U_1 \sqsubseteq x} + I_{U' \sqsubseteq x'}$. We continue recursively with U' and x'. The total running time of this procedure is $O(|x|)$. ■

Lemma 3.3 *Under element-wise independent and identical distributions over instance space $\mathcal{I} = \Sigma^n$, the conditional statistical query $\chi_{V,a}$ is legitimate and feasible at tolerance*

$$\tau = \frac{\epsilon^2}{40sn^2 + 4\epsilon}$$

Proof First of all, the function $\chi_{V,a}$ computes a binary mapping from labeled examples (x,y) to $\{0,1\}$ and satisfies the definition of a statistical query. Given

$\theta_{V,a}(x) = a$, that is, given $V \not\sqsubseteq x[1, n-1]$ or $x_{I_{V\sqsubseteq x}+1} = a$ if $V \sqsubseteq x[1, n-1]$, the query $\chi_{V,a}(x, y)$ returns 0 if x is a negative example ($y = -1$) or returns 1 if x is a positive example ($y = +1$).

From Lemma 1, evaluating the relation $V \sqsubseteq x$ and meanwhile determining $I_{V\sqsubseteq x}$ is feasible in time $O(n)$. Thus, $\theta_{V,a}(x)$ and then $\chi_{V,a}(x, y)$ can be efficiently evaluated.

For

$$\begin{aligned}\mathbb{P}(\theta_{V,a}(x) = a) =& \mathbb{P}(V \not\sqsubseteq x[1, n-1]) + \\ & \mathbb{P}(V \sqsubseteq x[1, n-1]) \cdot \mathbb{P}(x_{I_{V\sqsubseteq x}+1} = a \mid V \sqsubseteq x[1, n-1])\end{aligned}$$

in order to prove $\mathbb{P}(\theta_{V,a}(x) = a)$ not too small, we only need to show one of the two items in the sum is at least polynomially large.

We make an initial statistical query with tolerance $\tau = \epsilon^2/(40sn^2+4\epsilon)$ to estimate $\mathbb{P}(y = +1)$. If the answer is $\leq \epsilon - \tau$, then $\mathbb{P}(y = +1) \leq \epsilon$ and the algorithm outputs a hypothesis that all examples are negative. Otherwise, $\mathbb{P}(y = +1)$ is at least $\epsilon - 2\tau$, and the statistical query $\chi_{V,a}$ is used. As $V \sqsubseteq x[1, n-1] = U[1, \ell] \sqsubseteq x[1, n-1]$ is a necessary condition of $y = +1$, we have

$$\mathbb{P}(V \sqsubseteq x[1, n-1]) \geq \mathbb{P}(y = +1) \geq \epsilon - \frac{\epsilon^2}{20sn^2 + 2\epsilon}$$

Since $x_{I_{V\sqsubseteq x}+1}$ and $x[1, I_{V\sqsubseteq x}]$ are independent,

$$\mathbb{P}(x_{I_{V\sqsubseteq x}+1} = a \mid V \sqsubseteq x[1, n-1]) = \mathbb{P}(x_{I_{V\sqsubseteq x}+1} = a)$$

Because we don't have any knowledge of the distribution, we can't guarantee $\mathbb{P}(x_{I_{V\sqsubseteq x}+1} = a)$ is large enough for every $a \in \Sigma$. However, we notice that there is no need to consider symbols with small probabilities of occurrence. Now we show why

and how. For each $a \in \Sigma$, execute a statistical query

$$\chi'_a(x, y) = \mathbb{1}_{\{x_i = a\}} \tag{3.1}$$

at tolerance τ, where $\mathbb{1}_{\{\pi\}}$ represents the 0-1 truth value of the predicate π. Since the strings are element-wise i.i.d., the index i can be any integer between 1 and n. If the answer from oracle $STAT$ is $\leq \epsilon/(2sn) - \tau$, then $\mathbb{P}(x_i = a) \leq \epsilon/(2sn)$. For such an a, the probability that it shows up in a string is at most $\epsilon/(2s)$. Because there are at most $s - 1$ such symbols in Σ, the probability that any of them shows up in a string is at most $\epsilon/2$. Otherwise, $\mathbb{P}(x_i = a) \geq \epsilon/(2sn) - 2\tau$. Thus we only need to consider the symbols $a \in \Sigma$ such that $\mathbb{P}(x_i = a) \geq \epsilon/(2sn) - 2\tau$ and learn the ideal with error parameter $\epsilon/2$ so that the total error will be bounded within ϵ. For algebraic succinctness, we use a concise lower bound for $\mathbb{P}(x_i = a)$:

$$\mathbb{P}(x_i = a) \geq \frac{\epsilon}{2sn} - 2\tau = \frac{\epsilon}{2sn} - \frac{\epsilon^2}{20sn^2 + 2\epsilon} \geq \frac{\epsilon}{4sn} \tag{3.2}$$

Eventually we have

$$\begin{aligned} \mathbb{P}(\theta_{V,a}(x) = a) &\geq \mathbb{P}(V \sqsubseteq x[1, n-1]) \cdot \mathbb{P}(x_{I_{V \sqsubseteq x}+1} = a \mid V \sqsubseteq x[1, n-1]) \\ &\geq \left(1 - \frac{\epsilon}{20sn^2 + 2\epsilon}\right) \frac{\epsilon^2}{4sn} \end{aligned} \tag{3.3}$$

is polynomially large. Query $\chi_{V,a}$ is legitimate and feasible. ■

The correctness of the algorithm is based on the intuition that the query result $\mathbb{E}\chi_{V,a_+}$ of $a_+ \in U_{\ell+1}$ should be greater than that of $a_- \notin U_{\ell+1}$ and the difference is large enough to tolerate the noise from the oracle. To prove this, we first consider the exact learning case. Define an infinite string $U' = U[1, \ell]U[\ell + 2, L]U_{\ell+1}^{\infty}$ and let

$x' = x\Sigma^\infty$ be the extension of x obtained by padding it on the right with an infinite string generated from the same distribution as x. Let $Q(j,i)$ be the probability that the largest g such that $U'[1,g] \sqsubseteq x'[1,i]$ is j, or formally, $Q(j,i) = \mathbb{P}(U'[1,j] \sqsubseteq x'[1,i] \wedge U'[1,j+1] \not\sqsubseteq x'[1,i])$.

Proof (of Lemma 3.2) If the algorithm doesn't halt, U has not been completely recovered and $\ell < L$. By assumption, $V = U[1,\ell]$. If $V \not\sqsubseteq x[1,n-1]$ then x must be a negative example and $\chi_{V,a}(x,y) = 0$. Hence $\chi_{V,a}(x,y) = 1$ if and only if $V \sqsubseteq x[1,n-1]$ and $y = +1$.

Let random variable J be the largest value for which $U'[1,J]$ is a subsequence of $x[1,n-1]$. Consequently, $\mathbb{P}(J = j) = Q(j,n-1)$.

If $a \in U_{\ell+1}$, then $y = +1$ if and only if $J \geq L-1$. Thus we have

$$\mathbb{E}\chi_{V,a} = \sum_{j=L-1}^{n-1} Q(j,n-1)$$

If $a \notin U_{\ell+1}$, then $y = +1$ if and only if $U \sqsubseteq x[1, I_{V\sqsubseteq x}]x[I_{V\sqsubseteq x}+2, n]$. Since elements in a string are i.i.d., $\mathbb{P}(U \sqsubseteq x[1, I_{V\sqsubseteq x}]x[I_{V\sqsubseteq x}+2, n]) = \mathbb{P}(U'[1,L] \sqsubseteq x[1,n-1])$, which is exactly $\mathbb{P}(J \geq L)$. Thus we have

$$\mathbb{E}\chi_{V,a} = \sum_{j=L}^{n-1} Q(j,n-1)$$

The difference between these two values is $Q(L-1,n-1)$. In order to distinguish the target $U_{\ell+1}$ from other symbols, the query tolerance can be set to one fifth of the difference. The alphabet Σ will be separated into two clusters by the results of $\mathbb{E}\chi_{V,a}$: $U_{\ell+1}$ and the other symbols. The maximum difference (variance) inside a cluster is smaller than the minimum difference (gap) between the two clusters, making them distinguishable. As a consequence s statistical queries for each prefix of U suffice to learn U exactly. ■

Lemma 3.2 indicates bounding the quantity $Q(L-1, n-1)$ is the key to the tolerance for PAC learning. Unfortunately, the distribution $\{Q(j,i)\}$ doesn't seem of any strong properties we know of providing a polynomial lower bound. Instead we introduce new quantity $R(j,i) = \mathbb{P}(U'[1,j] \sqsubseteq x'[1,i] \wedge U'[1,j] \not\sqsubseteq x'[1,i-1])$ being the probability that the smallest g such that $U'[1,j] \sqsubseteq x'[1,g]$ is i. Now we show the strong unimodality of distribution $\{R(j,i)\}$. Denote $p_j = \mathbb{P}(x_i \in U'_j)$.

Lemma 3.4 *The convolution of two strongly unimodal discrete distributions is strongly unimodal.*

Proof The proof is obvious from the definition of strong unimodality and the associativity of convolution. Let $H_3 = H_2 * H_1$ be the convolution of two strongly unimodal distributions H_1 and H_2. For any unimodal distribution P_1, let $P_2 = H_1 * P_1$ be the convolution of H_1 and P_1. Because of the strong unimodality of distribution H_1, P_2 is a unimodal distribution. Also because of the strong unimodality of distribution H_2, the convolution of H_3 and P_1, $H_3 * P_1 = H_2 * H_1 * P_1 = H_2 * P_2$ is a unimodal distribution. Since P_1 can be an arbitrary unimodal distribution, H_3 is strongly unimodal according to the definition of strong unimodality. ■

Previous work [Ibr56] provided a useful equivalent statement of the strong unimodality of a distribution.

Lemma 3.5 [Ibr56] Distribution $\{H(i)\}$ is strongly unimodal if and only if $H(i)$ is log-concave. That is,

$$H(i)^2 \geq H(i+1) \cdot H(i-1)$$

for all i.

Since a distribution with all mass at zero is unimodal, an immediate consequence is

Corollary 3.1 *A strongly unimodal distribution is unimodal.*

We now prove the strong unimodality of distribution $\{R(j,i)\}$.

Lemma 3.6 *For any fixed j, distribution $\{R(j,i)\}$ is strongly unimodal with respect to i.*

Proof This proof can be done by induction on j as follows.

Basis: For $j = 1$, it is obvious that $\{R(1,i)\} = \{(1-p_1)^{i-1}p_1\}$ is a geometric distribution, which is strongly unimodal. According to Lemma 3.5, this is due to $R^2(1,i) = R(1,i-1) \cdot R(1,i+1)$ for all $i > 1$.

Inductive step: For $j > 1$, assume by induction $\{R(j-1,i)\}$ is strongly unimodal. Based on the definition of $R(j,i)$, we have

$$R(j,i) = \sum_{k=j-1}^{i-1} \Big(R(j-1,k) \cdot (1-p_j)^{i-k-1} p_j \Big) \tag{3.4}$$

Thus $R(j,i)$ is the convolution of distribution $\{R(j-1,i)\}$ and distribution $\{(1-p_j)^{i-1}p_j\}$, a geometric distribution just proved to be strongly unimodal. By assumption, $\{R(j-1,i)\}$ is strongly unimodal. From Lemma 3.4, distribution $\{R(j,i)\}$ is also strongly unimodal.

Conclusion: For any fixed j, distribution $\{R(j,i)\}$ is strongly unimodal with respect to i. ■

Combining Lemma 3.6 with Corollary 3.1, we have

Corollary 3.2 *For any fixed j, distribution $\{R(j,i)\}$ is unimodal with respect to i.*

Lemma 3.7 *Denote by $N(j)$ the mode of $\{R(j,i)\}$, then $N(j)$ is strictly increasing with respect to j. That is, for any $j > 1$, $N(j) > N(j-1)$.*

Proof According to Equation 3.4, $R(j,i)$ is the convolution of distribution $\{R(j-1,i)\}$ and distribution $\{(1-p_j)^{i-1}p_j\}$ so

$$R(j,i) = \sum_{k=j-1}^{i-1} \Big(R(j-1,k) \cdot (1-p_j)^{i-k-1}p_j\Big)$$

and

$$R(j,i+1) = \sum_{k=j-1}^{i} \Big(R(j-1,k) \cdot (1-p_j)^{i-k}p_j\Big)$$

Hence, we get

$$R(j,i+1) = p_j R(j-1,i) + (1-p_j)R(j,i) \tag{3.5}$$

Denote by $\Delta R(j,i)$ the difference $R(j,i) - R(j,i-1)$. From Equation 3.5, we have

$$\Delta R(j,i+1) = p_j \Delta R(j-1,i) + (1-p_j)\Delta R(j,i) \tag{3.6}$$

For any $j \geq 1$, we have $R(j,1) \geq R(j,0) = 0$ or $\Delta R(j,1) \geq 0$. From the definition of $N(j)$, $N(j)$ must be at least j and for any $i \leq N(j-1)$, the difference $\Delta R(j-1,i)$ is non-negative. Hence, if $\Delta R(j,i)$ is non-negative, then $\Delta R(j,i+1)$ is non-negative for Equation 3.6. So inductively, for any $i \leq N(j-1)+1$, we always have $\Delta R(j,i) \geq 0$. Recall that we define the mode of a distribution with multiple modes as the one with the largest index, thus $N(j) > N(j-1)$. ■

With the strong unimodality of distribution $\{R(j,i)\}$, we are able to present the PAC learnability of concept class III in the statistical query model.

Proof (of Theorem 3.1) From Lemma 3.3, statistical query $\chi_{V,a}$ is legitimate and feasible at tolerance $\tau = \epsilon^2/(40sn^2 + 4\epsilon)$ and our error parameter must be set to $\epsilon/2$ in order to have Inequality 3.2.

We modify the statistical query algorithm to make an initial statistical query with tolerance $\tau = \epsilon^2/(40sn^2+4\epsilon)$ to estimate $\mathbb{P}(y = +1)$. If the answer is $\leq \epsilon/2-\tau$, then $\mathbb{P}(y = +1) \leq \epsilon/2$ and the algorithm outputs a hypothesis that all examples are negative. If the answer is $\geq 1-\epsilon/2+\tau$, then $\mathbb{P}(y = +1) \geq 1-\epsilon/2$ and the algorithm outputs a hypothesis that all examples are positive.

Otherwise, $\mathbb{P}(y = +1)$ and $\mathbb{P}(y = -1)$ are both at least $\epsilon/2 - 2\tau$. We then do another statistical query at tolerance τ to estimate $\mathbb{P}(y = +1 \mid V \sqsubseteq x)$. Since $V \sqsubseteq x$ is a necessary condition of positivity, $\mathbb{P}(V \sqsubseteq x)$ must be at least $\mathbb{P}(y = +1) \geq \epsilon/2-2\tau$ and this statistical query is legitimate and feasible. If the answer is $\geq 1 - \epsilon/2 + \tau$, then $\mathbb{P}(y = +1 \mid V \sqsubseteq x) \geq 1 - \epsilon/2$. The algorithm outputs a hypothesis that all strings x such that $V \sqsubseteq x$ are positive and all strings x such that $V \not\sqsubseteq x$ are negative because $\mathbb{P}(y = -1 \mid V \not\sqsubseteq x) = 1$. If $\ell = L$, $\mathbb{P}(y = +1 \mid V \sqsubseteq x)$ must be 1 and the algorithm halts. Otherwise, $\ell < L$ and the first statistical query algorithm is used. We now show that $Q(L-1, n-1) \geq 5\tau$, establishing the bound on the query tolerance.

Let random variable I be the smallest value for which $U'[1, L]$ is a subsequence of $x'[1, I]$. Based on the definition of $R(j, i)$, we have $\mathbb{P}(I = i) = R(L, i)$. String x is a positive example if and only if $U'[1, L] \sqsubseteq x'[1, n]$, which is exactly $I \leq n$. As a consequence,

$$\mathbb{P}(y = +1) = \sum_{i=L}^{n} R(L, i) \tag{3.7}$$

From Corollary 3.2, distribution $\{R(L, i)\}$ is unimodal and assume its mode is $N(L)$. If $n \leq N(L)$ then $R(L, n)$ is at least as large as every term in the sum

$\mathbb{P}(y=+1)=\sum_{i=L}^{n} R(L,i)$. Hence we get

$$R(L,n) \geq \frac{\epsilon-4\tau}{2(n-L+1)} \geq \frac{\epsilon-4\tau}{2n} \geq \frac{5\epsilon^2}{40sn^2+4\epsilon} = 5\tau$$

If $n > N(L)$, according to Lemma 3.7, for any $j \leq L$ we have $n > N(j)$. That is, for any $j \leq L$, we have $R(j,n) \geq R(j,n+1)$.

From Equation 3.5,

$$R(j,n+1) = p_j R(j-1,n) + (1-p_j)R(j,n)$$

so

$$\begin{aligned} p_j R(j-1,n) + (1-p_j)R(j,n) \leq & R(j,n) \\ = & p_j R(j,n) + (1-p_j)R(j,n) \end{aligned}$$

We then have

$$R(j-1,n) \leq R(j,n)$$

This holds for any $j \leq L$ so $R(j,n)$ is non-decreasing with respect to j when $n > N(L)$. Inductively we get $R(L,n) \geq R(j,n)$ for any $j \leq L$.

Because $U'[1,L] \not\sqsubseteq x[1,n-1]$ is a necessary condition of $y=-1$ and

$$\mathbb{P}(U'[1,L] \not\sqsubseteq x[1,n-1]) = \sum_{j=0}^{L-1} Q(j,n-1)$$

we get

$$\sum_{j=0}^{L-1} Q(j,n-1) \geq \mathbb{P}(y=-1) \geq \frac{\epsilon-4\tau}{2}$$

Note that $R(j,n) = p_j Q(j-1,n-1)$, then

$$\sum_{j=1}^{L} \frac{R(j,n)}{p_j} \geq \frac{\epsilon-4\tau}{2}$$

Since

$$\mathbb{P}(y = +1) \geq \frac{\epsilon - 4\tau}{2} > 0$$

from Inequality 3.2, we must have $p_j \geq \epsilon/(4sn)$ for all j. Then we have

$$\frac{4sn}{\epsilon} \sum_{j=1}^{L} R(j,n) \geq \sum_{j=1}^{L} \frac{R(j,n)}{p_j} \geq \frac{\epsilon - 4\tau}{2}$$

Because $R(L,n) \geq R(j,n)$ for any $j \leq L$, we get

$$\frac{4sn}{\epsilon} LR(L,n) \geq \frac{\epsilon - 4\tau}{2}$$

and

$$R(L,n) \geq \frac{(\epsilon - 4\tau)\epsilon}{8sn^2} = \frac{5\epsilon^2}{40sn^2 + 4\epsilon} = 5\tau$$

Finally, we have

$$\begin{aligned} Q(L,n) &= (1 - p_{L+1})Q(L,n-1) + p_L Q(L-1,n-1) \\ &\geq p_L Q(L-1,n-1) = R(L,n) \geq \frac{5\epsilon^2}{40sn^2 + 4\epsilon} \end{aligned}$$

That is, $Q(L-1,n-1) \geq 5\tau$. For Lemma 3.2, we have $\tau = \epsilon^2/(40sn^2+4\epsilon)$. Inferring $\bar{\tau}$ from τ is trivial. Define general statistical query

$$\bar{\chi}_{V,a}(x,y) = \begin{cases} (y+1)/2 & \text{if } \theta_{V,a}(x) = a \\ 0 & \text{if } \theta_{V,a}(x) \neq a \end{cases} \tag{3.8}$$

Then for any a, the expected query result

$$\mathbb{E}\bar{\chi}_{V,a} = \mathbb{P}(\theta_{V,a}(x) = a) \cdot \mathbb{E}\chi_{V,a} + 0$$

and the difference between $\mathbb{E}\bar{\chi}_{V,a} \mid a \in U_{\ell+1}$ and $\mathbb{E}\bar{\chi}_{V,a} \mid a \notin U_{\ell+1}$ is $5\tau \cdot \mathbb{P}(\theta_{V,a}(x) = a)$. Hence, from Inequality 3.3,

$$\bar{\tau} = \left(1 - \frac{\epsilon}{20sn^2 + 2\epsilon}\right) \frac{\epsilon^4}{16sn(10sn^2 + \epsilon)}$$

This completes the proof. ■

3.2.3 A generalization to instance space $\Sigma^{\leq n}$

We have proved the extended class of shuffle ideals is PAC learnable from element-wise i.i.d. fixed-length strings. Nevertheless, in many real-world applications such as natural language processing and computational linguistics, it is more natural to have strings of varying lengths. Let n be the maximum length of the sample strings and as a consequence the instance space for learning is $\Sigma^{\leq n}$. Here we show how to generalize the statistical query algorithm in Section 3.2.1 to the more general instance space $\Sigma^{\leq n}$.

Let $\mathcal{A}_i$ be the algorithm in Section 3.2.1 for learning shuffle ideals from element-wise i.i.d. strings of fixed length i. Because instance space $\Sigma^{\leq n} = \bigcup_{i \leq n} \Sigma^i$, we divide the sample S into n subsets $\{S_i\}$ where $S_i = \{x \mid |x| = i\}$. An initial statistical query then is made to estimate probability $\mathbb{P}(|x| = i)$ for each $i \leq n$ at tolerance $\epsilon/(8n)$. We discard all subsets S_i with query answer $\leq 3\epsilon/(8n)$ in the learning procedure, because we know $\mathbb{P}(|x| = i) \leq \epsilon/(2n)$. There are at most $(n-1)$ such S_i of low occurrence probabilities. The total probability that an instance comes from one of these negligible sets is at most $\epsilon/2$. Otherwise, $\mathbb{P}(|x| = i) \geq \epsilon/(4n)$ and we apply algorithm $\mathcal{A}_i$ on each S_i with query answer $\geq 3\epsilon/(8n)$ with error parameter $\epsilon/2$. Because the probability of the condition is polynomially large, the algorithm is

feasible. Finally, the total error over the whole instance space will be bounded by ϵ and concept class Ш is PAC learnable from element-wise i.i.d. strings over instance space $\Sigma^{\leq n}$.

Corollary 3.3 *Under element-wise independent and identical distributions over instance space $\mathcal{I} = \Sigma^{\leq n}$, concept class Ш is approximately identifiable with $O(sn^2)$ conditional statistical queries from STAT$(\text{Ш}, \mathcal{D})$ at tolerance*

$$\tau = \frac{\epsilon^2}{160sn^2 + 8\epsilon}$$

or with $O(sn^2)$ statistical queries from STAT$(\text{Ш}, \mathcal{D})$ at tolerance

$$\bar{\tau} = \left(1 - \frac{\epsilon}{40sn^2 + 2\epsilon}\right) \frac{\epsilon^5}{512sn^2(20sn^2 + \epsilon)}$$

3.3 Learning principal shuffle ideals from Markovian strings

Markovian strings are widely studied in natural language processing and biological sequence modeling. Formally, a random string x is Markovian if the distribution of x_{i+1} only depends on the value of x_i: $\mathbb{P}(x_{i+1} \mid x_1 \ldots x_i) = \mathbb{P}(x_{i+1} \mid x_i)$ for any $i \geq 1$. If we denote by π_0 the distribution of x_1 and define $s \times s$ stochastic matrix M by $M(a_1, a_2) = \mathbb{P}(x_{i+1} = a_1 \mid x_i = a_2)$, then a random string can be viewed as a Markov chain with initial distribution π_0 and transition matrix M. We choose $\Sigma^{\leq n}$ as the instance space in this section and assume independence between the string length and the symbols in the string. We assume $\mathbb{P}(|x| = k) \geq t$ for all $1 \leq k \leq n$ and $\min\{M(\cdot, \cdot), \pi_0(\cdot)\} \geq c$ for some positive t and c. We will prove the PAC learnability of class ⨿ under this lower bound assumption. Denote by u the target pattern string

and let $L = |u|$.

3.3.1 Statistical query algorithm

Starting with empty string $v = \lambda$, the pattern string u is recovered one symbol at a time. Having recovered $v = u[1, \ell]$, we infer $u_{\ell+1}$ by $\Psi_{v,a} = \sum_{k=h+1}^{n} \mathbb{E}\chi_{v,a,k}$, where

$$\chi_{v,a,k}(x, y) = \frac{1}{2}(y + 1) \quad \text{given } I_{v \sqsubseteq x} = h,\ x_{h+1} = a \text{ and } |x| = k$$

$0 \leq \ell < L$ and h is chosen from $[0, n-1]$ such that the probability $\mathbb{P}(I_{v \sqsubseteq x} = h)$ is polynomially large. The statistical queries $\chi_{v,a,k}$ are made at tolerance τ claimed in Theorem 3.2 and the symbol with the largest query result of $\Psi_{v,a}$ is proved to be $u_{\ell+1}$. Again, the case where $\ell = L$ is addressed by query $\mathbb{P}(y = +1 \mid v \sqsubseteq x)$. The learning procedure is completed if the query result is close to 1.

3.3.2 PAC learnability

With query $\Psi_{v,a}$, we are able to recover the pattern string u approximately from $STAT(\sqcup\!\sqcup(u), \mathcal{D})$ at proper tolerance as stated in Theorem 3.2:

Theorem 3.2 *Under Markovian string distributions over instance space $\mathcal{I} = \Sigma^{\leq n}$, given $\mathbb{P}(|x| = k) \geq t > 0$ for $\forall\ 1 \leq k \leq n$ and $\min\{M(\cdot, \cdot), \pi_0(\cdot)\} \geq c > 0$, concept class $\sqcup\!\sqcup$ is approximately identifiable with $O(sn^2)$ conditional statistical queries from $STAT(\sqcup\!\sqcup, \mathcal{D})$ at tolerance*

$$\tau = \frac{\epsilon}{3n^2 + 2n + 2}$$

or with $O(sn^2)$ statistical queries from $STAT(\sqcup\!\sqcup, \mathcal{D})$ at tolerance

$$\bar{\tau} = \frac{3ctn\epsilon^2}{(3n^2 + 2n + 2)^2}$$

Due to the probability lower bound assumptions, the legitimacy and feasibility are obvious. To calculate the tolerance for PAC learning, we first consider the exact learning tolerance. Let x' be an infinite string generated by the Markov chain defined above. For any $0 \leq \ell \leq L - j$, we define quantity $R_\ell(j, i)$ to be the conditional probability

$$\mathbb{P}(u[\ell+1, \ell+j] \sqsubseteq x'[m+1, m+i] \wedge u[\ell+1, \ell+j] \not\sqsubseteq x'[m+1, m+i-1] \mid x'_m = u_\ell)$$

Intuitively, $R_\ell(j, i)$ is the probability that the smallest g such that $u[\ell+1, \ell+j] \sqsubseteq x'[m+1, m+g]$ is i, given $x'_m = u_\ell$. We have the following conclusion for the exact learning tolerance.

Lemma 3.8 *Under Markovian string distributions over instance space $\mathcal{I} = \Sigma^{\leq n}$, given $\mathbb{P}(|x| = k) \geq t > 0$ for $\forall\ 1 \leq k \leq n$ and $\min\{M(\cdot,\cdot), \pi_0(\cdot)\} \geq c > 0$, the concept class $\sqcup\!\sqcup$ is exactly identifiable with $O(sn^2)$ conditional statistical queries from STAT($\sqcup\!\sqcup, \mathcal{D}$) at tolerance*

$$\tau' = \min_{0 \leq \ell < L} \left\{ \frac{1}{3(n-h)} \sum_{k=h+1}^{n} R_{\ell+1}(L-\ell-1, k-h-1) \right\}$$

The algorithm first deals with the marginal case where $\mathbb{P}(y = +1) \leq \epsilon$ through query $\mathbb{P}(y = +1)$. If it doesn't halt, we know $\mathbb{P}(y = +1)$ is at least $(3n^2+2n)\epsilon/(3n^2+2n+2)$. We then make a statistical query $\chi'_h(x, y) = \frac{1}{2}(y+1) \cdot \mathbb{1}_{\{I_{v \sqsubseteq x} = h\}}$ for each h from ℓ to $n-1$. It can be shown that at least one h will give an answer $\geq (3n+1)\epsilon/(3n^2+2n+2)$. This implies lower bounds for $\mathbb{P}(I_{v \sqsubseteq x} = h)$ and $\mathbb{P}(y = +1 \mid I_{v \sqsubseteq x} = h)$. The former guarantees the feasibility while the latter can serve as a lower bound for the sum in Lemma 3.8 after some algebra and completes the proof.

The assumption on M and π_0 can be weakened to $M(u_{\ell+1}, u_\ell) = \mathbb{P}(x_2 = u_{\ell+1} \mid x_1 = u_\ell) \geq c$ and $\pi_0(u_1) \geq c$ for all $1 \leq \ell \leq L-1$. We first make a statistical query to estimate $M(a, u_\ell)$ for $\ell \geq 1$ or $\pi_0(a)$ for $\ell = 0$ for each symbol $a \in \Sigma$ at tolerance $c/3$. If the result is $\leq 2c/3$ then $M(a, u_\ell) \leq c$ or $\pi_0(a) \leq c$ and we won't consider symbol a at this position. Otherwise, $M(a, u_\ell) \geq c/3$ or $\pi_0(a) \geq c/3$ and the queries in the algorithm are feasible.

Corollary 3.4 *Under Markovian string distributions over instance space $\mathcal{I} = \Sigma^{\leq n}$, given $\mathbb{P}(|x| = k) \geq t > 0$ for $\forall 1 \leq k \leq n$, $\pi_0(u_1) \geq c$ and $M(u_{\ell+1}, u_\ell) \geq c > 0$ for $\forall 1 \leq \ell \leq L-1$, concept class $\sqcup\!\sqcup$ is approximately identifiable with $O(sn^2)$ conditional statistical queries from STAT$(\sqcup\!\sqcup, \mathcal{D})$ at tolerance*

$$\tau = \min\left\{\frac{\epsilon}{3n^2 + 2n + 2}, \frac{c}{3}\right\}$$

or with $O(sn^2)$ statistical queries from STAT$(\sqcup\!\sqcup, \mathcal{D})$ at tolerance

$$\bar{\tau} = \min\left\{\frac{ctn\epsilon^2}{(3n^2 + 2n + 2)^2}, \frac{tn\epsilon c^2}{3(3n^2 + 2n + 2)}\right\}$$

Now we present the complete proof.

Proof (of Lemma 3.8) If the algorithm doesn't halt, u has not been completely recovered and $\ell < L$. Again, we calculate the difference of $\Psi_{v,a}$ between the cases $a_+ = u_{\ell+1}$ and $a_- \neq u_{\ell+1}$.

For $a_- \neq u_{\ell+1}$, let p_j denote the probability that the first passage time from a_- to $u_{\ell+1}$ is equal to j. Notice that

$$\begin{aligned}\mathbb{E}\chi_{v,a_-,k} &= \sum_{j=1}^{k-h-1}\left(p_j \sum_{i=0}^{k-h-1-j} R_{\ell+1}(L-\ell-1, i)\right) \\ &\leq \sum_{j=1}^{k-h-1}\left(p_j \sum_{i=0}^{k-h-2} R_{\ell+1}(L-\ell-1, i)\right)\end{aligned}$$

We get

$$\mathbb{E}\chi_{v,a_-,k} \leq \sum_{i=0}^{k-h-2} R_{\ell+1}(L-\ell-1,i)$$

For $a_+ = u_{\ell+1}$, we have

$$\mathbb{E}\chi_{v,a_+,k} = \sum_{i=0}^{k-h-1} R_{\ell+1}(L-\ell-1,i)$$

Summing up all the items, we can get the difference

$$\begin{aligned}
\Psi_{v,a_+} - \Psi_{v,a_-} &= \sum_{k=h+1}^{n} \left(\mathbb{E}\chi_{v,a_+,k} - \mathbb{E}\chi_{v,a_-,k}\right) \\
&\geq \sum_{k=h+1}^{n} \left(\sum_{i=0}^{k-h-1} R_{\ell+1}(L-\ell-1,i) - \sum_{i=0}^{k-h-2} R_{\ell+1}(L-\ell-1,i) \right) \\
&= \sum_{k=h+1}^{n} R_{\ell+1}(L-\ell-1,k-h-1)
\end{aligned}$$

In order to distinguish the target $u_{\ell+1}$ from other symbols, the query tolerance can be set to one third of the difference so that the symbol with largest query result must be $u_{\ell+1}$. Thus the overall tolerance for $\Psi_{v,a}$ is $\sum_{k=h+1}^{n} R_{\ell+1}(L-\ell-1,k-h-1)/3$. Since $\Psi_{v,a}$ is the expectation sum of $(n-h)$ statistical queries, we can evenly distribute the overall tolerance on each $\chi_{v,a,k}$. So the final tolerance on each statistical query is

$$\tau' = \min_{0\leq\ell<L} \left\{ \frac{1}{3(n-h)} \sum_{k=h+1}^{n} R_{\ell+1}(L-\ell-1,k-h-1) \right\}$$

Taking minimum over $0 \leq \ell < L$ is because h depends on ℓ and the tolerance needs to be independent of h. As a consequence sn statistical queries for each prefix of U suffice to learn U exactly. ■

We then show how to choose a proper h from $[0, n-1]$.

Lemma 3.9 *Under Markovian string distributions over instance space* $\mathcal{I} = \Sigma^{\leq n}$, *given* $\mathbb{P}(|x| = k) \geq t > 0$ *for* $\forall 1 \leq k \leq n$ *and* $\min\{M(\cdot,\cdot), \pi_0(\cdot)\} \geq c > 0$, *the conditional statistical query* $\chi_{v,a,k}$ *is legitimate and feasible at tolerance*

$$\tau = \frac{\epsilon}{3n^2 + 2n + 2}$$

Proof First of all, the function $\chi_{v,a,k}$ computes a binary mapping from labeled examples (x, y) to $\{0, 1\}$ and satisfies the definition of a statistical query. Under the given conditions, $\chi_{v,a,k}$ returns 0 if x is a negative example ($y = -1$) or returns 1 if x is a positive example ($y = +1$).

From Lemma 3.1, evaluating the relation $v \sqsubseteq x$ and meanwhile determining $I_{v \sqsubseteq x}$ is feasible in time $O(n)$. Since $|x| \leq n$, determining $|x|$ also takes $O(n)$ time. Thus, $\chi_{v,a,k}(x, y)$ and then $\Psi_{v,a}$ can be efficiently evaluated.

According to the Markov property and the independence between string length and symbols in a string, we have

$$\begin{aligned}
&\mathbb{P}(I_{v \sqsubseteq x} = h,\ x_{h+1} = a \text{ and } |x| = k) \\
=&\mathbb{P}(I_{v \sqsubseteq x} = h) \cdot \mathbb{P}(x_{h+1} = a \mid I_{v \sqsubseteq x} = h) \cdot \mathbb{P}(|x| = k) \\
\geq&\mathbb{P}(I_{v \sqsubseteq x} = h) \cdot c \cdot t
\end{aligned}$$

The only problem left is to make sure $\mathbb{P}(I_{v \sqsubseteq x} = h)$ is polynomially large. Obviously this can't be guaranteed for all h between ℓ and $n - 1$ so h must be chosen carefully. We now show there must be such an h.

We make an initial statistical query with tolerance $\epsilon/(3n^2 + 2n + 2)$ to estimate $\mathbb{P}(y = +1)$. If the answer is $\leq (3n^2 + 2n + 1)\epsilon/(3n^2 + 2n + 2)$, then $\mathbb{P}(y = +1) \leq \epsilon$ and the algorithm outputs a hypothesis that all examples are negative. Otherwise, $\mathbb{P}(y = +1)$ is at least $(3n^2 + 2n)\epsilon/(3n^2 + 2n + 2)$, and the statistical queries $\{\chi_{v,a,k}\}$

are used. Since

$$\mathbb{P}(y=+1)=\sum_{h=\ell}^{n-1}\mathbb{P}(y=+1\wedge I_{v\sqsubseteq x}=h) \tag{3.9}$$

There must be at least one h so that

$$\begin{aligned}\mathbb{P}(y=+1\wedge I_{v\sqsubseteq x}=h)\geq&\frac{1}{n-h}\mathbb{P}(y=+1)\\ \geq&\frac{1}{n}\mathbb{P}(y=+1)\\ \geq&\frac{1}{n}\cdot\frac{(3n^2+2n)\epsilon}{3n^2+2n+2}\\ =&\frac{(3n+2)\epsilon}{3n^2+2n+2}\end{aligned}$$

As

$$\mathbb{P}(y=+1\wedge I_{v\sqsubseteq x}=h)=\mathbb{P}(y=+1\mid I_{v\sqsubseteq x}=h)\cdot\mathbb{P}(I_{v\sqsubseteq x}=h)$$

both $\mathbb{P}(y=+1\mid I_{v\sqsubseteq x}=h)$ and $\mathbb{P}(I_{v\sqsubseteq x}=h)$ must be at least $(3n+2)\epsilon/(3n^2+2n+2)$. This means there must be some h making our statistical queries legitimate.

We now show how to determine a proper value of h. We can do a statistical query

$$\chi'_h(x,y)=\frac{1}{2}(y+1)\cdot\mathbb{1}_{\{I_{v\sqsubseteq x}=h\}} \tag{3.10}$$

for each h from ℓ to $n-1$, where $\mathbb{1}_{\{\pi\}}$ represents the 0-1 truth value of the predicate π. It is easy to see $\mathbb{E}\chi'_h=\mathbb{P}(y=+1\wedge I_{v\sqsubseteq x}=h)$. According to our analysis above and due to the noise of the statistical query, there must be at least one h such that the answer is $\geq(3n+1)\epsilon/(3n^2+2n+2)$. If we choose such an h, it is guaranteed to have

$$\mathbb{P}(y=+1\wedge I_{v\sqsubseteq x}=h)\geq\frac{3n\epsilon}{3n^2+2n+2}$$

so that

$$\mathbb{P}(I_{v\sqsubseteq x} = h) \geq \frac{3n\epsilon}{3n^2 + 2n + 2}$$

and

$$\mathbb{P}(y = +1 \mid I_{v\sqsubseteq x} = h) \geq \frac{3n\epsilon}{3n^2 + 2n + 2} \tag{3.11}$$

After at most n statistical queries $\{\chi'_h\}$, we can determine the value of h in query $\chi_{v,a,k}$. Thus statistical queries $\{\chi_{v,a,k}\}$ and $\Psi_{v,a}$ are legitimate and feasible. ■

Below is the proof of Theorem 3.2.

Proof (of Theorem 3.2) From Lemma 3.9, statistical queries $\{\chi_{v,a,k}\}$ and $\Psi_{v,a}$ are legitimate and feasible at tolerance $\epsilon/(3n^2 + 2n + 2)$.

We modify the statistical query algorithm to make an initial statistical query with tolerance $\epsilon/(3n^2 + 2n + 2)$ to estimate $\mathbb{P}(y = +1)$. If the answer is $\leq (3n^2 + 2n + 1)\epsilon/(3n^2 + 2n + 2)$, then $\mathbb{P}(y = +1) \leq \epsilon$ and the algorithm outputs a hypothesis that all examples are negative. Otherwise, $\mathbb{P}(y = +1)$ is at least $(3n^2+2n)\epsilon/(3n^2+2n+2)$.

We then do another statistical query with tolerance $\epsilon/(3n^2 + 2n + 2)$ to estimate $\mathbb{P}(y = +1 \mid v \sqsubseteq x)$. Since $v \sqsubseteq x$ is a necessary condition of positivity, $\mathbb{P}(v \sqsubseteq x)$ must be at least $\mathbb{P}(y = +1) \geq (3n^2 + 2n)\epsilon/(3n^2 + 2n + 2)$ and this statistical query is legitimate and feasible. If the answer is $\geq 1 - (3n^2 + 2n)\epsilon/(3n^2 + 2n + 2)$, then $\mathbb{P}(y = +1 \mid v \sqsubseteq x) \geq 1 - \epsilon$. The algorithm outputs a hypothesis that all strings x such that $v \sqsubseteq x$ are positive and all strings x such that $v \not\sqsubseteq x$ are negative because $\mathbb{P}(y = -1 \mid v \not\sqsubseteq x) = 1$. If $\ell = L$, $\mathbb{P}(y = +1 \mid v \sqsubseteq x)$ must be 1 and the algorithm halts. Otherwise, $\ell < L$ and the first statistical query algorithm is used.

From the proof for Lemma 3.9, we then use $O(n)$ statistical queries

$$\chi'_h(x, y) = \frac{1}{2}(y + 1) \cdot \mathbb{1}_{\{I_{v\sqsubseteq x} = h\}}$$

to find an h such that Inequality 3.11 holds:

$$\mathbb{P}(y = +1 \mid I_{v \sqsubseteq x} = h) \geq \frac{3n\epsilon}{3n^2 + 2n + 2}$$

Similarly, let q_j denote the probability that the first passage time from u_ℓ to $u_{\ell+1}$ is equal to j. Notice that

$$\mathbb{P}(y = +1 \mid I_{v \sqsubseteq x} = h) \leq \sum_{j=1}^{n-h} \left(q_j \sum_{i=0}^{n-h-j} R_{\ell+1}(L - \ell - 1, i) \right)$$

We have

$$\begin{aligned}
\frac{3n\epsilon}{3n^2 + 2n + 2} \leq & \mathbb{P}(y = +1 \mid I_{v \sqsubseteq x} = h) \\
& \leq \sum_{j=1}^{n-h} \left(q_j \sum_{i=0}^{n-h-j} R_{\ell+1}(L - \ell - 1, i) \right) \\
& \leq \sum_{j=1}^{n-h} \left(q_j \sum_{i=0}^{n-h-1} R_{\ell+1}(L - \ell - 1, i) \right) \\
& \leq \sum_{i=0}^{n-h-1} R_{\ell+1}(L - \ell - 1, i) \\
& = \sum_{k=h+1}^{n} R_{\ell+1}(L - \ell - 1, k - h - 1)
\end{aligned}$$

From Lemma 3.8, the conditional tolerance is

$$\tau = \min_{0 \leq \ell < L} \left\{ \frac{1}{3(n-h)} \sum_{k=h+1}^{n} R_{\ell+1}(L - \ell - 1, k - h - 1) \right\} \geq \frac{\epsilon}{3n^2 + 2n + 2}$$

Similar to the proof of Theorem 3.1, define general statistical query

$$\bar{\chi}_{v,a,k}(x, y) = \begin{cases} (y+1)/2 & \text{if } I_{v \sqsubseteq x} = h,\ x_{h+1} = a \text{ and } |x| = k \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

and

$$\bar{\Psi}_{v,a} = \sum_{k=h+1}^{n} \mathbb{E}\bar{\chi}_{v,a,k} \tag{3.13}$$

Then the general tolerance $\bar{\tau}$ can be easily inferred from the conditional tolerance τ:

$$\bar{\tau} = \frac{3ctn\epsilon^2}{(3n^2+2n+2)^2}$$

Considering we have used n statistical queries to determine h, $(s+1)n$ statistical queries for each prefix of u suffice to PAC learn u. This completes the proof. ■

3.4 A constrained generalization to learning shuffle ideals under product distributions

A direct generalization from element-wise independent and identical distributions is product distributions. A random string, or a random vector of symbols under a product distribution has element-wise independence between its elements. That is, $\mathbb{P}(X = x) = \prod_{i=1}^{|x|} \mathbb{P}(X_i = x_i)$. Although strings under product distributions share many independence properties with element-wise i.i.d. strings, the algorithm in Section 3.2.1 is not directly applicable to this case as the distribution $\{R(j,i)\}$ defined above is not unimodal with respect to i in general. However, the intuition that given $I_{V \sqsubseteq x} = h$, the strings with $x_{h+1} \in U_{\ell+1}$ have higher probability of positivity than that of the strings with $x_{h+1} \notin U_{\ell+1}$ is still true under product distributions. Thus we generalize query $\chi_{V,a}$ and define for any $V \in (\Sigma^{\cup})^{\leq n}$, $a \in \Sigma$ and $h \in [0, n-1]$,

$$\tilde{\chi}_{V,a,h}(x,y) = \frac{1}{2}(y+1) \quad \text{given } I_{V \sqsubseteq x} = h \text{ and } x_{h+1} = a$$

where $y = c(x)$ is the label of example string x. To ensure the legitimacy and feasibility of the algorithm, we have to attach a lower bound assumption that $\mathbb{P}(x_i = a) \geq t > 0$, for $\forall 1 \leq i \leq n$ and $\forall a \in \Sigma$. This section provides a constrained algorithm based on this intuition. Let $P(+|a, h)$ denote $\mathbb{E}\tilde{\chi}_{V,a,h}$. If the difference $P(+|a_+, h) - P(+|a_-, h)$ is large enough for some h with nonnegligible $\mathbb{P}(I_{V \sqsubseteq x} = h)$, then we are able to learn the next element in U. Otherwise, the difference is very small and we will show that there is an interval starting from index $(h + 1)$ which we can skip with little risk. The algorithm is able to classify any string whose classification process skips $O(1)$ intervals.

Again the algorithm uses query $\mathbb{P}(y = +1 \mid V \sqsubseteq x)$ to tell whether it is time to halt. As before, let V be the partial pattern we have learned and the algorithm starts with $V = \lambda$. For $1 \leq i \leq n$ and $1 \leq j \leq L$, define probability $\tilde{Q}(j, i)$ as below.

$$\tilde{Q}(j,i) = \begin{cases} \text{if } 1 \leq j < L: \\ \mathbb{P}(U[L-j+1, L] \sqsubseteq x[n-i+1, n] \wedge U[L-j, L] \not\sqsubseteq x[n-i+1, n]) \\ \text{if } j = L: \\ \mathbb{P}(U \sqsubseteq x[n-i+1, n]) \end{cases}$$

Lemma 3.10 *Under product distributions over instance space $\mathcal{I} = \Sigma^n$, given $\mathbb{P}(x_i = a) \geq t > 0$ for $\forall 1 \leq i \leq n$ and $\forall a \in \Sigma$, concept class III is exactly identifiable with $O(sn)$ conditional statistical queries from $STAT(\text{III}, \mathcal{D})$ at tolerance*

$$\tau' = \frac{1}{5} \min \left\{ \tilde{Q}(L-1, n-1), \min_{1 \leq \ell \leq L} \max_{\ell \leq h \leq n-1} \tilde{Q}(L-\ell-1, n-h-1) \right\}$$

Proof If the algorithm doesn't halt, U has not been completely recovered and $\ell < L$. As before, we calculate the difference of $\mathbb{E}\tilde{\chi}_{V,a,h}$ between the cases $a_+ \in U_{\ell+1}$ and

$a_- \notin U_{\ell+1}$.

When $\ell = 0$ and $V = \lambda$, the value of $I_{V \sqsubseteq x}$ must be 0 so h is fixed to be 0 in the query. For symbol $a_+ \in U_1$, we have

$$\mathbb{E}\tilde{\chi}_{\lambda,a_+,0} = \tilde{Q}(L-1, n-1) + \tilde{Q}(L, n-1)$$

and for symbol $a_- \notin U_1$,

$$\mathbb{E}\tilde{\chi}_{\lambda,a_-,0} = \tilde{Q}(L, n-1)$$

Taking one fifth of the difference gives the tolerance $\tilde{Q}(L-1, n-1)/5$ for $\ell = 0$.

When $1 \leq \ell < L$ and $V = U[1, \ell]$, we have for symbol $a_+ \in U_{\ell+1}$,

$$\mathbb{E}\tilde{\chi}_{V,a_+,h} = \sum_{j=L-\ell-1}^{L} \tilde{Q}(j, n-h-1)$$

and for symbol $a_- \notin U_{\ell+1}$,

$$\mathbb{E}\tilde{\chi}_{V,a_-,h} = \sum_{j=L-\ell}^{L} \tilde{Q}(j, n-h-1)$$

Again taking one fifth of the difference gives the tolerance $\tilde{Q}(L-\ell-1, n-h-1)/5$. For a fixed $1 \leq \ell < L$, tolerance $\max_{\ell \leq h \leq n-1} \tilde{Q}(L-\ell-1, n-h-1)/5$ is enough to learn $U_{\ell+1}$ exactly. Taking the minimum tolerance among all $0 \leq \ell < L$ gives the overall tolerance in the statement. As a consequence s statistical queries for each prefix of U suffice to learn U exactly. ■

A more complicated algorithm is needed to PAC learn shuffle ideals under product

distributions. We first define two additional simple queries:

$$\chi'_{V,a,h,i}(x,y) = \mathbb{1}_{\{x_{h+i}=a\}} \quad \text{given } I_{V\sqsubseteq x} = h$$
$$\chi^{+}_{V,a,h,i}(x,y) = \mathbb{1}_{\{x_{h+i}=a\}} \quad \text{given } I_{V\sqsubseteq x} = h \text{ and } y = +1$$

whose expectations serve as empirical estimators for the distributions of the symbol at the next i-th position over all strings ($\chi'_{V,a,i}$) and over all positive strings ($\chi^{+}_{V,a,i}$), both conditioned on $I_{V\sqsubseteq x} = h$. Below is how the algorithm works, with $\bar{\epsilon}_{g+1}$ and ϵ' to be decided later in the proof.

First an initial query to estimate probability $\mathbb{P}(y = +1 \mid V \sqsubseteq x)$ is made. The algorithm will classify all strings such that $V \sqsubseteq x$ negative if the answer is close to 0, or positive if the answer is close to 1. To ensure the legitimacy and feasibility of the algorithm, we make another initial query to estimate the probability $\mathbb{P}(I_{V\sqsubseteq x} = h)$ for each h. The algorithm then excludes the low-probability cases such that any of the excluded ones happens with probability lower than $\epsilon/2$. Thus we only need to consider the cases with polynomially large $\mathbb{P}(I_{V\sqsubseteq x} = h)$ and learn the target ideal within error $\epsilon/2$. Otherwise, let $P(+|a,h)$ denote $\mathbb{E}\tilde{\chi}_{V,a,h}$ and we make a statistical query to estimate $P(+|a,h)$ for each $a \in \Sigma$. If the difference $P(+|a_+,h) - P(+|a_-,h)$, where a_+ is in the next element of U and a_- is not, is large enough for some h, then the results of queries for $P(+|a,h)$ will form two distinguishable clusters, where the maximum difference inside one cluster is smaller than the minimum gap between them, so that we are able to learn the next element in U.

Otherwise, for all h with nonnegligible $\mathbb{P}(I_{V\sqsubseteq x} = h)$, the difference $P(+|a_+,h) - P(+|a_-,h)$ is very small and we will show that there is an interval starting from index $h+1$ which we can skip with little risk for each case when $I_{V\sqsubseteq x} = h$. Problematic cases leading to misclassification will happen with very small probability within this interval. We are safe to skip the whole interval and move on. The remaining problem

is to identify the length of this interval, that is, to estimate the probability that an error happens if we skip an interval. Let $\mathcal{D}_{1:k}(h)$ be the distribution of $x[h+1, h+k]$ over all strings given $I_{V \sqsubseteq x} = h$ and $\mathcal{D}^+_{1:k}(h)$ be the corresponding distribution over all positive strings given $I_{V \sqsubseteq x} = h$. The probability that an error happens due to skipping the next k elements is the total variation distance between $\mathcal{D}_{1:k}(h)$ and $\mathcal{D}^+_{1:k}(h)$. Thanks to the independence between the elements in a string, it can be proved that $\|\mathcal{D}_{1:k}(h) - \mathcal{D}^+_{1:k}(h)\|_{TV}$ can be estimated within polynomially bounded error. Recall that the total variation distance $\|\cdot\|_{TV}$ between two distributions μ_1 and μ_2 is

$$\|\mu_1 - \mu_2\|_{TV} = \frac{1}{2}\|\mu_1 - \mu_2\|_1 = \min_{(Y,Z)} \mathbb{P}(Y \neq Z)$$

where $Y \sim \mu_1$ and $Z \sim \mu_2$ are random variables over μ_1 and μ_2 respectively. The minimum is taken over all joint distributions (Y, Z) such that the marginal distributions are still μ_1 and μ_2, i.e., $Y \sim \mu_1$ and $Z \sim \mu_2$.

Because the lengths of skipped intervals in cases with different $I_{V \sqsubseteq x}$ could be different, the algorithm branches the classification tree to determine the skipped interval according to the value of $I_{V \sqsubseteq x}$. The algorithm runs the procedure above recursively on each branch. Figure 3.3 demonstrates this skipping strategy of the algorithm, where parameter C is the maximum allowed number of skipped intervals on each path. Notice that the algorithm might not recover the complete pattern string U. Instead the hypothesis pattern string returned by the algorithm for one classification path is a subsequence of U with skipped intervals. We provide a toy example to explain the skipping logic. Let $n = 4$, $\Sigma = \{$a, b, c$\}$ and $U =$ 'ab'. Strings are drawn from a product distribution such that x_1, x_2 and x_4 are uniformly distributed over Σ but x_2 is almost surely 'a'. The algorithm first estimates $\mathbb{P}(y = +1 \mid x_1 = a)$ for each $a \in \Sigma$ and finds the value of x_1 matters little to the positivity.

1. Estimate probability $\mathbb{P}(y = +1 \mid V \sqsubseteq x)$ at tolerance $\epsilon'/3$. If the answer is $\leq 2\epsilon'/3$, classify all strings x such that $V \sqsubseteq x$ as negative and backtrack on the classification tree. If the answer is $\geq 1 - 2\epsilon'/3$, classify all strings x such that $V \sqsubseteq x$ as positive and backtrack. If the number of intervals skipped on the current path exceeds C, classify all strings x such that $V \sqsubseteq x$ as positive and backtrack. Otherwise go to Step 2.
2. For each h with nonnegligible $\mathbb{P}(I_{V \sqsubseteq x} = h)$, estimate $\mathbb{E}\chi_{V,a,h}$ at tolerance $\tau_1 = \bar{\epsilon}_{g+1}^2/384$ for each $a \in \Sigma$. Go to Step 3.
3. If the results for some h produce two distinguishable clusters, where the maximum difference inside one cluster is $\leq 4\tau_1$ while the minimum gap between two clusters is $> 4\tau_1$, then the set of all the symbols that belong to the cluster with larger query results is the next element in U. Update V and go to Step 1. Otherwise, branch the classification tree. For each h, let $k \leftarrow 1$ and $T \leftarrow 1$. Go to Step 4.
4. For each $a \in \Sigma$, estimate $\mathbb{E}\chi'_{V,a,h,k}$ and $\mathbb{E}\chi^+_{V,a,h,k}$ at tolerance $\tau_2 = \bar{\epsilon}_{g+1}/(8sn)$ so that we will have estimators $\widehat{\mathcal{D}}_k(h)$ and $\widehat{\mathcal{D}}_k^+(h)$. Go to Step 5.
5. $T \leftarrow (1 - \|\widehat{\mathcal{D}}_k(h) - \widehat{\mathcal{D}}_k^+(h)\|_{TV}) \cdot T$. If $1 - T \leq 3\bar{\epsilon}_{g+1}/4$, $k \leftarrow k + 1$ and go to Step 4. Otherwise, skip the interval from x_{h+1} to x_{h+k-1}. Update V and go to Step 1.

Figure 3.3: Approximately learning III under product distributions

It then estimates the distance between the distribution of x_1x_2 over all positive strings and that over all strings and finds the two distributions are close. However, when it moves on to estimate the distance between the distribution of $x_1x_2x_3$ over all positive strings and that over all strings, it gets a nonnegligible total variation distance. Therefore, the skipped interval is x_1x_2. The algorithm finally outputs the hypothesis pattern string 'ΣΣb' which means skipping the first two symbols and matching symbol 'b' in the rest of the string.

Theorem 3.3 *Under product distributions over instance space $\mathcal{I} = \Sigma^n$, given $\mathbb{P}(x_i = a) \geq t > 0$ for $\forall 1 \leq i \leq n$ and $\forall a \in \Sigma$, the algorithm PAC classifies any string that skips $C = O(1)$ intervals during the classification procedure with $O(sn^{C+2})$ condi-*

tional statistical queries from STAT$(\text{Ш}, \mathcal{D})$ *at tolerance*

$$\tau = \min\left\{\frac{\bar{\epsilon}_1^2}{384}, \frac{\bar{\epsilon}_1}{8sn}\right\}$$

or with $O(sn^{C+2})$ *statistical queries from STAT*$(\text{Ш}, \mathcal{D})$ *at tolerance*

$$\bar{\tau} = (\epsilon' - 2\tau) \cdot \min\left\{\frac{t\bar{\epsilon}_1^2}{384}, \frac{\bar{\epsilon}_1}{8sn}\right\}$$

where $\bar{\epsilon}_1 = (\epsilon'/3^{C+2})^{2^C}$ *and* $\epsilon' = \epsilon/(2n^C)$.

Proof For the sake of the legitimacy and feasibility of the algorithm, we make an initial query to estimate the probability $\mathbb{P}(I_{V\sqsubseteq x} = h)$ for each h at tolerance τ. Denote $\epsilon' = \epsilon/(2n^C)$. If the answer is $\leq \epsilon' - \tau$, then $\mathbb{P}(I_{V\sqsubseteq x} = h) \leq \epsilon'$ is negligible and we won't consider such cases because any of them happens with probability $\leq \epsilon/2$. Otherwise we have $\mathbb{P}(I_{V\sqsubseteq x} = h) \geq \epsilon' - 2\tau$. With the lower bound assumption that $\mathbb{P}(x_i = a) \geq t > 0$ for $\forall 1 \leq i \leq n$ and $\forall a \in \Sigma$, the legitimacy and feasibility are assured. Thus bounding the classification error in the nonnegligible cases within $\epsilon/2$ establishes a total error bound ϵ. Because there are at most n^C nonnegligible cases, the problem reduces to bounding the classification error for each within ϵ'.

In the learning procedure, the algorithm skips an interval $x[i_1, i_2]$ given $I_{V\sqsubseteq x} = h$ based on the assumption that the interval $x[i_1, i_2]$ matches some segment next to V in the pattern string U. Let ι_g be the indicator for the event that the assumption is false in the first g skipped intervals and denote probability $\epsilon_g = \mathbb{E}\iota_g$. Let $\epsilon_0 = 0$. Note that ϵ_g serves as an upper bound for the probability of misclassification due to skipping the first g intervals, because there are some lucky cases where the assumption doesn't hold but the algorithm still makes correct classifications. To ensure the accuracy of the algorithm, it suffices to prove ϵ_g is small. Let $\bar{\epsilon}_{g+1} = 8\sqrt{3\epsilon_g}$ for $g \geq 1$ and $\bar{\epsilon}_1$ as

defined in the theorem. We will prove $\epsilon_{g+1} \leq \bar{\epsilon}_{g+1}$ so that by induction and taking the minimum tolerance among all $g \leq C$ we then have the overall tolerances τ and $\bar{\tau}$ as claimed in the statement.

Let a_+, a'_+ be two (not necessarily distinct) symbols in the next element of U and a_-, a'_- be two (not necessarily distinct) symbols not in the next element of U. We have $|P(+|a_+, h) - P(+|a'_+, h)| \leq \epsilon_g$ and likewise $|P(+|a_-, h) - P(+|a'_-, h)| \leq \epsilon_g$. Let $P_i(+|a, h) = P(+|a, h, \iota_g = i)$ and denote $\Delta = P(+|a_+, h) - P(+|a_-, h)$ and $\Delta_i = P_i(+|a_+, h) - P_i(+|a_-, h)$ for $i \in \{0, 1\}$. As a consequence, $\Delta = \epsilon_g \Delta_1 + (1-\epsilon_g)\Delta_0$ and $\Delta_0 = \frac{\Delta - \epsilon_g \Delta_1}{1-\epsilon_g} \geq \frac{\Delta - \epsilon_g}{1-\epsilon_g}$. Therefore, $\Delta > \epsilon_g$ implies $\Delta_0 > 0$. In the other direction, $\Delta_0 = \frac{\Delta - \epsilon_g \Delta_1}{1-\epsilon_g} \leq 2(\Delta + \epsilon_g)$.

For each h we make a statistical query to estimate $P(+|a, h)$ for each $a \in \Sigma$ at tolerance $\tau_1 = \bar{\epsilon}^2_{g+1}/384$. If the minimum Δ among all pairs of (a_+, a_-), denoted by $\Delta_{\min}$, is $> 6\tau_1$, the results of queries for $P(+|a, h)$ must form two distinguishable clusters, where the maximum difference inside one cluster is $\leq 4\tau_1$ while the minimum gap between two clusters is $> 4\tau_1$. According to Lemma 3.10, the set of symbols with larger query answers is the next element in U because $\Delta > \epsilon_g$ holds for all pairs of (a_+, a_-).

Otherwise, the difference $\Delta_0 \leq 2(\Delta_{\min} + 2\epsilon_g + \epsilon_g) \leq \bar{\epsilon}^2_{g+1}/16$ for all h. Let $x' = xz$ where z is an infinite string under the uniform distribution. Let $E_h(1, i)$ be the event that matching the next element in U consumes exactly i symbols in string x' given $I_{V \sqsubseteq x'} = h$ and $\iota_g = 0$. Define probability $R_h(1, i) = \mathbb{P}(E_h(1, i))$. Let conditional probability $P_0(+|E_h(1, i))$ be the probability of positivity conditioned on event $E_h(1, i)$. For example, $P_0(+|a_+, h)$ is indeed $P_0(+|E_h(1, 1))$.

Denote by $P_0(+|h) = \mathbb{P}(y = +1 \mid I_{V \sqsubseteq x} = h \wedge \iota_g = 0)$. Because $P_0(+|h) \geq$

$P_0(+|a_-, h)$, we have

$$P_0(+|a_+, h) - P_0(+|h) \leq P_0(+|a_+, h) - P_0(+|a_-, h) < \frac{\bar{\epsilon}_{g+1}^2}{16}$$

while

$$P_0(+|a_+, h) - P_0(+|h) = \sum_{i=1}^{+\infty} R_h(1, i) \cdot (P_0(+|E_h(1, 1)) - P_0(+|E_h(1, i)))$$

Notice that probability $P_0(+|E_h(1, i))$ is monotonically non-increasing with respect to i. Then there must exist an integer $k \in [1, +\infty]$ such that $P_0(+|E_h(1, 1)) - P_0(+|E_h(1, i)) \leq \bar{\epsilon}_{g+1}/4$ for $\forall i \leq k$ and $P_0(+|E_h(1, 1)) - P_0(+|E_h(1, i)) \geq \bar{\epsilon}_{g+1}/4$ for $\forall i > k$. This implies

$$\begin{aligned} &\sum_{i \leq k} R_h(1, i) \left(P_0(+|E_h(1, 1)) - P_0(+|E_h(1, i))\right) \\ &+ \sum_{i > k} R_h(1, i) \left(P_0(+|E_h(1, 1)) - P_0(+|E_h(1, i))\right) \\ <& \frac{\bar{\epsilon}_{g+1}^2}{16} \end{aligned}$$

and

$$\frac{\bar{\epsilon}_{g+1}}{4} \sum_{i > k} R_h(1, i) < \frac{\bar{\epsilon}_{g+1}^2}{16}$$

Then we have $\sum_{i>k} R_h(1, i) < \bar{\epsilon}_{g+1}/4$. This means the next element in U almost surely shows up in this k-length interval. In addition, the difference $P_0(+|E_h(1, 1)) - P_0(+|E_h(1, i)) \leq \bar{\epsilon}_{g+1}/4$ for $\forall i \leq k$ means whether the next element in U first shows up at x_{h+1} or x_{h+k} has little effect on the probability of positivity. There are two cases where an error happens due to skipping the interval. The first case is that the next element in U doesn't occur within the interval, whose probability is $\sum_{i>k} R_h(1, i)$. The second case is that after matching the next element in U at x_{h+i} for some

$1 \leq i < k$, the value of $x[h+i+1, h+k]$ flips the class of the string. This happens with probability $\leq P_0(+|E_h(1,1)) - P_0(+|E_h(1,k))$. By union bound, the probability of the errors because of skipping the interval $x[h+1, h+k]$ is at most $\bar{\epsilon}_{g+1}/2$.

It is worth pointing out that k is an integer from 1 to $+\infty$ because when $i = 1$ the difference $P_0(+|E_h(1,1)) - P_0(+|E_h(1,i))$ is $0 \leq \bar{\epsilon}_{g+1}/4$ and surely $k \geq 1$. This means this interval is not empty and ensures the existence of the interval we want. On the other hand, the value k can be positive infinity but this makes no difference because the algorithm will skip everything until the end of a string.

After showing the existence of such an interval, we need to determine k and locate the interval. Let $\mathcal{D}_k(h)$ be the distribution of x_{h+k} and $\mathcal{D}_{1:k}(h)$ be the distribution of the $x[h+1, h+k]$ over all strings, both conditioned on $I_{V \sqsubseteq x} = h$. Also, let $\mathcal{D}^+_k(h)$ and $\mathcal{D}^+_{1:k}(h)$ be the corresponding distributions over all positive strings. We use $\hat{\cdot}$ as estimators for probabilities or distributions. The probability that an error happens due to skipping the next k letters is the total variation distance between $\mathcal{D}_{1:k}(h)$ and $\mathcal{D}^+_{1:k}(h)$.

Now let $Y \sim \mathcal{D}_{1:k}(h)$ and $Z \sim \mathcal{D}^+_{1:k}(h)$ be random strings over $\mathcal{D}_{1:k}(h)$ and $\mathcal{D}^+_{1:k}(h)$ respectively. Then

$$
\begin{aligned}
\|\mathcal{D}_{1:k}(h) - \mathcal{D}^+_{1:k}(h)\|_{TV} &= \min_{(Y,Z)} \mathbb{P}(Y \neq Z) \\
&= 1 - \max_{(Y,Z)} \mathbb{P}(Y = Z) \\
&= 1 - \max_{(Y,Z)} \prod_{i=1}^{k} \mathbb{P}(Y_i = Z_i) \\
&= 1 - \prod_{i=1}^{k} \max_{(Y,Z)} \mathbb{P}(Y_i = Z_i) \\
&= 1 - \prod_{i=1}^{k} \left(1 - \min_{(Y,Z)} \mathbb{P}(Y_i \neq Z_i)\right) \\
&= 1 - \prod_{i=1}^{k} \left(1 - \|\mathcal{D}_i(h) - \mathcal{D}^+_i(h)\|_{TV}\right)
\end{aligned}
$$

because of the independence between the symbols in a string and the fact that all minimums and maximums are taken over all joint distributions (Y, Z) such that the marginal distributions are still product distributions.

Thus we could estimate the global total variation distance $\|\mathcal{D}_{1:k}(h) - \mathcal{D}^{+}_{1:k}(h)\|_{TV}$ through estimating the local variation distance $\|\mathcal{D}_i(h) - \mathcal{D}^{+}_i(h)\|_{TV}$ for each $1 \leq i \leq k$. Assume $\widehat{p}_1$ and $\widehat{p}_2$ are estimates of two probabilities p_1 and p_2 from a statistical query at some tolerance τ_0. We have

$$\begin{aligned}
|p_1 p_2 - \widehat{p}_1 \widehat{p}_2| &= |p_1 p_2 - p_1 \widehat{p}_2 + p_1 \widehat{p}_2 - \widehat{p}_1 \widehat{p}_2| \\
&= |p_1 (p_2 - \widehat{p}_2) + (p_1 - \widehat{p}_1) \widehat{p}_2| \\
&\leq p_1 |p_2 - \widehat{p}_2| + |p_1 - \widehat{p}_1| \widehat{p}_2 \\
&\leq (p_1 + \widehat{p}_2) \tau_0 \leq 2\tau_0
\end{aligned}$$

By induction it can be proved that $\left| \prod_{i=1}^{k} p_i - \prod_{i=1}^{k} \widehat{p}_i \right| \leq k\tau_0$, which is a polynomial bound. For a probability q, let q_i be the corresponding probability conditioned on $\iota_g = i$ for $i \in \{0, 1\}$. We have $q = \epsilon_g q_1 + (1 - \epsilon_g) q_0$ and

$$q_0 = \frac{q - \epsilon_g q_1}{1 - \epsilon_g} \geq q - \epsilon_g q_1 \geq q - \epsilon_g$$

In the other direction,

$$\begin{aligned}
q_0 = \frac{q - \epsilon_g q_1}{1 - \epsilon_g} &= \frac{q + \epsilon_g - \epsilon_g^2 - \epsilon_g q - \epsilon_g + \epsilon_g^2 + \epsilon_g q - \epsilon_g q_1}{1 - \epsilon_g} \\
&= \frac{(q + \epsilon_g)(1 - \epsilon_g) - \epsilon_g (1 + q_1 - \epsilon_g - q)}{1 - \epsilon_g} \leq q + \epsilon_g
\end{aligned}$$

Note that here without loss of generality, we assume $\epsilon \leq \min\{(n-1)t, 24/(sn)\}$ so that $1 + q_1 - \epsilon_g - q \geq (n-1)t - \epsilon_g + q_1 > 0$ and $\epsilon_g \leq \bar{\epsilon}_{g+1}^2/192 < \bar{\epsilon}_{g+1}/(8sn)$. In the PAC learning model a polynomial upper bound for the error parameter ϵ is trivial,

because if a learning algorithm works with a small error bound, it automatically guarantees larger error bounds. As a consequence, $|q - q_0| \leq \epsilon_g$. In addition, using the definition of $\|\cdot\|_{TV}$,

$$\begin{aligned}
&| \, \|\mathcal{D}_i(h) - \mathcal{D}_i^+(h)\|_{TV} - \|\widehat{\mathcal{D}}_i(h) - \widehat{\mathcal{D}}_i^+(h)\|_{TV} \, | \\
=&\frac{1}{2} \, | \, \|\mathcal{D}_i(h) - \mathcal{D}_i^+(h)\|_1 - \|\widehat{\mathcal{D}}_i(h) - \widehat{\mathcal{D}}_i^+(h)\|_1 \, | \\
\leq&\frac{1}{2} \, | \, \|\mathcal{D}_i(h) - \mathcal{D}_i^+(h) - \widehat{\mathcal{D}}_i(h) + \widehat{\mathcal{D}}_i^+(h)\|_1 \, | \\
\leq&\frac{1}{2} \left(\|\mathcal{D}_i(h) - \widehat{\mathcal{D}}_i(h)\|_1 + \|\mathcal{D}_i^+(h) - \widehat{\mathcal{D}}_i^+(h)\|_1 \right) \\
\leq&\frac{s}{2} \left(\|\mathcal{D}_i(h) - \widehat{\mathcal{D}}_i(h)\|_\infty + \|\mathcal{D}_i^+(h) - \widehat{\mathcal{D}}_i^+(h)\|_\infty \right)
\end{aligned}$$

Hence, if we make statistical queries $\chi'_{V,a,h,i}$ and $\chi^+_{V,a,h,i}$ at tolerance $\tau_2 = \bar{\epsilon}_{g+1} \cdot 1/(8sn)$ and because $\bar{\epsilon}_{g+1}/(8sn) + \epsilon_g < \bar{\epsilon}_{g+1}/(4sn)$, the noise on $\|\mathcal{D}_i(h) - \mathcal{D}_i^+(h)\|_{TV}$ will be at most $\bar{\epsilon}_{g+1}/(4n)$ and we will be able to estimate $\|\mathcal{D}_{1:k}(h) - \mathcal{D}_{1:k}^+(h)\|_{TV}$ within error $k\bar{\epsilon}_{g+1}/(4n) \leq \bar{\epsilon}_{g+1}/4$. If $\|\widehat{\mathcal{D}}_{1:k}(h) - \widehat{\mathcal{D}}_{1:k}^+(h)\|_{TV} \geq 3\bar{\epsilon}_{g+1}/4$, then $\|\mathcal{D}_{1:k}(h) - \mathcal{D}_{1:k}^+(h)\|_{TV} \geq \bar{\epsilon}_{g+1}/2$. Otherwise, $\|\mathcal{D}_{1:k}(h) - \mathcal{D}_{1:k}^+(h)\|_{TV} < \bar{\epsilon}_{g+1}$ and we are still safe to increase k.

The algorithm does $O(sn^{C+2})$ queries $\chi_{V,a,h}$ at tolerance $\tau_1 = \bar{\epsilon}_{g+1}^2/384$, plus $O(sn^{C+2})$ queries $\chi'_{V,a,h,i}$ and $\chi^+_{V,a,h,i}$ at tolerance $\tau_2 = \bar{\epsilon}_{g+1}/(8sn)$. Thus by induction and taking the minimum tolerance among all $g \leq C$ we have the overall tolerances τ and $\bar{\tau}$ as claimed in the statement. ■

3.5 Learning shuffle ideals under general distributions

Although the string distribution is restricted (or sometimes even known) in most application scenarios, one might be interested in learning shuffle ideals under general unrestricted and unknown distributions without any prior knowledge. Unfortunately, under standard complexity assumptions, the answer is negative. Angluin et al. [AAEK13] have shown that a polynomial time PAC learning algorithm for principal shuffle ideals would imply the existence of polynomial time algorithms to break the RSA cryptosystem, factor Blum integers, and test quadratic residuosity.

Theorem 3.4 [AAEK13] *For any alphabet of size at least 2, given two disjoint sets of strings $S, T \subset \Sigma^{\leq n}$, the problem of determining whether there exists a string u such that $u \sqsubseteq x$ for each $x \in S$ and $u \not\sqsubseteq x$ for each $x \in T$ is NP-complete.*

As ideal ⨿ is a subclass of ideal Ш, we know learning ideal Ш is only harder. Is the problem easier over instance space Σ^n? The answer is again no.

Lemma 3.11 *Under general unrestricted string distributions, a concept class is PAC learnable using statistical queries over instance space $\Sigma^{\leq n}$ if and only if it is PAC learnable using statistical queries over instance space Σ^n.*

The proof of the if direction of Lemma 3.11 is similar to our generalization in Section 3.2.3 from instance space Σ^n to instance space $\Sigma^{\leq n}$. The only-if direction is an immediate consequence of the fact $\Sigma^n \subseteq \Sigma^{\leq n}$.

Note that Lemma 3.11 holds under general string distributions. It is not necessarily true when we have assumptions on the marginal distribution of string length. Notice that Lemma 3.11 requires algorithm $\mathcal{A}$ to be applicable to any $S_i \mid i \leq n$.

But this requirement can be weakened. There might not exist such a general algorithm $\mathcal{A}$. Instead we could have an algorithm $\mathcal{A}_i$ applicable to each subspace S_i with non-negligible occurrence probability $\mathbb{P}(|x| = i) \geq \epsilon/(4n)$, then it is easy to see that Lemma 3.11 still holds in this case. Moreover, Lemma 3.11 makes no assumption on the string distribution. In the cases under restricted string distributions, here are two conditions that suffice to make Lemma 3.11 hold: First, there is no assumption on the string length distribution; Second, we have an algorithm $\mathcal{A}_i$ applicable to instance space S_i over marginal distribution $\mathcal{D}_{|x|=i}$ for each $1 \leq i \leq n$ such that $\mathbb{P}(|x| = i)$ is polynomially large.

Despite the infeasibility of PAC learning a shuffle ideal in theory, it is worth exploring the possibilities to do the classification problem without theoretical guarantees, since in most applications we care more about the empirical performance than about theoretical results. For this purpose we propose a heuristic greedy algorithm for learning principal shuffle ideals based on a reward strategy as follows. Upon having recovered $v = \widehat{u}[1, \ell]$, for a symbol $a \in \Sigma$ and a string x of length n, we say a consumes k elements in x if $\min\{I_{va\sqsubseteq x}, n+1\} - I_{v\sqsubseteq x} = k$. The reward strategy depends on the ratio r_+/r_-: the algorithm receives r_- reward from each element it consumes in a negative example or r_+ penalty from each symbol it consumes in a positive string. A symbol is chosen as $\widehat{u}_{\ell+1}$ if it brings us the most reward. The algorithm will halt once $\widehat{u}$ exhausts any positive example and makes a false negative error, which means we have gone too far. Finally the ideal $\sqcup\,(\widehat{u}[1, \ell-1])$ is returned as the hypothesis. The performance of this greedy algorithm depends a great deal on the selection of parameter r_+/r_-. A clever choice is $r_+/r_- = \#(-)/\#(+)$, where $\#(+)$ is the number of positive examples x such that $\widehat{u} \sqsubseteq x$ and $\#(-)$ is the number of negative examples x such that $\widehat{u} \sqsubseteq x$. A more recommended but more complex strategy to determine the parameter r_+/r_- in practice is cross validation. Figure 3.4

Input: N labeled strings $\langle x^i, y^i \rangle$, string length n, alphabet Σ
Output: pattern string $\widehat{u}$

```
1.  û ← λ
2.  for ℓ ← 0 to n
3.      reward ← a vector of 0's of length |Σ|
4.      for each a ∈ Σ
5.          for i ← 1 to N
6.              if û ⊑ x^i
7.                  if y^i = +1
8.                      reward[a] ← reward[a]+
9.                                  (I_{û⊑x^i} − min{I_{ûa⊑x^i}, n + 1})r_+
10.                 else
11.                     reward[a] ← reward[a]+
12.                                 (min{I_{ûa⊑x^i}, n + 1} − I_{û⊑x^i})r_−
13.                 endif
14.             else
15.                 if y^i = +1
16.                     return û[1, ℓ − 1]
17.                 endif
18.             endif
19.         endfor
20.     endforeach
21.     û_{ℓ+1} ← argmax_{a∈Σ}{reward[a]}
22.     û ← ûû_{ℓ+1}
23. endfor
24. return û
```

Figure 3.4: A greedy algorithm for learning a principal shuffle ideal from example oracle EX

provides detailed pseudocode for this greedy method.

A recently studied approach to learning piecewise-testable regular languages is kernel machines [KCM08, KN09]. An obvious advantage of kernel machines over our greedy method is its broad applicability to general classification learning problems. Nevertheless, the time complexity of the kernel machine is $O(N^3 + n^2N^2)$ on a training sample set of size N [BL07], while our greedy method only takes $O(snN)$ time due to its great simplicity. Because N is usually large to ensure accuracy,

kernel machines suffer from low efficiency and long running time in practice. To make a comparison between the greedy method and kernel machines for empirical performance, we conducted a series of experiments on a real world dataset [BL13] with string length n as a variable. The experiment results demonstrate the empirical advantage for both efficiency and accuracy of the greedy algorithm over the kernel method, in spite of its simplicity.

Experiment settings and results

To make a comparison between the greedy method and kernel machines for empirical performance, we conducted a series of experiments in MATLAB on a workstation built with Intel i5-2500 3.30GHz CPU and 8GB memory. As discussed in Section 3.5, the running time of the kernel machine will be very large in practice when the sample size N and the string length n are large. Also, a pattern string u of improper length will lead to a degenerate sample set which contains only positive or only negative example strings. To prevent this less interesting case from happening, we set $|u| = \lceil ns^{-1} \rceil$. Intuitively, the sample set will be evenly partitioned into two classes in expectation under the uniform distribution. However, in this case n not being large demands the alphabet size s not being large either.

Combining all these constraints together, the experiment settings are: alphabet size $s = 8$, size of training set = size of testing set = 1024. We vary the string length n from 16 to 56 and let $|u| = \lceil ns^{-1} \rceil$. The pattern string u is generated uniformly at random from $\Sigma^{|u|}$. Our tests are run on the NSF Research Award Abstracts data set [BL13]. We use the abstracts of year 1993 as the training set and those of year 1992 as the testing set. The tests are case-insensitive and all the characters except the subset from 'a(A)' to 'h(H)' are removed from the texts. The result texts are then partitioned into a set of strings of length n, which serve as the example strings.

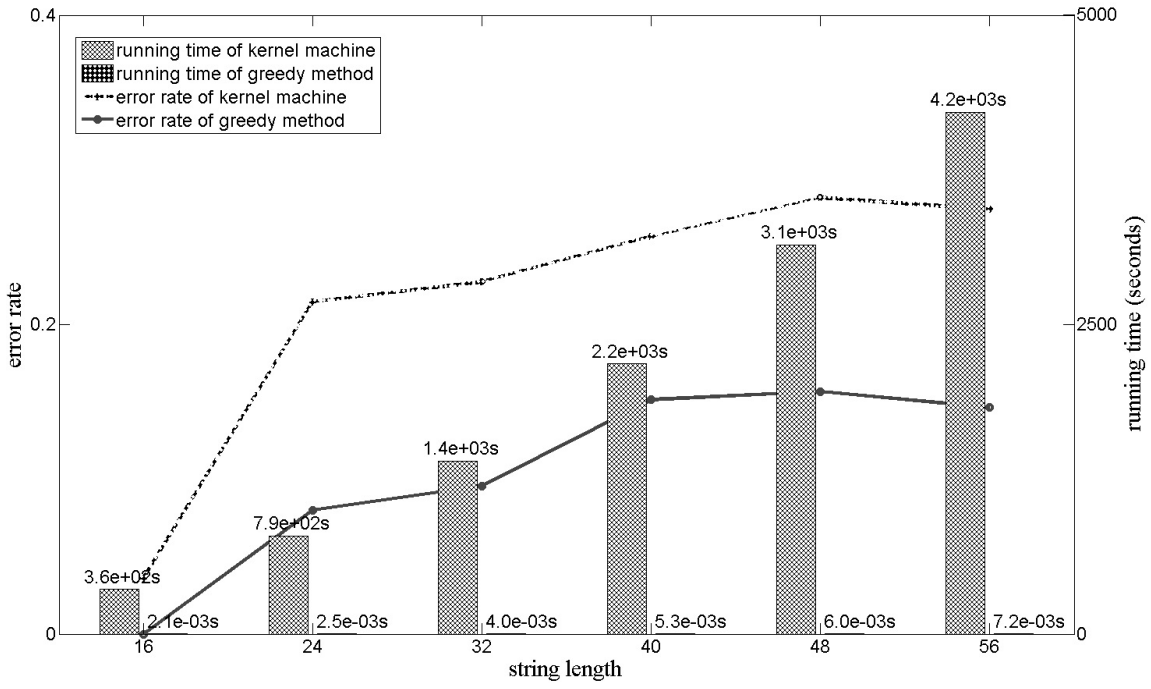

Figure 3.5: Experiment results with NSF abstracts data set (training 1993; testing 1992)

To be more robust against fluctuation from randomness, each test with a particular value of n is run for 10 times and the medians of error rates and running times are taken as the final performance scores. Both lines climb as n increases.

The experiment results are shown in Figure 3.5, with accuracy presented as line plot and efficiency demonstrated as bar chart. The overwhelming advantage of the greedy algorithm on efficiency is obvious. The kernel machine ran for hours in high dimensional cases, while the greedy method achieved even better accuracy within only milliseconds. The error rate of the greedy algorithm is always lower than that of the kernel machine as well.

It is worth noting that MATLAB started reporting a no-convergence error for the kernel method when the string length n reaches 56. Only successful runs of the kernel method were taken into account. Therefore, the performance of the kernel method

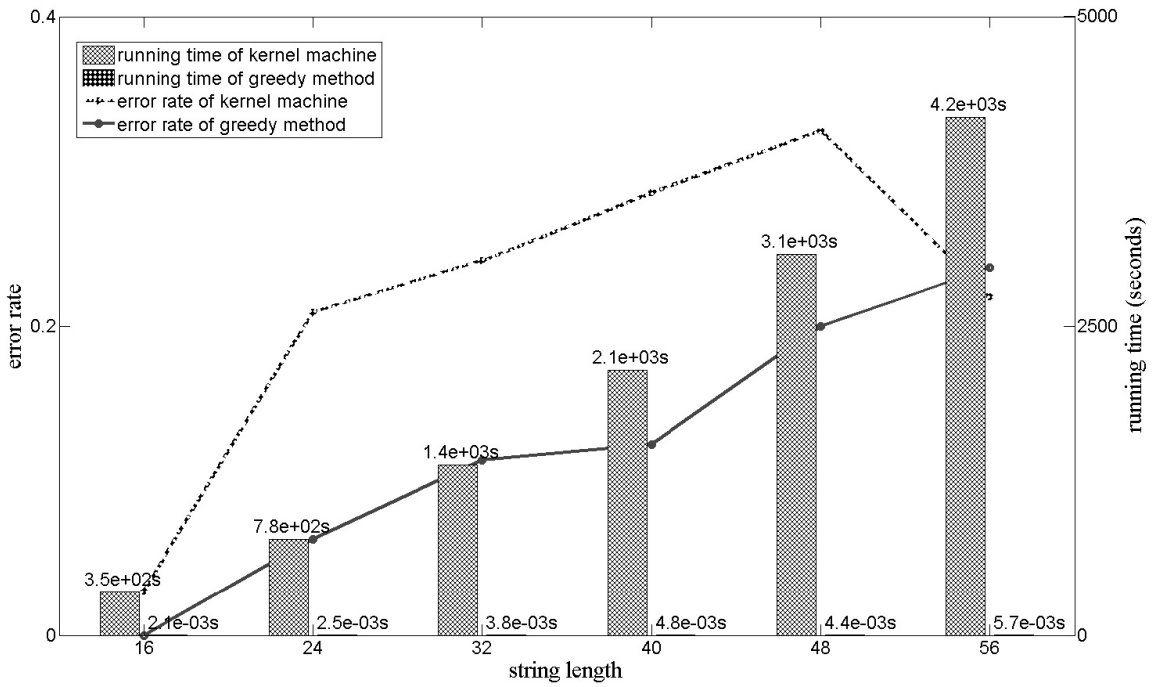

Figure 3.6: Experiment results with NSF abstracts data set (training 1999; testing 1998)

when $n = 56$ is very unstable over some datasets. Figure 3.6 is an example where kernel method became unpredictable when the no-convergence error happened. In this plot when $n = 56$ the kernel machine seems to have better accuracy than the greedy method, but considering that all the failed runs of the kernel machine were ruled out and only successful ones were taken into account, the apparent accuracy of the kernel method is shaky.

3.6 Discussion

We have shown positive results for learning shuffle ideals in the statistical query model under element-wise independent and identical distributions and Markovian distributions, as well as a constrained generalization to product distributions. It is

still open to explore the possibilities of learning shuffle ideals under less restricted distributions with weaker assumptions. Also a lot more work needs to be done on approximately learning shuffle ideals in applications with pragmatic approaches. In the negative direction, even a family of regular languages as simple as the shuffle ideals is not efficiently properly PAC learnable under general unrestricted distributions unless RP=NP. Thus, the search for a nontrivial properly PAC learnable family of regular languages continues. Another theoretical question that remains is how hard the problem of learning shuffle ideals is, or whether PAC learning a shuffle ideal is as hard as PAC learning a deterministic finite automaton.

Chapter 4

Learning a Random DFA from Uniform Strings and State Information

Deterministic finite automata (DFAs) 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 chapter 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. After introducing the preliminaries in Section 4.1, we present the

fast convergence of the random walks on a random DFA in Section 4.2. In addition to this positive property, a computationally efficient algorithm for learning random DFAs from uniform input strings in the statistical query model is proposed in Section 4.3, with a set of supporting experimental results.

The content of this chapter appears in [AC15].

4.1 Preliminaries

The *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 chapter we assume $s \geq 2$. Let $G = (V, E)$ be the underlying directed multigraph of the 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 vertex in G that corresponds to 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 chapter 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 the 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 $\Delta_{L_2}(t) \le 2\Delta_{TV}(t) \le \Delta_{\chi^2}(t)$, a convergence upper bound for $\Delta_{\chi^2}(t)$ implies ones for $\Delta_{L_2}(t)$ and $\Delta_{TV}(t)$.

Trakhtenbrot and Barzdin [TB73] first introduced the model of a 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 $\varphi(q,\sigma) = q'$.

4.2 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 [Gru73], 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 stated in Theorem 4.1.

Theorem 4.1 *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 [Gru73] first proved the irreducibility of a random automaton graph.

Lemma 4.1 [Gru73] 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 [Bal13] proved the aperiodicity.

Lemma 4.2 [Bal13] With probability $1 - o(1)$, the strongly connected component $\tilde{G}$ in Lemma 4.1 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 [Chu05] proved the connection between the Rayleigh quotient and the Laplacian matrix of a random walk.

Lemma 4.3 [Chu05]

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

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

From 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 4.4 *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 4.3 we know $\frac{1}{2}R(f) = \frac{\langle g\mathcal{L}, g\rangle}{\|g\|^2}$. From the symmetry of the 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 eigenvector corresponding to 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 on λ_1 in terms of the diameter and the maximum out-degree of the vertices in the graph.

Lemma 4.5 *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 Let $u_0 = \arg\max_{x \in V} \phi(x)$ 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 $1 = \sum_{x \in V} \phi(x) \leq n\phi(u_0) \leq n s_0^{Diam}\ \phi(v_0)$ and $\phi(v_0) \geq n^{-1}s_0^{-Diam}$.

From Lemma 4.4 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 $\vec{r} = (y, x_1, x_2 \ldots, x_{\ell-1}, z)$ be the shortest directed path from y to z, which must exist due to the strong connectivity. Then the length

of path $\vec{r}$ is ℓ. Therefore,

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

which completes the proof. ■

As a canonical technique, a lower bound on the smallest nonzero eigenvalue of the Laplacian matrix implies a lower bound on the convergence rate. Chung [Chu05] proved

Theorem 4.2 *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 4.5, we have

Theorem 4.3 *A random walk on a strongly connected and aperiodic directed graph G has convergence rate of order $2n \cdot Diam \cdot s_0^{1+Diam}(\log\left(ns_0^{Diam}\right))$, where $s_0 = \arg\max_{u \in V} d_u$ is the maximum out-degree of a vertex in G. Namely, after at most $t \geq 2n \cdot Diam \cdot s_0^{1+Diam}((\log\left(ns_0^{Diam}\right) + 2k))$ steps, we have $\Delta_{\chi^2}(t) \leq e^{-k}$.*

Now it remains to achieve a logarithmic upper bound for the diameter $Diam$. Fortunately, in our case $s_0 = s$ and Trakhtenbrot and Barzdin [TB73] proved the diameter of a random DFA is logarithmic.

Theorem 4.4 *With probability* $1-o(1)$*, the diameter of a random automaton graph is* $O(\log_s n)$*.*

With the logarithmic diameter we complete the proof of Theorem 4.1. The constant C in Theorem 4.1 is the constant used in the proof of Theorem 4.4 by Trakhtenbrot and Barzdin [TB73]. 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. In addition, Balle [Bal13] proved DFA minimization preserves aperiodicity. Now we also have Corollary 4.1.

Corollary 4.1 *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*.*

4.3 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 learning model (described in Section 3.1) with a theoretical guarantee on the maximum absolute error and supporting experimental results.

4.3.1 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 $y = y_1 y_2 \dots y_m$, define M_y to be the matrix product $M_y = M_{y_1} \cdot M_{y_2} \dots M_{y_m}$. 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 4.1 and 4.2, 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 $y = y_1 y_2 \dots y_m$, 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 4.4, 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 only need to 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 $\{p_y M_\sigma = p_{y\sigma} \mid y \in \Sigma^{\Theta(\log_s n)}\}$ has the same solution as $\{p_y M_\sigma = p_{y\sigma} \mid y \in \Sigma^{O(\log_s n)}\}$. 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 the 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 for q_0 is 1 and the rest are 0, for which

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

Theorem 4.1 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 $\mathbb{1}\{y$ 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 $\mathbb{E}\chi_{y,i}$, 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 $\|p_\lambda - \tilde{p}_\lambda\|_1 = 2\|\phi_t - \phi\|_{TV} \leq \Delta_{\chi^2}(t) \leq \frac{\tau}{2}$. Then for any string y, $\|p_y - \tilde{p}_y\|_\infty = \|(p_\lambda - \tilde{p}_\lambda)M_y\|_\infty \leq \|p_\lambda - \tilde{p}_\lambda\|_1 \leq \frac{\tau}{2}$. If we do the statistical queries with tolerance $\frac{\tau}{2}$, the maximum additive error will be $\|\tilde{p}_y - \widehat{p}_y\|_\infty \leq \frac{\tau}{2}$ for any string y. Thus we have $\|p_y - \widehat{p}_y\|_\infty \leq \tau$. To conclude a theoretical upper bound on the error, we use the following theorem by Björck [Bjö91], which was later refined by Higham [Hig94].

Theorem 4.5 *Let z be the optimal solution of 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 $\leq$ comparison, then*

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

when M has full column rank, or

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

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

Applying Theorem 4.5 to our case gives an upper bound on the maximum absolute error.

Corollary 4.2 *If P_A has full rank with high probability,*

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

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 $\|z\|_1 \leq \frac{(1+\varepsilon)\log ns}{\log\log ns}$ 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 $\mathbb{P}(\theta = n) \leq n \cdot n^{-n}$ is exponentially small. For any $k < n$,

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

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 $k \geq n$ then $\mathbb{P}(\theta \geq k)$ is exponentially small as discussed above. Otherwise we have $\left(1 - \frac{n}{k}\right)\log\left(1 - \frac{k}{n}\right) \leq 1$ in our case. Also notice that $\frac{1}{12n} \leq 1$. Let $k =$

$\frac{(1+\varepsilon)\log ns}{\log\log ns}$. The expression is upper bounded by

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

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 4.5 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 for 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.

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 convergence. We have no information for reconstructing the disconnected part. In the positive direction, due to Lemma 4.1, 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 4.2 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 esti-

mate $\tilde{p}_\lambda(q)$ 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 techniques. The full-rank assumption is reasonable because a random matrix is usually well conditioned and full-rank. From the empirical results in Section 4.3.2, 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 4.1, our algorithm is also applicable to learning a random DFA after minimization.

A toy example

The following toy example is to demonstrate how the algorithm works. 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 $(1/3, 4/9, 2/9)$. Since $\lceil \log_s n \rceil = \lceil \log_2 3 \rceil = 2$, 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

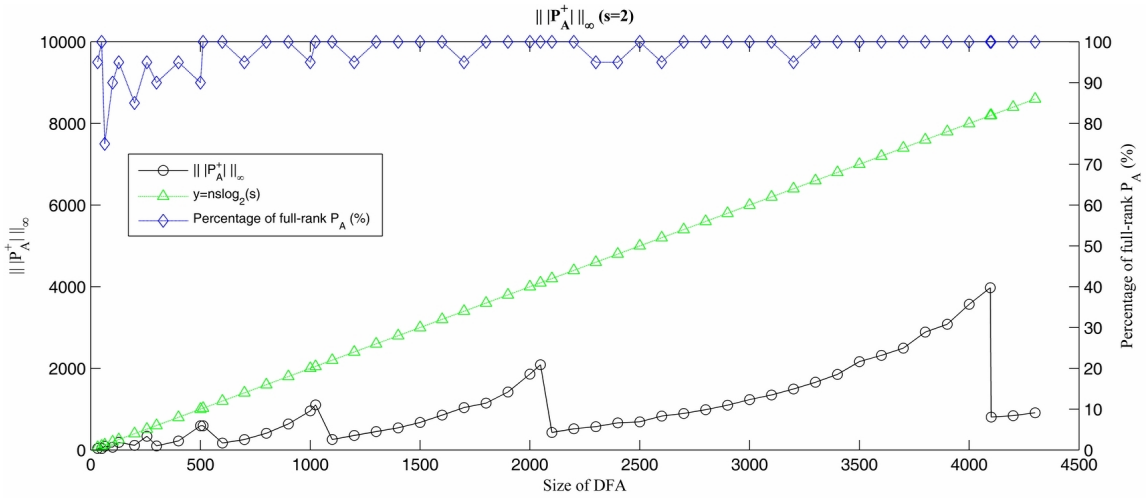

Figure 4.1: $|||P_A^\dagger|||_\infty$ versus n with fixed $s = 2$

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.

4.3.2 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 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 4.1 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.

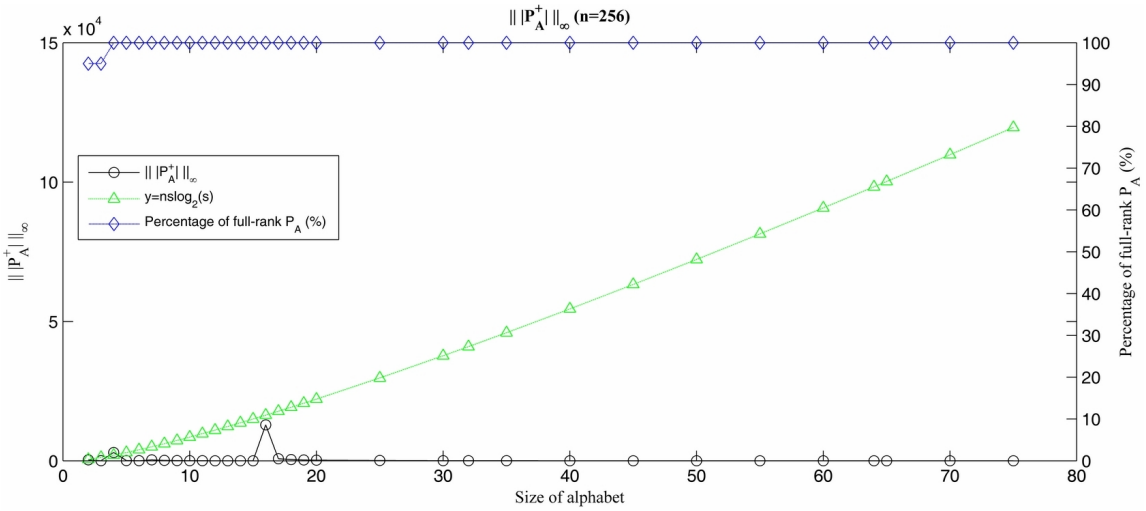

Figure 4.2: $\||\|P_A^\dagger\|\|_\infty$ versus s with fixed $n = 256$

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 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 4.2. 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 4.3 and 4.4 demonstrate the experimental results. Along with the error curve in each figure a function is plotted to approximate the asymptotic behavior of the error. An empirical error bound is $O(n^{-0.3})$ with fixed s and $O(s^{-0.3})$ with fixed n.

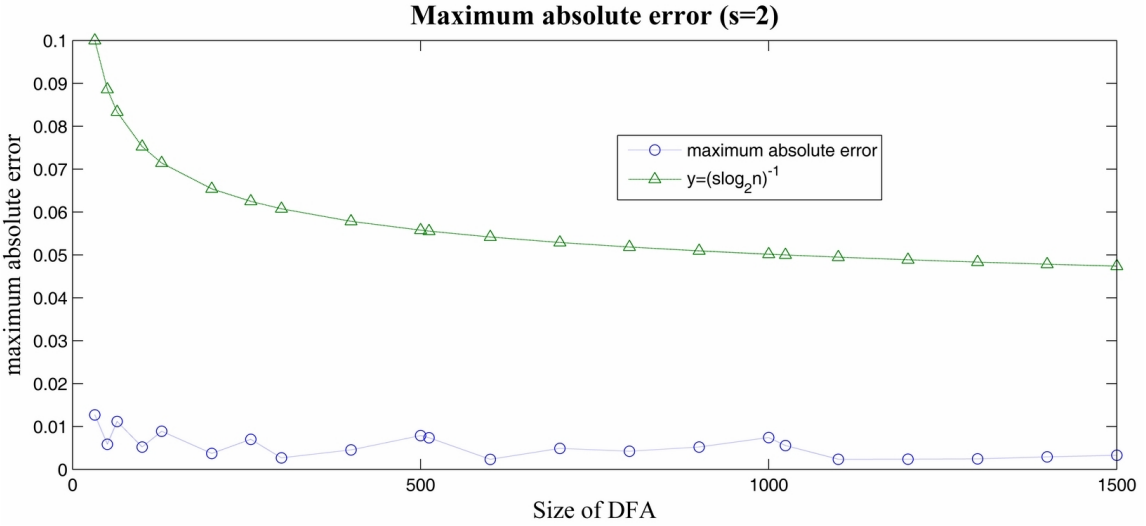

Figure 4.3: Maximum absolute error versus n with fixed $s = 2$

4.4 Discussion

In this chapter 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 [Thi68] 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 is continuous, not

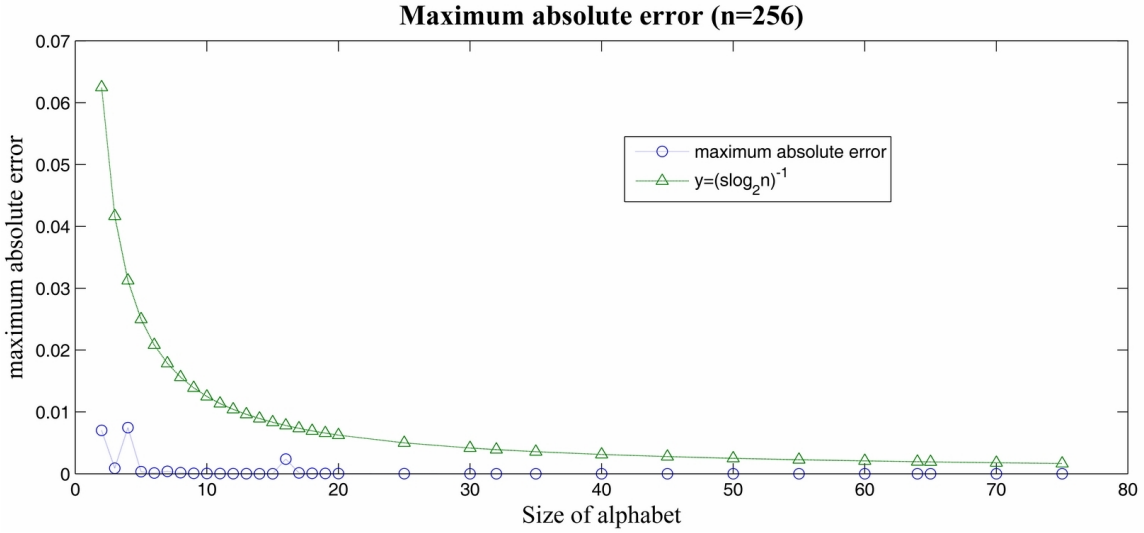

Figure 4.4: Maximum absolute error versus s with fixed $n = 256$

necessarily a binary vector.

Chapter 5

Learning Random Regular Graphs

The family of random regular graphs is a classic topic in the realms of graph theory, combinatorics and computer science. In this chapter we study the problem of learning random regular graphs from random paths. A random regular graph is generated uniformly at random and in a standard label-guided graph exploration setting, the edges incident from a node in the graph have distinct local labels. The input data to the statistical query oracle are path-vertex pairs (x, v) where x is a random uniform path (a random sequence of edge labels) and v is the vertex of the graph reached on the path x starting from a particular start vertex v_0. In Section 5.2 we present our main theorem on the fast convergence of random walks on random regular graphs. In addition to the theoretical results, we generalize our learning algorithm in Chapter 4 to learning random regular graphs from uniform paths in the statistical query model, in Section 5.3, with a group of experimental results. In Section 5.4 we discuss other applications and potential future work in computer science and machine learning.

The content of this chapter appears in [Che15].

5.1 Overview

Random walks on graphs have long served as a fundamental topic in the study of Markov chains and also as an important tool in machine learning research. On the other hand, regular graphs are widely studied in computer science for their important role in computational graph models and their applications. Because strong properties usually don't hold for all regular graphs, it is natural to ask whether we can pursue positive results for "almost all" regular graphs. This is addressed by studying high-probability properties of uniformly generated random regular graphs. In recent decades random regular graphs have gathered more and more attention in computer science, combinatorics and graph theory. Nevertheless, the study of random walks on random regular graphs is relatively limited. This chapter aims to fill this gap with a comprehensive study of the varieties of random regular graphs listed in Table 5.1. Detailed definitions of the random graph models are provided in Section 5.2.1. The notation in Table 5.1 will be used throughout this chapter. Our main contributions are the positive results on the fast convergence of random walks on random regular graphs, which fill the gap in the research on random regular graphs. With these positive theoretical results, we are able to generalize our learning algorithm in Chapter 4 to learning random regular graphs (i.e., almost all regular graphs) from random paths in the statistical query model.

Random out-regular multigraphs ($\mathrm{RMG}^+(s)$) are the most well-studied among the family of random regular graphs, mainly because the freedom and independence of the edge selections makes the analysis simple and direct. This is also due to the important role of deterministic finite automaton (DFA) in computer science, as the underlying automaton graph of a random DFA is exactly a $\mathrm{RMG}^+(s)$ [Gru73, TB73, AC15]. In the context of DFA learning, we have proved the fast convergence of

Random regular graph model	**Notation**
Random out-regular multigraph	$\mathrm{RMG}^+(s)$
Random out-regular simple graph	$\mathrm{RSG}^+(s)$
Random in-regular multigraph	$\mathrm{RMG}^-(s)$
Random in-regular simple graph	$\mathrm{RSG}^-(s)$
Random s-in s-out multigraph	$\mathrm{RMG}^\pm(s)$
Random s-in s-out simple graph	$\mathrm{RSG}^\pm(s)$
Random regular digraph	$\mathrm{RDG}(s)$
Random regular undirected graph	$\mathrm{RG}(s)$

Table 5.1: Random regular graphs with fixed degree s

random walks on a $\mathrm{RMG}^+(s)$ in Chapter 4. In this chapter, we first start with the slightly more restricted model, the random out-regular simple graphs ($\mathrm{RSG}^+(s)$), with less freedom and independence of the edges. Simple graphs are more natural in real-world applications like citation graphs and k-nearest neighbor graphs where self-loops and parallel edges are not allowed. We prove random walks on a $\mathrm{RSG}^+(s)$ converge to the stationary distribution polynomially fast with probability $1 - o(1)$. Based on the proofs for out-regular models, we then show similar properties for in-regular models. In-regular graphs are less popular and of limited interest in practice but their properties are helpful in studying the random s-in s-out graph models, first introduced by Fenner and Frieze [FF82], which can be viewed as the sum of a random out-regular graph and a random in-regular graph.

After that we study the two classes of regular graphs in usual sense: regular digraphs and regular undirected graphs. They are the most restricted graphs among these models but very widely studied in the literature. A random undirected sparse (i.e., $s = O(1)$) regular multigraph is known to be an expander graph with high probability. It is well known that expander graphs have well-bounded Laplacian eigenval-

ues. In this chapter RDG(s) and RG(s) are simple, not necessarily sparse graphs. In addition, polynomially bounding the Laplacian eigenvalues for RDG(s) and RG(s) is not hard and doesn't involve any randomness (including nonsparse cases, see Section 5.2.5 for detailed formal proofs), but bounding Laplacian eigenvalues is not sufficient for fast convergence. Most of our effort is spent on the aperiodicity, where the randomness in the models is formally dealt with. This is the major difficulty in our proof and requires a substantial amount of work. To the best of our knowledge, no work has been done on the ergodicity and convergence rate of the random walks on RDG(s) and previous results for RG(s) require $s = \lfloor \log^C n \rfloor$ for some constant $C \geq 2$, where n is the number of vertices in the graph. We present a complete proof for fast convergence of random walks on RDG(s) for $s \geq 2$ and random walks on RG(s) for $s \geq 3$ if n is odd and for $3 \leq s = o(\sqrt{n})$ or $s > \frac{1}{2}n$ if n is even.

5.2 Random walks on random regular graphs

In this section, we describe our main theoretical result. Concepts and notation used throughout this chapter are described in Section 5.2.1. The main theorem is presented in Section 5.2.2, followed by the proof.

5.2.1 Preliminaries

A *graph* is a tuple $G = (V, E)$, where V is a (finite) set whose elements are called *vertices* and E is a (finite) multiset of ordered pairs of V called *edges*. We denote by $n = |V|$. A graph is *undirected* if the vertex pairs in E are unordered, and is *simple* if it has no self-loops or parallel edges. If vertex v is reachable from another vertex u, the *distance* $d(u, v)$ from u to v is the minimum length of a path from u to v and $d(u, u) = 0$. The *diameter* of a graph is $\max\{d(u, v) \mid v = u \text{ or } v \text{ is reachable from } u\}$. A graph

G is *(cyclically) h-partite* if V can be partitioned into h subsets, $V_0, V_1, \ldots, V_{h-1}$, in such a way that all edges from V_i go to $V_{(i+1) \mod h}$. 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 component $V_0 \subset V$ is *isolated* if for any $u \in V_0$ and any v such that $(u, v) \in E$ or $(v, u) \in E$, we must have $v \in V_0$.

In an undirected graph, the *degree* of a vertex u is the number of edges incident to u. An undirected graph is *regular* if every vertex has the same degree. In a digraph, for a directed edge (u, v) in E, we say that vertex u has an *out-neighbor* v and vertex v has an *in-neighbor* u. The number of edges incident to a vertex u is the *in-degree* of u, denoted by d_u^-, and the number of edges incident from u is its *out-degree*, denoted by d_u^+. Unless otherwise stated, by default a *neighbor* refers to an out-neighbor and the degree of a vertex u denoted by d_u means its out-degree. A graph G is *out-regular* if $d_u = s$ for all $u \in V$; and is *in-regular* if $d_u^- = s$ for all $u \in V$. A digraph is *regular* if it is both in-regular and out-regular.

A *walk* on a graph G is a sequence of vertices $(v_0, v_1, \ldots, v_\ell)$ such that $(v_{i-1}, v_i) \in E$ for all $1 \leq i \leq \ell$. A *random walk* on a graph G is defined by a transition probability matrix P with $P(u, v) = \#\{(u, v) \in E\} \cdot d_u^{-1}$ 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 in the graph. A vertex (or equivalently a state of a random walk) u is *aperiodic* if $\gcd\{t \geq 1 \mid P^t(u, u) > 0\} = 1$. A graph G (or a random walk on G) is *irreducible* if for every u and v in V there exist a directed cycle in G containing 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 the 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*:

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

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

In this chapter we study the random graph models listed in Table 5.1. For each model, an instance is drawn uniformly at random from the instance space of the model. A random s-in s-out graph is generated as the sum of a random in-regular graph and a random out-regular graph [FF82]. A RDG(s) has no parallel edges but allows self-loops. A RG(s) is simple.

5.2.2 The main theorem

We prove positive results on the ergodicity and convergence rate of random walks on random regular graphs, as stated in the following theorem.

Theorem 5.1 *With probability* $1-o(1)$*, a random walk on a random regular graph has* $\Delta_{\chi^2}(t) \le e^{-k}$ *after* $t \ge t_0$ *steps, where*

1. *for RMG*$^+$*(*s*) and RSG*$^+$*(*s*):* $t_0 = 2C(C+1)sn^{1+C}(\log n + k)\cdot\log_s n$ *for some constant* $C > 0$ *when* $s \ge 2$*;*

2. *for RDG(*s*):* $t_0 = 2s(n-1)(\log n + 2k)$ *when* $s \ge 2$*;*

3. *for RG(*s*):* $t_0 = s(n-1)(\log n + 2k)$ *when* $s \ge 3$ *if* n *is odd; when* $3 \le s = o(\sqrt{n})$ *or* $s > \frac{1}{2}n$ *if* n *is even;*

4. *for* $RMG^{-}(s)$ *and* $RSG^{-}(s)$*:* $t_0 = 2C(C+1)s^C n^{1+C}(\log n + k)\cdot \log_s n$ *for some constant* $C > 0$ *when the walk is restricted to the unique irreducible component and there exists a constant* $C' \geq 1$ *such that* $s = \Omega\left(\left[\frac{\log n}{\log\log n}\right]^{1/C'}\right)$.

5. *for* $RMG^{\pm}(s)$ *and* $RSG^{\pm}(s)$*:* $t_0 = 2C(C+1)s^C n^{1+C}(\log n + k)\cdot \log_s n$ *for some constant* $C > 0$ *when there exists a constant* $C' \geq 1$ *such that* $s = \Omega\left(\left[\frac{\log n}{\log\log n}\right]^{1/C'}\right)$.

The constraints on s in the theorem are optimal. The low connectivity of 1-regular graphs makes them of little interest so we need the degree s to be at least 2. In the undirected case we have $s \geq 3$ because when $s = 2$ a connected 2-regular undirected graph (or component) can only be a simple cycle. That is, a RG(2) must be a set of isolated simple cycle(s). This not only breaks the irreducibility, but also violates the aperiodicity of the graph. The other constraint $s = o(\sqrt{n})$ for even n comes from the study of enumeration of RG(s). In the cases (4) and (5), a lower bound on s is needed because small in-degree s brings us large maximum out-degree (with respect to s). Unlike other models, the irreducible component in the in-regular cases in the theorem is not necessarily closed, and the fast convergence property only holds when the walk is restricted to the unique irreducible component.

5.2.3 Fast convergence on RMG^{+} and RSG^{+}

In Chapter 4 we have proved that random walks on a random DFA converge polynomially fast. Because the underlying graph of a random DFA is exactly a $RMG^{+}(s)$, the $RMG^{+}(s)$ case in the main theorem is established immediately by the work in Chapter 4.

A standard proof of fast convergence consists of three parts: irreducibility, aperiodicity and polynomial convergence rate. The irreducibility of $RSG^{+}(s)$ is built on

that of $\mathrm{RMG}^+(s)$, thanks to the similarities they share. A $\mathrm{RSG}^+(s)$ can be generated from a $\mathrm{RMG}^+(s)$ using a two-stage procedure. Stage 1: generate a $\mathrm{RMG}^+(s)$. Stage 2: for each vertex in the graph, check whether all its s neighbors are distinct nodes that are not itself. If not, keep choosing neighbors from V uniformly at random until it has exactly s distinct neighbors excluding itself. Finally, remove self-loops and merge parallel edges to simple edges. By this method a $\mathrm{RSG}^+(s)$ can be viewed as a $\mathrm{RMG}^+(s)$ adding more edges after removing self-loops and merging parallel edges. Together with the fact that a $\mathrm{RMG}^+(s)$ has a large closed and strongly connected component (Lemma 5.1), we achieve the irreducibility of $\mathrm{RSG}^+(s)$ (Lemma 5.2).

Denote by $p_h(\bar{n})$ the probability of existence of an h-partite component which consists of $\bar{n}$ vertices in a $\mathrm{RSG}^+(s)$. Let $\bar{G} = (\bar{V}, \bar{E})$ where $|\bar{V}| = \bar{n}$ be one such component. Note that $\bar{G}$ is h-partite if and only if $\bar{V}$ can be partitioned into h disjoint subsets $\bar{V}_0, \bar{V}_1, \ldots, \bar{V}_{h-1}$ such that all edges from $\bar{V}_i$ go to $\bar{V}_{(i+1) \bmod h}$. Algebra and combinatorics bounds give us that $p_h(\bar{n})$ is at most

$$\binom{n}{\bar{n}}\binom{\bar{n}-1}{h-1} \cdot \binom{\bar{n}}{\frac{\bar{n}}{h}, \frac{\bar{n}}{h}, \ldots, \frac{\bar{n}}{h}} \cdot \left(\frac{1}{h}\right)^{2\bar{n}+1} \cdot \left(\frac{\bar{n}}{\bar{n}-1}\right)^{2\bar{n}}$$

We further show that $p_h(\bar{n})$ is exponentially small for any $\bar{n} > 0.79n$ and any $h \geq 2$ so that the probability of periodicity $\leq \sum_{\bar{n}=\lceil 0.79n \rceil}^{n} \sum_{h=2}^{\bar{n}} p_h(\bar{n})$ goes to 0 when $n \to +\infty$ (Lemma 5.3).

The proof of the polynomial convergence rate is mainly done by showing that a $\mathrm{RSG}^+(s)$ has logarithmic diameter (of order $\Theta(\log_s n)$) with high probability. To do so, we generate a $\mathrm{RSG}^+(s)$ in a "level-wise" order. Initially we pick a start vertex $u_0 \in V$ and let *level* 0 be the set $\{u_0\}$. Then inductively, for each vertex u_i in level i, we choose its s neighbors from $\{V \setminus u_i\}$ uniformly at random without replacement. All the new chosen vertices form level $i+1$. This spanning procedure halts when no

new vertex is chosen as a neighbor of the boundary so the next level is empty. We call the set of vertices in all levels $\leq i$ the *ball* i. The final step is for each vertex not in the ball, uniformly choosing s distinct vertices as its neighbors. To accomplish the proof, we divide the above spanning procedure into six stages (see the proof of Theorem 5.2 for details). We show that the size of the spanning ball keeps increasing in the first 3 stages while the boundary of the ball starts shrinking in Stage 4 and finally the spanning procedure halts with an empty new level. The number of levels constructed in all stages is logarithmic, and so is the diameter of the graph.

Now we present the formal proof below.

Irreducibility

Since $\mathrm{RMG}^+(s)$ and $\mathrm{RSG}^+(s)$ share many similarities, we can achieve the irreducibility of $\mathrm{RSG}^+(s)$ based on that of $\mathrm{RMG}^+(s)$.

Lemma 5.1 [Gru73] *With probability* $1 - o(1)$*, a* $RMG^+(s)$ *has a unique strongly connected component, denote 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* $s \geq 7$*; b)* $\tilde{V}$ *is closed.*

The irreducibility of $\mathrm{RSG}^+(s)$ is proved in the following lemma.

Lemma 5.2 *With probability* $1 - o(1)$*, a* $RSG^+(s)$ *has a unique closed and strongly connected component, denoted by* $\tilde{G} = (\tilde{V}, \tilde{E})$*, of size* $\tilde{n}$ *when* $n \to +\infty$*, and* $\lim_{n\to+\infty} \frac{\tilde{n}}{n} \geq C$ *for some constant* $C > 0.7968$ *when* $s \geq 2$ *or some* $C > 0.999$ *when* $s \geq 7$*.*

Proof Recall that the only difference between $\mathrm{RSG}^+(s)$ from $\mathrm{RMG}^+(s)$ is that the s neighbors of each vertex are chosen without replacement so no self-loops or parallel

edges are allowed. We can consider the following two-stage procedure to generate a $\mathrm{RSG}^+(s)$ from a $\mathrm{RMG}^+(s)$. Stage 1: generate a $\mathrm{RMG}^+(s)$. Stage 2: for each vertex in the graph, check whether all its s neighbors are distinct nodes that are not itself. If not, keep choosing neighbors from V uniformly at random until it has exactly s distinct neighbors excluding itself. Finally, remove self-loops and merge parallel edges to simple edges. Because for every vertex $u \in V$, each $v \in V \setminus \{u\}$ will become one of the s neighbors of u with equal probability, the result of this procedure is a uniformly generated $\mathrm{RSG}^+(s)$.

Thus a $\mathrm{RMG}^+(s)$ can be viewed as a $\mathrm{RMG}^+(s)$ adding more edges after removing self-loops and merging parallel edges. This means the simple graph model has connectivity at least as good as the multigraph model. The size of the strongly connected component will only increase. After Stage 1 we have a $\mathrm{RMG}^+(s)$, denoted by $G_1 = (V, E_1)$ and let $\tilde{V}_1 \subseteq V$ be the closed strongly connected component of G_1 stated in Lemma 5.1. To show the irreducible component in a $\mathrm{RSG}^+(s)$ is also closed, note that for any $v \notin \tilde{V}_1$, there must exist at least one path from v to $\tilde{V}_1$. Otherwise there will be another strongly connected component in G_1, which contradicts Lemma 5.1. Thus in Stage 2, every time we add an edge from $\tilde{V}_1$ to some $u \notin \tilde{V}_1$, there must be some directed path(s) from u heading back to the irreducible component. All the vertices on this(these) path(s) are now strongly connected with $\tilde{V}_1$ and become new members of the irreducible component. Therefore, the irreducible component in the final simple graph will also be closed. ■

Aperiodicity

Lemma 5.3 *With probability* $1 - o(1)$*,* $\tilde{G}$ *in Lemma 5.2 is aperiodic.*

Proof Let $p_h(\bar{n})$ be the probability of existence of an h-partite component of size $\bar{n}$ in a RSG$^+(s)$. The proof is completed by showing $p_h(\bar{n})$ goes to 0 exponentially fast when $n \to +\infty$ for any $\bar{n} > 0.79n$ and $h \geq 2$ so that combining with Lemma 5.2 the probability of periodicity is $\leq \sum_{\bar{n}=\lceil 0.79n \rceil}^{n} \sum_{h=2}^{\bar{n}} p_h(\bar{n})$ and goes to 0 when $n \to +\infty$.

Let $\bar{G} = (\bar{V}, \bar{E})$ be a fixed component of size $\bar{n}$ in the graph. $\bar{G}$ is h-partite if $\bar{V}$ can be partitioned into h subsets, $\bar{V}_0, \bar{V}_1, \ldots, \bar{V}_{h-1}$, such that all edges from $\bar{V}_i$ go to $\bar{V}_{(i+1) \mod h}$. The number of such partitions is at most $h^{\bar{n}}$. The probability of forming a particular partition $\bar{V}_0, \bar{V}_1, \ldots, \bar{V}_{h-1}$ is

$$\begin{aligned}
\prod_{i=0}^{h-1} \left(\frac{\binom{|\bar{V}_{(i+1) \mod h}|}{s}}{\binom{n-1}{s}} \right)^{|\bar{V}_i|} &= \prod_{i=0}^{h-1} \left(\frac{\prod_{j=0}^{s-1} \left(|\bar{V}_{(i+1) \mod h}| - j \right)}{\prod_{j=0}^{s-1} (n-1-j)} \right)^{|\bar{V}_i|} \\
&\leq \prod_{i=0}^{h-1} \left(\prod_{j=0}^{s-1} \frac{|\bar{V}_{(i+1) \mod h}|}{n-1} \right)^{|\bar{V}_i|} \\
&= \prod_{i=0}^{h-1} \left(\frac{|\bar{V}_{(i+1) \mod h}|}{n-1} \right)^{s|\bar{V}_i|} \\
&\leq \left(\frac{\bar{n}}{h(n-1)} \right)^{s\bar{n}} \\
&\leq \left(\frac{\bar{n}}{h(n-1)} \right)^{2\bar{n}}
\end{aligned}$$

This is because the product $\prod_{i=0}^{h-1} x_{(i+1) \mod h}^{x_i}$, given $x_i > 0$ and $\sum_{i=0}^{h-1} x_i = \bar{n}$, is maximized for $x_i = \bar{n}/h, i = 0 \ldots h-1$. Thus

$$\begin{aligned}
p_h(\bar{n}) &\leq \binom{n}{\bar{n}} \cdot h^{\bar{n}} \cdot \left(\frac{\bar{n}}{h(n-1)} \right)^{2\bar{n}} \\
&= \binom{n}{\bar{n}} \cdot \left(\frac{1}{h} \right)^{\bar{n}} \cdot \left(\frac{\bar{n}}{n-1} \right)^{2\bar{n}} \\
&\leq \binom{n}{\bar{n}} \cdot \left(\frac{1}{2} \right)^{\bar{n}} \cdot \left(\frac{\bar{n}}{n-1} \right)^{2\bar{n}}
\end{aligned}$$

When $\bar{n} = n$, as $\lim_{n\to+\infty}\left(\frac{n}{n-1}\right)^{2n} = e^2$, apparently $p_h(n)$ goes to 0 exponentially fast.

When $0.79n < \bar{n} < n$, we have

$$\begin{aligned}
p_h(\bar{n}) \leq & \binom{n}{\bar{n}} \cdot \left(\frac{1}{2}\right)^{\bar{n}} \cdot \left(\frac{\bar{n}}{n-1}\right)^{2\bar{n}} \\
= & \frac{n!}{\bar{n}!(n-\bar{n})!} \cdot \left(\frac{1}{2}\right)^{\bar{n}} \cdot \left(\frac{\bar{n}}{n-1}\right)^{2\bar{n}} \\
\leq & \frac{\sqrt{2\pi n} \cdot n^n \cdot e^{n-\bar{n}+\frac{1}{12n}} \cdot e^{\bar{n}}}{\sqrt{2\pi(n-\bar{n})} \cdot e^n \cdot (n-\bar{n})^{n-\bar{n}} \cdot \sqrt{2\pi\bar{n}} \cdot \bar{n}^{\bar{n}}} \cdot \left(\frac{1}{2}\right)^{\bar{n}} \cdot \left(\frac{\bar{n}}{n-1}\right)^{2\bar{n}} \\
= & \sqrt{\frac{n}{2\pi\bar{n}(n-\bar{n})}} \cdot e^{\frac{1}{12n}} \cdot \left(\frac{\bar{n}^2}{2n^2}\right)^{\bar{n}} \cdot \frac{n^n}{\bar{n}^{\bar{n}} \cdot (n-\bar{n})^{n-\bar{n}}} \cdot \left(\frac{n}{n-1}\right)^{2\bar{n}} \\
\leq & \sqrt{\frac{n}{2\pi\bar{n}(n-\bar{n})}} \cdot e^{\frac{1}{12n}} \cdot \left(\frac{\bar{n}}{2n}\right)^{\bar{n}} \cdot \frac{n^{n-\bar{n}}}{(n-\bar{n})^{n-\bar{n}}} \cdot \left(\frac{n}{n-1}\right)^{2n} \\
= & \sqrt{\frac{n}{2\pi\bar{n}(n-\bar{n})}} \cdot e^{\frac{1}{12n}} \cdot \left[\left(\frac{\bar{n}}{2n}\right)^{\frac{\bar{n}}{n}} \cdot \left(1-\frac{\bar{n}}{n}\right)^{\frac{\bar{n}}{n}-1}\right]^n \cdot \left(\frac{n}{n-1}\right)^{2n}
\end{aligned}$$

Note that function $f(x) = (1-x)^{x-1} \cdot \left(\frac{x}{2}\right)^x < 0.7$ for all $0.79 < x < 1$. Hence, the probability $p_h(\bar{n})$ is exponentially small, which completes the proof. ■

Fast convergence

Based on Theorem 4.3, to accomplish the fast convergence of random walk on a $\mathrm{RSG}^+(s)$, we prove the diameter of a $\mathrm{RSG}^+(s)$ is logarithmic with high probability.

Theorem 5.2 *With probability* $1 - o(1)$*, the diameter of a RSG*$^+$*(s) is* $\Theta(\log_s n)$.

Proof The logarithmic lower bound is easy to prove. For a particular vertex $u \in V$, denote by $S_i(u)$ the set of vertices in G such that for any $v \in S_i(u)$ the distance from u to v is i. We know $S_0(u) = \{u\}$ and $n = \sum_{i=0}^{+\infty} |S_i(u)|$. According to the definition of diameter, $|S_i(u)| = 0$ for all $i > Diam$. Also notice that $|S_{i+1}(u)| \leq s|S_i(u)|$, for

which we have

$$n \leq 1 + s + s^2 + \ldots + s^{Diam} = \frac{s^{Diam+1} - 1}{s - 1}$$

After some algebra, $Diam \geq \log_s(n(s-1)+1) - 1 \geq \log_s(n(s-1)) - 1 = \log_s n + \log_s(s-1) - 1 \geq \log_s n - 1$ due to $\log_s(s-1) \geq 0$ for all $s \geq 2$. Hence, we have $Diam = \Omega(\log_s n)$. This lower bound holds for $\mathrm{RMG}^+(s)$ as well.

However, the proof of the upper bound is lengthy. It is well known that a $\mathrm{RMG}^+(s)$ has logarithmic diameter with high probability [TB73]. Although the proof for $\mathrm{RMG}^+(s)$ doesn't work for $\mathrm{RSG}^+(s)$ due to the dependence between its edge selections, our proof follows the framework of their proof.

Assume that we generate a $\mathrm{RSG}^+(s)$ in a "level-wise" order. We pick a vertex $u_0 \in V$ and let *level* 0 be the set $\{u_0\}$. Then choose its s neighbors from $V \setminus \{u_0\}$ uniformly at random without replacement. All the neighbors of u_0 form level 1. Inductively, for each vertex in level $i-1$ we choose its s neighbors uniformly excluding itself without replacement. All the new chosen vertices form level i. We call the set of vertices in level $\leq i$ the *ball* i. By intuition, level i is the set of vertices to which the distance from u_0 is i and ball i consists of all vertices to which the distance from u_0 is at most i. Obviously level i is the boundary of ball i. The spanning procedure halts when no new vertex is chosen as a neighbor of the boundary so the next level is empty. To completely generate a $\mathrm{RSG}^+(s)$, the final step is for each vertex not in the ball, uniformly choosing s distinct vertices as its neighbors. Let L_i be the size of level i and B_i be the size of ball i. At any time, we say a vertex is *occupied* if it has non-zero in-degree and unoccupied otherwise. During this process, *determining* a vertex refers to choosing its s neighbors.

In short, to accomplish the proof, we divide the above spanning procedure into six stages:

Stage 1 starts from the very beginning and ends at level ℓ_1 once $B_{\ell_1} \geq n^{\frac{1}{6}}$.

Stage 2 begins immediately after Stage 1 and ends at level ℓ_2 once $B_{\ell_2} \geq \frac{n}{s^4}$.

Stage 3 begins immediately after Stage 2 and ends at level ℓ_3 once $B_{\ell_3} \geq (1 - 2^{-s})\, n$.

Stage 4 begins immediately after Stage 3 and ends at level ℓ_4 once $L_{\ell_4} \leq (\log_2 n)^2$.

Stage 5 begins immediately after Stage 4 and ends at level ℓ_5 once $L_{\ell_5} \leq 120 \log_2 n$.

Stage 6 begins immediately after Stage 5 and ends at level ℓ_6 once $L_{\ell_6+1} = 0$. The spanning procedure halts.

Letting ℓ_0 be 0 and $\ell'_i = \ell_i - \ell_{i-1}$, $1 \leq i \leq 6$ be the number of new levels created in Stage i, we complete the proof by showing $\sum_{i=1}^{6} \ell'_i = O(\log_s n)$.

Now we start moving to the details. First we notice that the above level-wise procedure can also be used to generate a $\mathrm{RMG}^+(s)$ if we choose neighbors of a vertex with replacement and allow self-loops. To distinguish between the multi-graph case and the simple graph case, let $\widehat{L}_i$ be the size of level i and $\widehat{B}_i$ be the size of ball i in the multi-graph case so that we can make use of some partial results in the multi-graph case proved by Trakhtenbrot and Barzdin.

Consider a sequence of N Bernoulli trials with probability p for success and $1 - p$ for failure. Let $X(N, p)$ denote the random variable defined as the number of successful outcomes in this sequence. Trakhtenbrot and Barzdin proved that for any $p > 0$, any natural number N and any $pN < k \leq N$, $\mathbb{P}(X(N, p) \geq k) <$

$N \cdot [k/(pN)]^{(3+pN-k)/2}$. It's easy to see the following facts:

$$\begin{aligned}\mathbb{P}\left(X\left(ms, \frac{n-w}{n-s}\right) \leq k\right) &= \mathbb{P}\left(X\left(ms, \frac{(n-1)-(w-1)}{(n-1)-(s-1)}\right) \leq k\right) \\ &\leq \mathbb{P}(L_{i+1} \leq k \mid L_i = m \wedge B_i = w) \\ &\leq \mathbb{P}\left(X\left(ms, \frac{(n-1)-(w-1)-(ms-1)}{(n-1)-(s-1)}\right) \leq k\right) \\ &< \mathbb{P}\left(X\left(ms, \frac{n-w-ms}{n}\right) \leq k\right)\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}\left(X\left(ms, \frac{n-w}{n}\right) \leq k\right) &\leq \mathbb{P}(\widehat{L}_{i+1} \leq k \mid \widehat{L}_i = m \wedge \widehat{B}_i = w) \\ &< \mathbb{P}\left(X\left(ms, \frac{n-w-ms}{n}\right) \leq k\right)\end{aligned}$$

Imagine we choose the edges one by one in the above described level-wise order. Assuming the number of occupied nodes is t at the moment when we are choosing the i-th edge of vertex v, then the probability of choosing an unoccupied vertex as the destination (so that we have a new member of the next level) is $\frac{(n-1)-(t-1)}{n-i} = \frac{n-t}{n-i}$ under the simple graph model and is always $\frac{n-t}{n}$ under the multi-graph model. Therefore, under the same configuration, we will always have higher probability to choose an unoccupied vertex under the simple graph model than under the multi-graph model. We can easily conclude:

$$\mathbb{P}(L_{i+1} \leq k \mid L_i = m \wedge B_i = w) < \mathbb{P}(\widehat{L}_{i+1} \leq k \mid \widehat{L}_i = m \wedge \widehat{B}_i = w)$$

Similarly, imagine we determine the vertices one by one in the above described level-wise order and let $B(r)$ be the number of occupied vertices exactly after we have determined r vertices. Denote by $\widehat{B}(r)$ the corresponding quantity in the multi-graph case. From our analysis above it's easy to see $\mathbb{P}(B(r) \geq k) > \mathbb{P}(\widehat{B}(r) \geq k)$ for any $r \geq 1$. Below we will go through the six stages and show the number of new levels

constructed is small in every stage.

Stage 1: For any level $i \leq \lceil \frac{1}{6}\log_s n \rceil - 1$, we have $B_{i+1} \leq \sum_{j=0}^{j+1} s^j < s^{i+2} \leq s^2 n^{\frac{1}{6}}$. Thus the probability that an edge created on level i will point to an occupied vertex is less than $\frac{s^2 n^{\frac{1}{6}}-1}{n-1} < s^2 n^{-\frac{5}{6}}$. This means that the probability that more than one edge on the first $\lceil \frac{1}{6}\log_s n \rceil - 1$ levels will point to an occupied vertex is less than $\sum_{j=2}^{k} b(j,k,p)$ where k is the maximal possible number of edges on the first $\lceil \frac{1}{6}\log_s n \rceil - 1$ levels, $p = s^2 n^{-\frac{5}{6}}$ and $b(j,k,p)$ is the probability of j successful outcomes and $k-j$ failures in k Bernoulli trials with probability p for success. Obviously, $k < s^3 n^{\frac{1}{6}}$. Trakhtenbrot and Barzdin proved that for sufficiently large n, $\sum_{j=2}^{k} b(j,k,p) < n^{-\frac{8}{7}}$.

Hence, when $n \to +\infty$, with probability more than $1 - n^{-\frac{8}{7}}$, $\ell_1' \leq \lceil \frac{1}{6}\log_s n \rceil$ and $L_{\ell_1} \geq (s-1)n^{\frac{1}{6}}/s \geq n^{\frac{1}{6}}/2$.

Stage 2: Trakhtenbrot and Barzdin proved that when $\widehat{L}_{i-1} \geq n^{\frac{1}{6}}/2$ and $\widehat{B}_{i-1} < n/s^4$,

$$\mathbb{P}\left(\widehat{L}_i \geq \left(1 - \frac{s+2}{s^4}\right) s\widehat{L}_{i-1} \mid \widehat{L}_{i-1}, \widehat{B}_{i-1}\right) > 1 - n^{-C}$$

for any fixed C and sufficiently large n. We then have that when $L_{i-1} \geq n^{\frac{1}{6}}/2$ and $B_{i-1} < n/s^4$,

$$\begin{aligned}
&\mathbb{P}\left(L_i \geq \left(1 - \frac{s+2}{s^4}\right) sL_{i-1} \mid L_{i-1}, B_{i-1}\right) \\
>&\mathbb{P}\left(\widehat{L}_i \geq \left(1 - \frac{s+2}{s^4}\right) s\widehat{L}_{i-1} \mid \widehat{L}_{i-1} = L_{i-1}, \widehat{B}_{i-1} = B_{i-1}\right) \\
>&1 - n^{-C}
\end{aligned}$$

for any fixed $C > 1$ and sufficiently large n. Thus, with probability $> \left(1 - n^{-C}\right)^{\ell_2'} > \left(1 - n^{-C}\right)^n > 1 - n^{1-C}$, all the levels constructed at Stage 2 have growth factor at

least $s(1-(s+2)/s^4)$. With probability $> \left(1-n^{-\frac{8}{7}}\right)\left(1-n^{1-C}\right) \geq 1-n^{-\frac{9}{8}}$,

$$\ell_2 < \log_{(1-(s+2)/s^4)s} n = \frac{1}{1+\log_s(1-(s+2)/s^4)} \log_s n$$

and $B_{\ell_2} \geq n/s^4$ and for any $i \leq \ell_2$, $B_i > \left((1-(s+2)/s^4)\, s\right)^i$.

Stage 3: Trakhtenbrot and Barzdin proved that for sufficiently large n,

$$\prod_{r=n/s^5}^{(1-2^{-s})n} \mathbb{P}\left(\widehat{B}(r) \geq r + C_s n\right) > 1 - n^{-C}$$

for a constant $C > 0$ and another constant C_s only depending on s. We then know

$$\prod_{r=n/s^5}^{(1-2^{-s})n} \mathbb{P}\left(B(r) \geq r + C_s n\right) > \prod_{r=n/s^5}^{(1-2^{-s})n} \mathbb{P}\left(\widehat{B}(r) \geq r + C_s n\right) > 1 - n^{-C}$$

for a constant $C > 0$ and another constant C_s only depending on s. This means that all the levels constructed at Stage 3 have at least $C_s n$ vertices with high probability. Formally, when $n \to +\infty$, with probability greater than $1 - n^{-C}$, $\ell_3' < \frac{n}{C_s n} = \frac{1}{C_s}$.

So far, after Stage 3, there are only $n/2^s$ unoccupied vertices in the graph. If $s \geq \log_2 n - \log_2(C' \log_s n)$ for some constant $C' > 0$, we have $\frac{n}{2^s} \leq \frac{C' n \log_s n}{n} = O(\log_s n)$. That is, the number of unoccupied vertices is $O(\log_s n)$. No matter what will happen in Stage 4 to 6, in the worst case, the diameter of the graph will be at most $\ell_1' + \ell_2' + \ell_3' + O(\log_s n) = O(\log_s n)$ and we are done.

However, if $s < \log_2 n - \log_2(C' \log_s n)$, we have to move on to the later stages.

Stage 4: We prove the boundary of the spanning ball starts shrinking in Stage 4.

$$\begin{aligned}
\mathbb{P}\left(L_i \le \frac{1.5s}{2^s} L_{i-1} \mid L_{i-1}, B_{i-1}\right) &\ge \mathbb{P}\left(X\left(sL_{i-1}, \frac{n-B_{i-1}}{n-s}\right) \le \frac{1.5s}{2^s} L_{i-1}\right) \\
&\ge \mathbb{P}\left(X\left(sL_{i-1}, \frac{n}{(n-s)2^s}\right) \le \frac{1.5s}{2^s} L_{i-1}\right) \\
&= 1 - \mathbb{P}\left(X\left(sL_{i-1}, \frac{n}{(n-s)2^s}\right) > \frac{1.5s}{2^s} L_{i-1}\right) \\
&\ge 1 - sL_{i-1} \cdot \left(\frac{n-s}{n} \cdot 1.5\right)^{\left(\frac{n}{n-s}-1.5\right)\left(\frac{sL_{i-1}}{2^{s+1}}\right)+\frac{3}{2}}
\end{aligned}$$

Because $s < \log_2 n - \log_2(C' \log_s n)$ and $L_{i-1} > (\log_2 n)^2$, it follows that when $n \to +\infty$, the above probability is at least $1 - n^{-C}$ for some constant $C > 1$. Formally, with probability at least $\left(1 - n^{-C}\right)^n > 1 - n^{1-C}$, all the levels constructed at Stage 4 have growth factor at most $\frac{1.5s}{2^s}$ and

$$\ell_4' < \log_{2^s/(1.5s)} n = \frac{\log_2 s}{s - \log_2(1.5s)} \log_s n$$

Stage 5: We show the growth factor at this stage is at most 2/3. Using the fact that $s \ge 2$, $L_{i-1} > 120 \log_2 n$ and $s < \log_2 n - \log_2(C' \log_s n)$, for sufficiently large n,

$$\begin{aligned}
\mathbb{P}\left(L_i \le \frac{2}{3} L_{i-1} \mid L_{i-1}, B_{i-1}\right) &\ge 1 - \mathbb{P}\left(X\left(sL_{i-1}, \frac{n}{(n-s)2^s}\right) > \frac{2}{3} L_{i-1}\right) \\
&\ge 1 - sL_{i-1} \cdot \left(\frac{(n-s)2^{s+1}}{3ns}\right)^{\left(\frac{ns}{(n-s)2^{s+1}}-\frac{1}{3}\right)L_{i+1}+\frac{3}{2}} \\
&> 1 - sL_{i-1} \cdot 2^{1-\frac{1}{30}L_{i-1}} \\
&> 1 - sL_{i-1} \cdot 2^{1-4\log_2 n} \\
&> 1 - n^{-3}
\end{aligned}$$

This implies that all levels constructed at Stage 5 have growth at most 2/3 and $\ell_5' < \log_{\frac{3}{2}}(\log_2 n)^2 < (4 \log_2 s) \log_s \log_2 n$ with probability greater than $(1 - n^{-3})^n >$

$1 - n^{-2}$.

Stage 6: We construct a logarithmic upper bound for the number of new vertices occupied at Stage 6. For some constant $C > 0$,

$$\begin{aligned}
&\mathbb{P}\left(B_{\ell_6} - B_{\ell_5} > \frac{C\log_2 n}{s}\right)\\
\leq&\mathbb{P}\left(L_i > \frac{C\log_2 n}{s} \mid L_{i-1} = 120\log_2 n + \frac{C\log_2 n}{s}, B_{i-1}\right)\\
\leq&\mathbb{P}\left(X\left(120s\log_2 n + C\log_2 n, \frac{n}{(n-s)2^s}\right) > \frac{C\log_2 n}{s}\right)\\
\leq&(120s + C)\log_2 n \cdot \left(\frac{C(n-s)2^s}{ns(120s+C)}\right)^{\left((120s+C)2^{-s} - \frac{C}{s}\right)\log_2\sqrt{n} + \frac{3}{2}}\\
\leq&(120s + C)\log_2 n \cdot 2^{\left(s + \log_2 \frac{C(n-s)}{ns(120s+C)}\right)\left((120s+C)2^{-s} - \frac{C}{s}\right)\log_2\sqrt{n} + \frac{3}{2}\left(s + \log_2 \frac{C(n-s)}{ns(120s+C)}\right)}
\end{aligned}$$

Simple algebra gives

$$\begin{aligned}
&\left(s + \log_2 \frac{C(n-s)}{ns(120s+C)}\right)\left(\frac{120s+C}{2^s} - \frac{C}{s}\right)\\
=&-C + \frac{120s+C}{2^s}\log_2 \frac{C(n-s)}{ns(120s+C)} - \frac{C}{s}\log_2 \frac{C(n-s)}{ns(120s+C)} + \frac{120s^2 + Cs}{2^s}
\end{aligned}$$

For any $s < \log_2 n - \log_2(C'\log_s n)$, all the addends expect the first item approach zero as s increases. Therefore, there exists some constant C_0 such that when $n \to +\infty$, $\mathbb{P}\left(B_{\ell_6} - B_{\ell_5} > \frac{C_0\log_2 n}{s}\right) < n^{-2}$. Formally, with probability greater than $1 - n^{-2}$,

$$\ell_6' \leq B_{\ell_6} - B_{\ell_5} \leq \frac{C_0\log_2 n}{s} = \frac{C_0\log_2 s}{s}\log_s n$$

Conclusion: With probability greater than $1 - n^{-\frac{10}{9}}$, the diameter of a RSG$^+(s)$ is at most $\sum_{i=1}^{6}\ell_i' = O(\log_s n)$. ■

With Theorem 4.3 and 5.2, we reach the fast convergence argument on the RSG$^+(s)$ model.

5.2.4 Fast convergence on RMG^{-}, RSG^{-}, $\mathrm{RMG}^{\pm}$ and $\mathrm{RSG}^{\pm}$

The conclusions drawn for random walks on random out-regular graphs can be generalized to the in-regular cases. Let A be the adjacency matrix of graph G. Denote by $G^{\top}$ the *transpose* of G defined by adjacency matrix $A^{\top}$. We can see that (1) $G^{\top}$ has exactly the same irreducible components as G; (2) The aperiodicity of G implies the aperiodicity of $G^{\top}$; (3) The diameter of the transpose graph is equal to the diameter of the original graph. These give us the properties of irreducibility, aperiodicity and logarithmic diameter for the in-regular models. Note that the irreducible component of a random in-regular graph is usually not closed. Hence, the fast convergence argument only holds when the walk is restricted to the unique irreducible component. According to Theorem 4.3, it remains to bound the maximum out-degree $s_0 = \arg\max_{u \in V} d_u$. This requires the lower bound assumption on the in-degree s as stated in the main theorem, because small in-degree results in large maximum out-degree of the graph (with respect to s).

A random s-in s-out graph can be viewed as the sum of a random out-regular graph and a random in-regular graph, generated independently of each other. Thus logarithmic diameter is trivial. The original paper by Fenner and Frieze [FF82] has already shown the strong connectivity of the random s-in s-out graphs for $s \geq 2$. As the entire graph is strongly connected, the connected component is surely closed and unique. Aperiodicity is established by the fact that sum graph retains all directed cycles in the original graphs.

Formal proof

The following facts are immediate observations from the definitions.

Fact 5.1 *For any $u, v \in V$, u and v are strongly connected in G if and only if they*

are strongly connected in $G^\top$.

Fact 5.2 *Graph* G *is* h*-partite if and only if graph* $G^\top$ *is* h*-partite.*

Fact 5.3 *The distance from* $u \in V$ *to* $v \in V$ *in* G *is equal to the distance from* v *to* u *in* $G^\top$.

Fact 5.1 tells us $G^\top$ has exactly the same irreducible components as G and Fact 5.2 shows the equivalence of the aperiodicity of G and $G^\top$. Fact 5.3 leads to $Diam(G) = Diam(G^\top)$. Because a random in-regular graph can be created by transposing a corresponding random out-regular graph, we can conclude the following statements.

Corollary 5.1 *With probability* $1 - o(1)$*, a RMG*$^-(s)$ *has a strongly connected component, denoted by* $\tilde{G} = (\tilde{V}, \tilde{E})$*, of size* $\tilde{n}$ *when* $n \to +\infty$*, 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* $s \geq 7$*; b) a random walk on* $\tilde{G}$ *is aperiodic.*

Corollary 5.2 *With probability* $1 - o(1)$*, the diameter of a RMG*$^-(s)$ *is* $\Theta(\log_s n)$.

Corollary 5.3 *With probability* $1 - o(1)$*, a RSG*$^-(s)$ *has a strongly connected component, denoted by* $\tilde{G} = (\tilde{V}, \tilde{E})$*, of size* $\tilde{n}$ *when* $n \to +\infty$*, and a)* $\lim_{n\to+\infty} \frac{\tilde{n}}{n} \geq C$ *for some constant* $C > 0.7968$ *when* $s \geq 2$ *or some* $C > 0.999$ *when* $s \geq 7$*; b) a random walk on* $\tilde{G}$ *is aperiodic.*

Corollary 5.4 *With probability* $1 - o(1)$*, the diameter of a RSG*$^-(s)$ *is* $\Theta(\log_s n)$.

Note that in these cases the irreducible component is usually not closed. Hence, the fast convergence argument only holds when the walk is restricted to the unique irreducible component. According to Theorem 4.3, to bound the convergence rate we still need the maximum out-degree $s_0 = \arg\max_{u\in V} d_u$. To prove fast convergence, we need a lower-bound assumption on the in-degree s.

Lemma 5.4 *Let* $s_0 = \arg\max_{u \in V} d_u$ *be the maximum out-degree of a RMG*$^-(s)$ *with* $s = \Omega\left(\left[\frac{\log n}{\log\log n}\right]^{1/C'}\right)$ *for some constant* $C' \geq 1$. *With probability* $1 - o(1)$, $s_0 = O(s^{C'+\varepsilon})$ *for any constant* $\varepsilon > 0$.

Proof According to the properties of a RMG$^-(s)$, the probability of $s_0 > ns$ is 0 and $\mathbb{P}(s_0 = ns) \leq n \cdot n^{-ns}$ is exponentially small. For any $k < ns$,

$$
\begin{aligned}
&\mathbb{P}(s_0 \geq k) \leq n \cdot \mathbb{P}(\text{a particular vertex has out-degree at least } k) \\
\leq & n \cdot \binom{ns}{k} \left(\frac{1}{n}\right)^k \\
\leq & \frac{\sqrt{2\pi ns}\left(\frac{ns}{e}\right)^{ns} e^{\frac{1}{12ns}}}{\sqrt{2\pi k}\left(\frac{k}{e}\right)^{k} e^{\frac{1}{12k+1}} \cdot \sqrt{2\pi(ns-k)}\left(\frac{ns-k}{e}\right)^{ns-k} e^{\frac{1}{12(ns-k)+1}}} \cdot n\left(\frac{1}{n}\right)^k \\
\leq & \sqrt{\frac{n^3 s}{2\pi k(ns-k)}} \cdot \frac{e^{\frac{1}{12ns}}(ns)^{ns}}{(nk)^k (ns-k)^{ns-k}} \\
\leq & \sqrt{\frac{1}{2\pi}} \cdot \exp\Bigg(\log n + ns\log(ns) - k\log k \\
& - (ns-k)\log(ns-k) - k\log n + \frac{1}{12ns}\Bigg)
\end{aligned}
$$

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

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

Let $k = s^c$ where $c = C' + \varepsilon$. If $k \geq ns$ then $\mathbb{P}(s_0 \geq k)$ is exponentially small as discussed above. Otherwise we have $\left(1 - \frac{ns}{k}\right)\log\left(1 - \frac{k}{ns}\right) \leq 1$ in our case. Also notice that $\frac{1}{12ns} \leq 1$. The exponent is then upper bounded by $\log n + s^c - s^c(c - 1)\log s + 1$. Letting $\log n \leq s^c(c - 1 - 0.5\varepsilon)\log s$ gives

$$
s \geq \left[\frac{c\log n}{(c-1-0.5\varepsilon)W\left(\frac{c\log n}{c-1-0.5\varepsilon}\right)}\right]^{\frac{1}{c}} = o\left(\left[\frac{\log n}{\log\log n}\right]^{\frac{1}{C'}}\right)
$$

where $W(x)$ is the Lambert W-function [Lam58], defined by $W(x)e^{W(x)} = x$ for $x \geq -e^{-1}$. ■

Combining Lemma 5.4 with Theorem 4.3, we reach the fast convergence of a random walk on a $\mathrm{RMG}^{-}(s)$ with $s = \Omega\left(\left[\frac{\log n}{\log\log n}\right]^{1/C'}\right)$.

The same convergence property holds on a $\mathrm{RSG}^{-}(s)$.

Lemma 5.5 *Let $s_0 = \arg\max_{u \in V} d_u$ be the maximum out-degree of a RSG^{-}(s) with $s = \Omega\left(\left[\frac{\log n}{\log\log n}\right]^{1/C'}\right)$ for some constant $C' \geq 1$. With probability $1 - o(1)$, $s_0 = O(s^{C'+\varepsilon})$ for any constant $\varepsilon > 0$.*

Proof From the definition of a $\mathrm{RSG}^{-}(s)$, the probability of $s_0 \geq n$ is 0. If we have large $s = \Theta(n)$, then the argument automatically holds because $s_0 \leq n - 1 = O(s)$. Otherwise $s = o(n)$, $\mathbb{P}(s_0 = n-1) \leq n \cdot \left(\frac{s}{n-1}\right)^{n-1}$ is exponentially small. For any $k < n - 1$, using the union bound,

$$
\begin{aligned}
&\mathbb{P}(s_0 \geq k) \leq n \cdot \mathbb{P}(\text{a particular vertex has at least } k \text{ neighbors}) \\
\leq & n \cdot \binom{n-1}{k} \left[\frac{\binom{1}{1}\binom{n-2}{s-1}}{\binom{n-1}{s}}\right]^k \\
= & n \cdot \binom{n-1}{k} \left[\frac{s}{n-1}\right]^k \\
\leq & \frac{n \cdot \sqrt{2\pi(n-1)}\left(\frac{n-1}{e}\right)^{n-1} e^{\frac{1}{12(n-1)}}}{\sqrt{2\pi k}\left(\frac{k}{e}\right)^k e^{\frac{1}{12k+1}} \cdot \sqrt{2\pi(n-k-1)}\left(\frac{n-k-1}{e}\right)^{n-k-1} e^{\frac{1}{12(n-k-1)+1}}} \left(\frac{s}{n-1}\right)^k \\
\leq & \sqrt{\frac{n^2(n-1)}{2\pi k(n-k-1)}} \cdot \frac{e^{\frac{1}{12(n-1)}}(n-1)^{n-k-1}s^k}{k^k(n-k-1)^{n-k-1}} \\
\leq & \sqrt{\frac{1}{2\pi}} \cdot \exp\Bigg(\log n + \frac{1}{12(n-1)} + (n-k-1)\log(n-1) \\
& + k\log s - k\log k - (n-k-1)\log(n-k-1)\Bigg)
\end{aligned}
$$

Again we choose a value of k such that the exponent in the last expression goes to $-\infty$. The exponent can be reshaped as

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

Because $\frac{1}{12(n-1)}$ and $\left(1 - \frac{n-1}{k}\right)\log\left(1 - \frac{k}{n-1}\right)$ are both at most 1 in our case, letting $c = C' + \varepsilon$ and $k = s^c$ gives us

$$\log n + 1 - s^c(c-1)\log s + s^c$$

For $s = \Omega\left(\left[\frac{\log n}{\log\log n}\right]^{1/C'}\right)$, the expression goes to $-\infty$ and completes the proof. ■

Thus we have proved the $\mathrm{RSG}^-(s)$ case in the main theorem.

The model of random s-in s-out graphs is a random graph model first introduced by Fenner and Frieze [FF82], which can be viewed as the sum of a random out-regular graph and a random in-regular graph, generated independently of each other. We provide a brief proof for $\mathrm{RMG}^{\pm}(s)$ by simply combining the previously proved arguments for $\mathrm{RMG}^+(s)$ and $\mathrm{RMG}^-(s)$. The same result for $\mathrm{RSG}^{\pm}(s)$ can be similarly achieved based on the arguments for $\mathrm{RSG}^+(s)$ and $\mathrm{RSG}^-(s)$.

The original paper by Fenner and Frieze [FF82] has already proved the strong connectivity of the random s-in s-out graphs for $s \geq 2$. As the entire graph is strongly connected, the connected component is surely closed and unique. As for aperiodicity, since $\tilde{V}$ is strongly connected, we only need to show one of the $v \in \tilde{V}$ is aperiodic. Without loss of generality, let $v \in \tilde{V}^+$ and then v is aperiodic in the $\mathrm{RMG}^+(s)$ with high probability, which means that there exists a sufficiently large ℓ_0 such that for all $\ell \geq \ell_0$, there is a directed cycle of length ℓ over v. Because we only add edges onto the graph when generating the $\mathrm{RMG}^-(s)$, the sum graph $\mathrm{RMG}^{\pm}(s)$

still retains such cycles and v is aperiodic. The logarithmic diameter of $\mathrm{RMG}^{\pm}(s)$ is due to

$$Diam(G_1 + G_2) \leq Diam(G_1) + Diam(G_2)$$

for any graphs G_1 and G_2.

Again, combining with Theorem 4.3 we reach the fast convergence property stated in the main theorem, and the same argument holds on a $\mathrm{RSG}^{\pm}(s)$.

5.2.5 Fast convergence on RDG and RG

Among all the models in this chapter, $\mathrm{RDG}(s)$ is the most constrained one, due to the strong dependence and strict restrictions on the edge selections (same in the undirected model) and the lack of symmetry (while the undirected model has symmetry). Unlike the previous cases, the proof is based on enumeration.

Previous works have contributed the irreducibility of $\mathrm{RDG}(s)$. The proof of aperiodicity starts with the asymptotic enumeration of regular digraphs. The key to this first step is the bijection between regular digraphs and binary square matrices with equal line sums. Let $N(n, s)$ be the number of s-regular digraphs with n vertices. We present an asymptotic formula for $N(n, s)$, by unifying the asymptotic results on binary square matrices with equal line sums (Lemma 5.7). We also observe the bijection between regular digraphs with n vertices and colored regular bipartite (undirected) graphs with $2n$ vertices. Let $G = (V, E)$ be a regular digraph of fixed degree s. We construct a regular bipartite graph $G' = (V', E')$ where $|V'| = 2|V|$ as follows. Without loss of generality, let $V = \{v_1, v_2, \ldots, v_n\}$ and $V' = \{v'_1, v'_2, \ldots, v'_{2n}\}$ with $\{v'_1, v'_2, \ldots, v'_n\}$ of one color and $\{v'_{n+1}, v'_{n+2}, \ldots, v'_{2n}\}$ of the other. Let $(v'_i, v'_{n+j}) \in E'$ if and only if $(v_i, v_j) \in E$. We can see such a regular bipartite graph G' is unique for each regular digraph G and vice versa.

To show the aperiodicity of RDG(s), we again need to inverse-exponentially upper-bound the probability of the graph being h-partite, denoted by p_h. If V can be partitioned into h disjoint subsets $V_0, V_1, \ldots, V_{h-1}$, such that all edges from V_i go to $V_{(i+1) \mod h}$, because the graph is both in-regular and out-regular, we must have $|V_0| = |V_1| = \ldots = |V_{h-1}| = \frac{n}{h}$ and $h \leq \frac{n}{s}$. Notice that the number of possible edge combinations from V_i going to $V_{(i+1) \mod h}$ is exactly the number of colored s-regular bipartite (undirected) graphs of size $\frac{n}{h}$, which is $N(\frac{n}{h}, s)$. Based on the bijection we constructed above, this gives

$$p_h \leq \frac{1}{h} \cdot \binom{n}{\frac{n}{h}, \frac{n}{h}, \ldots, \frac{n}{h}} \cdot \frac{\left[N\left(\frac{n}{h}, s\right)\right]^h}{N(n, s)}$$

With the asymptotic enumeration results we complete the proof (Lemma 5.8).

Unlike all the previous cases where we achieve fast convergence by proving logarithmic diameter, for random regular digraphs the polynomial convergence rate follows from a lower bound on the first non-zero eigenvalue of the Laplacian matrix. Note that the walk matrix $P = \frac{1}{s}A$ of a random walk on a RDG(s) is a doubly stochastic matrix, and so is the matrix $\frac{1}{2}(P + P^\top)$. Also observe that the Perron vector of any regular digraph is always the uniform distribution over the vertices. Using a spectral lower bound for doubly stochastic matrices due to Fiedler [Fie72], we complete the proof.

Random regular undirected graphs are much more widely studied than directed ones, mainly owing to the symmetry of undirected graphs. Previous works have established connectivity and enumeration results. Because the only periodic case for an undirected graph is being bipartite, we only need to bound the probability p_2. This is again done by enumeration. In the proof of the preceding digraph case we have already deduced an asymptotic formula for the number of bipartite s-regular

undirected graphs with n vertices, which is $\binom{n}{\frac{n}{2}} \cdot N(\frac{n}{2}, s)$. Denote by $N'(n, s)$ the number of s-regular undirected graphs with n vertices. We have

$$p_2 \leq \frac{1}{2}\binom{n}{\frac{n}{2}} \cdot \frac{N(\frac{n}{2}, s)}{N'(n, s)}$$

Using the same spectral lower bound for doubly stochastic matrices as in the preceding digraph case, we have the polynomial convergence rate. Proof details are presented below.

Proof of Theorem 5.1 for random regular digraphs

In this section we study random walks on RDG(s). Because the edges in this case are no longer chosen independently, the proof is done mainly by enumeration.

Irreducibility and aperiodicity

Previous works have shown the irreducibility [Wor99].

Lemma 5.6 *With probability $1 - o(1)$, a RDG(s) is strongly connected when $s \geq 2$.*

Now we prove aperiodicity, starting with the asymptotic enumeration of regular digraphs.

Lemma 5.7 *Let $N(n, s)$ be the number of s-regular digraphs of size n.*

$$N(n, s) = \begin{cases} \frac{(ns)!}{(s!)^{2n}} \exp\left[-\frac{(s-1)^2}{2} + O\left(\frac{s^3}{n}\right)\right] & \text{if } 1 \leq s \leq \frac{n}{2} \\ N(n, n-s) & \text{if } \frac{n}{2} < s < n \\ 1 & \text{if } s = n \end{cases}$$

$N(n, s)$ is also the number of colored s-regular bipartite (undirected) graphs of size $2n$.

Proof We first show the bijection between regular digraphs of size n and colored regular bipartite (undirected) graphs of size $2n$. Let $G = (V, E)$ be a regular digraph of fixed degree s. We construct a regular bipartite graph $G' = (V', E')$ where $|V'| = 2|V|$ as follows. Without loss of generality, let $V = \{v_1, v_2, \ldots, v_n\}$ and $V' = \{v'_1, v'_2, \ldots, v'_{2n}\}$ with $\{v'_1, v'_2, \ldots, v'_n\}$ of one color and $\{v'_{n+1}, v'_{n+2}, \ldots, v'_{2n}\}$ of the other. Let $(v'_i, v'_{n+j}) \in E'$ if and only if $(v_i, v_j) \in E$. We can see such a regular bipartite graph G' is unique for each regular digraph G and vice versa. Note that this bijection is connectivity-preserving. To see this, consider that for a directed graph there are two cases of being disconnected. The first case is that there exist nonempty $V_1 \subset V$ and $V_2 \subset V$ with only edges going from V_1 to V_2 and no edge going back. This is impossible in a regular digraph because the in-degree of V_1 must be equal to its out-degree. The other case is no edge between V_1 and V_2, where the corresponding bipartite graph G' is also disconnected.

In order to prove the aperiodicity of a RDG(s), we first need to do enumeration for regular digraphs. It's easy to see another bijection: the one between regular digraphs and binary square matrices with equal line sums. Although little previous work has been done on the enumeration of regular digraphs, we are fortunate to have asymptotic results on the enumeration of binary square matrices with equal line sums. Let $N(n, s)$ be the number of regular digraphs with n vertices of fixed in-degree and out-degree equal to s, which is also the number of $n \times n$ binary matrices with equal line sums s and the number of regular bipartite graphs. McKay [McK84] proved that for $1 \leq s < \frac{1}{6}n$,

$$N(n, s) = \frac{(ns)!}{(s!)^{2n}} \exp\left[-\frac{(s-1)^2}{2} + O\left(\frac{s^3}{n}\right)\right] \tag{5.1}$$

and Canfield and McKay [CM05] showed for $s \leq \frac{1}{2}n$ and $s = \Theta(n)$,

$$N(n,s) = \frac{\binom{n}{s}^{2n}}{\binom{n^2}{ns}} \left(1 - \frac{1}{n}\right)^{n-1} \exp\left(\frac{1}{2} + o(1)\right) \tag{5.2}$$

We are able to unify these two asymptotic results and show that the latter case also satisfies the former formula. For $s \leq \frac{1}{2}n$ and $s = \Theta(n)$,

$$\begin{aligned}
N(n,s) =& \frac{\binom{n}{s}^{2n}}{\binom{n^2}{ns}} \left(1 - \frac{1}{n}\right)^{n-1} \exp\left(\frac{1}{2} + o(1)\right) \\
=& \frac{\left(\frac{n!}{s!(n-s)!}\right)^{2n}}{\frac{(n^2)!}{(ns)!(n^2-ns)!}} \exp\left(o(1) - \frac{1}{2}\right) \\
=& \frac{(ns)!}{(s!)^{2n}} \cdot \frac{\left(\frac{n!}{(n-s)!}\right)^{2n}}{\frac{(n^2)!}{(n^2-ns)!}} \exp\left(O(1)\right) \\
=& \frac{(ns)!}{(s!)^{2n}} \cdot \frac{\left(\frac{\sqrt{2\pi n}\cdot n^n \cdot e^{n-s}}{e^n \cdot \sqrt{2\pi(n-s)}\cdot (n-s)^{n-s}}\right)^{2n}}{\frac{\sqrt{2\pi n^2}\cdot n^{2n^2}\cdot e^{n^2-ns}}{e^{n^2}\cdot\sqrt{2\pi(n^2-ns)}\cdot(n^2-ns)^{n^2-ns}}} \exp\left(O(1)\right) \\
=& \frac{(ns)!}{(s!)^{2n}} \cdot \frac{n^{2n^2} e^{ns} (n^2-ns)^{n^2-ns}}{e^{2ns}(n-s)^{2n(n-s)} n^{2n^2}} \cdot \left(\frac{n}{n-s}\right)^{n-\frac{1}{2}} \exp\left(O(1)\right) \\
=& \frac{(ns)!}{(s!)^{2n}} \cdot \frac{n^{n^2-ns}(n-s)^{n^2-ns} n^{n-\frac{1}{2}}}{e^{ns}(n-s)^{2n(n-s)}(n-s)^{n-\frac{1}{2}}} \exp\left(O(1)\right) \\
=& \frac{(ns)!}{(s!)^{2n}} \cdot \frac{n^{n^2-(s-1)n-\frac{1}{2}}}{e^{ns}(n-s)^{n^2-(s-1)n-\frac{1}{2}}} \exp\left(O(1)\right) \\
=& \frac{(ns)!}{(s!)^{2n}} \cdot \left(1 - \frac{s}{n}\right)^{(s-1)n+\frac{1}{2}-n^2} \exp\left(O(1) - ns\right)
\end{aligned}$$

Let $C = \frac{s}{n} \leq \frac{1}{2}$. Since $s = \Theta(n)$,

$$\begin{aligned}
N(n,s) =& \frac{(ns)!}{(s!)^{2n}} \exp\left(\log(1-C) \cdot \left((s-1)n + \frac{1}{2} - n^2\right) + O(1) - ns\right) \\
=& \frac{(ns)!}{(s!)^{2n}} \exp\left((C-1)n^2 \log(1-C) - n\log(1-C) + O(1) - ns\right) \\
=& \frac{(ns)!}{(s!)^{2n}} \exp\left[-\frac{(s-1)^2}{2} + O\left(\frac{s^3}{n}\right)\right]
\end{aligned}$$

Using complement graphs, it is apparent that $N(n,s) = N(n,n-s)$ for $\frac{n}{2} \leq s < n$. When $s = n$, the only possible regular digraph in this case is the complete graph so $N(n,n) = 1$. Combining all the above cases completes the proof. ■

Lemma 5.8 *With probability* $1 - o(1)$*, a RDG(s) is aperiodic.*

Proof A regular digraph $G = (V, E)$ is h-partite if V can be partitioned into h subsets, $V_0, V_1, \ldots, V_{h-1}$, such that all edges from V_i go to $V_{(i+1) \mod h}$. Because the graph is regular, we must have $|V_0| = |V_1| = \ldots = |V_{h-1}| = \frac{n}{h}$ and $h \leq \frac{n}{s}$. Also we notice that the number of possible edge combinations from V_i going to $V_{(i+1) \mod h}$ is exactly the number of colored s-regular bipartite (undirected) graphs of size $\frac{n}{h}$, which is $N(\frac{n}{h}, s)$. Denote by p_h the probability of a RDG(s) being h-partite. The proof is done by showing p_h goes to 0 exponentially fast for all $2 \leq h \leq \frac{n}{s}$. The case where $s > \frac{n}{2}$ is trivial. Thus below we only consider $s \leq \frac{n}{2}$. We first prove the argument holds when $s = o(n)$. For $2 \leq h \leq \frac{n}{2s}$,

$$p_h \leq \frac{1}{h} \cdot \binom{n}{\frac{n}{h}, \frac{n}{h}, \ldots, \frac{n}{h}} \cdot \frac{\left[N\left(\frac{n}{h}, s\right)\right]^h}{N(n,s)}$$

According to Lemma 5.7,

$$\begin{aligned} N(n,s) =& \frac{(ns)!}{(s!)^{2n}} \exp\left[-\frac{(s-1)^2}{2} + O\left(\frac{s^3}{n}\right)\right] \\ =& \frac{\sqrt{2\pi ns}(ns)^{ns}e^{2ns}}{e^{ns}\left[\sqrt{2\pi s} \cdot s^s\right]^{2n}} \exp\left[-\frac{(s-1)^2}{2} + O\left(\frac{s^3}{n}\right)\right] \\ =& \frac{\sqrt{ns}}{s^n(2\pi)^{n-\frac{1}{2}}} \cdot \left(\frac{en}{s}\right)^{ns} \exp\left[-\frac{(s-1)^2}{2} + O\left(\frac{s^3}{n}\right)\right] \end{aligned}$$

and

$$\left[N\left(\frac{n}{h},s\right)\right]^h = \left[\frac{\sqrt{s\frac{n}{h}}}{s^{\frac{n}{h}}(2\pi)^{\frac{n}{h}-\frac{1}{2}}}\cdot\left(\frac{en}{hs}\right)^{s\frac{n}{h}}\exp\left[-\frac{(s-1)^2}{2}+O\left(\frac{s^3h}{n}\right)\right]\right]^h$$
$$=\frac{\left(\frac{ns}{h}\right)^{\frac{1}{2}h}}{s^n(2\pi)^{n-\frac{h}{2}}}\cdot\left(\frac{en}{hs}\right)^{ns}\exp\left[-\frac{h(s-1)^2}{2}+O\left(\frac{s^3h^2}{n}\right)\right]$$

Also,

$$\binom{n}{\frac{n}{h},\frac{n}{h},\ldots,\frac{n}{h}} = \frac{n!}{\left(\frac{n}{h}!\right)^h} \le \frac{\sqrt{2\pi n}\cdot n^n e^{n+\frac{1}{12n}}}{e^n\left(\sqrt{2\pi\frac{n}{h}}\left(\frac{n}{h}\right)^{\frac{n}{h}}\right)^h}$$
$$=\frac{\sqrt{2\pi n}\cdot h^n e^{\frac{1}{12n}}}{\left(\frac{2\pi n}{h}\right)^{\frac{1}{2}h}} = (2\pi n)^{\frac{1}{2}(1-h)}\cdot h^{n+\frac{1}{2}h}\cdot e^{\frac{1}{12n}}$$

We then have

$$\begin{aligned}
p_h \le & \frac{h^{n+\frac{1}{2}h-1}}{(2\pi n)^{\frac{1}{2}(h-1)}}\cdot\frac{\left(\frac{ns}{h}\right)^{\frac{1}{2}h}s^n(2\pi)^{n-\frac{1}{2}}}{s^n(2\pi)^{n-\frac{1}{2}h}\sqrt{ns}}\cdot\left(\frac{1}{h}\right)^{ns} \\
& \cdot\exp\left[\frac{(s-1)^2}{2}-\frac{h(s-1)^2}{2}+O\left(\frac{s^3h^2}{n}\right)\right] \\
= & \frac{h^{n+\frac{1}{2}h-1}}{(2\pi n)^{\frac{1}{2}(h-1)}}\cdot\frac{(2\pi ns)^{\frac{1}{2}(h-1)}}{h^{\frac{1}{2}h}}\cdot\left(\frac{1}{h}\right)^{ns} \\
& \cdot\exp\left[-\frac{(h-1)(s-1)^2}{2}+O\left(\frac{s^3h^2}{n}\right)\right] \\
= & \frac{s^{\frac{1}{2}(h-1)}}{h^{ns-n+1}}\exp\left[-\frac{(h-1)(s-1)^2}{2}+O\left(\frac{s^3h^2}{n}\right)\right] \\
= & \exp\left\{-\frac{1}{2}(h-1)[(s-1)^2-\log s]-(ns-n+1)\log h+O\left(\frac{s^3h^2}{n}\right)\right\}
\end{aligned}$$

For $s \ge 2$, we have $(s-1)^2 - \log s > 0$. When $h = O(1)$, $O\left(\frac{s^3h^2}{n}\right) = O\left(\frac{s^3}{n}\right) = o(ns)$ since $s = o(n)$. When $h = \omega(1)$, as $h < \frac{n}{s}$, $O\left(\frac{s^3h^2}{n}\right) = O(ns) = o(ns\log h)$. Hence, p_h goes to 0 exponentially fast.

For $\frac{n}{2s} < h < \frac{n}{s}$, surely $h = \omega(1)$ as $s = o(n)$. Also, $\left(\frac{n}{h} - s\right)^3 < s^3$ since $\frac{n}{2h} < s < \frac{n}{h}$. Then we have

$$\begin{aligned}\left[N\left(\frac{n}{h}, \frac{n}{h} - s\right)\right]^h =& \left[\frac{\sqrt{\frac{n}{h}\left(\frac{n}{h} - s\right)}}{\left(\frac{n}{h} - s\right)^{\frac{n}{h}} (2\pi)^{\frac{n}{h} - \frac{1}{2}}} \cdot \left(\frac{en}{h\left(\frac{n}{h} - s\right)}\right)^{\frac{n}{h}\left(\frac{n}{h} - s\right)} \right. \\ & \left. \cdot \exp\left[-\frac{\left(\frac{n}{h} - s - 1\right)^2}{2} + O\left(\frac{\left(\frac{n}{h} - s\right)^3 h}{n}\right)\right]\right]^h \\ =& \frac{\left(\frac{n}{h}\left(\frac{n}{h} - s\right)\right)^{\frac{1}{2}h}}{\left(\frac{n}{h} - s\right)^n (2\pi)^{n - \frac{h}{2}}} \cdot \left(\frac{en}{h\left(\frac{n}{h} - s\right)}\right)^{n\left(\frac{n}{h} - s\right)} \\ & \cdot \exp\left[-\frac{h\left(\frac{n}{h} - s - 1\right)^2}{2} + O\left(\frac{s^3 h^2}{n}\right)\right]\end{aligned}$$

so that

$$\begin{aligned}p_h \leq& \frac{1}{h} \cdot \binom{n}{\frac{n}{h}, \frac{n}{h}, \ldots, \frac{n}{h}} \cdot \frac{\left[N(\frac{n}{h}, \frac{n}{h} - s)\right]^h}{N(n, s)} \\ =& \frac{h^{n + \frac{1}{2}h - 1}}{(2\pi n)^{\frac{1}{2}(h-1)}} \cdot \frac{\left(\frac{n}{h}\left(\frac{n}{h} - s\right)\right)^{\frac{1}{2}h} s^n (2\pi)^{n - \frac{1}{2}}}{\left(\frac{n}{h} - s\right)^n (2\pi)^{n - \frac{1}{2}h} \sqrt{ns}} \cdot \left(\frac{en}{h\left(\frac{n}{h} - s\right)}\right)^{n\left(\frac{n}{h} - s\right)} \\ & \cdot \left(\frac{en}{s}\right)^{-ns} \cdot \exp\left[\frac{(s-1)^2}{2} - \frac{h\left(\frac{n}{h} - s - 1\right)^2}{2} + O\left(\frac{s^3 h^2}{n}\right)\right]\end{aligned}$$

Notice that function $\left(\frac{ey}{x}\right)^x$ with constraint $0 < x \leq \frac{1}{2}y$ reaches its maximum at $x = \frac{1}{2}y$, which implies

$$\left(\frac{en}{h\left(\frac{n}{h} - s\right)}\right)^{\frac{n}{h} - s} \leq (2e)^{\frac{n}{2h}}$$

and

$$
\begin{aligned}
p_h \leq & h^{n+\frac{1}{2}h-1} \cdot \frac{\sqrt{2\pi n}}{\sqrt{ns}} \left(\frac{n}{2\pi hn}\right)^{\frac{1}{2}h} \cdot \frac{(2\pi)^{\frac{1}{2}(h-1)}}{\left(\frac{n}{h}-s\right)^{n-\frac{1}{2}h}} \cdot s^n \cdot \left(\frac{2s}{n}\right)^{ns} \\
& \cdot (2e)^{\frac{n^2}{2h}-ns} \cdot \exp\left[\frac{(s-1)^2}{2} - \frac{h\left(\frac{n}{h}-s-1\right)^2}{2} + O\left(\frac{s^3h^2}{n}\right)\right] \\
= & s^{n-\frac{1}{2}} \left(\frac{n}{h}-s\right)^{\frac{1}{2}h-n} \cdot h^{n-1}(2e)^{\frac{n^2}{2h}-ns} \left(\frac{2s}{n}\right)^{ns} \\
& \cdot \exp\left[\frac{(s-1)^2}{2} - \frac{h\left(\frac{n}{h}-s-1\right)^2}{2} + O\left(\frac{s^3h^2}{n}\right)\right] \\
= & s^{-\frac{1}{2}}h^{-1} \left(\frac{n}{h}-s\right)^{\frac{1}{2}h-n} \cdot (2e)^{\frac{n^2}{2h}-ns} \cdot \exp\Bigg[n\log s + n\log h + ns\log 2 + ns\log s \\
& - ns\log n - \frac{n^2}{2h} - \frac{hs^2}{2} - hs - \frac{h}{2} + ns + n + \frac{1}{2}s^2 - s + O\left(\frac{s^3h^2}{n}\right)\Bigg] \\
= & s^{-\frac{1}{2}}h^{-1} \left(\frac{n}{h}-s\right)^{\frac{1}{2}h-n} \cdot (2e)^{\frac{n^2}{2h}-ns} \cdot \exp\Bigg[n\log s + ns\log 2 + ns\log s \\
& - \left(1 - \frac{\log h}{s\log n}\right) ns\log n - \frac{n^2}{2h} - \frac{hs^2}{2} - hs - \frac{h}{2} + ns + n + \frac{1}{2}s^2 \\
& - s + O\left(\frac{s^3h^2}{n}\right)\Bigg]
\end{aligned}
$$

Notice that $O\left(\frac{s^3h^2}{n}\right) = O(ns)$ for $h < \frac{n}{s}$ and $\frac{n^2}{2h} - ns < 0$ for $s > \frac{n}{2h}$. Also, $1 - \frac{\log h}{s\log n} > 0$ for any $s \geq 2$, we have p_h going to 0 exponentially fast.

The case where $h = \frac{n}{s}$ is deferred to the end of this proof.

Now we study the case when $s = \Theta(n) < \frac{1}{2}n$ and $\frac{n}{h} - s = \Theta(n)$. In this case we have $\epsilon n \leq s \leq \left(\frac{1}{h} - \epsilon\right) n$ for some positive constant $\epsilon > 0$ and $2 \leq h < \frac{n}{s}$ is surely $O(1)$. Let $C = \frac{s}{n} < \frac{1}{2}$ so $0 < \epsilon \leq \lim_{n\to+\infty} C \leq \frac{1}{h} - \epsilon < \frac{1}{2}$. When $2 \leq h \leq \frac{n}{2s}$, we have $s = \Theta(n) = \Theta(\frac{n}{h})$.

$$
p_h \leq \frac{1}{h} \cdot \binom{n}{\frac{n}{h}, \frac{n}{h}, \ldots, \frac{n}{h}} \cdot \frac{\left[N\left(\frac{n}{h}, s\right)\right]^h}{N(n,s)}
$$

According to Lemma 5.7,

$$\begin{aligned}
N(n,s) =& \frac{(ns)!}{(s!)^{2n}} \exp\Big((C-1)n^2\log(1-C) - n\log(1-C) + O(1) - ns\Big) \\
=& \frac{\sqrt{2\pi ns}(ns)^{ns}e^{2ns}}{e^{ns}\left[\sqrt{2\pi s}\cdot s^s\right]^{2n}} \exp\Big((C-1)n^2\log(1-C) - n\log(1-C) \\
& + O(1) - ns\Big) \\
=& \frac{\sqrt{ns}}{s^n(2\pi)^{n-\frac{1}{2}}} \cdot \left(\frac{n}{s}\right)^{ns} \exp\Big((C-1)n^2\log(1-C) \\
& - n\log(1-C) + O(1)\Big)
\end{aligned}$$

and

$$\begin{aligned}
\left[N\left(\frac{n}{h}, s\right)\right]^h =& \left[\frac{\sqrt{s\frac{n}{h}}}{s^{\frac{n}{h}}(2\pi)^{\frac{n}{h}-\frac{1}{2}}} \cdot \left(\frac{n}{hs}\right)^{s\frac{n}{h}} \exp\left((hC-1)\left(\frac{n}{h}\right)^2\log(1-hC)\right.\right. \\
& \left.\left. - \log(1-hC)\cdot\frac{n}{h} + O(1)\right)\right]^h \\
=& \frac{\left(\frac{ns}{h}\right)^{\frac{1}{2}h}}{s^n(2\pi)^{n-\frac{h}{2}}} \cdot \left(\frac{n}{hs}\right)^{ns} \exp\left((hC-1)\frac{n^2}{h}\log(1-hC)\right. \\
& \left. - n\log(1-hC) + O(h)\right)
\end{aligned}$$

so that

$$\begin{aligned}
p_h \leq & \frac{h^{n+\frac{1}{2}h-1}}{(2\pi n)^{\frac{1}{2}(h-1)}} \cdot \frac{\left(\frac{ns}{h}\right)^{\frac{1}{2}h} s^n (2\pi)^{n-\frac{1}{2}}}{s^n (2\pi)^{n-\frac{1}{2}h}\sqrt{ns}} \cdot \left(\frac{1}{h}\right)^{ns} \cdot \exp\Bigg[(hC-1)\frac{n^2}{h}\log(1-hC) \\
& - n\log(1-hC) + O(h) - (C-1)n^2\log(1-C) + n\log(1-C) - O(1)\Bigg] \\
= & \frac{h^{n+\frac{1}{2}h-1}}{(2\pi n)^{\frac{1}{2}(h-1)}} \cdot \frac{(2\pi ns)^{\frac{1}{2}(h-1)}}{h^{\frac{1}{2}h}} \cdot \left(\frac{1}{h}\right)^{ns} \cdot \exp\Bigg[\Bigg(\left(C-\frac{1}{h}\right)\log(1-hC) \\
& - (C-1)\log(1-C)\Bigg)n^2 + (\log(1-C) - \log(1-hC))n + O(h)\Bigg] \\
= & \frac{s^{\frac{1}{2}(h-1)}}{h^{ns-n+1}} \cdot \exp\Bigg[\left(\left(C-\frac{1}{h}\right)\log(1-hC) - (C-1)\log(1-C)\right)n^2 \\
& + (\log(1-C) - \log(1-hC))n + O(h)\Bigg]
\end{aligned}$$

Notice that function $\left(y-\frac{1}{x}\right)\log(1-xy)$ with constraints $xy \leq \frac{1}{2}$ and $x \geq 2$ reaches its maximum at $x = 2$. Thus

$$\left(C-\frac{1}{h}\right)\log(1-hC) \leq \left(C-\frac{1}{2}\right)\log(1-2C)$$

Also note that function $f(x) = (1-x)\log(1-x) + \left(x-\frac{1}{2}\right)\log(1-2x) < 0$ for any $0 < x < \frac{1}{2}$. Therefore, p_h goes to 0 exponentially fast.

When $\frac{n}{2s} < h < \frac{n}{s}$, as $\frac{n}{h} - s = \Theta(n) = \Theta(\frac{n}{h})$,

$$
\begin{aligned}
&\left[N\left(\frac{n}{h}, \frac{n}{h} - s\right)\right]^h \\
=&\left[\frac{\sqrt{\frac{n}{h}\left(\frac{n}{h} - s\right)}}{\left(\frac{n}{h} - s\right)^{\frac{n}{h}} (2\pi)^{\frac{n}{h} - \frac{1}{2}}} \cdot \left(\frac{n}{h\left(\frac{n}{h} - s\right)}\right)^{\frac{n}{h}\left(\frac{n}{h} - s\right)} \cdot \exp\left(O(1)\right.\right. \\
&\left.\left. + (1 - hC - 1)\left(\frac{n}{h}\right)^2 \log(1 - (1 - hC)) - \frac{n}{h}\log(1 - (1 - hC))\right)\right]^h \\
=&\frac{\left(\frac{n}{h}\left(\frac{n}{h} - s\right)\right)^{\frac{1}{2}h}}{\left(\frac{n}{h} - s\right)^n (2\pi)^{n - \frac{h}{2}}} \cdot \left(\frac{n}{h\left(\frac{n}{h} - s\right)}\right)^{n\left(\frac{n}{h} - s\right)} \\
&\cdot \exp\left(-Cn^2 \log(hC) - n\log(hC) + O(h)\right)
\end{aligned}
$$

and

$$
\begin{aligned}
p_h \leq & \frac{1}{h} \cdot \binom{n}{\frac{n}{h}, \frac{n}{h}, \ldots, \frac{n}{h}} \cdot \frac{\left[N(\frac{n}{h}, \frac{n}{h} - s)\right]^h}{N(n,s)} \\
= & \frac{h^{n+\frac{1}{2}h-1}}{(2\pi n)^{\frac{1}{2}(h-1)}} \cdot \frac{\left(\frac{n}{h}\left(\frac{n}{h}-s\right)\right)^{\frac{1}{2}h} s^n (2\pi)^{n-\frac{1}{2}}}{\left(\frac{n}{h}-s\right)^n (2\pi)^{n-\frac{1}{2}h}\sqrt{ns}} \cdot \left(\frac{n}{h\left(\frac{n}{h}-s\right)}\right)^{n\left(\frac{n}{h}-s\right)} \\
& \cdot \left(\frac{n}{s}\right)^{-ns} \cdot \exp\Bigg(- Cn^2 \log(hC) - n\log(hC) + O(h) \\
& - (C-1)n^2 \log(1-C) + n\log(1-C) - O(1)\Bigg) \\
= & h^{n-1} s^{n-\frac{1}{2}} \cdot \left(\frac{n}{h} - s\right)^{\frac{1}{2}h-n} \cdot (1-hC)^{\left(C-\frac{1}{h}\right)n^2} \cdot C^{Cn^2} \\
& \exp\Bigg((-C\log(hC) - (C-1)\log(1-C))n^2 + (\log(1-C) - \log(hC))n \\
& + O(h)\Bigg) \\
= & \frac{h^{n-1}}{\sqrt{s}} \cdot \left(\frac{n}{h} - s\right)^{\frac{1}{2}h-n} \cdot \exp\Bigg(\Bigg(\left(C - \frac{1}{h}\right)\log(1-hC) + C\log C \\
& - C\log(hC) - (C-1)\log(1-C)\Bigg)n^2 + n\log n + (\log(1-C) - \log h)n \\
& + O(h)\Bigg)
\end{aligned}
$$

where

$$
\begin{aligned}
& \left(C - \frac{1}{h}\right)\log(1-hC) + C\log C - C\log(hC) - (C-1)\log(1-C) \\
= & \left(C - \frac{1}{h}\right)\log(1-hC) - C\log h - (C-1)\log(1-C)
\end{aligned}
$$

Notice that

$$
\frac{\partial}{\partial h}\left[\left(C - \frac{1}{h}\right)\log(1-hC) - C\log h\right] = \frac{\log(1-Ch)}{h^2} < 0
$$

as $0 < Ch = \frac{sh}{n} < 1$. Due to $\frac{1}{2C} = \frac{n}{2s} < h < \frac{n}{s} = \frac{1}{C}$,

$$\begin{aligned}
&\left(C - \frac{1}{h}\right)\log(1-hC) + C\log C - C\log(hC) - (C-1)\log(1-C) \\
\leq &(C-2C)\log\left(1 - C\frac{1}{2C}\right) - C\log\frac{1}{2C} - (C-1)\log(1-C) \\
= &C\log 2 + C\log(2C) - (C-1)\log(1-C) \\
= &C\log(4C) - (C-1)\log(1-C) < 0
\end{aligned}$$

for any $0 < \epsilon \leq C \leq \frac{1}{2} - \epsilon < \frac{1}{2}$. Therefore, we again have p_h going to 0 exponentially fast.

When $s = \Theta(n) < \frac{n}{2}$ and $\frac{n}{h} - s = o(n)$, let $C = \frac{s}{n} < \frac{1}{h}$ but $\lim_{n\to+\infty} C = \frac{1}{h}$. In this case $2 \leq h < \frac{n}{s}$ is $O(1)$ and surely $\frac{n}{2s} < h < \frac{n}{s}$. Otherwise, $h \leq \frac{n}{2s}$ implies $\frac{n}{h} - s \geq \frac{n}{2h} = \Theta(n)$. Also, due to $\frac{n}{h} - s = o(n)$ and $h = O(1)$, $O\left(\frac{\left(\frac{n}{h}-s\right)^3 h^2}{n}\right) = o(n^2)$.

$$
\begin{aligned}
p_h \leq & \frac{1}{h} \cdot \binom{n}{\frac{n}{h}, \frac{n}{h}, \ldots, \frac{n}{h}} \cdot \frac{\left[N(\frac{n}{h}, \frac{n}{h} - s)\right]^h}{N(n, s)} \\
= & \frac{h^{n+\frac{1}{2}h-1}}{(2\pi n)^{\frac{1}{2}(h-1)}} \cdot \frac{\left(\frac{n}{h}\left(\frac{n}{h} - s\right)\right)^{\frac{1}{2}h} s^n (2\pi)^{n-\frac{1}{2}}}{\left(\frac{n}{h} - s\right)^n (2\pi)^{n-\frac{1}{2}h}\sqrt{ns}} \cdot \left(\frac{en}{h\left(\frac{n}{h} - s\right)}\right)^{n\left(\frac{n}{h}-s\right)} \\
& \cdot \left(\frac{n}{s}\right)^{-ns} \cdot \exp\left(- \frac{h\left(\frac{n}{h} - s - 1\right)^2}{2} + O\left(\frac{\left(\frac{n}{h} - s\right)^3 h^2}{n}\right)\right. \\
& \left. - (C-1)n^2 \log(1-C) + n\log(1-C) - O(1)\right) \\
= & h^{n-1} s^{n-\frac{1}{2}} \cdot \left(\frac{n}{h} - s\right)^{\frac{1}{2}h-n} \cdot \left(\frac{1-hC}{e}\right)^{\left(C-\frac{1}{h}\right)n^2} \cdot C^{Cn^2} \exp\left(- \frac{n^2}{2h} - \frac{hs^2}{2}\right. \\
& \left. - hs - \frac{h}{2} + ns + n - (C-1)n^2 \log(1-C) + n\log(1-C) + o(n^2)\right) \\
= & \frac{h^{n-1}}{\sqrt{s}} \cdot \left(\frac{n}{h} - s\right)^{\frac{1}{2}h-n} \cdot \exp\left(\left(\left(\frac{1}{h} - C\right) - \left(\frac{1}{h} - C\right)\log(1-hC)\right.\right. \\
& \left.\left. + C\log C - (C-1)\log(1-C) - \frac{1}{2h} - \frac{hC^2}{2} + C\right)n^2 + o(n^2)\right)
\end{aligned}
$$

where $\lim_{n\to+\infty} C = \frac{1}{h}$ and

$$
\begin{aligned}
& \left(\frac{1}{h} - C\right) - \left(\frac{1}{h} - C\right)\log(1-hC) + C\log C \\
& - (C-1)\log(1-C) - \frac{1}{2h} - \frac{hC^2}{2} + C \\
= & 0 + 0 - \frac{1}{h}\log h - \left(1 - \frac{1}{h}\right)\log\left(\frac{h}{h-1}\right) - \frac{1}{2h} - \frac{1}{2h} + \frac{1}{h} \\
= & -\frac{1}{h}\log h - \left(1 - \frac{1}{h}\right)\log\left(\frac{h}{h-1}\right) < 0
\end{aligned}
$$

for which p_h goes to 0 exponentially fast.

There are still two cases to be addressed, where $h = \frac{n}{s}$ or $s = \frac{1}{2}n$. Notice that when $s = \frac{1}{2}n$, the only possible h is $2 = \frac{n}{s}$ so we can handle both by handling the

former. In this case, we have

$$p_{\frac{n}{s}} \leq \frac{s}{n}\binom{n}{s,s,\ldots,s} \cdot \frac{1}{N(n,s)}$$

where

$$\binom{n}{s,s,\ldots,s} = \frac{n!}{(s!)^{\frac{n}{s}}} \leq \frac{\sqrt{2\pi n}\cdot n^n e^{n+\frac{1}{12n}}}{e^n(\sqrt{2\pi s}\cdot s^s)^{\frac{n}{s}}} = \frac{\sqrt{2\pi n}}{(2\pi s)^{\frac{n}{2s}}}\left(\frac{n}{s}\right)^n e^{\frac{1}{12n}}$$

For $s = o(n)$,

$$\begin{aligned}
p_{\frac{n}{s}} \leq & \frac{\sqrt{2\pi n}}{(2\pi s)^{\frac{n}{2s}}}\left(\frac{n}{s}\right)^{n-1} e^{\frac{1}{12n}} \cdot \frac{s^n(2\pi)^{n-\frac{1}{2}}}{\sqrt{ns}}\left(\frac{en}{s}\right)^{-ns} \exp\left(\frac{(s-1)^2}{2}\right) \\
= & \frac{(2\pi)^n}{\sqrt{s}(2\pi s)^{\frac{n}{2s}}}\left(\frac{s}{n}\right)^{ns-n+1} \exp\left(n\log s - ns + \frac{(s-1)^2}{2} + \frac{1}{12n}\right) \\
\leq & \frac{1}{\sqrt{s}(2\pi s)^{\frac{n}{2s}}}\left(\frac{2\pi s}{n}\right)^{n+1} \exp\left(n\log s - ns + \frac{(s-1)^2}{2} + \frac{1}{12n}\right)
\end{aligned}$$

which goes to 0 exponentially fast.

For $s = \Theta(n)$ and $s \leq \frac{1}{2}n$,

$$\begin{aligned}
p_{\frac{n}{s}} \leq & \frac{\sqrt{2\pi n}}{(2\pi s)^{\frac{n}{2s}}}\left(\frac{n}{s}\right)^{n-1} e^{\frac{1}{12n}} \cdot \frac{s^n(2\pi)^{n-\frac{1}{2}}}{\sqrt{ns}} \cdot \left(\frac{n}{s}\right)^{-ns} \\
& \exp\Big(-(C-1)n^2\log(1-C) + n\log(1-C) - O(1)\Big) \\
= & \frac{(2\pi)^n}{\sqrt{s}(2\pi s)^{\frac{n}{2s}}}\left(\frac{s}{n}\right)^{ns-n+1} \exp\Big(-(C-1)n^2\log(1-C) \\
& + n\log n + (\log(1-C) + \log C)n + o(1)\Big)
\end{aligned}$$

which goes to 0 exponentially fast as well. Combining all the above cases completes the proof. ■

Fast convergence

According to Theorem 4.3, the proof of fast convergence can be done by either bounding the diameter of the graph or directly bounding the first non-zero eigenvalue of the Laplacian matrix. In this section we present the fast convergence of random walks on $\mathrm{RSG}^+(s)$ via the spectral method.

First note that the walk matrix P of a random walk on a $\mathrm{RDG}(s)$ is doubly stochastic matrix, so is $\frac{1}{2}(P + P^\top)$. Fiedler [Fie72] proved a very useful theorem:

Theorem 5.3 *Let Q be a doubly stochastic $n \times n$ matrix ($n \geq 2$) and $\lambda \neq 1$ be any non-stochastic eigenvalue of Q.*

$$|1 - \lambda| \geq \varphi_n[\mu(Q)]$$

where

$$\mu(Q) = \min_{\emptyset \neq M \subset [n]} \sum_{i \in M, j \notin M} Q_{ij}$$

and

$$\varphi_n(x) = \begin{cases} 2\left(1 - \cos\frac{\pi}{n}\right)x & \text{if } 0 \leq x \leq \frac{1}{2} \\ 1 - 2(1-x)\cos\frac{\pi}{n} - (2x-1)\cos\frac{2\pi}{n} & \text{if } \frac{1}{2} < x \leq 1 \end{cases}$$

The same paper also presented the following lemma.

Lemma 5.9 *For any doubly stochastic matrix Q, $0 \leq \mu(Q) \leq 1$. Q is reducible if and only if $\mu(Q) = 0$.*

Now we show the fast convergence of random walks on $\mathrm{RDG}(s)$.

Theorem 5.4 *With probability $1 - o(1)$, a random walk on a RDG(s) has $\Delta_{\chi^2}(t) \leq e^{-k}$ after at most $t \geq 2s(n-1)(\log n + 2k)$ steps.*

Proof As P has been shown irreducible with probability $1 - o(1)$, so is $\frac{1}{2}(P + P^\top)$. Then for Lemma 5.9 $0 < \mu(\frac{1}{2}(P + P^\top)) \leq 1$. The fact that any non-zero entry in P

is at least $\frac{1}{s}$ gives $\mu(\frac{1}{2}(P+P^\top)) \geq \frac{1}{2s}$. For Theorem 5.3,

$$\left|1-\lambda_{\frac{1}{2}(P+P^\top)}\right| \geq 2\left(1-\cos\frac{\pi}{n}\right)\frac{1}{2s} > \frac{1}{s(n-1)}$$

for all non-stochastic eigenvalues $\lambda_{\frac{1}{2}(P+P^\top)} \neq 1$ of matrix $\frac{1}{2}(P+P^\top)$, due to the fact $\cos x < 1 - \frac{x}{\pi - x}$ for all $x \in \left(0, \frac{\pi}{2}\right)$. Also observing that the stationary distribution on a RDG(s) is always the uniform distribution, we have the Laplacian matrix

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

and $|\lambda_1(\mathcal{L})| \geq \left|1-\lambda_{\frac{1}{2}(P+P^\top)}\right| > \frac{1}{s(n-1)}$ where $\lambda_1(\mathcal{L})$ is the smallest nonzero eigenvalue of $\mathcal{L}$. Combining with $\phi(u) = \frac{1}{n}$ for any $u \in V$ we complete the proof. ■

Proof of Theorem 5.1 for random regular undirected graphs

Random regular undirected graphs are much more widely studied than directed ones, mainly because of the symmetry of undirected graphs. However, the study of the convergence of random walks on RG(s) is still very limited. Hildebrand [Hil94] proved fast convergence with constraint $s = \lfloor \log^C n \rfloor$ for some constant $C \geq 2$. Cooper and Frieze [CF05] studied the cover time of RG(s) with fixed constant $s = O(1)$ but no convergence result was provided. In this section we present a more general result with constraint $3 \leq s = o(\sqrt{n})$ or $s > \frac{1}{2}n$. This constraint comes from the enumeration of RG(s) and the proof could be generalized if we had better results on the enumeration problem in the future.

Cooper et al. [CFR02] and Krivelevich et al. [KSVW01] together proved the connectivity of RG(s) for $s \geq 3$.

Lemma 5.10 *With probability* $1-o(1)$*, a RG(s) is connected when* $s \geq 3$*.*

Now we prove the aperiodicity as below.

Lemma 5.11 *With probability* $1-o(1)$*, a RG(s) is aperiodic when* $s \geq 3$ *for odd* n*;* $3 \leq s = o(\sqrt{n})$ *or* $s > \frac{1}{2}n$ *for even* n*.*

Proof When n is odd, the graph is surely aperiodic because for undirected graphs the only periodic case is being bipartite and for regular undirected graphs the only bipartite partition is an even partition. Also, the aperiodicity is trivial when $s > \frac{1}{2}n$. Below we will prove the nontrivial case where n is even and $3 \leq s \leq \frac{1}{2}n$. Denote by $N'(n,s)$ the number of s-regular undirected graphs of size n. McKay and Wormald [MW91] proved an enumeration result for $s = o(\sqrt{n})$ that

$$N'(n,s) = \frac{(sn)!}{\left(\frac{1}{2}sn\right)! \cdot 2^{\frac{1}{2}ns}(s!)^n} \exp\left[\frac{1-s^2}{4} - \frac{s^3}{12n} + O\left(\frac{s^2}{n}\right)\right]$$

Since $s = o(\sqrt{n}) < \frac{1}{4}n$, the probability of a RG(s) being periodic p_2 is bounded by

$$\begin{aligned}
p_2 \leq & \frac{1}{2}\binom{n}{\frac{n}{2}} \cdot \frac{N(\frac{n}{2},s)}{N'(n,s)} \\
= & \frac{1}{2}\frac{n!}{\left(\frac{n}{2}!\right)^2} \cdot \frac{\left(\frac{ns}{2}\right)!\left(\frac{ns}{2}\right)! \cdot 2^{\frac{ns}{2}}(s!)^n}{(s!)^n \cdot (ns)!} \exp\left[-\frac{(s-1)^2}{2} + O\left(\frac{s^3}{n}\right) + \frac{s^2}{4} - \frac{1}{4}\right] \\
= & \frac{\sqrt{2\pi n} \cdot n^n e^n}{2 \cdot e^n \pi n \left(\frac{n}{2}\right)^n} \cdot \frac{\left[\left(\frac{ns}{2}\right)!\right]^2 \cdot 2^{\frac{ns}{2}}}{(ns)!} \exp\left[-\frac{1}{4}s^2 + s + O\left(\frac{s^3}{n}\right) - \frac{3}{4}\right] \\
= & \frac{2^{n+\frac{1}{2}ns}}{\sqrt{2\pi n}} \cdot \frac{\pi ns \cdot \left(\frac{1}{2}ns\right)^{ns} e^{ns}}{e^{ns}(ns)^{ns}\sqrt{2\pi ns}} \exp\left[-\frac{1}{4}s^2 + s + O\left(\frac{s^3}{n}\right) - \frac{3}{4}\right] \\
= & 2^{-\frac{1}{2}ns+n-1} \cdot \sqrt{s} \cdot \exp\left[-\frac{1}{4}s^2 + s + O\left(\frac{s^3}{n}\right) - \frac{3}{4}\right]
\end{aligned}$$

When $s = \omega(1)$ and $s = o(\sqrt{n})$, p_2 goes to 0 exponentially fast because $O\left(\frac{s^3}{n}\right) =$

$o(s)$. When $s = O(1)$ and $s \geq 3$, $-\frac{1}{2}s + 1 < 0$ and p_2 goes to 0 exponentially fast as well, which completes the proof. ■

The fast convergence argument for RG(s) can be proved using the same proof for RDG(s). The only difference is that P is symmetric and $\frac{1}{2}(P + P^\top) = P$ so $|\lambda_1(\mathcal{L})| \geq |1 - \lambda_P| > \frac{2}{s(n-1)}$.

5.3 Reconstructing random regular graphs from random paths

The positive theoretical results in Section 5.2 establish the generalization of our learning algorithm in Chapter 4 to learning random regular graphs. Because the nature of the algorithm requires the graph to be out-regular, we only apply this algorithm to the models with fixed out-degree s, namely RMG$^+$(s), RSG$^+$(s), RDG(s) and RG(s).

5.3.1 Preliminaries

In this section we study the problem of learning regular graphs in the statistical query model. In a typical label-guided graph exploration setting [FIP$^+$04, Rei05, BS94, BFR$^+$98], in a regular graph with fixed out-degree s, the s edges incident from a node are associated with s distinct *port numbers* in $\Sigma = \{1, 2, \ldots, s\}$, in a one-to-one manner. Each edge of a node is labeled with the associated port number. Note that port numbering is *local*, i.e., there is no relation between port numbers at u and at v. In the undirected case RG(s), every undirected edge (u, v) has two labels corresponding to its port numbers at u and at v respectively, which are not

necessarily identical. A *path* on the graph is a sequence of edge labels. The input data to the statistical query oracle are path-destination pairs of the form (x, v) where $x \in \Sigma^t$ is a random uniform path and v is the vertex on the graph reached on the path x starting from a particular *start vertex* v_0. Here $t = poly(n, s)$ is the length of the example paths. The learner has access to the oracle $STAT$ and algorithms are designed to reconstruct the graph (or the unique closed irreducible component for $\mathrm{RMG}^+(s)$ and $\mathrm{RSG}^+(s)$) from statistical queries.

5.3.2 The learning algorithm

A uniform path $x \in \Sigma^t$ corresponds to a random walk of length t on the graph G starting from the start vertex v_0. Since all these four types of random regular graphs have been proved to have one unique closed irreducible component with high probability and due to the main theorem, the walk will converge to the stationary distribution p_λ polynomially fast, with any start vertex. Define a collection of $n \times n$ binary matrices M_σ indexed by labels $\sigma \in \Sigma$ as follows. For each pair of vertices u and v, the element $M_\sigma(u, v)$ is 1 if $(u, v) \in E$ and is labeled with σ at vertex u, and 0 otherwise. For a path $y = y_1 y_2 \ldots y_m$ of length m, define M_y to be the matrix product $M_y = M_{y_1} \cdot M_{y_2} \ldots M_{y_m}$. Also define the distribution vector p_y over V obtained by starting with the stationary distribution p_λ and walking along the path y on the graph. That is, $p_y = p_\lambda M_y$. Let z be the i-th column of matrix M_σ, P_A be the $s^D \times n$ coefficient matrix whose rows are $\{p_y \mid y \in \Sigma^D\}$ and b be the vector consisting of $\{p_{y\sigma}(i) \mid y \in \Sigma^D\}$ corresponding to each y in P_A. Here D is an upper bound on the diameter. From Theorem 5.2 we have $D = \Theta(\log_s n)$ and the concrete constant can be inferred from the proof, which depends on s and approaches unity with increasing s. The algorithm recovers the structure of the strongly connected component by solving the linear equation system $P_A z = b$ for each column z in each

M_σ.

By setting $k = \log \frac{2}{\tau}$ in the main theorem, after t_0 steps the random walk converges to the stationary distribution p_λ within χ-square distance $\frac{\tau}{2}$ with high probability. Observe that $2\|\phi_t - \phi\|_{TV} \leq \Delta_{\chi^2}(t)$, where ϕ_t is the distribution vector over V after t steps of random walk. We can estimate the stationary distribution for a vertex i by the fraction of examples (x, v) such that $v = i$. In general, for any path y, we can estimate the value of p_y for a vertex i as the ratio between the number of pairs (x, v) such that y is a suffix of x and $v = i$ and the number of examples (x, v) where y is a suffix of x. In the statistical query model this is done with a conditional statistical query $\chi_{y,i}(x, v) = \mathbb{1}\{v = i \mid y \text{ is a suffix of } x\}$ at tolerance $\frac{\tau}{2}$, where $\mathbb{1}$ is the boolean indicator function. Denote by vector $\widehat{p}_y$ the query result returned by oracle *STAT* where $\widehat{p}_y(i)$ is the estimated $\mathbb{E}\chi_{y,i}$, and by $\widehat{P}_A$ and $\widehat{b}$ the estimates for P_A and b respectively. We have $\|p_y - \widehat{p}_y\|_\infty \leq \tau$ for any path y. The algorithm approximates z by solving the perturbed linear least squares problem: $\min_z \|\widehat{P}_A z - \widehat{b}\|_2$. Let vector $\widehat{z}$ be the solution. Then from Chapter 4 we have

Lemma 5.12 *If P_A has full rank with high probability, for all columns z in all matrices M_σ, $\|z - \widehat{z}\|_\infty \leq \|z\|_1 \||P_A^\dagger|\|_\infty \tau + O(\tau^2)$ with probability $1 - o(1)$.*

For RMG$^+(s)$, it is proved in Chapter 4 with high probability $\|z\|_1 \leq \frac{(1+\varepsilon)\log ns}{\log\log ns}$ for any constant $\varepsilon > 0$. We show this also holds for RSG$^+(s)$. For RDG(s) and RG(s), we have $\|z\|_1 = s$.

Theorem 5.5 *If P_A has full rank with high probability,*

1. *for RMG$^+(s)$ and RSG$^+(s)$, $\|z - \widehat{z}\|_\infty \leq \frac{(1+\varepsilon)\log ns}{\log\log ns} \||P_A^\dagger|\|_\infty \tau + O(\tau^2)$ for any constant $\varepsilon > 0$*

2. *for RDG(s) and RG(s), $\|z - \widehat{z}\|_\infty \leq s \||P_A^\dagger|\|_\infty \tau + O(\tau^2)$*

holds for all columns z in all matrices M_σ with probability $1 - o(1)$.

Proof The RMG$^+(s)$ case has been proved in Chapter 4 and the RSG$^+(s)$ case can be proved in a similar way too. Here we provide a quick proof based on our proof in Chapter 4 to bypass the lengthy algebra.

Let θ be the largest 1-norm of the columns in M_σ. According to the properties of a RSG$^+(s)$, the probability of $\theta > n-1$ is 0 and $\mathbb{P}(\theta = n) \leq n \cdot (n-1)^{-(n-1)}$ is exponentially small. For any $k < n-1$,

$$\begin{aligned}&\mathbb{P}(\theta \geq k) \leq n \cdot \mathbb{P}(\text{a particular column has 1-norm at least } k)\\ \leq & n \cdot \binom{n-1}{k} \left(\frac{1}{n-1}\right)^k \leq 2(n-1) \cdot \binom{n-1}{k} \left(\frac{1}{n-1}\right)^k\end{aligned}$$

In Chapter 4 we proved when $k = \frac{(1+\varepsilon)\log ns}{\log\log ns}$, $n \cdot \binom{n}{k}\left(\frac{1}{n}\right)^k = \frac{1}{s} \cdot o(1)$. Thus, in our case when $k = \frac{(1+\varepsilon)\log(n-1)s}{\log\log(n-1)s} \leq \frac{(1+\varepsilon)\log ns}{\log\log ns}$, we have $(n-1) \cdot \binom{n-1}{k}\left(\frac{1}{n-1}\right)^k = \frac{1}{s} \cdot o(1)$ so that $\mathbb{P}(\theta \geq k) \leq \frac{1}{s} \cdot o(1)$. 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 $1 - o(1)$. ■

This further implies that if we set the tolerance $\tau = \frac{\log\log ns}{3|||P_A^\dagger|||_\infty \log ns}$ for RMG$^+(s)$ and RSG$^+(s)$, and $\tau = \frac{1}{3s|||P_A^\dagger|||_\infty}$ for RDG(s) and RG(s), the solution error $\|z - \widehat{z}\|_\infty < \frac{1}{2}$ with high probability. Based on the prior knowledge we have for 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 $\tilde{z}(v) = z(v)$ for any vertex v. We provide a toy example here to demonstrate how the learning algorithm works on a concrete regular graph.

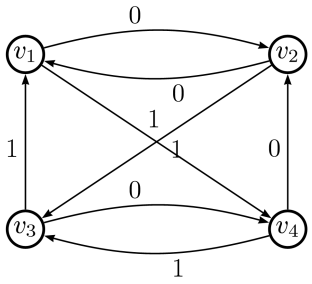

Figure 5.1: A 2-regular digraph with 4 vertices

A toy example

Suppose we consider the 2-regular digraph in Figure 5.1 whose transition matrices are

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

For any regular digraph, the stationary distribution p_λ is always the uniform distribution. As $\log_s n = \log_2 4 = 2$, the coefficient matrix P_A is

$$P_A = \begin{pmatrix} p_{00} \\ p_{01} \\ p_{10} \\ p_{11} \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.75 & 0.25 \\ 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.25 & 0.25 \end{pmatrix}$$

Denote by $z = (M_0(1,1), M_0(2,1), M_0(3,1), M_0(4,1))^\top$ the the first column of matrix M_0. Let vector b be $(p_{000}(1), p_{010}(1), p_{100}(1), p_{110}(1))^\top = (0.5, 0, 0.5, 0)^\top$ as defined in the algorithm. The algorithm recovers z by solving the equation system

$P_A z = b$, that is, solving

$$\begin{cases} 0.5M_0(1,1) + 0.5M_0(2,1) + 0M_0(3,1) + 0M_0(4,1) = 0.5 \\ 0M_0(1,1) + 0M_0(2,1) + 0.75M_0(3,1) + 0.25M_0(4,1) = 0 \\ 0M_0(1,1) + 0.5M_0(2,1) + 0M_0(3,1) + 0.5M_0(4,1) = 0.5 \\ 0.5M_0(1,1) + 0M_0(2,1) + 0.25M_0(3,1) + 0.25M_0(4,1) = 0 \end{cases}$$

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

5.3.3 Experiments and empirical results

In this section we present experimental results to illustrate the empirical performance of the learning algorithm. To be more robust against fluctuation from randomness, each test was run for 20 times and the medians were taken. The graphs are generated uniformly at random as defined 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 from $[-\tau, \tau]$. Since Chapter 4 already included experiments on learning a random DFA, whose underlying graph is exactly $\mathrm{RMG}^+(s)$, we don't duplicate the experiments for $\mathrm{RMG}^+(s)$.

The generating procedure of a $\mathrm{RSG}^+(s)$ is standard. Each node $v \in V$ independently chooses s neighbors from $\{V \setminus v\}$ without replacement uniformly at random. However, to the best of our knowledge, there is no algorithm that efficiently generates a $\mathrm{RDG}(s)$ or a $\mathrm{RG}(s)$. In our experiments, we use the celebrated pairing model first introduced by Bollobás [Bol80]. In an undirected regular graph, each vertex has

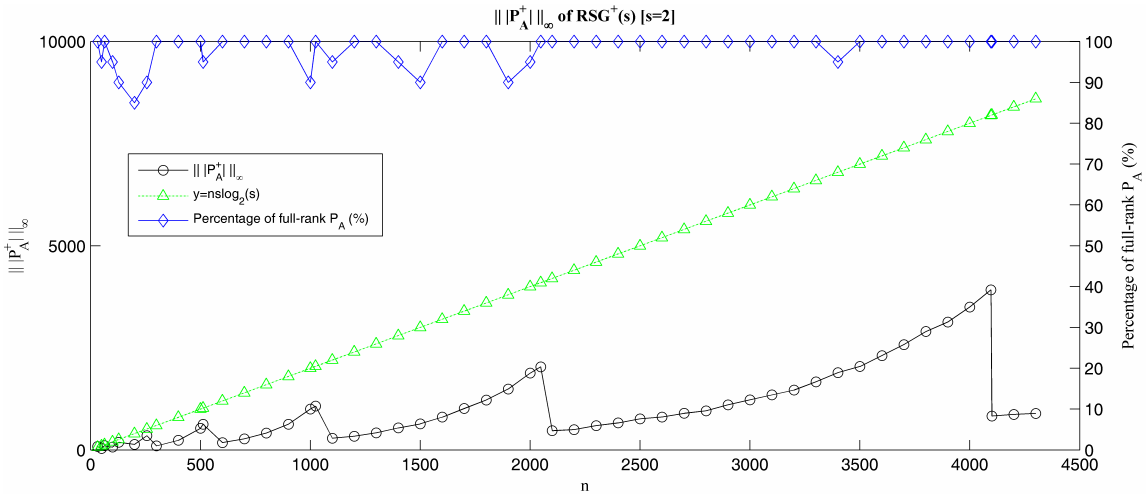

Figure 5.2: $\|\|\|P_A^\dagger|\|\|_\infty$ of $\mathrm{RSG}^+(s)$, versus n with fixed $s = 2$

s ports associated to its s edges. It is well known that the necessary and sufficient conditions for an s-regular graph with n vertices to exist are that $n \geq s + 1$ and that ns is even. To generate a $\mathrm{RG}(s)$, we uniformly pick a perfect matching of the ns ports into $\frac{1}{2}ns$ pairs. Adding an edge between each pair of ports gives a (not necessarily simple) regular graph. Repeat this procedure until it produces a simple graph. Likewise we generate a $\mathrm{RDG}(s)$ by uniformly matching ns out-ports (corresponding to outgoing edges) with ns in-ports (corresponding to incoming edges) until we get a regular digraph with no parallel edges. This method is not efficient owing to the unbounded number of repetitions, especially when s grows. Hence, with large s this generating method is extremely slow. Note that this limitation comes from the existing generating methods. Our learning algorithm is efficient.

The experiments start with an empirical estimate for the norm $\|\|\|P_A^\dagger|\|\|_\infty$. For $\mathrm{RSG}^+(s)$ we first vary the graph size n from 32 to 4300 with fixed out-degree $s = 2$. Figure 5.2 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

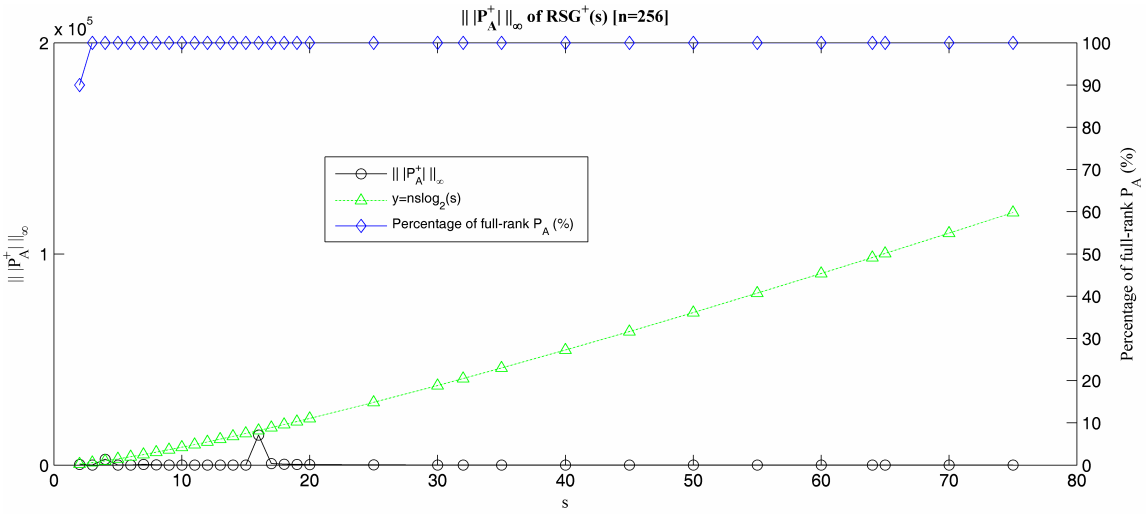

Figure 5.3: $|||P_A^\dagger|||_\infty$ of RSG$^+(s)$, versus s with fixed $n = 256$

overdetermined with many extra equations. Thus it is robust to perturbation and well-conditioned. When n 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 graph size reaches $n = s^i$ for 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 5.3. 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 the coefficient matrix P_A is measured to support the full-rank assumption. Both figures suggest an upper bound $ns \log s$ for $|||P_A^\dagger|||_\infty$ of RSG$^+(s)$. Figures 5.8 and 5.9 demonstrate the experimental results for the maximum absolute error. Along with the error curve a function is plotted to approximate the behavior of the error. An empirical error bound is $O(\log^{-1} n)$ with fixed s and $O(1/\sqrt{s})$ with fixed n.

Because generating a RDG(s) and generating a RG(s) are extremely slow with large s, the range of s where we can efficiently conduct the experiments is very limited. For RDG(s) we first vary n from 32 to 4300 with fixed $s = 2$ (Figure 5.4) as

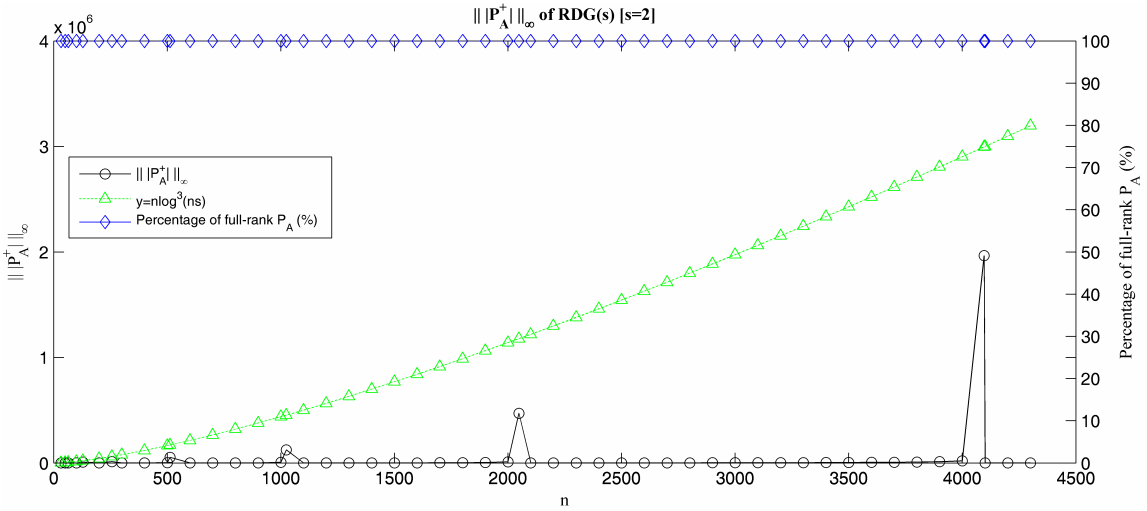

Figure 5.4: $|||P_A^\dagger|||_\infty$ of RDG(s), versus n with fixed $s = 2$

before but with fixed $n = 256$ we vary s from 2 to 6 (Figure 5.5). The norm $|||P_A^\dagger|||_\infty$ of RDG(s) is bounded by $n \log^3(ns)$ and an empirical error bound is $O(\log^{-1} n)$ with fixed s (Figure 5.10) and $O(1/s)$ with fixed n (Figure 5.11). For RG(s) we vary n from 26 to 3000 with fixed $s = 3$ (Figure 5.6) and vary s from 3 to 8 with fixed $n = 242$ (Figure 5.7). As the existence of a regular undirected graph requires even ns and s is fixed to be 3 when varying n, we only run experiments with even n. For critical points where $n = 3^i$, experiments are run with $n = 3^i - 1$ and $n = 3^i + 1$. This explains why we start with $n = 26$ instead of $n = 27$ with fixed $s = 3$, and also why we fix $n = 242$ rather than $n = 243$ when varying s. The norm $|||P_A^\dagger|||_\infty$ of RG(s) is bounded by $sn^{1.6}$ and an empirical error bound is $O(\log n/\sqrt{n})$ with fixed s (Figure 5.12) and $O(1/s)$ with fixed n (Figure 5.13).

5.4 Other applications and discussion

With the broad applications of regular graphs in computer science and machine learning, our theoretical results can be applied to other research areas such as distributed

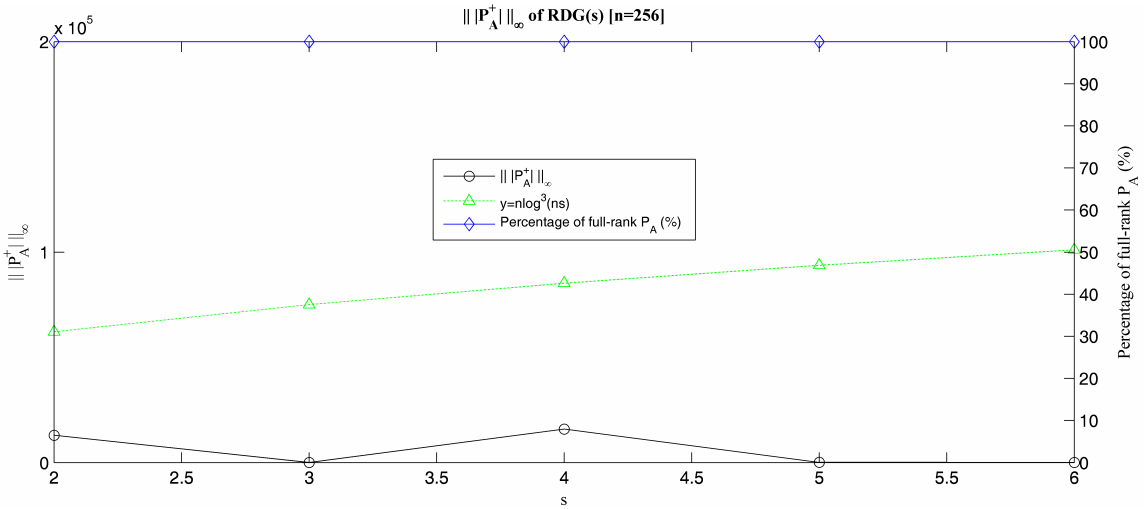

Figure 5.5: $\||P_A^\dagger|\|_\infty$ of RDG(s), versus s with fixed $n = 256$

networks and social network graphs. Performing random walks on distributed networks is an active area of research (see [BBSB06] for a comprehensive survey). High connectivity, bounded degree and low diameter are very common properties of (well designed) distribution network models. Theorem 4.3 explicitly provides fast convergence for random walks on these models. For instance, Pandurangan et al. [PRU03] proposed a protocol which ensures that the network is connected and has logarithmic diameter with high probability, and has always bounded degree. A simpler, fully decentralized model named SWAN was proposed by Bourassa and Holt [BH03] based on random walks, which produces a random regular graph. In another direction, random walks have proven to be a simple, yet powerful mathematical tool for extracting information from large scale and complex social networks (see [SM11] for a comprehensive survey). Social network graphs also have the above properties (high connectivity, small degree and low diameter) so that the random walks will converge fast as we proved. One application of fast convergence is the capability of uniformly sampling the graph, which is very important in many graph learning problems.

In this chapter we have shown positive theoretical results on random walks on

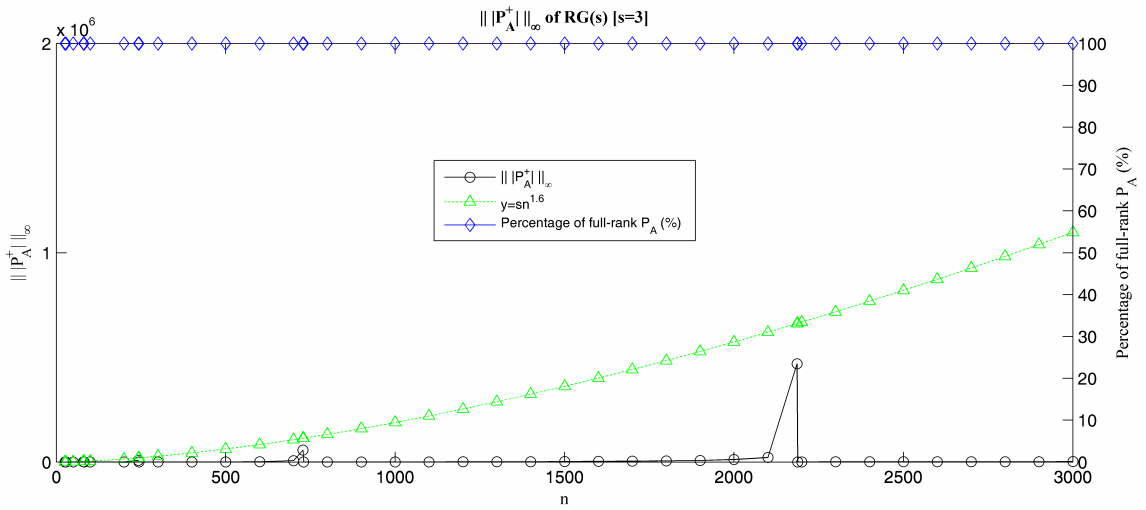

Figure 5.6: $|||P_A^\dagger|||_\infty$ of RG(s), versus n with fixed $s = 3$

random regular graphs, and generalized our algorithm in Chapter 4 to learning random regular graphs from random paths. One technical question concerning the fast convergence result is whether it can be generalized to weighted random walks on random regular graphs. An immediate benefit from this generalization is the release from the requirement of uniform paths in the learning algorithm. However, we conjecture this requires a polynomial lower bound on the edge weights in the graph, to avoid exponentially small nonzero elements in the walk matrix P. Another potential future work is to apply this algorithm to learning a more general class of graphs. Note that any generalization of the algorithm needs not only fast convergence, but also asymmetry of the target graph. The class of permutation automata [Thi68] is one example that has symmetric graph structure and degenerate P_A. Also, there is possibility of relaxing the constraint on s in the RG(s) case if advances in the enumeration of regular undirected graphs are made.

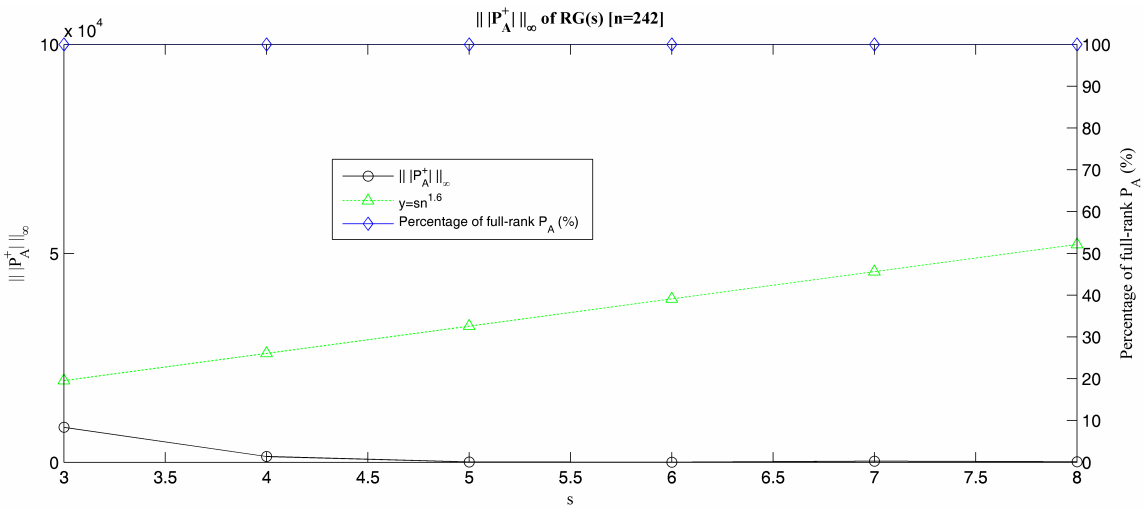

Figure 5.7: $|||P_A^{\dagger}|||_{\infty}$ of RG(s), versus s with fixed $n = 242$

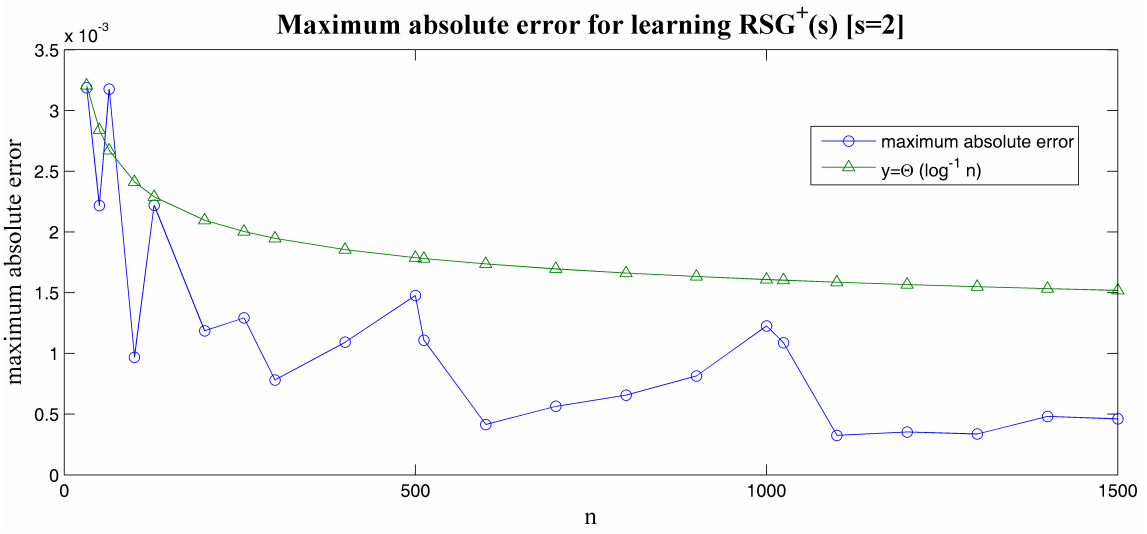

Figure 5.8: Maximum absolute error for learning a RSG$^{+}(s)$, versus n with fixed $s = 2$

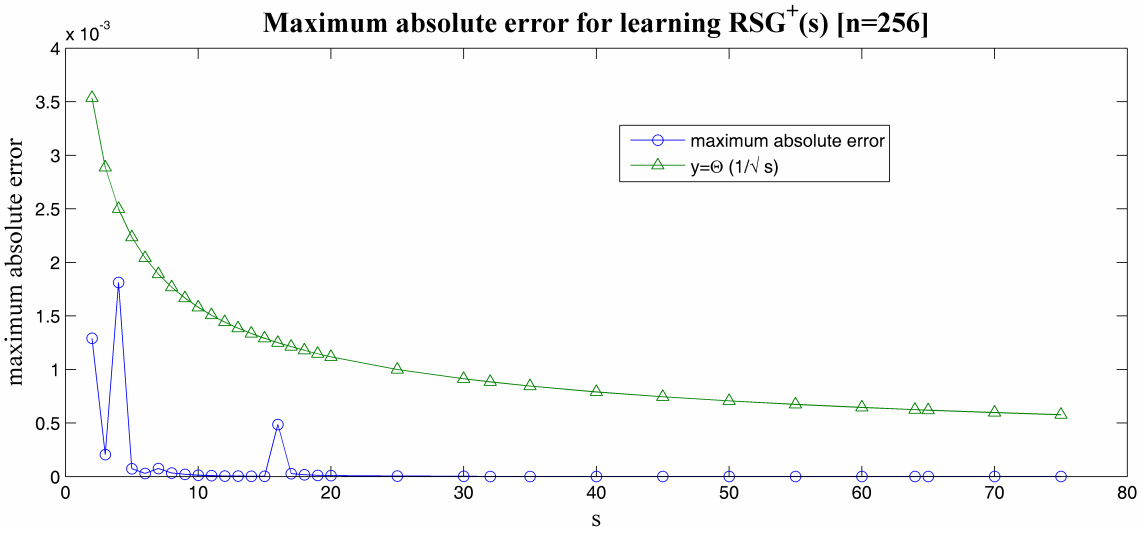

Figure 5.9: Maximum absolute error for learning a $\mathrm{RSG}^{+}(s)$, versus s with fixed $n = 256$

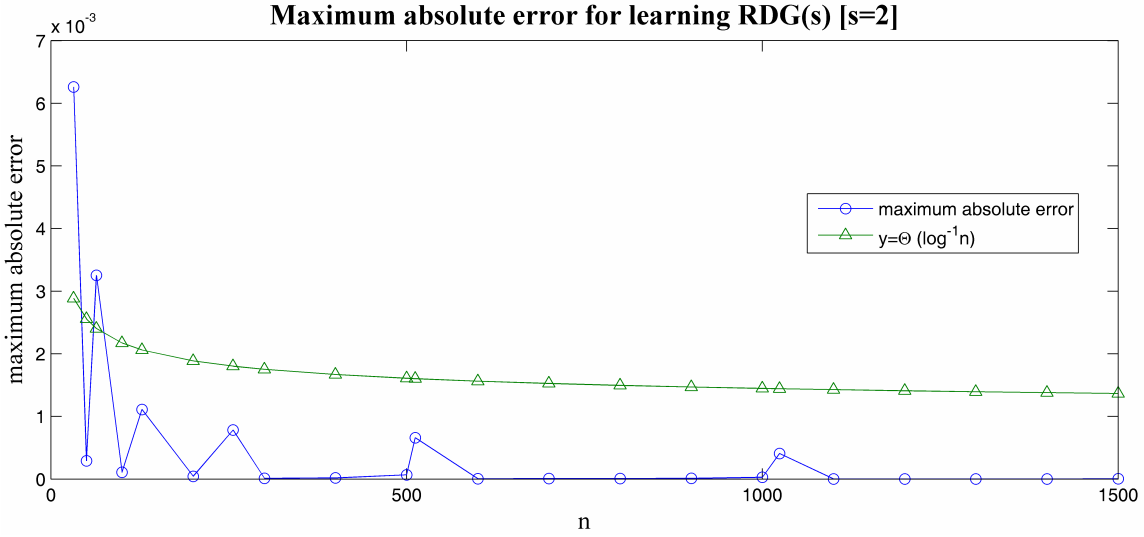

Figure 5.10: Maximum absolute error for learning a $\mathrm{RDG}(s)$, versus n with fixed $s = 2$

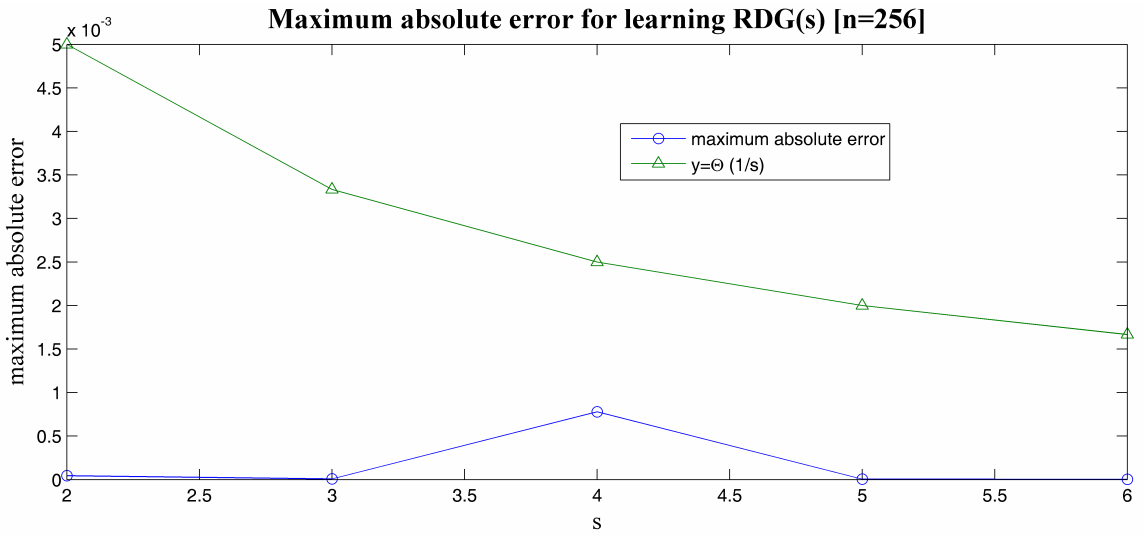

Figure 5.11: Maximum absolute error for learning a RDG(s), versus s with fixed $n = 256$

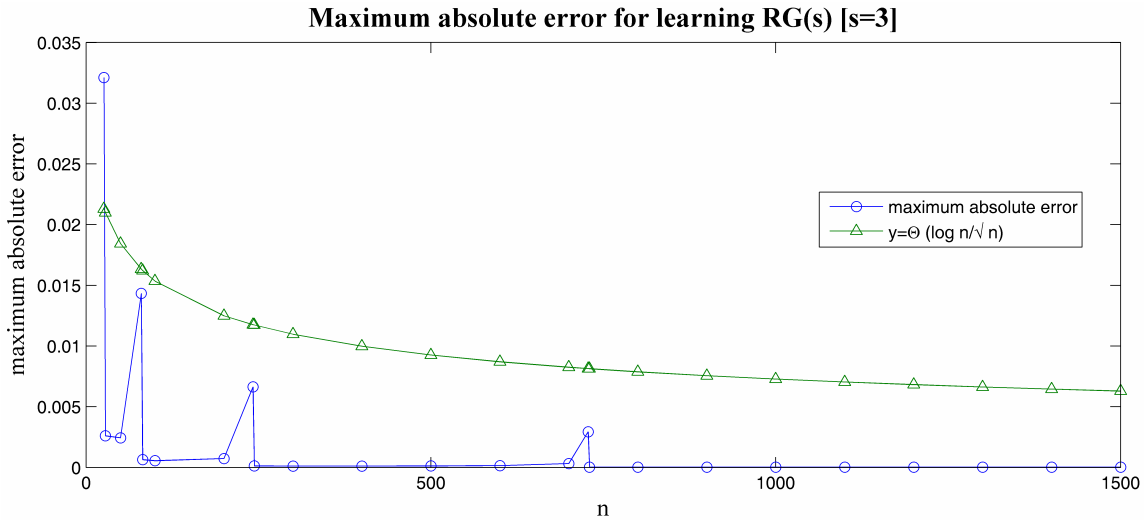

Figure 5.12: Maximum absolute error for learning a RG(s), versus n with fixed $s = 3$

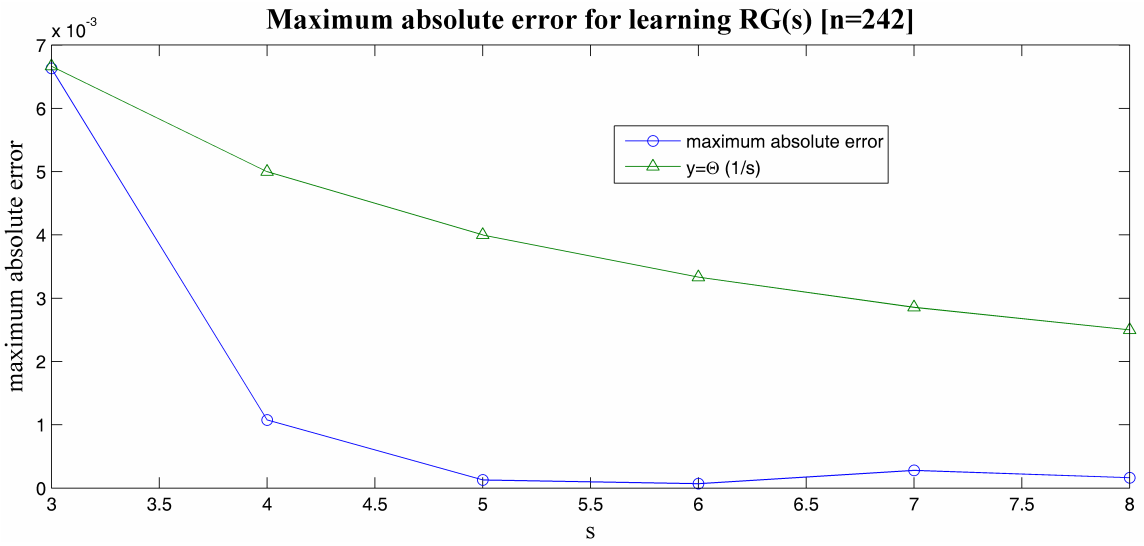

Figure 5.13: Maximum absolute error for learning a RG(s), versus s with fixed $n = 242$

Bibliography

[AAC16] Dana Angluin, James Aspnes, and Dongqu Chen. A population protocol for binary signaling consensus. *In Preparation*, 2016.

[AAD+06] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed computing*, 18(4):235–253, 2006.

[AAE07] Dana Angluin, James Aspnes, and David Eisenstat. A simple population protocol for fast robust approximate majority. In *Distributed Computing*, pages 20–32. Springer, 2007.

[AAEK13] Dana Angluin, James Aspnes, Sarah Eisenstat, and Aryeh Kontorovich. On the learnability of shuffle ideals. *The Journal of Machine Learning Research*, 14(1):1513–1531, 2013.

[AC15] Dana Angluin and Dongqu Chen. Learning a random DFA from uniform strings and state information. In *Proceedings of the 26th International Conference on Algorithmic Learning Theory*, 2015.

[AEKR10] Dana Angluin, David Eisenstat, Leonid Kontorovich, and Lev Reyzin. Lower bounds on learning random structures with statistical queries.

In *The 21st International Conference on Algorithmic Learning Theory*, pages 194–208. Springer, 2010.

[Ald89] David J. Aldous. Lower bounds for covering times for reversible Markov chains and random walks on graphs. *Journal of Theoretical Probability*, 2(1):91–100, 1989.

[Ang78] Dana Angluin. On the complexity of minimum inference of regular sets. *Information and Control*, 39(3):337–350, 1978.

[Ang87] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987.

[AR09] James Aspnes and Eric Ruppert. An introduction to population protocols. In *Middleware for Network Eccentric and Mobile Applications*, pages 97–120. Springer, 2009.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[Bal13] Borja Balle. Ergodicity of random walks on random DFA. *arXiv preprint arXiv:1311.6830*, 2013.

[BBSB06] Marc Bui, Thibault Bernard, Devan Sohier, and Alain Bui. Random walks in distributed computing: A survey. In Thomas Böhme, Victor M. Larios Rosillo, Helena Unger, and Herwig Unger, editors, *Innovative Internet Community Systems*, volume 3473 of *Lecture Notes in Computer Science*, pages 1–14. Springer Berlin Heidelberg, 2006.

[BCN+15] Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Riccardo Silvestri. Plurality consensus in the gossip model. In *Pro-*

ceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 371–390. Society for Industrial and Applied Mathematics, 2015.

[BE76] Béla Bollobás and Paul Erdös. Cliques in random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 80:419–427, November 1976.

[Ben74] Edward A. Bender. The asymptotic number of non-negative integer matrices with given row and column sums. *Discrete Mathematics*, 10(2):217 – 223, 1974.

[BFR+98] Michael A. Bender, Antonio Fernández, Dana Ron, Amit Sahai, and Salil Vadhan. The power of a pebble: Exploring and mapping directed graphs. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 269–278. Association for Computing Machinery, 1998.

[BGPS06] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, 2006.

[BH03] Virgil Bourassa and Fred Holt. Swan: Small-world wide area networks. In *Proceeding of International Conference on Advances in Infrastructures (SSGRR 2003w), LAquila, Italy*, 2003.

[Bjö91] Åke Björck. Component-wise perturbation analysis and error bounds for linear least squares solutions. *BIT Numerical Mathematics*, 31(2):237–244, 1991.

[BL07] Léon Bottou and Chih-Jen Lin. Support vector machine solvers. *Large scale kernel machines*, pages 301–320, 2007.

[BL13] Kevin Bache and Moshe Lichman. National Science Foundation research award abstracts 1990-2003 data set. *University of California, Irvine Machine Learning Repository*, 2013.

[BM96] Richard K. Belew and Melanie Mitchell, editors. *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1996.

[Bol80] Béla Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311 – 316, 1980.

[Bol88] Béla Bollobás. The chromatic number of random graphs. *Combinatorica*, 8(1):49–55, 1988.

[BS94] Michael A. Bender and Donna K Slonim. The power of team exploration: Two robots can learn unlabeled directed graphs. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 75–85. IEEE, 1994.

[Bsh97] Nader H. Bshouty. Exact learning of formulas in parallel. *Machine Learning*, 26(1):25–41, January 1997.

[CCN12] Luca Cardelli and Attila Csikász-Nagy. The cell cycle switch computes approximate majority. *Scientific reports*, 2, 2012.

[CF05] Colin Cooper and Alan Frieze. The cover time of random regular graphs. *SIAM Journal on Discrete Mathematics*, 18(4):728–740, 2005.

[CF08] Colin Cooper and Alan Frieze. The cover time of the giant component of a random graph. *Random Structures & Algorithms*, 32(4):401–439, 2008.

[CF09] Colin Cooper and Alan Frieze. Random walks on random graphs. In Maggie Cheng, editor, *Nano-Net*, volume 3 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 95–106. Springer Berlin Heidelberg, 2009.

[CFR02] Colin Cooper, Alan Frieze, and Bruce Reed. Random regular graphs of non-constant degree: connectivity and hamiltonicity. *Combinatorics, Probability & Computing*, 11(03):249–261, 2002.

[Che14] Dongqu Chen. Learning shuffle ideals under restricted distributions. In *Advances in Neural Information Processing Systems*, pages 757–765, 2014.

[Che15] Dongqu Chen. Learning random regular graphs. *Yale University Technical Report YALEU/DCS/TR-1518*, September 2015.

[Cho81] Noam Chomsky. Principles and parameters in syntactic theory I. In Norbert Hornstein and David Lightfoot (red.), editors, *Explanation in Linguistics: the Logical Problem of Language Acquisition*. London: Longman, 1981.

[Cho93] Noam Chomsky. *Lectures on Government and Binding: The Pisa Lectures*. Studies in generative grammar. Mouton de Gruyter, 1993.

[Chu97] Fan-Rong King Chung. *Spectral Graph Theory*. Number no. 92 in CBMS Regional Conference Series. Conference Board of the Mathematical Sciences, 1997.

[Chu05] Fan Chung. Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19, 2005.

[CM05] Rodney E. Canfield and Brendan D. McKay. Asymptotic enumeration of dense 0-1 matrices with equal row sums and equal column sums. *The Electronic Journal of Combinatorics [electronic only]*, 12(1):null, 2005.

[CS10] Marie Coppola and Ann Senghas. The emergence of deixis in nicaraguan signing. *Sign languages: A Cambridge language survey*, pages 543–569, 2010.

[DLH05] Colin De La Higuera. A bibliographical study of grammatical inference. *Pattern recognition*, 38(9):1332–1348, September 2005.

[ER59] Paul Erdös and Alfréd Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[ER60] Paul Erdös and Alfréd Rényi. On the evolution of random graphs. In *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[ER61a] Paul Erdös and Alfréd Rényi. On a classical problem of probability theory. *Magyar. Tud. Akad. Mat. Kutató Int. Közl.*, 6:215–220, 1961.

[ER61b] Paul Erdös and Alfréd Rényi. On the evolution of random graphs II. *Bulletin de l'Institut international de statistique*, 38(4):343–347, 1961.

[ER64] Paul Erdös and Alfréd Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Academiae Scientiarum Hungarica*, 12(1-2):261–267, 1964.

[Erd47] Paul Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53(4):292–294, 04 1947.

[Erd59] Paul Erdös. Graph theory and probability. *Canadian Journal of Mathematics*, 1959.

[Erd60] Paul Erdös. Graph theory and probability. II. *Canadian Journal of Mathematics*, 1960.

[Fei95a] Uriel Feige. A tight lower bound on the cover time for random walks on graphs. *Random Structures & Algorithms*, 6(4):433–438, 1995.

[Fei95b] Uriel Feige. A tight upper bound on the cover time for random walks on graphs. *Random Structures & Algorithms*, 6(1):51–54, 1995.

[FF82] Trevor I. Fenner and Alan M. Frieze. On the connectivity of random m-orientable graphs and digraphs. *Combinatorica*, 2(4):347–359, 1982.

[Fie72] Miroslav Fiedler. Bounds for eigenvalues of doubly stochastic matrices. *Linear Algebra and Its Applications*, 5(3):299–310, 1972.

[FIP+04] Pierre Fraigniaud, David Ilcinkas, Guy Peer, Andrzej Pelc, and David Peleg. Graph exploration by a finite automaton. In *Mathematical Foundations of Computer Science 2004*, pages 451–462. Springer, 2004.

[FKR+97] Yoav Freund, Michael Kearns, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. Efficient learning of typical finite automata from random walks. *Information and Computation*, 138(1):23 – 48, October 1997.

[Fla82] Leopold Flatto. Limit theorems for some random variables associated with urn models. *The Annals of Probability*, pages 927–934, 1982.

[Gal05] Bruno Galantucci. An experimental study of the emergence of human communication systems. *Cognitive science*, 29(5):737–767, 2005.

[GFR03] Bruno Galantucci, Carol A. Fowler, and Michael J. Richardson. Experimental investigations of the emergence of communication procedures. *Studies in perception and action VII*, pages 120–124, 2003.

[GK49] Boris Vladimirovich Gnedenko and Andrey Nikolaevich Kolmogorov. Limit distributions for sums of independent random variables. *Addison-Wesley series in statistics*, 1949.

[Gol78] E. Mark Gold. Complexity of automaton identification from given data. *Information and control*, 37(3):302–320, 1978.

[Gru73] Aleksandr Aleksandrovich Grusho. Limit distributions of certain characteristics of random automaton graphs. *Mathematical Notes of the Academy of Sciences of the USSR*, 14(1):633–637, 1973.

[GW94] Edward Gibson and Kenneth Wexler. Triggers. *Linguistic inquiry*, pages 407–454, 1994.

[Hig94] Nicholas J. Higham. A survey of componentwise perturbation theory in numerical linear algebra. In *Proceedings of symposia in applied mathematics*, volume 48, pages 49–77, 1994.

[Hil94] Martin Hildebrand. Random walks on random regular simple graphs. In *Institute for Mathematics and its Applications (IMA) Preprint Series*, 1994.

[Ibr56] Il'dar Abdullovich Ibragimov. On the composition of unimodal distributions. *Theory of Probability & Its Applications*, 1(2):255–260, 1956.

[JLSW08] Jeffrey C. Jackson, Homin K. Lee, Rocco A. Servedio, and Andrew Wan. Learning random monotone DNF. In *Approximation, Randomization and*

Combinatorial Optimization. Algorithms and Techniques, pages 483–497. Springer, 2008.

[Jon98] Johan Jonasson. On the cover time for random walks on random graphs. *Combinatorics, Probability and Computing*, 7(3):265–279, September 1998.

[JS05] Jeffrey C. Jackson and Rocco A. Servedio. Learning random log-depth decision trees under uniform distribution. *SIAM Journal on Computing*, 34(5):1107–1128, 2005.

[KCM08] Leonid Aryeh Kontorovich, Corinna Cortes, and Mehryar Mohri. Kernel methods for learning languages. *Theoretical Computer Science*, 405(3):223–236, 2008.

[KDG07] Simon Kirby, Mike Dowman, and Thomas L Griffiths. Innateness and culture in the evolution of language. *Proceedings of the National Academy of Sciences*, 104(12):5241–5245, 2007.

[Kea98] Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, November 1998.

[KN09] Leonid Aryeh Kontorovich and Boaz Nadler. Universal kernel-based learning with applications to regular languages. *The Journal of Machine Learning Research*, 10:1095–1129, 2009.

[Kos83] Kimmo Koskenniemi. Two-level model for morphological analysis. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, volume 83, pages 683–685, 1983.

[KP08] Ondřej Klíma and Libor Polák. Hierarchies of piecewise testable languages. *Proceedings of the 12th International Conference on Developments in Language Theory*, pages 479–490, 2008.

[KRS03] Leonid (Aryeh) Kontorovich, Dana Ron, and Yoram Singer. A Markov model for the acquisition of morphological structure. *Technical Report, Carnegie Mellon University, CMU-CS-03-147*, 10, June 2003.

[KSVW01] Michael Krivelevich, Benny Sudakov, Van H Vu, and Nicholas C Wormald. Random regular graphs of high degree. *Random Structures & Algorithms*, 18(4):346–363, 2001.

[Kur81] Thomas G. Kurtz. *Approximation of population processes*, volume 36. Society for Industrial and Applied Mathematics, 1981.

[KV94] Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

[Lam58] Johann Heinrich Lambert. Observationes variae in mathesin puram. *Acta Helvetica*, 3(1):128–168, 1758.

[Lig91] David Lightfoot. *How to set parameters: Arguments from language change*. Cambridge Univ Press, 1991.

[Lot83] M. Lothaire. Combinatorics on words. *Encyclopedia of Mathematics and Its Applications - Vol 17, Addison-Wesley*, 1983.

[Lov93] László Lovász. Random walks on graphs: A survey, 1993.

[LS90] László Lovász and Miklós Simonovits. The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In *Foundations*

of Computer Science, 1990. Proceedings., 31st Annual Symposium on, pages 346–354. IEEE, 1990.

[LW86] Tomasz Łuczak and John C. Wierman. The chromatic number of random graphs at the double-jump threshold. *Combinatorica*, 9(1):39–49, 1986.

[McK84] Brendan D. McKay. Asymptotics for 0-1 matrices with prescribed line sums. *Enumeration and Design,(Academic Press, 1984)*, pages 225–238, 1984.

[MMW10] Mehryar Mohri, Pedro J. Moreno, and Eugene Weinstein. Efficient and robust music identification with weighted finite-state transducers. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(1):197–207, 2010.

[Moh96] Mehryar Mohri. On some applications of finite-state automata theory to natural language processing. *Journal of Natural Language Engineering*, 2(01):61–80, March 1996.

[Moh97] Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational linguistics*, 23(2):269–311, 1997.

[MPR02] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[MSPA10] Irit Meir, Wendy Sandler, Carol Padden, and Mark Aronoff. Emerging sign languages. *Oxford handbook of deaf studies, language, and education*, 2:267–280, 2010.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[MW91] Brendan D. McKay and Nicholas C. Wormald. Asymptotic enumeration by degree sequence of graphs with degrees $o(n^{1/2})$. *Combinatorica*, 11(4):369–382, 1991.

[NS60] Donald J. Newman and Lawrence Shepp. The double dixie cup problem. *The American Mathematical Monthly*, 67(1):58–61, 1960.

[PRU03] Gopal Pandurangan, Prabhakar Raghavan, and Eli Upfal. Building low-diameter peer-to-peer networks. *Selected Areas in Communications, IEEE Journal on*, 21(6):995–1002, 2003.

[PVV09] Etienne Perron, Dinkar Vasudevan, and Milan Vojnović. Using three states for binary consensus on complete graphs. In *INFOCOM 2009, IEEE*, pages 2527–2535. IEEE, 2009.

[PW93] Leonard Pitt and Manfred K. Warmuth. The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of the ACM (JACM)*, 40(1):95–142, January 1993.

[RBB+02] Owen Rambow, Srinivas Bangalore, Tahir Butt, Alexis Nasr, and Richard Sproat. Creating a finite-state parser with application semantics. In *Proceedings of the 19th international conference on Computational linguistics-Volume 2*, pages 1–5. Association for Computational Linguistics, 2002.

[Rei05] Omer Reingold. Undirected st-connectivity in log-space. In *Proceedings*

of the Thirty-seventh Annual ACM Symposium on Theory of Computing, STOC '05, 2005.

[RG96] José Ruiz and Pedro Garcia. Learning k-piecewise testable languages from positive data. In *Grammatical Interference: Learning Syntax from Sentences*, pages 203–210. Springer, 1996.

[Rol79] Hans-Anton Rollik. Automaten in planaren graphen. In *Theoretical Computer Science 4th GI Conference*, pages 266–275. Springer, 1979.

[RYC14] Russell Richie, Charles Yang, and Marie Coppola. Modeling the emergence of lexicons in homesign systems. *Topics in cognitive science*, 6(1):183–195, 2014.

[Sel08] Linda Sellie. Learning random monotone DNF under the uniform distribution. In *The 21st Annual Conference on Learning Theory (COLT 2008)*, pages 181–192. Citeseer, 2008.

[Sel09] Linda Sellie. Exact learning of random DNF over the uniform distribution. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 45–54. Association for Computing Machinery, 2009.

[SGSC96] Richard Sproat, William Gale, Chilin Shih, and Nancy Chang. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational linguistics*, 22(3):377–404, 1996.

[Sha51] Claude E. Shannon. Presentation of a maze-solving machine. In *The 8th Conference of the Josiah Macy Jr. Found (Cybernetics)*, pages 173–180, March 1951.

[Sim75] Imre Simon. Piecewise testable events. In *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*, pages 214–222. Springer, 1975.

[SM11] Purnamrita Sarkar and Andrew W. Moore. Random walks in social networks and their applications: A survey. In Charu C. Aggarwal, editor, *Social Network Data Analytics*, pages 43–77. Springer US, 2011.

[TB73] Boris (Boaz) Avraamovich Trakhtenbrot and Ian Martynovich Barzdin. Finite Automata: Behavior and Synthesis. *Fundamental Studies in Computer Science, V. 1*, 1973.

[Thi68] Gabriel Thierrin. Permutation automata. *Theory of Computing Systems*, 2(1):83–90, 1968.

[Val84] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Wor81] Nicholas C. Wormald. The asymptotic distribution of short cycles in random regular graphs. *Journal of Combinatorial Theory, Series B*, 31(2):168 – 182, 1981.

[Wor99] Nicholas C. Wormald. Models of random regular graphs. In John D. Lamb and Donald A. Preece, editors, *Surveys in Combinatorics, 1999*, pages 239–298. Cambridge University Press, 1999. Cambridge Books Online.

[WS98] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, (393):440–442, 1998.

[WVO12] Marco Wiering and Martijn Van Otterlo. *Reinforcement Learning: State-of-the-Art*, volume 12. Springer, 2012.