Some Machine Learning Methods from Sequential Input

by

Kwang-Sung Jun

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN–MADISON

2015

Date of final oral examination: June 26, 2015

The dissertation is approved by the following members of the Final Oral Committee:
Xiaojin (Jerry) Zhu, Associate Professor, Computer Sciences
Mark Craven, Professor, Biostatistics and Medical Informatics
Robert Nowak, Professor, Electrical and Computer Engineering
Timothy Rogers, Professor, Psychology
Stephen Wright, Professor, Computer Sciences

ACKNOWLEDGMENTS

First, I would like to thank my advisor Professor Xiaojin (Jerry) Zhu. His enthusiasm, relentless pursuit of excellence, and ability to examine problems from every single angle were a great inspiration for me. It was Jerry who gave me an opportunity to start research in the first semester of my graduate program and warmed me up with a nonparametric Bayesian model. He happily let me read his books, and sat down with me to explain the math. He also taught me how to debug mathematical codes. He cared so much about my career and experience that he encouraged me to apply for summer internships. The industrial job experience and the networks I made through these internships are now an invaluable asset to me. Whenever I needed to give a talk at an important place, his feedbacks on my presentations were truly a million dollar lesson. He also invited me to numerous meetings with other researchers, which became a great opportunity to collaborate. Thanks to him, I met Professor Tim Rogers, Professor Robert Nowak, Kevin Jamieson, and Professor Ming Yuan who all became my coauthor. He also encouraged me to start a reading group on non-i.i.d. learning, which became a firm base for my research. Organizing and leading this group were a priceless experience.

More than half of my thesis could not have happened without Professor Timothy Rogers. He introduced me to the world of cognitive psychology. He happily shared his data and analysis for our research in a neat format, and provided insight on human cognition. Sections he wrote for publications were also a great lesson on academic writing.

I also like to thank Professor Robert Nowak who collaborated with me throughout my Ph.D. program. I met Rob when Jerry invited him to our meeting on a multi-armed bandit project. He provided tremendous help with intuition on the problem we are facing. He was patient with my clumsy proof and walked me through how to make it concise.

I'd like to thank my preliminary exam and defense committee. Professor Benjamin Snyder provided great insight on my research topics from a natural language processing point of view. Professor Charles Dyer worked on an iPad application project with me where his feedbacks and enthusiasm for research were a great inspiration for me. Professor Mark Craven introduced me to the world of bioinformatics and how machine

learning methods are used in the domain. Professor Stephen Wright taught me the fundamental knowledge of optimization. The lecture note I compiled from his class is still a good reference on the background of optimization.

There are many coauthors and collaborators to whom I owe a great deal. Kevin Jamieson has been a tremendous help on the multi-armed bandit project. He kindly spared his time to answer my questions on his paper. He was enthusiastic enough to read my proofs that are probably not in the best shape. Burr Settles has kindly shared his feature volunteering data. He made a video introduction of our paper, which helped promoting our research. His writing advice was invaluable when I first wrote a draft of our paper for ICML. Professor Ming Yuan provided great guidance on writing proofs on our INVITE random walk model. Junming Sui (Xu) has been my partner throughout my graduate program who sat down next to me in the office. We worked together on many projects and reading groups. Having a discussion with him was always insightful and enjoyable. Bryan Gibson was my first collaborator at UW-Madison. I learned Matlab and Dirichlet process from him. He was also generous enough to read my writing for my preliminary exam and give me feedback on English composition. Zhuoran Yang was a passionate mathematician who would not hesitate to start writing a proof on whiteboard in the middle of a discussion.

I thank my fellow graduate students. I was able to learn so much from people I met here on various aspects of graduate life in Madison such as daily life hacks, course advice, thesis templates, project ideas, and math understanding. In alphabetical order, I want to thank: David Andrzejewski, Manish Bansal, Deborah Chasman, Gautam Dasarathy, Andrew Goldberg, Hyunwoo Kim, Taedong Kim, Wonhwa Kim, Finn Kuusisto, Sangkyun Lee, Shike Mei, Yimin Tan, Zhuoran Yang, Jongwon Yoon, and Wei Zhang.

I also like to thank Professor Kyu-Baek Hwang for introducing me to the world of machine learning, encouraging me to submit my first paper as an undergraduate, and writing a recommendation letter for me. I'd like to thank many professors from my undergraduate time who gave me valuable advice and wrote recommendation letters for me: Professor Jaeyoung Choi, Professor Youngtaek Park, and Professor Seungmin Yang. My graduate student life was financially stable thanks to the Korea Foundation for Advanced Studies and the chairman Tae-Won Chey. I owe a lot to the foundation, and that motivates me to work harder.

Finally, I could not have completed the program without the care and love from my family. I'd like to thank my parents for an endless support and encouragement.

CONTENTS

Contents iv

List of Tables vi

List of Figures vii

Abstract viii

1 Introduction 1

2 Sampling with Reduced Replacement 4
- *2.1 Sampling With Reduced Replacement* 4
- *2.2 Relation to Other Sampling Paradigms* 5
- *2.3 Efficient Parameter Estimation* 7
- *2.4 Application I: Feature Volunteering for Text Classification* 9
- *2.5 Application II: Verbal Fluency for Brain Damaged Patients* 17
- *2.6 Related Works* 20

3 Initial-Visit Emitting Random Walk 23
- *3.1 The INVITE Random Walk* 23
- *3.2 Computing the INVITE likelihood* 25
- *3.3 Consistency of the MLE* 27
- *3.4 Parameter Estimation via Regularized Maximum Likelihood* 29
- *3.5 Toy Experiments* 33
- *3.6 Application: Human Memory Search* 35
- *3.7 Related Work* 38

4 Top Arm Identification in Multi-Armed Bandit with Batch Arm Pulls 41
- *4.1 Related Works* 42
- *4.2 The Fixed Confidence Setting* 43
- *4.3 The Fixed Budget Setting* 51
- *4.4 Experiments* 57

5 Conclusion 63

A Appendices 66

A.1 Results Required for Theorem 3.7 66

A.2 Proof of (4.9) 73

Bibliography 74

LIST OF TABLES

2.1 An example feature volunteering response for building sports document classifier 10
2.2 Domains in the feature volunteering application. 14
2.3 Text categorization results. 16
2.4 An example verbal fluency response for the animal category 18

3.1 Statistics of the verbal fluency data. 37
3.2 Verbal fluency test set log likelihood. 37

4.1 Toy experiment result: the speedup in the number of batches in the fixed confidence setting. 59

LIST OF FIGURES

2.1 Screenshot of the feature volunteering interface. 15

2.2 Verbal fluency experimental results. SWIRL distributions for patients, controls, and general word frequency on the World Wide Web (using Google 1T 5-gram data) are denoted by $\mathbf{s}_P$, $\mathbf{s}_C$, and $\mathbf{s}_W$, respectively. 22

3.1 Example Markov chains 24

3.2 An INVITE log likelihood plot showing its nonconcavity 30

3.3 Toy experiment results where the error is measured with the Frobenius norm. 35

4.1 Microwell experiment result. (b-e) plots the false negative rate vs. budget B. 60

4.2 (a) Twitter daily pattern (b) Expected rewards of users (c) Experiment result 62

ABSTRACT

Along with an exponentially growing amount of data, machine learning methods have been very successful and popular in a variety of tasks including regression, classification, clustering, etc. Much research has focused on batch data where each item is independent and identically distributed (i.i.d.). Recently, there has been growing interest in machine learning settings that go beyond the batch i.i.d. data. In my thesis, we tackle some problems that arise from sequential inputs where observations are often non-i.i.d. or coupled with sequential decision making. Such problems carry a number of unique computational challenges. Finding solutions to these problems would have a significant impact in a variety of applications.

We propose a new sampling paradigm called sampling with reduced replacement (SWIRL) that resembles the Pólya urn model in its sampling procedure, but is critically different in that it is non-exchangeable. Thus, the order of the item matters in SWIRL. When applied to a machine learning application, SWIRL learns from a list of simple rules expressing domain knowledge to help improving classification accuracy. In addition, SWIRL learns some important aspects of humans who generated the list and can help effective diagnosis of cognitive impairments.

Although SWIRL successfully models a non-i.i.d., non-exchangeable data, the model intrinsically captures zeroth-order information of items. In applications, a group of items tend to be near each other in a list, indicating dependency between them in the generative process. This motivates us to consider a higher-order model. We study a random walk with a repeat-censoring mechanism that outputs an item only when it is visited for the first time. We call this model *initial-visit emitting* (INVITE) random walk. Though such a repeat-censoring mechanism was proposed earlier and shown to effectively model human behavior in a cognitive task, a computational study on INVITE such as parameter estimation was lacking in prior works. We propose the first tractable parameter estimation method for INVITE and prove its theoretical properties such as consistency. In a cognitive psychology application, the parameter estimation of INVITE opens up new horizons in the study of human memory search.

Finally, we introduce a new setting in the multi-armed bandit (MAB) problem where arms are pulled in batches. In other words, one has to make b arm pulls as a single action. This new setting is well-motivated by real-world applications, and is

not subsumed by any existing settings such as delayed feedback or multiple plays. A natural cost here is the number of batches instead of the number of arm pulls, the standard cost in MAB problems. Given an algorithm that finds the top-k arms with T *arm pulls* under the standard setting, can we propose an algorithm that does the same with T/b *batches*? This might not be possible due to the limited adaptivity in batch arm pulls. Interestingly, we show that one can almost achieve the T/b batches with carefully desigined algorithms and their analysis. We propose two efficient algorithms and address the fundamental question of what we gain or loose by pulling multiple arms in batches when compared to the standard setting of pulling one arm at a time.

1 INTRODUCTION

There has been growing interest in sequential input in the machine learning community. Sequential input is abundant, and provides us with unique challenges. When data is generated in a sequential manner, the likelihood of observing a particular item in the next time often correlates with the items generated so far. For example, Higher-order Markov models consider a few most recently generated items. Online learning algorithms address issues arising from a situation where data points arrive one at a time. In the problem of concept drift, the underlying data distribution may change over time. In active learning, the environment offers the opportunity for the learner to decide which data to ask for, which can significantly reduce the required number of labeled data to perform well. In multi-armed bandit problems, a learner must choose one of the many arms at a time. Then, the learner receives a reward from the chosen arm but not from the other arms. While repeating this action-feedback loop, the learner tries to maximize its cumulative reward. Reinforcement learning introduces a state space where taking an action results in not only a reward but also a state transition.

Finding solutions to all these problems above is important not only for its valuable applications but also for expanding the boundary of machine learning. *In this thesis, we consider some problems from sequential inputs that are motivated by important applications and propose novel models and algorithms that address the key issues so as to excel in both theory and practice.* This thesis consists of two parts. In the first part, we propose two distinct methods (Chapter 2 and 3 respectively) for non-i.i.d., non-exchangeable list of items. In the second part (Chapter 4), we introduce a new multi-armed bandit (MAB) problem and propose algorithms that are provably efficient in solving it.

The two methods in the first part are both motivated by tasks that are of significant interest in cognitive psychology and machine learning. One such task arises from a scenario where one wants to train a machine learner to classify sports articles. She might begin by generating a list of relevant phrase⇒label rules (e.g., "touchdown⇒football, home-run⇒baseball"). Incorporating such *feature volunteering* as training data in a machine learning algorithm has been an area of considerable research interest [17, 41, 53]. Another task asks humans to respond to the following: given a category label as a cue (e.g. animals, vehicles, etc.) generate as many example words as possible in 60 seconds without repetition. For example, "cat, dog, cow, horse, otter, wolf, deer, antelope,...". This task is called *verbal fluency*, which has been of great interest among cognitive scientists since it yields rich information about human semantic memory [30, 31] and has been widely adopted in neurology to aid in the diagnosis of cognitive dysfunction [42, 54, 51].

We collectively call the two tasks above as *human-generated lists*. In both tasks, the generation of an item is non-i.i.d. and non-exchangeable since previously-generated items must not be generated afterwards and the order of items matters. Furthermore, they share interesting traits stemming from the unique human cognitive mechanism and memory structure. The standard sampling paradigms such as sampling with or without replacement are not a good fit since they are exchangeable. Incorporating the distinct characteristics of the human-generated lists into a computational model is a new challenge for the machine learning community.

In Chapter 2, we propose a new sampling paradigm called *sampling with reduced replacement* (SWIRL). SWIRL is comparable to the Pólya urn model since items are drawn from an urn and then replaced. The replacement is, however, performed in a way that is in between sampling with replacement and without replacement. This procedure turns out to be the key to introducing non-exchangeability that can model item orders and the repeats in lists at the same time. We present SWIRL and its efficient parameter estimation (i.e. learning). In the two applications introduced above, we show that the estimated parameters of SWIRL are successfully employed to show its effectiveness.

In Chapter 3, we consider a random walk with a repeat-censoring mechanism that outputs a state only when it is visited for the first time. The observation is thus a permutation of the items or a prefix of it. We call this model *initial-visit emitting* (INVITE) random walk. Abbott et al. have shown that such a random walk exhibits the human behavior of clustering and switching cluster in verbal fluency tasks [1]. However, no study has provided a proper way to perform parameter estimation. A naive approach to perform parameter estimation is intractable since there are infinitely many underlying random walk trajectories that might have produced an observed, censored list. We propose the first tractable way for computing the INVITE maximum likelihood estimate (MLE), and show its modeling power on real-world verbal fluency responses.

In the second part of the thesis, we turn to a new multi-armed bandit (MAB) problem in which arms must be sampled in batches of size b, rather than one at a time. This batch arm pull setting is motivated by two important applications. In social media, one might want to find the top-k users who are most active about a certain topic. If one has access to all the tweets, the problem reduces to a counting problem. In reality, publicly available APIs typically rate-limit the number of users one can follow at the same time, which means there exists a batch constraint of size b (e.g. b=5000 in Twitter). In biology experimentation for finding genes that are important in virus replication, a useful device is microwell array that can perform multiple b gene experiments at a time (e.g. b=384). In both applications, the batch constraints naturally arise, and the sampling cost is the number of batches. Existing algorithms fail to operate provably correctly under the batch arm pull setting since they are designed to consider the number of *arm pulls* rather than the number of *batches*.

In Chapter 4, we propose algorithms that operate under the batch arm pull setting and analyze their performance in the number of batches. The analysis addresses an important question of whether or not one has to spend more number of arm pulls under the batch constraint than under the standard setting due to the batch constraint. We also address under what MAB problems one achieves b factor reduction in the number of batches when compared to the number of arm pulls in pursuing a target level of identification error. Our theoretical result indicates that such a b factor reduction is almost achieved with carefully designed algorithms. We demonstrate the new batch MAB algorithms in the two real-world applications introduced above and show that they outperform baseline methods.

2 SAMPLING WITH REDUCED REPLACEMENT

We propose a new sampling paradigm called *sampling with reduced replacement* (SWIRL). Informally, SWIRL is “in-between” sampling with and without replacement, since a drawn item is replaced but with its probability discounted. This brings non-exchangeability and allows us to model order and repetition simultaneously. We formally define SWIRL and provide the maximum likelihood estimate to learn model parameters from human-generated lists. Though not in closed form, our likelihood function is convex and easily optimized. We present a machine learning application called *feature volunteering* where one volunteers a list of simple rules of the form word⇒ label (e.g. “puck” ⇒ “hockey”) that she believes to help classifying documents in a domain of interest (e.g. sports documents). We train SWIRL on a crowdsourced set of feature volunteering lists to incorporate the estimated parameters into downstream text classifiers. We perform the incorporation via popular frameworks where the estimated parameters of SWIRL improve classification accuracy upon baseline methods. We then present a psychology application: verbal fluency, where the estimated parameters correctly capture different mental process of the two groups, and are used to classify healthy vs. brain damaged populations.

2.1 Sampling With Reduced Replacement

Let $\mathcal{V}$ denote the vocabulary of items for a task, and $\mathbf{z} = (z_1, \ldots, z_n)$ be an *ordered* list of n items where $z_t \in \mathcal{V}$ and the z's are not necessarily distinct.

We formally define the generative process of SWIRL. Assume that there exists an unnormalized distribution over the items $\mathcal{V}$. That is, let $s_i \geq 0$ be the “initial size” of item i for $i \in \mathcal{V}$, not necessarily normalized. SWIRL selects item i with probability proportional to s_i. Critically, the size of the selected item (say, z_1) will be *discounted* by a factor $\alpha \in [0, 1]$ for the next draw: $s_{z_1} \leftarrow \alpha s_{z_1}$. This reduces the chance that item z_1 will be selected again in the future. To make it a full generative model, we assume a Poisson(λ) list length distribution. The process of generating a single list $\mathbf{z} = (z_1, \ldots, z_n)$ is specified in Algorithm 1.

The set of N lists produced independently is written $\mathbf{z}^{(1)} = (z_1^{(1)}, \ldots, z_{n^{(1)}}^{(1)})$, $\ldots$, $\mathbf{z}^{(N)} = (z_1^{(N)}, \ldots, z_{n^{(N)}}^{(N)})$.

Algorithm 1 The SWIRL Model

```
Parameters: $\lambda, \mathbf{s} = \{s_i \mid i \in \mathcal{V}\}, \alpha$.
$n \sim \text{Poisson}(\lambda)$
for $t = 1, \ldots, n$ do
    $z_t \sim \text{Multinomial}\left(\frac{s_i}{\sum_{j \in \mathcal{V}} s_j} \mid i \in \mathcal{V}\right)$
    $s_{z_t} \leftarrow \alpha s_{z_t}$
end for
```

2.2 Relation to Other Sampling Paradigms

Obviously, $\alpha = 1$ recovers sampling with replacement, while $\alpha = 0$ appears to be sampling without replacement. However, the nuance is better revealed if we consider an urn model. Consider an urn with colored balls in it. There are $|\mathcal{V}|$ distinct colors. For each color $i \in \mathcal{V}$, let there be m_i balls with that color. The total number of balls in the urn is $\sum_{i \in \mathcal{V}} m_i$. Let $\mathbf{m} = \{m_i \mid i \in \mathcal{V}\}$. Furthermore, let s_i be the ball size for color i. All m_i balls of color i have this same size s_i. Let the probability of drawing a ball be proportional to its size. For example, the probability that the first draw from the urn produces a ball with color i is

$$\Pr(\text{ball has color } i) = \frac{s_i m_i}{\sum_{j \in \mathcal{V}} s_j m_j}. \tag{2.1}$$

We can now contrast several sampling paradigms:

- **Sampling without replacement when $|\mathcal{V}| = 2$.** This is the standard and simplest sampling without replacement paradigm. There are only two colors (black and white, or success and failure) with equal size: $s_1 = s_2$. One draws $n \leq m_1 + m_2$ balls from the urn without replacement (i.e., draws are according to (2.1); if a draw produces a ball with color i then $m_i \leftarrow m_i - 1$ for the next draw). The traditional quantity of interest is k, the number of black balls in the n draws. Note that k is a count that summarizes the sequence of n draws. It is well known that k follows a hypergeometric distribution hypg$(k; \mathbf{m}, n)$, where the population size is $\sum_{i \in \mathcal{V}} m_i$, the number of successes in the population is m_1, and the number of draws is n.

- **Sampling without replacement when $|\mathcal{V}| > 2$.** When there are multiple colors and when the sizes s_i are the same for all colors $i \in \mathcal{V}$, the summary statistic of n sample without replacement draws is a vector $\mathbf{k} = \{k_i \mid i \in \mathcal{V}\}$, and $\mathbf{k}$ follows the familiar multivariate hypergeometric distribution mhypg$(\mathbf{k}; \mathbf{m}, n)$.

• **Sampling without replacement when $|\mathcal{V}| > 2$ and the sizes may be different.** Again, balls are drawn according to (2.1) and are not replaced. $\mathbf{k}$ is the summary count vector for n draws. The distribution of $\mathbf{k}$ now follows the so-called multivariate Wallenius' noncentral hypergeometric distribution $\text{mwnchypg}(\mathbf{k}; \mathbf{m}, \mathbf{s}, n)$, which is a generalization to the multivariate hypergeometric distribution [55, 13, 21]. "Noncentral" refers to the fact that the s_i's may be different. It is worth noting that after drawing a ball of color i, we *subtract* a fixed amount s_i from the "total size" of color i.

• **SWIRL.** We only have one ball of each color: $m_i = 1, \forall i \in \mathcal{V}$. However, each size s_i may be different. Balls are drawn according to (2.1) but are replaced: Upon drawing a ball with color i, we "trim" its size by *multiplying* a discount factor α with s_i. This results in a geometric — rather than arithmetic — reduction in that color's total size. Furthermore, in our applications we are interested in a full sequence of n draws (i.e., the ordered list $\mathbf{z}$) rather than a summary count vector $\mathbf{k}$.

The following example further illustrates the difference between the paradigms.

Example (Non-exchangeability)**.** Sampling without replacement *is well-known to be exchangeable. This is not true for SWIRL. Let there be two colors $\mathcal{V} = \{1, 2\}$. Consider two experiments with matching total size for each color:*

(Experiment 1) There are $m_1 = a$ balls of color 1 and $m_2 = b$ balls of color 2. Let $s_1 = s_2 = 1$. We perform sampling without replacement. Let z_i be the color of the ith draw. Then

$$\begin{aligned}
P(z_1 = 1) &= \frac{a}{a+b} \\
P(z_2 = 1) &= P(z_2 = 1 \mid z_1 = 1)P(z_1 = 1) + P(z_2 = 1 \mid z_1 = 2)P(z_1 = 2) \\
&= \frac{a-1}{a-1+b} \cdot \frac{a}{a+b} + \frac{a}{a+b-1} \cdot \frac{b}{a+b} \\
&= \frac{a}{a+b}
\end{aligned}$$

That is, the marginals $P(z_1 = 1) = P(z_2 = 1)$. This extends to marginals on all $z_t, t \leq a + b$ and is the classic exchangeability result.

(Experiment 2) There is $m_1 = 1$ ball of color 1 with size $s_1 = a$, and $m_2 = 1$ ball

of color 2 with size $s_2 = b$. We perform SWIRL with discounting factor α. Then

$$\begin{aligned} P(z_1 = 1) &= \frac{a}{a+b} \\ P(z_2 = 1) &= P(z_2 = 1 \mid z_1 = 1)P(z_1 = 1) + P(z_2 = 1 \mid z_1 = 2)P(z_1 = 2) \\ &= \frac{\alpha a}{\alpha a + b} \cdot \frac{a}{a+b} + \frac{a}{a + \alpha b} \cdot \frac{b}{a+b} \end{aligned}$$

In general, $P(z_1 = 1) \neq P(z_2 = 1)$. For instance, when $\alpha = 0$ we have $P(z_2 = 1) = b/(a+b)$.

2.3 Efficient Parameter Estimation

Recall that there are three parameters in SWIRL: size $\mathbf{s}$, Poisson intensity λ, and discount factor α. Given observed lists $\mathbf{z}^{(1)} \ldots \mathbf{z}^{(N)}$, the log likelihood is

$$\ell = \sum_{j=1}^{N} n^{(j)} \log \lambda - \lambda + \sum_{t=1}^{n^{(j)}} \log P\left(z_t^{(j)} \mid z_{1:t-1}^{(j)}, \mathbf{s}, \alpha\right),$$

where $n^{(j)}$ is the length of the jth list. The key quantity here is

$$P\left(z_t^{(j)} \mid z_{1:t-1}^{(j)}, \mathbf{s}, \alpha\right) = \frac{\alpha^{c(z_t^{(j)}, j, t)} s_{z_t^{(j)}}}{\sum_{i \in \mathcal{V}} \alpha^{c(i,j,t)} s_i} \tag{2.2}$$

where $c(i, j, t)$ is the count of item i in the jth prefix list of length $t-1$: $z_1^{(j)}, \ldots, z_{t-1}^{(j)}$. This repeated discounting and renormalization couples α and $\mathbf{s}$, making the MLE difficult to solve in closed form.

The λ parameter is independent of $\mathbf{s}$ and α, so the MLE $\widehat{\lambda}$ is the usual average list length. To simplify notation, we omit the Poisson and focus on the log likelihood of $\mathbf{s}$ and α, namely

$$\ell(\mathbf{s}, \alpha) = \sum_{j=1}^{N} \sum_{t=1}^{n^{(j)}} c(z_t^{(j)}, j, t) \log \alpha + \log s_{z_t^{(j)}} - \log \sum_{i \in \mathcal{V}} \alpha^{c(i,j,t)} s_i.$$

We transform the variables into the log domain which is easier to work with: $\boldsymbol{\beta} \equiv \log \mathbf{s}$, and $\gamma \equiv \log \alpha$. The log likelihood can now be written as $\ell(\boldsymbol{\beta}, \gamma) =$

$$\sum_{j=1}^{N} \sum_{t=1}^{n^{(j)}} c(z_t^{(j)}, j, t)\gamma + \beta_{z_t^{(j)}} - \log \sum_{i \in \mathcal{V}} \exp(c(i, j, t)\gamma + \beta_i).$$

Due to the log-sum-exp form, $\ell(\boldsymbol{\beta}, \gamma)$ is concave in $\boldsymbol{\beta}$ and γ [8]. Note the initial sizes $\mathbf{s}$ are scale invariant. We remove this invariance by setting $s_{MFI} = 1$ where MFI is the most frequent item in $\mathbf{z}^{(1)} \ldots \mathbf{z}^{(N)}$. Equivalently, $\beta_{MFI} = 0$.

The complete convex optimization problem for finding the MLE of SWIRL is

$$\begin{aligned} \min_{\boldsymbol{\beta}, \gamma} \quad & -\ell(\boldsymbol{\beta}, \gamma) && (2.3) \\ \text{s.t.} \quad & \beta_{MFI} = 0 && (2.4) \\ & \gamma \leq 0. && (2.5) \end{aligned}$$

The MLE is readily computed by quasi-Newton methods such as LBFGS, where the required gradient for (2.3) is computed by

$$\begin{aligned} \frac{\partial \ell}{\partial \beta_i} &= -c(i) + \sum_{j=1}^{N} \sum_{t=1}^{n^{(j)}} \frac{\exp(c(i, j, t)\gamma + \beta_i)}{\sum_{i' \in \mathcal{V}} \exp(c(i', j, t)\gamma + \beta_{i'})} \\ \frac{\partial \ell}{\partial \gamma} &= \sum_{j=1}^{N} \sum_{t=1}^{n^{(j)}} -c(z_t^{(j)}, j, t) + \frac{\sum_{i \in \mathcal{V}} \exp(c(i, j, t)\gamma + \beta_i) c(i, j, t)}{\sum_{i' \in \mathcal{V}} \exp(c(i', j, t)\gamma + \beta_{i'})}, \end{aligned}$$

where $c(i)$ is the total count of occurrences for item i in the lists $\mathbf{z}^{(1)} \ldots \mathbf{z}^{(N)}$, for $i \in \mathcal{V}$.

Maximum A Posteriori Estimate Additionally, it is easy to compute the MAP estimate when we have prior knowledge on $\boldsymbol{\beta}$ and γ. Suppose we believe the initial sizes should be approximately proportional to $\mathbf{s}_0$. For example, in the vehicle verbal fluency task, $\mathbf{s}_0$ may be the counts of various vehicles in a large corpus. We can define $\beta_{0_i} \equiv \log(s_{0_i}/s_{0_{MFI}}); \forall i \in \mathcal{V}$, and adopt a Gaussian prior on $\boldsymbol{\beta}$ centered around $\boldsymbol{\beta}_0$ (equivalently, $\mathbf{s}$ follows a log-Normal distribution). Since this prior removes the scale invariance on $\boldsymbol{\beta}$, we no longer need the constraint $\beta_{MFI} = 0$. Similarly, we may have prior knowledge of γ_0. The MAP estimate is the solution to

$$\min_{\boldsymbol{\beta}, \gamma} \quad -\ell(\boldsymbol{\beta}, \gamma) + \tau_1 \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|^2 + \tau_2(\gamma - \gamma_0)^2 \qquad (2.6)$$

$$\text{s.t.} \quad \gamma \le 0 \ ,$$

where τ_1, τ_2 are appropriate regularization terms to prevent overfitting.

2.4 Application I: Feature Volunteering for Text Classification

We now turn to a machine learning application of SWIRL: training text classifiers from human-volunteered *feature labels* (rather than documents). A feature label is a simple rule stating that the presence of a word or phrase indicates a particular class label. For example, “touchdown⇒football” implies that documents containing the word “touchdown” probably belong to the class “football.” Some prior work exploits a *bag* of volunteered features from users, where each has an equal importance. Although such feature labels help classification [41], the order of volunteered words is not taken into account. The order turns out to be very useful, however, as we will show later. Other prior work solicits labeled features through a form of feature *queries*: the computer, via unsupervised corpus analysis (e.g., topic modeling), proposes a list of high-probability candidate features for a human to label [17].

Departing from previous works, we point out that the human teacher, upon hearing the categorization goal (e.g., the classes to be distinguished), can *volunteer* an ordered list of feature labels without first consulting a corpus; see Table 2.1. This “feature volunteering” procedure is particularly attractive when a classifier is promptly needed for a novel task, since humans can be recruited quickly via crowdsourcing or other means, even before a corpus is fully compiled. Another possibility, which we recommend for practitioners[1], is to treat feature volunteering as a crucial first step in a chain of progressively richer interactive supervision, followed by queries on both features and documents. Queries can be selected by the computer in order to build better classifiers over time. Such a combination has been studied recently [53].

In this section, we show that feature volunteering can be successfully combined with

[1] Feature volunteering and feature query labeling are complementary ways of obtaining feature labels. The former provides a way to capture importance of feature labels (e.g., by their order), while the latter can consult extra resources (e.g., unlabeled data) to harvest more feature labels, as pointed out by **(author?)** [41].

Order	Item
1	goal ⇒ soccer
2	net ⇒ hockey
3	touch down ⇒ football
4	home run ⇒ baseball
5	pitcher ⇒ baseball
6	dribble ⇒ basketball
7	puck ⇒ hockey
8	quarterback ⇒ football
9	bat ⇒ baseball
10	first base ⇒ baseball
11	outfield ⇒ baseball
12	defense ⇒ football
13	wide reciever ⇒ football
14	hockey stick ⇒ hockey
15	point guard ⇒ basketball
16	second base ⇒ baseball
17	field goal ⇒ football
18	off sides ⇒ soccer

Table 2.1: An example feature volunteering response for building sports document classifier

two existing frameworks for training classifiers with labeled features: (*i*) Generalized Expectation (GE) for logistic regression [17] and (*ii*) informative Dirichlet priors (IDP) for multinomial naïve Bayes [53]. We also show that "order matters" by highlighting the value of SWIRL as a model for feature volunteering. That is, by endowing each volunteered feature with its size s_i as estimated in Section 2.3, we can build better classifiers than by treating the volunteered features equally under both of these machine learning frameworks.

Generalized Expectation (GE)

Let $y \in \mathcal{Y}$ be a class label, and $\mathbf{x} \in \mathbb{R}^{|\mathcal{F}^+|}$ be a vector describing a text document using feature set $\mathcal{F}^+$, which is a super set of volunteered feature set $\mathcal{F}$. Consider the conditional probability distributions realizable by multinomial logistic regression

$$\Delta_\Theta = \{p_{\boldsymbol{\theta}}(y \mid \mathbf{x}) \mid \boldsymbol{\theta} \in \mathbb{R}^{|\mathcal{F}^+| \times |\mathcal{Y}|}\}, \tag{2.7}$$

where

$$p_{\boldsymbol{\theta}}(y \mid \mathbf{x}) = \frac{\exp(\theta_y^\top \mathbf{x})}{\sum_{y' \in \mathcal{Y}} \exp(\theta_{y'}^\top \mathbf{x})}. \tag{2.8}$$

Generalized Expectation (GE) seeks a distribution $p^* \in \Delta_\Theta$ that matches a set of given reference distributions $\widehat{p_f}(y)$: distributions over the label y if the feature $f \in \mathcal{F}$ is present. We will construct $\widehat{p_f}(y)$ from feature volunteering and compare it against other constructions in the next section. Before that, we specify the matching sought by GE [17].

We restrict ourselves to sufficient statistics based on counts, such that $x_f \in \mathbf{x}$ is the number of times feature f occurs in the document. Let $\mathcal{U}$ be an unlabeled corpus. The empirical mean conditional distribution on documents where $x_f > 0$ is given by

$$M_f\left[p_{\boldsymbol{\theta}}(y \mid \mathbf{x})\right] \equiv \frac{\sum_{\mathbf{x} \in \mathcal{U}} \mathbb{1}\{x_f > 0\} p_{\boldsymbol{\theta}}(y \mid \mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{U}} \mathbb{1}\{x'_f > 0\}}. \tag{2.9}$$

GE training minimizes a regularized objective based on the KL-divergence of these distributions:

$$p^* = \operatorname*{argmin}_{p_{\boldsymbol{\theta}} \in \Delta_\Theta} \sum_{f \in \mathcal{F}} \mathrm{KL}\Big(\widehat{p_f}(y) \,\Big\|\, M_f[p_{\boldsymbol{\theta}}(y \mid \mathbf{x})]\Big) + \frac{\|\boldsymbol{\theta}\|^2}{2}. \tag{2.10}$$

In other words, GE seeks to make the reference and empirical distributions as similar as possible.

Constructing GE Reference Distributions $\widehat{p_f}(y)$ with SWIRL

Recall that in feature volunteering, the N human users produce multiple ordered lists $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}$. Each item z in these lists is a $f \Rightarrow y$ (feature⇒label) pair. It is possible that the same f appears multiple times in different z's, mapping to different y's (e.g., “goalie⇒soccer” and “goalie⇒hockey”). In this case we say feature f co-occurs with multiple y's.

For each list $\mathbf{z}$, we split it into $|\mathcal{Y}|$ sublists by item labels. This produces one ordered sublist per class. We collect all N sublists for a single class y, and treat them as N lists generated by SWIRL from the yth urn. We find the MLE of this yth urn using (2.3), and normalize it to sum to one. We are particularly interested in the size estimates $\mathbf{s}_y = \{s_{f \Rightarrow y} \mid f \in \mathcal{F}\}$. This is repeated for all $|\mathcal{Y}|$ urns, so we have $\mathbf{s}_1, \ldots, \mathbf{s}_{|\mathcal{Y}|}$.

We construct reference distributions using

$$\widehat{p_f}(y) = \frac{s_{f \Rightarrow y}}{\sum_{y' \in \mathcal{Y}} s_{f \Rightarrow y'}}; \ \forall f \in \mathcal{F} \ , \tag{2.11}$$

where $\mathcal{F}$ is the union of features appearing in the lists $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}$ and $s_{f \Rightarrow y} = 0$ if feature f is absent from the yth list. For example, imagine "goalie⇒soccer" appears near the top of a list, so $s_{\text{goalie} \Rightarrow \text{soccer}}$ is large (say 0.4), "goalie⇒hockey" appears much later in another list, so $s_{\text{goalie} \Rightarrow \text{hockey}}$ is small (say 0.1), and "goalie" is never associated with "baseball". Then by (2.11), $\hat{p}_{\text{goalie}}(\text{soccer}) = 0.8$, $\hat{p}_{\text{goalie}}(\text{hockey}) = 0.2$, and $\hat{p}_{\text{goalie}}(\text{baseball}) = 0$.

In our experiments, we compare three ways of creating reference distributions. **GE/SWIRL** is given by Equation (2.11). **GE/Equal** is defined as

$$\widehat{p_f}(y) = \frac{\mathbb{1}\{s_{f \Rightarrow y} > 0\}}{\sum_{y' \in \mathcal{Y}} \mathbb{1}\{s_{f \Rightarrow y'} > 0\}}; \ \ \forall i \in \mathcal{F}, \tag{2.12}$$

which is similar to (2.11), except that all features co-occurring with y have equal size. This serves as a baseline to investigate whether order matters. **GE/Schapire** is the reference distribution used in previous work [17]:

$$\widehat{p_f}(y) = \begin{cases} q/m \quad \text{if feature } f \text{ co-occurs with } y \\ (1-q)/(|\mathcal{Y}| - m) \quad \text{otherwise,} \end{cases} \tag{2.13}$$

where m is the number of distinct labels co-occurring with feature f in $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}$, and q is a smoothing parameter. We use $q = 0.9$ as in prior work.

Informative Dirichlet Priors (IDP)

Another way to use feature volunteering is by training multinomial Naïve Bayes models, where feature⇒label rules are adapted as informative priors on feature parameters. The class distribution $p(y)$ is parameterized by π_y, and the likelihood of a document $\mathbf{x}$ given a class label y is modeled as $p(\mathbf{x} \mid y) = \prod_f (\phi_{fy})^{x_f}$, where x_f is the frequency count of feature f and ϕ_{fy} is multinomial parameter for feature f under class y.

Dirichlet priors are placed on each class-conditional multinomial parameter, where the hyperparameter is denoted by d_{fy} for phrase f under class y.

We estimate π_y by class proportion of labeled documents, and ϕ_{fy} by posterior expectation as follows:

$$\phi_{fy} \; \propto \; d_{fy} + \sum_{\mathbf{x}} p(y \mid \mathbf{x}) x_f, \tag{2.14}$$

where ϕ_{fy} is normalized over phrases to sum to one for each y. When learning from only labeled instances, $p(y \mid \mathbf{x}) \in \{0, 1\}$ indicates the true labeling of instance $\mathbf{x}$. When learning from both labeled and unlabeled instances, we run EM as follows. First, initialize ϕ_{fy}'s by (2.14) using only the d_{fy} hyperparameters. Second, repeatedly apply (2.14) until convergence, where the second summation term is over both labeled and unlabeled instances and $p(y \mid \mathbf{x})$ for unlabeled instance $\mathbf{x}$ is computed using Bayes rule.

Constructing IDP Priors d_{fy} with SWIRL

We compare two approaches for incorporating prior knowledge into naïve Bayes by feature volunteering. **IDP/SWIRL** sets the hyperparameters as follows: $d_{fy} = 1 + k n_y s_{f \Rightarrow y}$, where f is a feature, k a parameter, and n_y is the number of unique features in y's list. Again, we compute $s_{f \Rightarrow y}$ via SWIRL as in Section 2.4.

Note that replacing $s_{f \Rightarrow y}$ with $\mathbb{1}\{s_{f \Rightarrow y} > 0\}/n_y$ recovers prior work [53]. In this method, only the association between a feature f and a class y is taken into account, rather than relative importance of these associations. This baseline, **IDP/Settles**, sets $d_{fy} = 1 + k\mathbb{1}\{s_{f \Rightarrow y} > 0\}$ and serves to investigate whether order matters in human-generated lists.

Experiments

We conduct feature volunteering text classification experiments in three different domains: **sports** (sports articles), **movies** (movie reviews), and **webkb** (university web pages). The classes, number of human participants (N), and the number of distinct list features they produced ($|\mathcal{F}|$) are listed in Table 2.2.

Participants. A total of 135 undergraduate students from the University of Wisconsin-Madison participated for partial course credit. No one participated in more

Corpus	Class Labels	N	$\|\mathcal{F}\|$	$\|\mathcal{F}^+\|$
sports	baseball, basketball, football, hockey, soccer	52	594	2948
movies	negative, positive	27	382	2514
webkb	course, faculty, project, student	56	961	2521

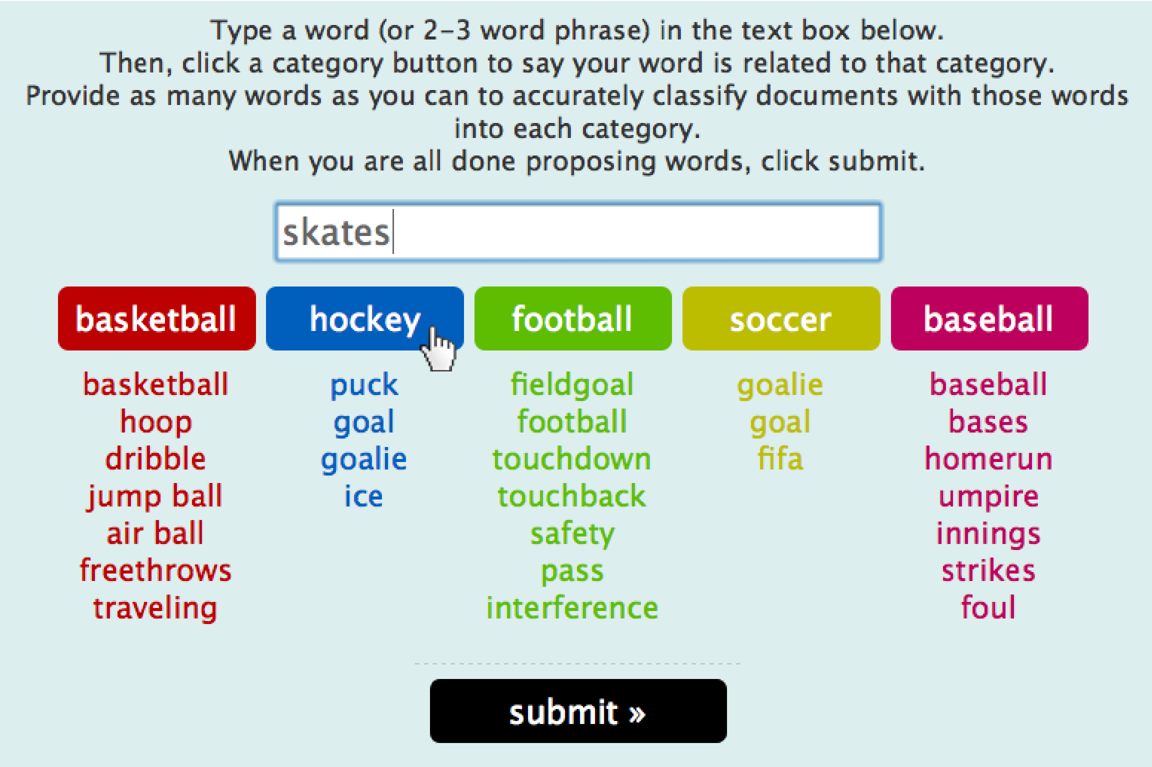
Table 2.2: Domains in the feature volunteering application.

than one domain. All human studies in this paper were approved by the institutional review board.

Procedure. Participants were informed of only the class labels, and were asked to provide as many words or short phrases as they thought would be necessary to accurately classify documents into the classes. They volunteered features using the web-based computer interface illustrated in Figure 2.1 (shown here for the sports domain). The interface consists of a text box to enter features, followed by a series of buttons corresponding to labels in the domain. When the participant clicks on a label (e.g., "hockey" in the figure), the phrase is added to the bottom of the list below the corresponding button, and erased from the input box. The feature⇒label pair is recorded for each action in sequence. The label order was randomized for each subject to avoid presentation bias. Participants had 15 minutes to complete the task.

Data cleaning. We normalized the volunteered phrases by case-folding, punctuation removal, and space normalization. We manually corrected obvious misspellings. We also manually mapped different forms of a feature to its dictionary canonical form: for example, we mapped "lay-up" and "lay up" to "layup." The average (and maximum) list length is 39 (91) for sports, 20 (46) for movies, and 40 (85) for webkb. Most participants volunteered features at a fairly uniform speed for the first five minutes or so; some then exhausted ideas. This suggests the importance of combining feature volunteering with feature and document querying, as mentioned earlier. This combination is left for future work.

Unlabeled corpus $\mathcal{U}$. Computing (2.9) requires an unlabeled corpus. We produce $\mathcal{U}$ for the **sports** domain by collecting 1123 Wikipedia documents via a shallow web crawl starting from the top-level wiki-category for the five sport labels



Type a word (or 2-3 word phrase) in the text box below.
Then, click a category button to say your word is related to that category.
Provide as many words as you can to accurately classify documents with those words into each category.
When you are all done proposing words, click submit.

skates

basketball	hockey	football	soccer	baseball
basketball	puck	fieldgoal	goalie	baseball
hoop	goal	football	goal	bases
dribble	goalie	touchdown	fifa	homerun
jump ball	ice	touchback		umpire
air ball		safety		innings
freethrows		pass		strikes
traveling		interference		foul

Figure 2.1: Screenshot of the feature volunteering interface.

(e.g., "Category:Baseball"). We produce a matching $\mathcal{U}$ for the **movies** and **webkb** domains from the standard movie sentiment corpus [45] (2000 instances) and the WebKB corpus [15] (4199 instances), respectively. Note that $\mathcal{U}$'s are treated as unlabeled for training our models, however, we use each $\mathcal{U}$'s in a transductive fashion to serve as our test sets as well.

Training with GE. We define the feature set for learning $\mathcal{F}^+$ (i.e., the dimensionality in (2.7)) to be the union of $\mathcal{F}$ (volunteered phrases) plus all unigrams occurring at least 50 times in the corpus $\mathcal{U}$ for that domain. That is, we include all volunteered phrases, even if they are not a unigram or appear fewer than 50 times in the corpus. $|\mathcal{F}^+|$ for each domain is listed in Table 2.2. We construct the reference distributions according to **GE/SWIRL**, **GE/Equal**, and **GE/Schapire** as in section 2.4, and find the optimal logistic regression models $p^*(y \mid \mathbf{x})$ by solving (2.10) with LBFGS for each domain and reference distribution.

Corpus	SWIRL	Equal	Schapire	FV
sports	**0.865**	0.847	0.795	**0.875**
movies	**0.733**	**0.733**	**0.725**	0.681
webkb	**0.463**	0.444	0.429	0.426

(a) GE accuracies for logistic regression

Corpus	SWIRL	Settles	FV
sports	**0.911**	0.901	0.875
movies	**0.687**	0.656	**0.681**
webkb	**0.659**	**0.651**	0.426

(b) IDP accuracies for multinomial naïve Bayes

Table 2.3: Text categorization results.

Training with IDP. We use the same $\mathcal{F}^+$ in GE training, construct IDP hyperparameters according to **IDP/SWIRL** and **IDP/Settles**, and learn MNB classifiers as in section 2.4 using uniform π_y. Following prior work, we apply one-step EM with $k = 50$.

Feature Voting Baseline (FV). We also include a simple baseline for both frameworks. To classify a document $\mathbf{x}$, FV scans through unique volunteered features for which $x_f > 0$. Each class y for which $f \Rightarrow y$ exists in $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}$ receives one vote. At the end, FV predicts the label as the one with the most votes. Ties are broken randomly. Accuracy of FV is measured by averaging over 20 trials due to this randomness.

Results. Text classifiers built from volunteered features and SWIRL consistently outperform the baselines. Classification accuracies of the different models under GE and IDP are shown in Table 2.3(a,b). For each domain, we show the best accuracy in bold face, as well as any accuracies whose difference from the best is not statistically significant[2]. Under the GE framework, **GE/SWIRL** is the best on movies and webkb, and is indistinguishable from the best on sports. Under the IDP framework, **IDP/SWIRL** consistently outperforms all baselines.

The fact that both **GE/SWIRL** and **IDP/SWIRL** are better than (or on par with) the baselines under both frameworks strongly indicates that *order matters*. That is, when working with human-generated lists, the item order carries information that can be useful to machine learning algorithms. Such information can be extracted by

[2] Using paired two-tailed t-tests, $p < 0.05$.

SWIRL parameter estimates and successfully incorporated into a secondary classification task. Although dominated by SWIRL-based approaches, **FV** is reasonably strong and may be a quick stand-in due to its simplicity.

A caveat: the human participants were only informed of the class labels and did not know $\mathcal{U}$. Mildly amusing mismatch ensued. For example, the webkb corpus was collected in 1997 (before the "social media" era), but the volunteered feature labels included "facebook⇒student," "dropbox⇒course," "reddit⇒faculty," and so on. Our convenient but outdated choice of $\mathcal{U}$ quite possibly explains the low accuracy of all methods in the webkb domain.

2.5 Application II: Verbal Fluency for Brain Damaged Patients

One human list-generation task that has received detailed examination in cognitive psychology is "verbal fluency." Human participants are asked to say as many examples of a category as possible in one minute with no repetitions [22].[3] For instance, participants may be asked to generate examples of a semantic category (e.g., "animals" or "furniture"), a phonemic or orthographic category (e.g., "words beginning with the letter F"), or an ad-hoc category (e.g., "things you would rescue from a burning house"). See Table 2.4 for an example. Verbal fluency has been widely adopted in neurology to aid in the diagnosis of cognitive dysfunction since the task is sensitive to different cognitive disorders of a participant. We review related works in Section 4.1.

Despite the widespread adoption of the verbal fluency task, standard methods for analyzing the data are comparatively primitive: correct responses and different error types are counted, while sequential information is typically discarded. We propose to use SWIRL as a computational model of the verbal fluency task, since we are unaware of any other such models. We show that, though overly simplified in some respects, SWIRL nevertheless estimates key parameters that correspond to cognitive mechanisms. We further show that these estimates differ in healthy populations versus patients with temporal-lobe epilepsy, a neurological disorder thought to disrupt

[3] Despite the instruction, people still do repeat.

Order	Item
1	cat
2	dog
3	cow
4	horse
5	otter
6	wolf
7	deer
8	antelope
9	gorrilla
10	lion
11	yak
12	armadillo
13	porcupine
14	raccoon
15	fox
16	wolf
17	bear

Table 2.4: An example verbal fluency response for the animal category

semantic knowledge. Finally, we report promising classification results, indicating that our model could be useful in aiding diagnosis of cognitive dysfunction in the future.

Participants. We investigated fluency data generated from two populations: a group of 27 patients with temporal-lobe epilepsy (a disorder thought to disrupt semantic abilities), and a group of 24 healthy controls matched to the patients in age, education, sex, nonverbal IQ and working-memory span. Patients were recruited through an epilepsy clinic at the University of Wisconsin-Madison. Controls were recruited through fliers posted throughout Madison, Wisconsin.

Procedure. We conducted four category-fluency tasks: animals, vehicles, dogs, and boats. In each task, participants were shown a category name on a computer screen and were instructed to verbally list as many examples of the category as possible in 60 seconds without repetition. The audio recordings were later transcribed by lab technicians to render word lists. We normalize the lists by expanding abbreviations (“lab” → “labrador”), removing inflections (“birds” → “bird”), and discarding junk utterances and interjections. The average (and maximum) list length is 20 (37) for animals, 14 (33) for vehicles, 11 (23) for dogs, and 11 (20) for boats.

Results. We estimated SWIRL parameters $\lambda, \mathbf{s}, \alpha$ using (2.3) for the patient and control groups on each task separately, and observed that:

1. *Patients produced shorter lists.* Figure 2.2(a) shows that the estimated Poisson intensity $\hat{\lambda}$ (i.e., the average list length) is smaller for patients on all four tasks. This is consistent with the psychological hypothesis that patients suffering from temporal-lobe epilepsy produce items at a slower rate, hence fewer items in the time-limit.

2. *Patients and controls have similar lexicon distributions, but both deviate from word usage frequency.* Figure 2.2(b) shows the top 10 lexicon items and their probabilities (normalized $\mathbf{s}$) for patients ($\mathbf{s}_P$) and controls ($\mathbf{s}_C$) in the animals task, sorted by $(\mathbf{s}_{Pi} + \mathbf{s}_{Ci})/2$. $\mathbf{s}_P$ and $\mathbf{s}_C$ are qualitatively similar. We also show corpus probabilities $\mathbf{s}_W$ from the Google Web 1T 5-gram data set [9] for comparison (normalized on items appearing in human-generated lists). Not only does $\mathbf{s}_W$ have smaller values, but its order is very different: horse (0.06) is second largest, while other top corpus words like fish (0.05) and mouse (0.03) are not even in the human top 10. This challenges the psychological view that verbal fluency largely follows real-world lexicon usage frequency. Both observations are supported quantitatively by comparing the Jensen-Shannon divergence (JSD) [4] between the whole distributions $\mathbf{s}_P$, $\mathbf{s}_C$, $\mathbf{s}_W$; see Figure 2.2(c). Clearly, $\mathbf{s}_P$ and $\mathbf{s}_C$ are relatively close, and both are far from $\mathbf{s}_W$.

3. *Patients discount less and repeat more.* Figure 2.2(d) shows the estimated discount factor $\hat{\alpha}$. In three out of four tasks, the patient group has larger $\hat{\alpha}$. Recall that a larger $\hat{\alpha}$ leaves an item's size relatively unchanged, thus increasing the item's chance to be sampled again — in other words, more repeats. This is consistent with the psychological hypothesis that patients have a reduced ability to inhibit items already produced.

Healthy vs. Patient Classification. We conducted additional experiments where we used SWIRL parameters to build down-stream healthy vs. patient classifiers. Specifically, we performed leave-one-out (LOO) classification experiments for each of the four verbal fluency tasks. Given a task, each training set (minus one person's list for testing) was used for learning two separate SWIRL models with MAP estimation (2.6): one for patients and the other for healthy participants. We set $\tau_1 = \infty$ and $\boldsymbol{\beta}_0 = \mathbf{1}$ due to the finding that both populations have similar lexicon distributions, and $\tau_2 = 0$. We classified the held-out test list by likelihood ratio threshold at 1 for the patient vs. healthy models. The LOO accuracies of SWIRL on animals, vehicles, dogs and

[4] Each group's MLE of $\mathbf{s}$ (and hence p) has zero probability on items only in the other groups. We use JSD since it is symmetric and allows disjoint supports.

boats were 0.647, 0.706, 0.784, and 0.627, respectively. In contrast, the majority vote baseline has accuracy $27/(24+27) = 0.529$ for all tasks. The improvement for the dogs task over the baseline approach is statistically significant[5].

2.6 Related Works

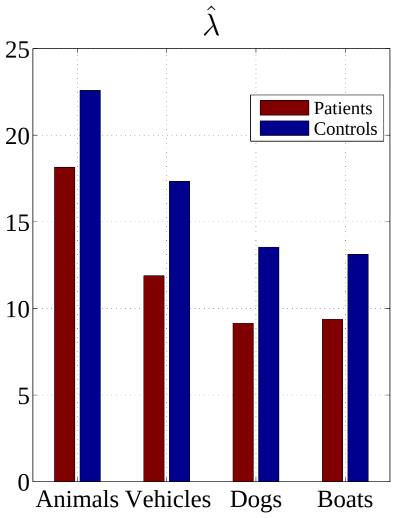
SWIRL resembles many existing sampling paradigms as we have shown in Section 2.2. Pólya urn model generalizes sampling with replacement and without replacement. Suppose that there are x black balls and y white balls in an urn. At each trial, the model draws a ball uniformly at random, then replaces it back with c additionally new balls of the same color. Although c is usually taken as a nonnegative number that models "the rich get richer" phenomenon, $c = 0$ makes the model sampling with replacement, and $c = -1$ makes the model sampling without replacement. Now, take the number of balls of the same color as the "size" notion as in SWIRL. We can view it as having two items "black" and "white" with their size x and y respectively. In this view, the Pólya urn model with $c = -1$ (sampling without replacement) is comparable to SWIRL since sampling a ball (e.g. a black ball) reduces the size of the item (its color, e.g. "black") by subtracting one from it (e.g. $x \leftarrow x - 1$). Such an additive discounting is different from the multiplicative discounting of SWIRL that multiplies a discount factor $\alpha \in [0, 1]$ to the size. This subtle point turns out to make a huge difference in their exchangeability; the Pólya urn model is exchangeable whereas SWIRL is not. The non-exchangeability of SWIRL is the key property that allows modeling order preference in a list.

There are numerous works that builds a classifier by features and their labels, which are called labeled features. Learning from labeled features is attractive since it is a cheaper form of supervision than document labels. Liu et al. proposed a procedure that computes a ranked list of words under each category [41]. Then, the users are prompted to choose words that they believe to be important in classifying documents. Druck et al. also compute a list of words and present it to ask users to assign a label to each word [17]. Both works prepare a list of features to prompt users to provide supervision where the order information of items is not collected. In contrast, our application lets users to *volunteer* words and their labels as a list where the order

[5] Using paired two-tailed t-tests, $p < 0.05$.

information is naturally preserved. Wu et al. proposed a variant of support vector machine that can learn with labeled features [56]. Although the authors asked humans to volunteer words for each category, they ignored the order information of the words all together, unlike using SWIRL in feature volunteering.

Verbal fluency has been widely adopted in neurology to aid in the diagnosis of cognitive dysfunction [42, 54, 51]. Performance in verbal fluency is generally sensitive to a variety of neurological disorders [49], but different syndromes also give rise to different patterns of impairment, making it useful for diagnosis [46]. Category and letter fluency in particular are sensitive to a broad range of cognitive disorders resulting from brain damage [50]. For instance, patients with prefrontal injuries are prone to inappropriately repeating the same item several times (*perseverative* errors) [5], whereas patients with pathology in the anterior temporal cortex are more likely to generate incorrect responses (*semantic* errors) and produce many fewer items overall [32, 50]. Despite these observations and widespread adoption of the task, standard methods for analyzing the data are comparatively primitive: correct responses and different error types are counted, while sequential information is typically discarded. In contrast, our work proposes a systematic way for diagnosing patients by employing a computational model.



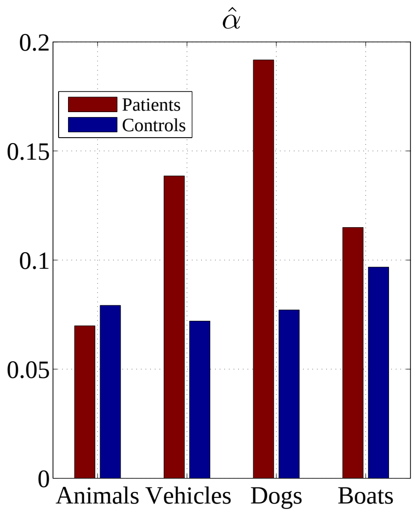
(a) list length $\hat{\lambda}$

Item	$\mathbf{s}_P$	$\mathbf{s}_C$	$\mathbf{s}_W$
cat	.15	.13	.05
dog	.17	.11	.08
lion	.04	.04	.01
tiger	.04	.03	.01
bird	.04	.02	.03
elephant	.03	.03	.01
zebra	.01	.04	.00
bear	.03	.03	.03
snake	.02	.02	.01
horse	.02	.03	.06
⋮	⋮	⋮	⋮

(b) top 10 “animal” words

Task	Pair	JSD
animals	$(\mathbf{s}_P, \mathbf{s}_C)$	.09
	$(\mathbf{s}_P, \mathbf{s}_W)$	.17
	$(\mathbf{s}_C, \mathbf{s}_W)$	.19
vehicles	$(\mathbf{s}_P, \mathbf{s}_C)$	.18
	$(\mathbf{s}_P, \mathbf{s}_W)$	.23
	$(\mathbf{s}_C, \mathbf{s}_W)$	.25
dogs	$(\mathbf{s}_P, \mathbf{s}_C)$	.15
	$(\mathbf{s}_P, \mathbf{s}_W)$	.54
	$(\mathbf{s}_C, \mathbf{s}_W)$	.55
boats	$(\mathbf{s}_P, \mathbf{s}_C)$	.15
	$(\mathbf{s}_P, \mathbf{s}_W)$	.54
	$(\mathbf{s}_C, \mathbf{s}_W)$	.52

(c) distribution comparisons



(d) discount factor $\hat{\alpha}$

Figure 2.2: Verbal fluency experimental results. SWIRL distributions for patients, controls, and general word frequency on the World Wide Web (using Google 1T 5-gram data) are denoted by $\mathbf{s}_P$, $\mathbf{s}_C$, and $\mathbf{s}_W$, respectively.

3 INITIAL-VISIT EMITTING RANDOM WALK

We consider a random walk with a repeat-censoring mechanism that outputs an item only when it is visited for the first time. Abbott et al. showed that such a random walk generates items in clusters [1]; items are generated in multiple runs of similar items, exhibiting clustering and switching cluster repeatedly. We call this model *initial-visit emitting* (INVITE) random walk. Though simple and attractive, no studies have proposed a proper and tractable way for estimating the INVITE parameters. We present the first tractable parameter estimation of INVITE, and address its theoretical properties such as identifiability and consistency. In a cognitive science application of human memory search, we show that the INVITE random walk successfully models human responses from a human memory task, which sheds light on using it as a tool for inferring the underlying semantic network of humans.

3.1 The INVITE Random Walk

Consider a random walk on a set of n states S with an initial distribution $\boldsymbol{\pi} > 0$ (entry-wise) and an arbitrary transition matrix $\mathbf{P}$ where P_{ij} is the probability of jumping from state i to j. A surfer starts from a random initial state drawn from $\boldsymbol{\pi}$. She outputs a state if it is the first time she visits that state. Upon arriving at an already visited state, however, she does not output the state. The random walk continues indefinitely. Therefore, the output consists of states in the order of their first-visit; the underlying entire walk trajectory is hidden. We further assume that the time steps of each output are unobserved. For example, consider the random walk over four states in Figure 3.1(a). If the underlying random walk takes the trajectory $(1, 2, 1, 3, 1, 2, 1, 4, 1, \ldots)$, the observation is $(1, 2, 3, 4)$.

We call this random walk model the *initial-visit emitting* (INVITE) random walk. We say that the observation produced by INVITE is a *censored* list since non-initial visits are censored. It is easy to see that a censored list is a permutation of the n states or a prefix thereof (more on this later). We denote a censored list by $\mathbf{a} = (a_1, a_2, \ldots, a_M)$ where $M \leq n$. A censored list is not Markovian since the probability of a transition in censored list depends on the whole history rather than just the current state. It is worth noting that INVITE is distinct from Broder's

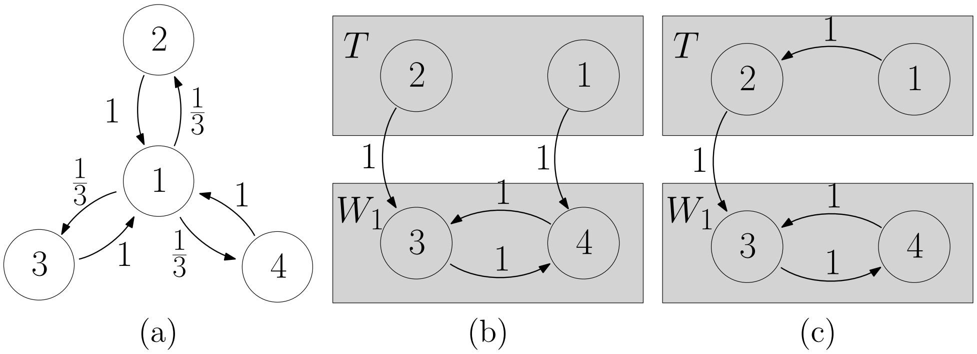


Figure 3.1: Example Markov chains

algorithm for generating random spanning trees [10], or the self-avoiding random walk [20], or cascade models of infection. We discuss the technical difference to related works in Section 4.1.

We characterize the type of output INVITE is capable of producing, given that the underlying uncensored random walk continues indefinitely. A state s is said to be *transient* if a random walk starting from s has nonzero probability of not returning to itself in finite time and *recurrent* if such probability is zero. A set of states A is *closed* if a walk cannot exit A; i.e., if $i \in A$ and $j \notin A$, then a random walk from i cannot reach j. A set of states B is *irreducible* if there exists a path between every pair of states in B; i.e., if $i, j \in B$, then a random walk from i can reach j. Define $[M] = \{1, 2, \ldots, M\}$. The following theorem states that a finite state Markov chain can be uniquely decomposed into disjoint sets.

Theorem 3.1. *[18] If the state space S is finite, then S can be written as a disjoint union $T \cup W_1 \cup \ldots \cup W_K$, where T is a set of transient states that is possibly empty and each W_k, $k \in [K]$, is a nonempty closed irreducible set of recurrent states.*

The following theorem states what a censored list should look like. We use $a_{1:M}$ as a shorthand for $a_1, \ldots, a_M$.

Theorem 3.2. *Consider a Markov chain $\mathbf{P}$ with the decomposition $S = T \cup W_1 \cup \ldots \cup W_K$ as in Theorem 3.1. A censored list $\mathbf{a} = (a_{1:M})$ generated by INVITE on $\mathbf{P}$ has zero or more transient states, followed by all states in one and only one closed irreducible set. That is, $\exists \ell \in [M]$ s.t. $\{a_{1:\ell-1}\} \subseteq T$ and $\{a_{\ell:M}\} = W_k$ for some $k \in [K]$.*

Proof. We first claim that *(i)* there must be a recurrent state i in a censored list where $i \in W_k$ for some k. Then, it suffices to show that given *(i)* is true, *(ii)* recurrent states outside W_k cannot appear, *(iii)* every states in W_k must appear, and *(iv)* a transient state cannot appear after a recurrent state.

(i): suppose there is no recurrent state in a censored list $\mathbf{a} = (a_{1:M})$. Then, every state a_i, $i \in [M]$, is a transient state. Since the underlying random walk runs indefinitely in finite state space, there must be a state a_j, $j \in [M]$, that is visited infinitely many times. This contradicts the fact that a_j is a transient state.

Suppose a recurrent state $i \in W_k$ was visited. Then,

(ii): the random walk cannot escape W_k since W_k is closed.

(iii): the random walk will reach to every state in W_k in finite time since W_k is finite and irreducible.

(iv): the same reason as *(iii)*.

□

As an example, when the graph is fully connected INVITE is capable of producing all $n!$ permutations of the n states as the censored lists. As another example, in Figure 3.1 (b) and (c), both chains has two transient states $T = \{1, 2\}$ and two recurrent states $W_1 = \{3, 4\}$. (b) has no path that visits both 1 and 2, and thus every censored list must be a prefix of a permutation. However, (c) has a path that visits both 1 and 2, thus can generate (1,2,3,4), a full permutation.

In general, each INVITE run generates a permutation of n states, or a prefix of a permutation. Let $\mathrm{Sym}(n)$ be the symmetric group on $[n]$. Then, the data space $\mathcal{D}$ of censored lists is $\mathcal{D} \equiv \{(a_{1:k}) \mid \mathbf{a} \in \mathrm{Sym}(n), k \in [n]\}$.

3.2 Computing the INVITE likelihood

Learning and inference under the INVITE model is challenging due to its likelihood function. A naive method to compute the probability of a censored list $\mathbf{a}$ given $\boldsymbol{\pi}$ and $\mathbf{P}$ is to sum over all uncensored random walk trajectories $\mathbf{x}$ which produces $\mathbf{a}$: $\mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) = \sum_{\mathbf{x} \text{ produces } \mathbf{a}} \mathbb{P}(\mathbf{x}; \boldsymbol{\pi}, \mathbf{P})$. This naive computation is intractable since the summation can be over an infinite number of trajectories $\mathbf{x}$'s that might have produced the censored list $\mathbf{a}$. For example, consider the censored list $\mathbf{a} = (1, 2, 3, 4)$

generated from Figure 3.1(a). There are infinite uncensored trajectories to produce $\mathbf{a}$ by visiting states 1 and 2 arbitrarily many times before visiting state 3, and later state 4.

The likelihood of $\boldsymbol{\pi}$ and $\mathbf{P}$ on a censored list $\mathbf{a}$ is

$$\mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) = \begin{cases} \pi_{a_1} \prod_{k=1}^{M-1} \mathbb{P}(a_{k+1} \mid a_{1:k}; \mathbf{P}) & \\ \quad \text{if } \mathbf{a} \text{ cannot be extended} & \\ 0 \quad \text{otherwise.} & \end{cases} . \tag{3.1}$$

Note we assign zero probability to a censored list that is not completed yet, since the underlying random walk must run forever. We say a censored list $\mathbf{a}$ is *valid* (*invalid*) under $\boldsymbol{\pi}$ and $\mathbf{P}$ if $\mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) > 0$ $(= 0)$.

We first review the *fundamental matrix* in the absorbing random walk. A state that transits to itself with probability 1 is called an *absorbing state*. Given a Markov chain $\mathbf{P}$ with absorbing states, we can rearrange the states into $\mathbf{P}' = \begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$, where $\mathbf{Q}$ is the transition between the nonabsorbing states, $\mathbf{R}$ is the transition from the nonabsorbing states to absorbing states, and the rest trivially represent the absorbing states. Theorem 3.3 presents the *fundamental matrix*, the essential tool for the tractable computation of the INVITE likelihood.

Theorem 3.3. *[16] The fundamental matrix of the Markov chain $\mathbf{P}'$ is $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$. N_{ij} is the expected number of times that a chain visits state j before absorption when starting from i. Furthermore, define $\mathbf{B} = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R}$. Then, B_{ik} is the probability of a chain starting from i being absorbed by k. In other words, $B_{i\cdot}$ is the absorption distribution of a chain starting from i.*

As a tractable way to compute the likelihood, we propose a novel formulation that turns an INVITE random walk into a series of absorbing random walks. Although INVITE itself is not an absorbing random walk, each segment that produces the next item in the censored list can be modeled as one. That is, for each $k = 1 \ldots M - 1$ consider the segment of the uncensored random walk starting from the previous output a_k until the next output a_{k+1}. For this segment, we construct an absorbing random walk by keeping $a_{1:k}$ nonabsorbing and turning the rest into the absorbing states. A random walk starting from a_k is eventually absorbed by a state in $S \setminus \{a_{1:k}\}$. The probability of being absorbed by a_{k+1} is exactly the probability of outputting a_{k+1} after outputting $a_{1:k}$ in INVITE. Formally, we construct an absorbing random walk $\mathbf{P}^{(k)}$:

$$\mathbf{P}^{(k)} = \begin{pmatrix} \mathbf{Q}^{(k)} & \mathbf{R}^{(k)} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \tag{3.2}$$

where the states are ordered as $a_{1:M}$. Corollary 3.4 summarizes our computation of the INVITE likelihood.

Corollary 3.4. *The k-th step INVITE likelihood for $k \in [M-1]$ is*

$$\mathbb{P}(a_{k+1} \mid a_{1:k}, \mathbf{P}) = \begin{cases} [(\mathbf{I} - \mathbf{Q}^{(k)})^{-1}\mathbf{R}^{(k)}]_{k1} & \textit{if } (\mathbf{I} - \mathbf{Q}^{(k)})^{-1} \textit{ exists} \\ 0 & \textit{otherwise} \end{cases} \tag{3.3}$$

Suppose we observe m independent realizations of INVITE:

$$D_m = \left\{ \left(a_1^{(1)}, ..., a_{M_1}^{(1)}\right), ..., \left(a_1^{(m)}, ..., a_{M_m}^{(m)}\right) \right\},$$

where M_i is the length of the i-th censored list. Then, the INVITE log likelihood is $\ell(\boldsymbol{\pi}, \mathbf{P}; D_m) = \sum_{i=1}^{m} \log \mathbb{P}(\mathbf{a}^{(i)}; \boldsymbol{\pi}, \mathbf{P})$.

3.3 Consistency of the MLE

Identifiability is an essential property for a model to be consistent. Theorem 3.5 shows that allowing self-transitions in $\mathbf{P}$ cause INVITE to be unidentifiable. Then, Theorem 3.6 presents a remedy. Let diag($\mathbf{q}$) be a diagonal matrix whose i-th diagonal entry is q_i.

Theorem 3.5. *Let $\mathbf{P}$ be an $n \times n$ transition matrix without any self-transition ($P_{ii} = 0, \forall i$), and $\mathbf{q} \in [0,1)^n$. Define $\mathbf{P}' = diag(\mathbf{q}) + (\mathbf{I} - diag(\mathbf{q}))\mathbf{P}$, a scaled transition matrix with self-transition probabilities $\mathbf{q}$. Then, $\mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) = \mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}')$, for every censored list $\mathbf{a}$.*

Proof. It suffices to show that $\mathbb{P}(a_{k+1} \mid a_{1:k}; \mathbf{P}) = \mathbb{P}(a_{k+1} \mid a_{1:k}; \mathbf{P}')$, where $\mathbf{a} = (a_1, \ldots, a_M)$ and $k \leq M-1$. Define submatrices $(\mathbf{Q}, \mathbf{R})$ and $(\mathbf{Q}', \mathbf{R}')$ from $\mathbf{P}$ and

$\mathbf{P}'$, respectively, as in (3.2). Note that $\mathbf{Q}' = \text{diag}(q_{1:k}) + (\mathbf{I} - \text{diag}(q_{1:k}))\mathbf{Q}$ and $\mathbf{R}' = (\mathbf{I} - \text{diag}(q_{1:k}))\mathbf{R}$.

$$\begin{aligned}\mathbb{P}(a_{k+1} \mid a_{1:k}; \mathbf{P}') &= \left(\mathbf{I} - \text{diag}(q_{1:k}) - (\mathbf{I} - \text{diag}(q_{1:k}))\mathbf{Q}\right)^{-1} (\mathbf{I} - \text{diag}(q_{1:k}))\mathbf{R} \\ &= (\mathbf{I} - \mathbf{Q})^{-1}(\mathbf{I} - \text{diag}(q_{1:k}))^{-1}(\mathbf{I} - \text{diag}(q_{1:k}))\mathbf{R} \\ &= \mathbb{P}(a_{k+1} \mid a_{1:k}; \mathbf{P})\end{aligned}$$

□

For example, consider a censored list $\mathbf{a} = (1, j)$ where $j \neq 1$. Using the fundamental matrix, $\mathbb{P}(a_2 \mid a_1; \mathbf{P}) = (1 - P_{11})^{-1}P_{1j} = (\sum_{j' \neq 1} P_{1j'})^{-1}P_{1j} = (\sum_{j' \neq 1} cP_{1j'})^{-1}cP_{1j}, \forall c$. This implies that multiplying a constant c to P_{1j} for all $j \neq 1$ and renormalizing the first row $\mathbf{P}_{1\cdot}$ to sum to 1 does not change the likelihood.

Theorem 3.6. *Assume the initial distribution* $\boldsymbol{\pi} > 0$ *elementwise. In the space of transition matrices* $\mathbf{P}$ *without self-transitions, INVITE is identifiable.*

Proof. Suppose $(\boldsymbol{\pi}, \mathbf{P}) \neq (\boldsymbol{\pi}', \mathbf{P}')$. We show that there exists a censored list $\mathbf{a}$ such that $\mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) \neq \mathbb{P}(\mathbf{a}; \boldsymbol{\pi}', \mathbf{P}')$.

Case 1: $\boldsymbol{\pi} \neq \boldsymbol{\pi}'$.

It follows that $\pi_i \neq \pi'_i$ for some i. Note that the marginal probability of observing i as the first item in a censored list is $\sum_{\mathbf{a} \in \mathcal{D}: a_1 = i} \mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) = \pi_i$. Then,

$$\sum_{\mathbf{a} \in \mathcal{D}: a_1 = i} \mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) = \pi_i \neq \pi'_i = \sum_{\mathbf{a} \in \mathcal{D}: a_1 = i} \mathbb{P}(\mathbf{a}; \boldsymbol{\pi}', \mathbf{P}').$$

which implies that there exists a censored list $\mathbf{a}$ for which $\mathbb{P}(\mathbf{a}; \pi, \mathbf{P}) \neq \mathbb{P}(\mathbf{a}; \pi', \mathbf{P}')$.

Case 2: $\boldsymbol{\pi} = \boldsymbol{\pi}'$ but $\mathbf{P} \neq \mathbf{P}'$.

It follows that $P_{ij} \neq P'_{ij}$ for some i and j. Then, we compute the marginal probability of observing (i, j) as the first two items in a censored list, which results in

$$\sum_{\mathbf{a} \in \mathcal{D}: \substack{a_1 = i, \\ a_2 = j}} \mathbb{P}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) = \pi_i P_{ij} \neq \pi'_i P'_{ij} = \sum_{\mathbf{a} \in \mathcal{D}: \substack{a_1 = i, \\ a_2 = j}} \mathbb{P}(\mathbf{a}; \boldsymbol{\pi}', \mathbf{P}').$$

Then, there exists a censored list $\mathbf{a}$ for which $\mathbb{P}(\mathbf{a}; \pi, \mathbf{P}) \neq \mathbb{P}(\mathbf{a}; \pi', \mathbf{P}')$.

□

Let $\Delta^{n-1} = \{\mathbf{p} \in \mathbb{R}^n \mid p_i \geq 0, \forall i, \sum_i p_i = 1\}$ be the probability simplex. For brevity, we pack the parameters of INVITE into one vector $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{\theta} \in \boldsymbol{\Theta} = \{(\boldsymbol{\pi}^\top, \mathbf{P}_{1\cdot}, \ldots, \mathbf{P}_{n\cdot})^\top \mid \boldsymbol{\pi}, \mathbf{P}_{i\cdot} \in \Delta^{n-1}, P_{ii} = 0, \forall i\}.$$

Let $\boldsymbol{\theta}^* = (\boldsymbol{\pi}^{*\top}, \mathbf{P}^*_{1\cdot}, \ldots, \mathbf{P}^*_{n\cdot})^\top \in \boldsymbol{\Theta}$ be the true model. Given a set of m censored lists D_m generated from $\boldsymbol{\theta}^*$, the average log likelihood function and its pointwise limit are

$$\widehat{\mathcal{Q}}_m(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \log \mathbb{P}(\mathbf{a}^{(i)}); \boldsymbol{\theta}) \qquad \text{and} \qquad \mathcal{Q}^*(\boldsymbol{\theta}) = \sum_{\mathbf{a} \in \mathcal{D}} \mathbb{P}(\mathbf{a}; \boldsymbol{\theta}^*) \log \mathbb{P}(\mathbf{a}; \boldsymbol{\theta}). \tag{3.4}$$

For brevity, we assume that the true model $\boldsymbol{\theta}^*$ is strongly connected; the analysis can be easily extended to remove it. Under the Assumption A1, Theorem 3.7 states the consistency result.

Assumption A1. *Let $\boldsymbol{\theta}^* = (\boldsymbol{\pi}^{*\top}, \mathbf{P}^*_{1\cdot}, \ldots, \mathbf{P}^*_{n\cdot})^\top \in \boldsymbol{\Theta}$ be the true model. $\boldsymbol{\pi}^*$ has no zero entries. Furthermore, $\mathbf{P}^*$ is strongly connected.*

Theorem 3.7. *Assume A1. The MLE of INVITE $\widehat{\boldsymbol{\theta}}_m \equiv \max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \widehat{\mathcal{Q}}_m(\boldsymbol{\theta})$ is consistent.*

Proof. The proof relies on Lemma A.6 and Lemma A.2 that are presented in the appendix. Since $\boldsymbol{\Theta}$ is compact, the sequence $\{\widehat{\boldsymbol{\theta}}_m\}$ has a convergent subsequence $\{\widehat{\boldsymbol{\theta}}_{m_j}\}$. Let $\boldsymbol{\theta}' = \lim_{j\to\infty} \widehat{\boldsymbol{\theta}}_{m_j}$. Since $\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta}^*) \leq \widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})$,

$$\mathcal{Q}^*(\boldsymbol{\theta}^*) = \lim_{j\to\infty} \widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta}^*) \leq \lim_{j\to\infty} \widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j}) = \mathcal{Q}^*(\boldsymbol{\theta}'),$$

where the last equality is due to Lemma A.6. By Lemma A.2, $\boldsymbol{\theta}^*$ is the unique maximizer of $\mathcal{Q}^*$, which implies $\boldsymbol{\theta}' = \boldsymbol{\theta}^*$. Note that the subsequence was chosen arbitrarily. Since every convergent subsequence converges to $\boldsymbol{\theta}^*$, $\widehat{\boldsymbol{\theta}}_m$ converges to $\boldsymbol{\theta}^*$. □

3.4 Parameter Estimation via Regularized Maximum Likelihood

We present a regularized MLE (RegMLE) of INVITE. We first extend the censored lists that we consider. Now we allow the underlying walk to terminate after finite

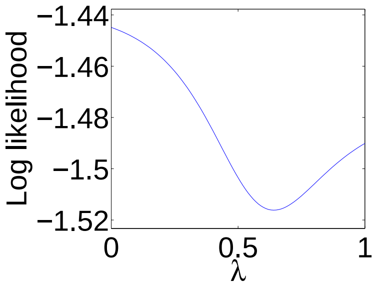


Figure 3.2: An INVITE log likelihood plot showing its nonconcavity

steps because in real-world applications the observed censored lists are often truncated. That is, the underlying random walk can be stopped before exhausting every states the walk could visit. For example, in verbal fluency, participants have limited time to produce a list. Consequently, we use the *prefix likelihood*

$$\mathcal{L}(\mathbf{a}; \boldsymbol{\pi}, \mathbf{P}) = \pi_{a_1} \prod_{k=1}^{M-1} \mathbb{P}(a_{k+1} \mid a_{1:k}; \mathbf{P}). \tag{3.5}$$

We find the RegMLE by maximizing the prefix log likelihood plus a regularization term on $\boldsymbol{\pi}, \mathbf{P}$. Note that, $\boldsymbol{\pi}$ and $\mathbf{P}$ can be separately optimized. For $\boldsymbol{\pi}$, we place a Dirichlet prior and find the maximum a posteriori (MAP) estimator $\widehat{\boldsymbol{\pi}}$ by $\widehat{\pi}_j \propto \sum_{i=1}^m \mathbb{1}_{a_1^{(i)}=j} + C_{\boldsymbol{\pi}}, \forall j$.

Directly computing the RegMLE of $\mathbf{P}$ requires solving a constrained optimization problem, because the transition matrix $\mathbf{P}$ must be row stochastic. We re-parameterize $\mathbf{P}$ which leads to a more convenient unconstrained optimization problem. Let $\boldsymbol{\beta} \in \mathbb{R}^{n \times n}$. We exponentiate $\boldsymbol{\beta}$ and row-normalize it to derive $\mathbf{P}$: $P_{ij} = e^{\beta_{ij}} / \sum_{j'=1}^n e^{\beta_{ij'}}, \forall i, j$. We fix the diagonal entries of $\boldsymbol{\beta}$ to $-\infty$ to disallow self-transitions. We place squared ℓ_2 norm regularizer on $\boldsymbol{\beta}$ to prevent overfitting. The unconstrained optimization problem is:

$$\min_{\boldsymbol{\beta}} \quad -\sum_{i=1}^{m} \sum_{k=1}^{M_i-1} \log \mathbb{P}(a_{k+1}^{(i)} \mid a_{1:k}^{(i)}; \boldsymbol{\beta}) + \frac{1}{2} C_{\boldsymbol{\beta}} \sum_{i \neq j} \beta_{ij}^2 \quad , \tag{3.6}$$

where $C_{\boldsymbol{\beta}} > 0$ is a regularization parameter.

We point out that the objective function of (3.6) is not convex in $\boldsymbol{\beta}$ in general.

Let $n = 5$ and suppose we observe two censored lists $(5, 4, 3, 1, 2)$ and $(3, 4, 5, 1, 2)$. We found with random starts two different local optima $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\beta}^{(2)}$ of (3.6). We plot the prefix log likelihood of $(1-\lambda)\boldsymbol{\beta}^{(1)} + \lambda\boldsymbol{\beta}^{(2)}$, where $\lambda \in [0, 1]$ in Figure 3.2. Nonconcavity of this 1D slice implies nonconcavity of the prefix log likelihood surface in general.

The derivative of the prefix log likelihood w.r.t. $\boldsymbol{\beta}$ Given a censored list $\mathbf{a}$, define a mapping σ that maps a state to its position in $\mathbf{a}$; that is, $\sigma(a_i) = i$. Let $\mathbf{N}^{(k)} = (\mathbf{I} - \mathbf{Q}^{(k)})^{-1}$. Hereafter, we drop the superscript (k) from $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{N}$ when it's clear from the context. Using $\partial(A^{-1})_{k\ell}/\partial A_{ij} = -(A^{-1})_{ki}(A^{-1})_{j\ell}$, the following identity becomes useful:

$$\begin{aligned}
\frac{\partial N_{k\ell}}{\partial Q_{ij}} &= \frac{\partial((\mathbf{I}-\mathbf{Q})^{-1})_{k\ell}}{\partial Q_{ij}} \\
&= \sum_{c,d} \frac{\partial((\mathbf{I}-\mathbf{Q})^{-1})_{k\ell}}{\partial(\mathbf{I}-\mathbf{Q})_{cd}} \frac{\partial(\mathbf{I}-\mathbf{Q})_{cd}}{\partial Q_{ij}} \\
&= \sum_{c,d} ((\mathbf{I}-\mathbf{Q})^{-1})_{kc}((\mathbf{I}-\mathbf{Q})^{-1})_{d\ell} \mathbb{1}_{\{c=i,d=j\}} \\
&= ((\mathbf{I}-\mathbf{Q})^{-1})_{ki}((\mathbf{I}-\mathbf{Q})^{-1})_{j\ell} \\
&= N_{ki}N_{jl}.
\end{aligned}$$

The derivative of $\mathbf{P}$ w.r.t. $\boldsymbol{\beta}$ is given as follows:

$$\begin{aligned}
\frac{\partial P_{rc}}{\partial \beta_{ij}} &= \mathbb{1}\{r = i\} \left(\frac{\mathbb{1}\{j = c\} e^{\beta_{ic}} (\sum_{\ell=1}^{n} e^{\beta_{i\ell}}) - e^{\beta_{ic}} e^{\beta_{ij}}}{(\sum_{\ell=1}^{n} e^{\beta_{i\ell}})^2} \right) \\
&= \mathbb{1}\{r = i\}(-P_{ic}P_{ij} + \mathbb{1}\{j = c\}P_{ic}).
\end{aligned}$$

The derivative of $\log \mathbb{P}(a_{k+1} \mid a_{1:k})$ with respect to $\boldsymbol{\beta}$ is

$$\begin{aligned}
\frac{\partial \log \mathbb{P}(a_{k+1} \mid a_{1:k})}{\partial \beta_{ij}} &= \mathbb{P}(a_{k+1} \mid a_{1:k})^{-1} \sum_{\ell=1}^{k} \frac{\partial(N_{k\ell}R_{\ell 1})}{\partial \beta_{ij}} \\
&= \mathbb{P}(a_{k+1} \mid a_{1:k})^{-1} \left(\sum_{\ell=1}^{k} \frac{\partial N_{k\ell}}{\partial \beta_{ij}} R_{\ell 1} + N_{k\ell} \frac{\partial R_{\ell 1}}{\partial \beta_{ij}} \right)
\end{aligned}$$

We need to compute $\frac{\partial N_{k\ell}}{\partial \beta_{ij}}$:

$$\begin{aligned}
\frac{\partial N_{k\ell}}{\partial \beta_{ij}} &= \sum_{c,d=1}^{k} \frac{\partial((\mathbf{I}-\mathbf{Q})^{-1})_{k\ell}}{\partial(\mathbf{I}-\mathbf{Q})_{cd}} \frac{\partial(\mathbf{I}-\mathbf{Q})_{cd}}{\partial \beta_{ij}} \\
&= \sum_{c,d=1}^{k} (-1) N_{kc} N_{d\ell} \cdot (-1) \mathbb{1}_{\{a_c = i\}} (-P_{ia_d} P_{ij} + \mathbb{1}_{\{a_d = j\}} P_{ia_d}) \\
&= \mathbb{1}_{\{\sigma(i) \leq k\}} N_{k\sigma(i)} \sum_{d=1}^{k} N_{d\ell} (-P_{ia_d} P_{ij} + \mathbb{1}_{\{a_d = j\}} P_{ia_d}),
\end{aligned}$$

where $\sigma(i) \leq k$ means item i appeared among the first k items in the censored list $\mathbf{a}$. Then,

$$\begin{aligned}
\sum_{\ell=1}^{k} \frac{\partial N_{k\ell}}{\partial \beta_{ij}} R_{\ell 1} &= \mathbb{1}_{\{\sigma(i) \leq k\}} N_{k\sigma(i)} \sum_{\ell,d=1}^{k} N_{d\ell} (-P_{ia_d} P_{ij} + \mathbb{1}_{\{a_d = j\}} P_{ia_d}) R_{\ell 1} \\
&= \mathbb{1}_{\{\sigma(i) \leq k\}} N_{k\sigma(i)} \left(-P_{ij} \sum_{d=1}^{k} P_{ia_d} \sum_{\ell=1}^{k} N_{d\ell} R_{\ell 1} + \sum_{d=1}^{k} \mathbb{1}_{\{a_d = j\}} P_{ia_d} \sum_{\ell=1}^{k} N_{d\ell} R_{\ell 1} \right) \\
&= \mathbb{1}_{\{\sigma(i) \leq k\}} N_{k\sigma(i)} \left(-P_{ij} (\mathbf{QNR})_{\sigma(i)1} \; + \mathbb{1}_{\{\sigma(j) \leq k\}} P_{ij} (\mathbf{NR})_{\sigma(j)1} \right)
\end{aligned}$$

and

$$\begin{aligned}
\sum_{\ell=1}^{k} N_{k\ell} \frac{\partial R_{\ell 1}}{\partial \beta_{ij}} &= \sum_{\ell=1}^{k} N_{kl} \mathbb{1}_{\{\ell = \sigma(i)\}} \left(-P_{ia_{k+1}} P_{ij} + \mathbb{1}_{\{a_{k+1} = j\}} P_{ia_{k+1}} \right) \\
&= \mathbb{1}_{\{\sigma(i) \leq k\}} N_{k\sigma(i)} \left(-P_{ia_{k+1}} P_{ij} + \mathbb{1}_{\{a_{k+1} = j\}} P_{ia_{k+1}} \right).
\end{aligned}$$

Putting everything together,

$$\begin{aligned}
&\frac{\partial \log \mathbb{P}(a_{k+1} \mid a_{1:k})}{\partial \beta_{ij}} \\
&= \frac{\mathbb{1}_{\{\sigma(i) \leq k\}} N_{k\sigma(i)}}{\mathbb{P}(a_{k+1} \mid a_{1:k})} (-P_{ij} (\mathbf{QNR})_{\sigma(i)1} + \mathbb{1}_{\{\sigma(j) \leq k\}} P_{ij} (\mathbf{NR})_{\sigma(j)1} \\
&\qquad\qquad - P_{ia_{k+1}} P_{ij} + \mathbb{1}_{\{a_{k+1} = j\}} P_{ia_{k+1}}) \\
&= \frac{\mathbb{1}_{\{\sigma(i) \leq k\}} N_{k\sigma(i)} P_{ij}}{\mathbb{P}(a_{k+1} \mid a_{1:k})} \left(-(\mathbf{QNR})_{\sigma(i)1} + \mathbb{1}_{\{\sigma(j) \leq k\}} (\mathbf{NR})_{\sigma(j)1} \; P_{ia_{k+1}} \left(\frac{\mathbb{1}_{\{a_{k+1} = j\}}}{P_{ij}} - 1 \right) \right)
\end{aligned}$$

for all $i \neq j$.

Efficient Optimization using Averaged Stochastic Gradient Descent Given a censored list $\mathbf{a}$ of length M, computing the derivative of $\mathbb{P}(a_{k+1} \mid a_{1:k})$ w.r.t. $\boldsymbol{\beta}$ takes $O(k^3)$ time for matrix inversion. There are n^2 entries in $\boldsymbol{\beta}$, so the time complexity per item is $O(k^3+n^2)$. This computation needs to be done for $k = 1, ..., (M-1)$ in a list and for m censored lists, which makes the overall time complexity $O(mM(M^3+n^2))$. In the worst case, M is as large as n, which makes it $O(mn^4)$. Even the state-of-the-art batch optimization method such as LBFGS takes a very long time to find the solution for a moderate problem size such as $n \approx 500$.

For a faster computation of the RegMLE (3.6), we turn to averaged stochastic gradient descent (ASGD) [52, 48]. ASGD processes the lists sequentially by updating the parameters after every list. The per-round objective function for $\boldsymbol{\beta}$ on the i-th list is

$$f(\mathbf{a}^{(i)}; \boldsymbol{\beta}) \equiv -\sum_{k=1}^{M_i-1} \log \mathbb{P}(a_{k+1}^{(i)} \mid a_{1:k}^{(i)}; \boldsymbol{\beta}) + \frac{C_\beta}{2m} \sum_{i \neq j} \beta_{ij}^2.$$

We randomly initialize $\boldsymbol{\beta}_0$. At round t, we update the current solution $\boldsymbol{\beta}_t$ with $\boldsymbol{\beta}_t \leftarrow \boldsymbol{\beta}_{t-1} - \eta_t \nabla f(a^{(i)}; \boldsymbol{\beta})$ and the average estimate $\overline{\boldsymbol{\beta}}_t$ with $\overline{\boldsymbol{\beta}}_t \leftarrow \frac{t-1}{t}\overline{\boldsymbol{\beta}}_{t-1} + \frac{1}{t}\boldsymbol{\beta}_t$. We let $\eta_t = \gamma_0(1+\gamma_0 a t)^{-c}$. We use $a = C_\beta/m$ and $c = 3/4$ following [7] and choose γ_0 by running the algorithm on a small subsample of the train set. After running ASGD for a fixed number of epochs, we take the final average estimate $\overline{\boldsymbol{\beta}}_t$ as the solution.

3.5 Toy Experiments

We compare INVITE against two other popular estimators of $\mathbf{P}$: naive random walk (RW) and First-Edge (FE). RW is the regularized MLE of the naive random walk, pretending the censored lists are the underlying uncensored walk trajectory:

$$\widehat{P}_{rc}^{(RW)} \propto \left(\sum_{i=1}^{m} \sum_{j=1}^{M_i-1} \mathbb{1}_{(a_j^{(i)}=r) \wedge (a_{j+1}^{(i)}=c)} \right) + C_{RW},$$

for all row r and column c. Though simple and popular, RW is a biased estimator due to the model mismatch.

FE is the First-Edge estimator proposed in [2] for graph structure recovery in cascade model. FE uses only the first two items in each censored list:

$$\widehat{P}_{rc}^{(FE)} \propto \left(\sum_{i=1}^{m} \mathbb{1}_{(a_1^{(i)}=r) \wedge (a_2^{(i)}=c)} \right) + C_{FE}.$$

Because the first transition in a censored list is always the same as the first transition in the underlying trajectory, FE is a consistent estimator of $\mathbf{P}$ (assuming $\boldsymbol{\pi}$ has no zero entries). In fact, FE is equivalent to the RegMLE of the length two prefix likelihood of the INVITE model. However, we expect FE to waste information since it discards the rest of the censored lists.

In this section, we compare the three estimators INVITE, RW, and FE on toy datasets, where the observations are indeed generated by an initial-visit emitting random walk. We construct three undirected, unweighted graphs of $n = 25$ nodes each:

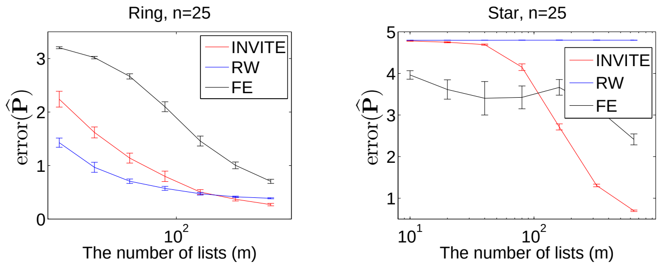
- **Ring**: A ring graph
- **Star**: $n - 1$ nodes each connected to a "hub" node
- **Grid**: A 2-dimensional $\sqrt{n} \times \sqrt{n}$ lattice.

The initial distribution $\boldsymbol{\pi}^*$ is uniform, and the transition matrix $\mathbf{P}^*$ at each node has an equal transition probability to its neighbors. For each graph, we generate datasets with $m \in \{10, 20, 40, 80, 160, 320, 640\}$ censored lists. Each censored list has length n. We note that, in the star graph a censored list contains many apparent transitions between leaf nodes, although such transitions are not allowed in its underlying uncensored random walk. This will mislead RW. This effect is less severe in the grid graph and the ring graph.

For each estimator, we perform 5-fold cross validation (CV) for finding the best smoothing parameters C_β, C_{RW}, C_{FE} on the grid $10^{-2}, 10^{-1.5}, \ldots, 10^{1}$, respectively, with which we compute each estimator. Then, we evaluate the three estimators using the Frobenius norm between $\widehat{\mathbf{P}}$ and the true transition matrix $\mathbf{P}^*$:

$$\text{error}(\widehat{\mathbf{P}}) = \sqrt{\sum_{i,j} (\widehat{P}_{ij} - P_{ij}^*)^2}.$$

Note the error must approach 0 as m increases for consistent estimators. We repeat the same experiment 20 times where each time we draw a new set of censored lists.



(a) (b) (c)

Figure 3.3: Toy experiment results where the error is measured with the Frobenius norm.

Figure 3.3 shows how $\text{error}(\widehat{\mathbf{P}})$ changes as the number of censored lists m increases. The error bars are 95% confidence bounds. We make three observations:

1. *INVITE tends towards 0 error.* This is expected given the consistency of INVITE in Theorem 3.7.

2. *RW is biased.* In all three plots, RW tends towards some positive number, unlike INVITE and FE. This is because RW has the wrong model on the censored lists.

3. *INVITE outperforms FE.* On the ring and grid graphs INVITE dominates FE for every training set size. On the star graph FE is better than INVITE with a small m, but INVITE eventually achieves lower error. This reflects the fact that, although FE is unbiased, it discards most of the censored lists and therefore has higher variance compared to INVITE.

3.6 Application: Human Memory Search

A key goal for cognitive science has been to understand the mental structures and processes that underlie human semantic memory search. Verbal fluency has provided the central paradigm for this work: given a category label as a cue (e.g. animals, vehicles, etc.) participants must generate as many example words as possible in 60 seconds without repetition. The task is useful because, while exceedingly easy to administer, it yields rich information about human semantic memory. Participants do not generate responses in random order but produce "bursts" of related items, beginning with the highly frequent and prototypical, then moving to subclusters of related items. This ordinal structure sheds light on associative structures in memory: retrieval of a given item promotes retrieval of a related item, and so on, so that the temporal proximity of items in generated lists reflects the degree to which the two items are related in memory [30, 12, 26].

The task also places demands on other important cognitive contributors to memory search: for instance, participants must retain a mental trace of previously-generated items and use it to refrain from repetition, so that the task draws upon working memory and cognitive control in addition to semantic processes. For these reasons the task is a central tool in all commonly-used metrics for diagnosing cognitive dysfunction (see e.g. [14]). Category and letter fluency in particular are sensitive to a broad range of cognitive disorders resulting from brain damage [50]. For instance, patients with prefrontal injuries are prone to inappropriately repeating the same item several times (*perseverative* errors) [5], whereas patients with pathology in the anterior temporal cortex are more likely to generate incorrect responses (*semantic* errors) and produce many fewer items overall [32, 50]. For all these reasons the task has been very widely employed both in basic science and applied health research.

Nevertheless, the representations and processes that support category fluency remain poorly understood. Beyond the general observation that responses tend to be clustered by semantic relatedness, it is not clear what ordinal structure in produced responses reveals about the structure of human semantic memory, in either healthy or disordered populations. In the past few years researchers in cognitive science have begun to fill this gap by considering how search models from other domains of science might explain patterns of responses observed in fluency tasks [24, 25, 31]. We review related works in Section 4.1.

In this section, we propose to use INVITE as a model of human memory search and perform our parameter estimation procedure since its repeat-censoring mechanism is known to match the human behaviors of generating runs of semantically related items [1] in verbal fluency task. Our proposal is significantly different from prior works since we learn the transition matrix directly from the verbal fluency responses rather than constructing it from a separate source such as a standard text corpus [25], or a set of responses in a completely different task [1].

		Animal	Food
n		274	452
m		4710	4622
Length	Min.	2	1
	Max.	36	47
	Mean	18.72	20.73
	Median	19	21

Table 3.1: Statistics of the verbal fluency data.

	Model	Test set mean neg. loglik.
Animal	INVITE	**60.18 (±1.75)**
	RW	69.16 (±2.00)
	FE	72.12 (±2.17)
Food	INVITE	**83.62 (±2.32)**
	RW	94.54 (±2.75)
	FE	100.27 (±2.96)

Table 3.2: Verbal fluency test set log likelihood.

Data The data used to assess human memory search consists of two verbal fluency datasets from the Wisconsin Longitudinal Survey (WLS). The WLS is a longitudinal assessment of many sociodemographic and health factors that has been administered to a large cohort of Wisconsin residents every five years since the 1950s. Verbal fluency for two semantic categories, animals and foods, was administered in the last two testing rounds (2005 and 2010), yielding a total of 4714 lists for animals and 4624 lists for foods collected from a total of 5674 participants ranging in age from their early-60's to mid-70's. The raw lists included in the WLS were preprocessed by expanding abbreviations ("lab" → "labrador"), removing inflections ("cats" → "cat"), correcting spelling errors, and removing response errors like unintelligible items. Though instructed to not repeat, some human participants did occasionally produce repeated words. We removed the repetitions from the data, which consist of 4% of the word token responses. Finally, the data exhibits a Zipfian behavior with many idiosyncratic, low count words. We removed words appearing in less than 10 lists. In total, the process resulted in removing 5% of the total number of word token responses. The statistics of the data after preprocessing is summarized in Table 3.1.

Procedure We randomly subsample 10% of the lists as the test set, and use the rest as the training set. We perform 5-fold CV on the training set for each estimator to find the best smoothing parameter $C_{\beta}, C_{RW}, C_{FE} \in \{10^1, 10^{.5}, 10^0, 10^{-.5}, 10^{-1}, 10^{-1.5}, 10^{-2}\}$ respectively, where the validation measure is the prefix log likelihood for INVITE and the standard random walk likelihood for RW. For the validation measure of FE we use the INVITE prefix log likelihood since FE is equivalent to the length two prefix likelihood of INVITE. Then, we train the final estimator on the whole training set using the fitted regularization parameter.

Result The experiment result is summarized in Table 3.2. For each estimator, we measure the average per-list negative prefix log likelihood on the test set for INVITE and FE, and the standard random walk per-list negative log likelihood for RW. The number in the parenthesis is the 95% confidence interval. Boldfaced numbers mean that the corresponding estimator is the best and the difference from the others is statistically significant under a two-tailed paired t-test at 95% significance level. In both animal and food verbal fluency tasks, the result indicates that human-generated fluency lists are better explained by INVITE than by either RW or FE. Furthermore, RW outperforms FE. We believe that FE performs poorly despite being consistent because the number of lists is too small (compared to the number of states) for FE to reach a good estimate.

3.7 Related Work

Though behavior in semantic fluency tasks has been studied for many years, few computationally explicit models of the task have been advanced. Influential models in the psychological literature, such as the widely-known "clustering and switching" model of Troyer et al. [54], have been articulated only verbally. Efforts to estimate the structure of semantic memory from fluency lists have mainly focused on decomposing the structure apparent in distance matrices that reflect the mean inter-item ordinal distances across many fluency lists [12]—but without an account of the processes that generate list structure it is not clear how the results of such studies are best interpreted. In the recent few years researchers in cognitive science have begun to focus on explicit model of the processes by which fluency lists are generated. In these works, the structure of semantic memory is first modeled either as a graph or as a continuous multidimensional space estimated from word co-occurrence statistics in large corpora of natural language. Researchers then assess whether structure in fluency data can be understood as resulting from a particular search process operating over the specified semantic structure. Models explored in this vein include simple random walk over a semantic network, with repeated nodes omitted from the sequence produced [24], the PageRank algorithm employed for network search by Google [25], and foraging algorithms designed to explain the behavior of animals searching for food [31]. Each example reports aspects of human behavior that are well-explained by the respective search process, given accompanying assumptions about the nature of the underlying semantic structure. However, these works do not learn their model directly from the fluency lists, which is the key difference from our study.

Broder's algorithm **Generate** [10] for generating random spanning tree is similar to INVITE's generative process. Given an undirected graph, the algorithm runs a random walk and outputs each transition to an unvisited node. Upon transiting to an already visited node, however, it does not output the transition. The random walk stops after visiting every node in the graph. In the end, we observe an ordered list of transitions. For example, in Figure 3.1(a) if the random walk trajectory is (2,1,2,1,3,1,4), then the output is (2→1, 1→3, 1→4). Note that if we take the starting node of the first transition and the arriving nodes of each transition, then the output list reduces to a censored list generated from INVITE with the same underlying random walk. Despite the similarity, to the best of our knowledge, the censored list derived from the output of the algorithm Generate has not been studied, and there has been no parameter estimation task discussed in prior works.

Self-avoiding random walk, or non-self-intersecting random walk, performs random walk while avoiding already visited node [20]. For example, in Figure 3.1(a), if a self-avoiding random walk starts from state 2 then visits 1, then it can only visit states 3 or 4 since 2 is already visited. Insofar as not visiting the same node twice, self-avoiding walk is similar to INVITE. However, a key difference is that self-avoiding walk cannot produce a transition $i \to j$ if $P_{ij} = 0$. In contrast, INVITE can appear to have such "transitions" in the censored list. Such behavior is a core property that allows INVITE to *switch* clusters in modeling human memory search.

INVITE resembles cascade models in many aspects [40, 23]. In a cascade model, the information or disease spreads out from a seed node to the whole graph by infections that occur from an infected node to its neighbors. [23] formulates a graph learning problem where an observation is a list, or so-called trace, that contains infected nodes along with their infection time. Although not discussed in the present paper, it is trivial for INVITE to produce time stamps for each item in its censored list, too. However, there is a fundamental difference in how the infection occurs. A cascade model typically allows multiple infected nodes to infect their neighbors in parallel, so that infection can happen simultaneously in many parts of the graph. On the other hand, INVITE contains a *single surfer* that is responsible for all the infection via a random walk. Therefore, infection in INVITE is necessarily sequential. This results in INVITE exhibiting clustering behaviors in the censored lists, which is well-known in human memory search tasks [54].

4 TOP ARM IDENTIFICATION IN MULTI-ARMED BANDIT WITH BATCH ARM PULLS

We consider the top-k pure-exploration problem for stochastic multi-armed bandits (MAB). Formally, we are given n arms that produce a stochastic reward when pulled. The reward of arm i is an i.i.d. sample from a distribution ν_i whose support is in $[0,1]$. The bounded-support assumption can be generalized to the σ-sub-Gaussian assumption. Denote by $\mu_i = \mathbb{E}_{X\sim\nu_i}X$ the expected reward of the arm i. We assume a unique top-k set; i.e., $\mu_1 \geq \mu_2 \geq ... \geq \mu_k > \mu_{k+1} \geq ... \geq \mu_n$.

In this chapter, we introduce a new setting in which arms must be pulled in batches of size b. We also consider the additional constraint that any one arm can be pulled at most $r \leq b$ times in a batch (if $r = b$ then there is no constraint). Said another way, at each round, a multiset I of size at most b must be chosen where the multiplicity of an arm is at most r. Consequently, the action space $\mathcal{A}$ is defined as follows:

$$\mathcal{A} := \left\{ I \in \mathbb{Z}_{0+}^n \,\middle|\, \sum_{i=1}^n I_i \leq b, I_j \leq r, \forall j \right\}. \tag{4.1}$$

We call this the (b,r)-*batch* MAB setting. Note that this encompasses the standard MAB setting, which corresponds to $b = 1, r = 1$. The general (b,r)-batch setting is fundamentally different, since sampling decisions must be made one batch at a time, rather than one sample at a time. This loss in flexibility could hinder performance, but our main theoretical results show that in most cases there is no significant increase in sample complexity.

There are many real-world applications where the batch constraint arises. For example, suppose the arms represent Twitter users and the goal is to find users who tweet the most about a topic of interest. Using Twitter's free API, one can follow up to $b = 5000$ users at a time, which amounts to a batch of that size. Also, a user is either observed or not, so $r = 1$ in this application. The batch is the fundamental sampling unit in such applications, and thus this paper is interested in methods that aim to reduce the *batch complexity*: the total number of *batches*, or rounds, to correctly identify the top-k arms. As another example, we consider the problem of using microwell array experiments to identify the top k genes involved in virus replication processes. Here the arms are genes (specifically single knockdown cell strains) and a batch consists of $b = 384$ microwells per array. In this case, we may repeat the same gene (cell strain) in every microwell, so $r = 384$.

Returning to the discussion of theory and algorithms, we consider two settings: fixed confidence and fixed budget. In the fixed confidence setting, given a target failure rate δ the goal is to identify the top-k arms with probability at least $1 - \delta$ using the fewest number of batches as possible. In the fixed budget setting, given a budget number of batches B, the goal is to identify the top-k arms with as high probability as possible. For the fixed confidence (budget) setting, we propose the BatchRacing (BatchSAR) algorithm and prove its theoretical guarantee in Section 4.2 (Section 4.3). Our analysis shows that batch MABs have the same sample complexity as unconstrained MABs as long as r is not too small.

An alternative to our batch MAB algorithms is to use standard MAB algorithms as follows. One could use each batch to obtain a sample from an particular arm of interest, and effectively ignore the rest of the data from each batch. This would allow one to use standard MAB algorithms. For example, in the Twitter application we could follow just one user at a time, but we can follow 5000 for the same cost (free). Our results show that in most cases this (obviously naive) reduction to the standard MAB would have a sample complexity b times greater than that of our batch MAB algorithms.

In section 4.4, we validate the analysis of the proposed algorithms with a toy experiment, and evaluate the algorithms on real-world data from the two applications introduced above.

4.1 Related Works

The pure exploration problem in MAB has a long history back to the '50s with the work of [6]. The top-k exploration, or subset selection, in stochastic MAB has received much attention recently. In the fixed confidence setting including the Probably Approximately Correct (PAC) setup, the Racing algorithm for the top-1 exploration problem was proposed by Maron and Moore [44] and by Even-Dar et al. [19] independently. The Racing algorithm works in a uniform sampling manner on surviving arms while deactivating infeasible arms based on confidence intervals. Racing was generalized to top-k identification by Heidrich-Meisner and Igel [29], and its sample complexity was analyzed in [39]. Median Elimination (ME) [19, 37] runs in stages where each stage eliminates the bottom half of the surviving arms. Under the PAC setup, the sample complexity of ME matches the lower bound. Going beyond elimination-based algorithms with uniform sampling, LUCB proposed by Kalyanakrishnan et al. [38] adaptively chooses which arm to pull based on confidence intervals without removing arms. Kaufmann et al. improve both Racing and LUCB with confidence intervals based on Chernoff information, though restricted to exponentially distributed rewards [39]. In the fixed budget setting, Successive Accepts and Rejects (SAR) proposed by Bubeck et al. [11] runs in $n - 1$ phases with predetermined rounds of uniform sampling where after each phase the empirically best arm is accepted or the empirically worst arm is rejected. LUCB-E by Kaufmann et al. [39] runs in a more adaptive way with confidence intervals without eliminations, although the problem hardness parameter must be known ahead to run the algorithm with guarantee.

Our proposed (b, r)-batch setting overlaps with various existing settings, but is, to the best of our knowledge, never subsumed by them. One popular setting, though studied under the regret setting, is semi-bandit feedback [4] where an action $I \in \mathcal{A} \subseteq \{0, 1\}^n$ indicates coordinate-wise whether or not ($I_j = 1$ or 0) one pulls an arm. (b,r=1)-batch is a special case of semi-bandit feedback where the action space $\mathcal{A}$ is defined to be all b-sized subsets of arms. This is called multiple plays in [3]. A variant of the multiple plays was studied in [43] where the player pulls b times in a round, but is allowed to pull an arm more than once, up to b times. However, the author assumes that pulling the same arm more than once in a round produces the same reward, whereas in our (b,r=b)-batch setting repeated pulls produce independent rewards.

In delayed feedback setting [36], the reward at time t is revealed after τ_t time steps. One can relate this setting to our (b, r)-batch setting by assuming that rewards are revealed in blocks after every b rounds: $\tau_t = b - 1 - (t - 1 \mod b)$. If b is known to the player, this construction is exactly the $(b, r = b)$-batch setting. Nevertheless, delayed feedback has only been considered in regret setting.

4.2 The Fixed Confidence Setting

Algorithm 2 BatchRacing

```
**Input**: $n$ arms, $\delta \in (0,1)$, $k$: number of top arms, $b$: batch size, $r$: maximum repeats
**Output**: $k$ arms.
$t \leftarrow 1$, $S_1 \leftarrow [n]$, $R_1 \leftarrow \emptyset$, $A_1 \leftarrow \emptyset$, $T_i(0) \leftarrow 0, \forall i$
**while** $S_t \neq \emptyset$ **do**
  $I \leftarrow \mathbf{0} \in \mathbb{R}^n$
  **for** $\tau = 1 \dots \min\{b, |S_t|r\}$ **do**
    $j \leftarrow \arg\min_{\ell \in S_s : I_\ell \leq r} T_\ell(t-1) + I_\ell$ (break ties arbitrarily)
    $I_j \leftarrow I_j + 1$
  **end for**
  Pull by $I$ (pull arm $j$ $I_j$ times, $\forall j$).
  $T_i(t) \leftarrow T_i(t-1) + I_i, \forall i$
  $k' \leftarrow k - |A_t|$
  $A_{t+1} \leftarrow A_t \bigcup \{i \in S_t \mid L_i(t,\delta) > \max^{(k'+1)}_{j \in S_t} U_j(t,\delta)\}$
  $R_{t+1} \leftarrow R_t \bigcup \{i \in S_t \mid U_i(t,\delta) < \max^{(k')}_{j \in S_t} L_j(t,\delta)\}$
  $S_{t+1} \leftarrow S_t \backslash (R_{t+1} \bigcup A_{t+1})$
  $t \leftarrow t + 1$
**end while**
Output $A_t$.
```

In the fixed confidence setting, given a target failure rate δ, one tries to find the top-k arms with probability at least $1 - \delta$ in the fewest number of batches as possible. An algorithm must output the correct top-k arms w.p. $\geq 1 - \delta$, and its performance is measured in the batch complexity.

Denote by $\max^{(k)} S$ the k-th largest member of S. Let $X_{i,j}$ be the j-th reward of arm i. Define the empirical mean of arm i up to τ samples as $\hat{\mu}_{i,\tau} = \frac{1}{\tau}\sum_{j=1}^{\tau} X_{i,j}$. The key success of an algorithm in the fixed confidence setting often relies on the confidence bound on the true mean μ_i; the tighter the bounds are, the less arm pulls we spend. We adopt a confidence bound proposed in [34] that resembles the asymptotic deviation in the law of the iterated logarithm (which we present later in Lemma 4.1). Define a deviation function

$$D(\tau, \omega) = \sqrt{\frac{4\log(\log_2(2\tau)/\omega)}{\tau}}.$$

Let $T_i(t)$ be the number of arm pulls of arm i at round t. We define the lower confidence bound (LCB) $L_i(t,\delta)$ and the upper confidence bound (UCB) $U_i(t,\delta)$ of arm i at round t as follows:

$$L_i(t,\delta) = \hat{\mu}_{i,T_i(t)} - D\left(T_i(t), \sqrt{\delta/(6n)}\right) \tag{4.2}$$

$$U_i(t,\delta) = \hat{\mu}_{i,T_i(t)} + D\left(T_i(t), \sqrt{\delta/(6n)}\right). \tag{4.3}$$

Now that the environment allows the (b, r)-batch arm pull, can we propose an algorithm whose batch complexity achieves b factor reduction from the sample complexity of the state-of-the-art algorithms? Inspired by the Racing algorithm [44, 29], we propose a BatchRacing algorithm for the (b, r)-batch setting, see Algorithm 2. The algorithm maintains a set of active or surviving arms that is initialized as $S_1 = [n]$ before the first round. At round t, the algorithm chooses b arm pulls that keeps the pull count of each arm in the surviving set S_t as uniform as possible. Then, the algorithm checks if there is any arm that is certainly top-k or certainly not top-k using the LCB and UCB as follows. Let A_t (R_t) be the set of accepted (rejected) arms at round t and $A_1 = R_1 = \emptyset$. Let $k' = k - |A_t|$, the remaining number of top arms to identify. Any arm i whose LCB is greater than the UCB of $|S_t| - k'$ arms is moved to the accept set: $A_{t+1} \leftarrow A_t \cup \{i\}$. Symmetrically, any arm i whose UCB is smaller than the LCB of k' arms is moved to the reject set $R_{t+1} \leftarrow R_t \cup \{i\}$. Those accepted or rejected arms are removed from the surviving set S_t. The process is repeated until it accepts k arms ($|A_t| = k$), and A_t is the output.

We define the gap Δ_i of arm i and denote by $\sigma(j)$ the arm with j-th smallest gap such that

$$\Delta_i = \begin{cases} \mu_i - \mu_{k+1} & \text{if } i \le k \\ \mu_k - \mu_i & \text{if } i > k \end{cases} \quad \text{and} \quad \Delta_{\sigma(1)} = \Delta_{\sigma(2)} \le \Delta_{\sigma(3)} \le \ldots \le \Delta_{\sigma(n)}, \tag{4.4}$$

where $\Delta_{\sigma(1)} = \Delta_{\sigma(2)}$ by definition. Let $\omega = \sqrt{\delta/(6n)}$. Define

$$\overline{T}_i = 1 + \left\lfloor 64\Delta_i^{-2} \log((2/\omega)\log_2(192\Delta_i^{-2}/\omega)) \right\rfloor.$$

Before presenting the batch complexity of BatchRacing, we discuss the key ingredients of the analysis. Lemma 4.1 proposed in [34] is a non-asymptotic form of the law of the iterated logarithm for σ-sub-Gaussian random variables.

Lemma 4.1. *(non-asymptotic law of the iterated logarithm) [34]*[1] *Let* $X_1, X_2, \ldots$ *be i.i.d. zero-mean sub-Gaussian random variables with scale* $\sigma > 0$*; i.e.,* $\mathbb{E}e^{\lambda X_i} \le e^{\frac{\lambda^2\sigma^2}{2}}$.

[1] A tighter version can be found in [34].

Let $\omega \in (0, \sqrt{1/6})$. *Then,*

$$\mathbb{P}\left(\forall \tau \geq 1, \left|\sum_{s=1}^{\tau} X_s\right| \leq 4\sigma\sqrt{\tau \log(\log_2(2\tau)/\omega)}\right) \geq 1 - 6\omega^2. \tag{4.5}$$

Proof. Let $S_\tau = \sum_{s=1}^{\tau} X_s$. Note that the upper and lower bound around S_τ is symmetric. Thus, it suffices to show that the upper bound on S_τ fails with probability at most $3\omega^2$. Show that $\{e^{S_\tau}\}$ is sub-Martingale, apply Doob's maximal inequality to it, then use the definition of the sub-Gaussian and Markov's inequality to have $\mathbb{P}\left(\max_{\tau=1,\ldots,m} S_\tau \geq x\right) \leq e^{-\frac{x^2}{2m\sigma^2}}$. Define

$$\mathcal{E}_1 = \bigcap_{k \geq 0} \left\{S_{2^k} \leq 2\sigma\sqrt{2^k \log(\log_2(2 \cdot 2^k)/\omega)}\right\}$$
$$\mathcal{E}_2 = \bigcap_{k \geq 1} \bigcap_{\tau=2^k+1}^{2^{k+1}} \left\{S_\tau - S_{2^k} \leq 2\sigma\sqrt{2^k \log(\log_2(2 \cdot 2^k)/\omega)}\right\}.$$

One can verify that $\mathcal{E}_1 \cap \mathcal{E}_2$ implies the upper bound stated in the Lemma since *(i)* for $\tau = 1$ and 2, $\mathcal{E}_1$ implies the bound and *(ii)* for all $\tau \geq 3$ such that $2^k < \tau \leq 2^{k+1}$ for some k,

$$S_\tau \leq S_{2^k} + 2\sigma\sqrt{2^k \log(\log_2(2 \cdot 2^k)/\omega)} \leq 4\sigma\sqrt{2^k \log(\log_2(2 \cdot 2^k)/\omega)} < 4\sigma\sqrt{\tau \log(\log_2(2\tau)/\omega)}.$$

The failure probability of the bound is at most $\mathbb{P}(\overline{\mathcal{E}_1}) + \mathbb{P}(\overline{\mathcal{E}_2})$, which we bound with $3\omega^2$ as follows:

$$\mathbb{P}(\overline{\mathcal{E}_1}) \leq \sum_{k \geq 0} e^{-2\log(\log_2(2^{k+1})/\omega)} \leq \frac{\pi^2}{6}\omega^2 < 2\omega^2$$
$$\mathbb{P}(\overline{\mathcal{E}_2}) \leq \sum_{k \geq 1} \mathbb{P}\left(\bigcup_{\tau=2^k+1}^{2^{k+1}} \left\{S_\tau - S_{2^k} \geq 2\sigma\sqrt{2^k \log(\log_2(2 \cdot 2^k)/\omega)}\right\}\right)$$
$$= \sum_{k \geq 1} \mathbb{P}\left(\bigcup_{\tau=2^k+1}^{2^{k+1}} \left\{\sum_{i=2^k+1}^{\tau} X_i \geq 2\sigma\sqrt{2^k \log(\log_2(2 \cdot 2^k)/\omega)}\right\}\right)$$
$$= \sum_{k \geq 1} \mathbb{P}\left(\bigcup_{j=1}^{2^k} \left\{S_j \geq 2\sigma\sqrt{2^k \log(\log_2(2 \cdot 2^k)/\omega)}\right\}\right)$$

$$\leq \sum_{k\geq 1} \left(\frac{\omega}{k+1}\right)^2 = (\pi^2/6 - 1)\omega^2 < \omega^2.$$

□

Lemma 4.2 states that the true mean of arm i can be bounded by the empirical top-i arm.

Lemma 4.2. *Denote by $\widehat{i}$ the index of the arm whose empirical mean is i-th largest: i.e., $\hat{\mu}_{\widehat{1}} \geq \ldots \geq \hat{\mu}_{\widehat{n}}$. Assume that the empirical means of the arms are controlled by ϵ: i.e., $\forall i, |\hat{\mu}_i - \mu_i| \leq \epsilon$. Then,*

$$\forall i, \mu_i - \epsilon \leq \hat{\mu}_{\widehat{i}} \leq \mu_i + \epsilon.$$

Proof. To show $\mu_i - \epsilon \leq \hat{\mu}_{\widehat{i}}$, consider two cases: *(i)* empirically i-th largest arm is true top-i, i.e. $\widehat{i} \leq i$, and *(ii)* is not true top-i, i.e. $\widehat{i} > i$. Case *(i)* implies that $\mu_{\widehat{i}} \geq \mu_i$. Then,

$$\hat{\mu}_{\widehat{i}} \geq \mu_{\widehat{i}} - \epsilon \geq \mu_i - \epsilon.$$

Case *(ii)* implies that there exists an arm ℓ that is true top-i, but not empirical top-i, i.e. $\mu_\ell \geq \mu_i$ and $\hat{\mu}_\ell \leq \hat{\mu}_{\widehat{i}}$, since otherwise $\widehat{i}$ (the empirical top-i) must have been squeezed out of the empirical top-i, which is a contradiction. Then,

$$\hat{\mu}_{\widehat{i}} \geq \hat{\mu}_\ell \geq \mu_\ell - \epsilon \geq \mu_i - \epsilon.$$

One can show $\hat{\mu}_{\widehat{i}} \leq \mu_i + \epsilon$ by symmetry. □

Let $\mathcal{E}_i(\delta) = \{\forall t \geq 1, L_i(t, \delta) \leq \mu_i \leq U_i(t, \delta)\}$ be the probability that the LCB and UCB of arm i defined in (4.2) and (4.3) traps the true mean μ_i for all $t \geq 1$. Let $\omega = \sqrt{\delta/(6n)}$ and define

$$\overline{T}_i = 1 + \left\lfloor 64\Delta_i^{-2} \log((2/\omega)\log_2(192\Delta_i^{-2}/\omega)) \right\rfloor. \tag{4.6}$$

Lemma 4.3 shows a sufficient condition for an arm to be removed to either reject set R_t or accept set A_t.

Lemma 4.3. *Assume $\cap_{i=1}^n \mathcal{E}_i(\delta)$. In Algorithm 2, let $T'(t) = \min_{i \in S_t} T_i(t)$ and $k' = k - |A_t|$. Then,*

$$\forall t, \forall i > k, \left(T'(t) \geq \overline{T}_i \implies U_i(t,\delta) < \overset{(k')}{\max_{j \in S_t}} L_j(t,\delta)\right) \tag{4.7}$$

$$\forall t, \forall i \leq k, \left(T'(t) \geq \overline{T}_i \implies L_i(t,\delta) > \overset{(k'+1)}{\max_{j \in S_t}} U_j(t,\delta)\right). \tag{4.8}$$

Proof. Hereafter, we use $L_i(t)$ and $U_i(t)$ as shorthands for $L_i(t,\delta)$ and $U_i(t,\delta)$ respectively. It suffices to show for the case where A_t and R_t are empty since otherwise the problem is equivalent to removing rejected or accepted arms from consideration and starting a new problem with $n \leftarrow (n - |A_t| - |R_t|)$ and $k \leftarrow (k - |A_t|)$ while maintaining the samples collected so far.

To prove (4.7), let $i > k$ and $\widehat{k}$ be the arm with k-th largest empirical mean at t-th round. We prove the contrapositive. Assume the RHS is false: $U_i(t) \geq \max_{j \in S_t}^{(k-|A_t|)} L_j(t)$. Note that using $D(T_i(t),\omega) \leq D(T'(t),\omega))$,

$$U_i(t) \leq \hat{\mu}_{i,T_i(t)} + D(T'(t),\omega) \leq \mu_i + 2D(T'(t),\omega),$$

and

$$U_i(t) \geq \overset{(k-|A_t|)}{\max_{j \in S_t}} L_j(t) = L_{\widehat{k}}(t) \geq \hat{\mu}_{\widehat{k},T_{\widehat{k}}(t)} - D(T'(t),\omega),$$

which implies $\mu_i + 2D(T'(t),\omega) \geq \hat{\mu}_{\widehat{k},T_{\widehat{k}}(t)} - D(T'(t),\omega)$. Using $\hat{\mu}_{\widehat{k},T_{\widehat{k}}(t)} \geq \mu_k - D(T'(t),\omega)$ that is due to Lemma 4.2,

$$\begin{aligned} \mu_i + 2D(T'(t),\omega) \geq \mu_k - 2D(T'(t),\omega) \\ \Delta_i \leq 4D(T'(t),\omega) = 4\sqrt{(4/T'(t))\log(\log_2(2T'(t))/\omega)} \\ T'(t) \leq 64\Delta_i^{-2}\log(\log_2(2T'(t))/\omega). \end{aligned}$$

Invert this using

$$\tau \leq c\log\left(\frac{\log_2 2\tau}{\omega}\right) \implies \tau \leq c\log\left(\frac{2}{\omega}\log_2\left(\frac{3c}{\omega}\right)\right) \tag{4.9}$$

with $c = 64\Delta_i^{-2}$ to have $T'(t) > 64\Delta_i^{-2}\log(\log_2(2T'(t))/\omega)$. We provide the proof of (4.9) in the appendix. Then, $T'(t) \geq 1 + \lfloor 64\Delta_i^{-2} \log(\log_2(2T'(t))/\omega) \rfloor = \overline{T}_i$. This completes the proof of (4.7). By symmetry, one can show (4.8) as well. □

Note that even if $r > \lfloor b/2 \rfloor$, BatchRacing pulls the same arm at most $\lfloor b/2 \rfloor$ times in a batch. Thus, our analysis relies on the effective repeated pull limit $r' = \min\{r, \lfloor b/2 \rfloor\}$ instead of r.

Theorem 4.4. *If $b \geq 2$, with probability at least $1-\delta$, Algorithm 2 outputs the top-k arms $\{1, \ldots, k\}$ after at most*

$$\frac{1}{r'}\overline{T}_{\sigma(1)} + \frac{1}{b}\left(\sum_{i=\lfloor b/r' \rfloor+1}^{n} \overline{T}_{\sigma(i)}\right) + \log n + \frac{n}{b} + \frac{1}{r'} + 2 \tag{4.10}$$

$$= O\left(\frac{1}{r'}\Delta_{\sigma(1)}^{-2} \log\left(\frac{n\log(\Delta_{\sigma(1)}^{-2})}{\delta}\right) + \frac{1}{b}\left(\sum_{i=\lfloor b/r' \rfloor+1}^{n} \Delta_{\sigma(i)}^{-2} \log\left(\frac{n\log(\Delta_{\sigma(i)}^{-2})}{\delta}\right)\right) + \log n\right)$$

batches. In the case of $b = r = 1$, the algorithm does so after at most

$$\sum_{i=1}^{n} \overline{T}_{\sigma(i)} = O\left(\sum_{i=1}^{n} \Delta_{\sigma(i)}^{-2} \log\left(\frac{n\log(\Delta_{\sigma(i)}^{-2})}{\delta}\right)\right)$$

batches.

Proof. The key insight in proving the batch complexity of BatchRacing is that once there are only $\lfloor b/r \rfloor$ or less surviving arms, there is no choice but to pull each arm r times at every batch, possibly wasting some arm pulls. Before then, the algorithm fully exploits pulling b times in a round and enjoys a full factor of b reduction. Note a random variable X such that $X \in [0,1]$ w.p. 1 is sub-Gaussian with scale $\sigma = 1/2$. By Lemma 4.1, $\mathbb{P}(\cap_{i=1}^n \mathcal{E}_i(\omega)) \geq 1-\delta$. Thus, it is safe to assume that $\cap_{i=1}^n \mathcal{E}_i(\omega)$; the true mean of each arm is trapped by the LCB and UCB.

Correctness We show that if an arm i is accepted (rejected) then i is (not) top-k. It suffices to prove for round t such that A_t and R_t are empty for the same reason stated in the proof of Lemma 4.3. An arm i being accepted implies that there are $(n-k)$ arms whose true means are smaller than μ_i since the LCB and UCB of each arm trap its true mean. Thus, i is a top-k arm. Similarly, if an arm j is rejected then j is not top-k.

Batch Complexity Let $T'(t) = \min_{i \in S_t} T_i(t)$. When $T'(t)$ goes above $\overline{T}_i$ for some arm i, then we know by Lemma 4.3 that arm i is either accepted or rejected and is removed from S_t. We consider the worst case where an arm i is removed only after satisfying $T'(t) \geq \overline{T}_i$, and the order of the removal is arm $\sigma(n), \sigma(n-1), \ldots, \sigma(3)$, and $\sigma(2)$ or $\sigma(1)$. Any early removal of an arm and switching the order of the removal necessarily make the rest of the problem easier: i.e., required number of arm pulls are reduced.

By Lemma 4.3, when $T'(t)$ reaches $\overline{T}_i$, arm $\sigma(i)$ is removed. If $b = r = 1$, the batch complexity can be trivially obtained by summing up $\overline{T}_i$ for $i \in [n]$. Thus, we consider $b \geq 2$ for the rest of the proof. Let t_i be the round that $T'(t_i)$ just passed $\overline{T}_{\sigma(i)}$: i.e., $T'(t_i - 1) < \overline{T}_i$ and $T'(t_i) \geq \overline{T}_{\sigma(i)}$. Note that $T'(t_i)$ can be greater than $\overline{T}_{\sigma(i)}$ due to the "batch effect" where the increment of the number of samples of an arm can be greater than 1. Such discrepancy can be large when b is much larger than $|S_{t_i}|$. It is easy to see that $T'(t_i) \leq \overline{T}_i + \lceil b/|S_{t_i}| \rceil - 1$. Then,

$$T_{\sigma(i)}(t_i) \leq T'(t_i) + 1 \leq \overline{T}_{\sigma(i)} + \left\lceil \frac{b}{|S_{t_i}|} \right\rceil = \overline{T}_{\sigma(i)} + \left\lceil \frac{b}{(n-i+1)} \right\rceil . \tag{4.11}$$

Consider the round $t' \equiv t_{(\lfloor b/r' \rfloor + 1)}$ where arm $\sigma(\lfloor b/r' \rfloor + 1)$ is just removed. The total number of samples used up to t' is

$$bt' = \sum_{i=1}^{n} T_{\sigma(i)}(t') \leq \left\lfloor \frac{b}{r'} \right\rfloor (T'(t') + 1) + \sum_{i=\lfloor b/r' \rfloor + 1}^{n} \overline{T}_{\sigma(i)} + \left\lceil \frac{b}{n-i+1} \right\rceil .$$

At round $t' + 1$, there are only $\lfloor b/r' \rfloor$ surviving arms. The algorithm then has no choice but to pull every surviving arm r' times in each round until we remove the hardest two arms $\sigma(1)$ and $\sigma(2)$ during which arm $\sigma(3), \ldots, \sigma(\lceil b/r' \rceil)$ are removed as a byproduct. The extra number of batches that must be performed until termination is then $\lceil \frac{1}{r'}(\overline{T}_{\sigma(2)} - T'(t')) \rceil$. Therefore, the total number of batches is

$$\begin{aligned} t' + \left\lceil \frac{1}{r'}(\overline{T}_{\sigma(2)} - T'(t')) \right\rceil &\leq \frac{1}{r'}\overline{T}_{\sigma(2)} + \frac{1}{b} \sum_{i=\lfloor b/r' \rfloor + 1}^{n} \left(\overline{T}_{\sigma(i)} + \left\lceil \frac{b}{n-i+1} \right\rceil \right) + \frac{1}{r'} + 1 \\ &\leq \frac{1}{r'}\overline{T}_{\sigma(2)} + \frac{1}{b} \left(\sum_{i=\lfloor b/r' \rfloor + 1}^{n} \overline{T}_{\sigma(i)} \right) + 1 + \log n + \frac{n}{b} + \frac{1}{r'} + 1 \end{aligned}$$

$$= O\left(\frac{1}{r'}\overline{T}_{\sigma(2)} + \frac{1}{b}\sum_{i=\lfloor b/r' \rfloor+1}^{n} \overline{T}_{\sigma(i)} + \log n\right).$$

□

Note that in the case of b=r=1, BatchRacing is exactly the Racing algorithm, and the batch complexity is equivalent to the sample complexity. The sample complexity of Racing stated in Theorem 4.4 is the best known in sofaras the exact top-k identification problem, although it does not match the best known lower bound $\Omega(\sum_{i=1}^{n} \Delta_i^{-2})$ [39]. One can verify that when $r \geq \lfloor b/2 \rfloor$, the batch complexity (4.10) reduces to $\frac{1}{b}$ fraction of the sample complexity of Racing except the additive $\log n$ term. The $\log n$ term is the price to pay for performing batch arm pull, which limits adaptivity. Note that the batch complexity is at least n/b since each arm must be pulled at least once. Thus, unless b is large enough to satisfy $n/b \ll \log(n)$, the additive $\log(n)$ is negligible. For simplicity, we ignore the additive $\log n$ from the discussion. For $r < \lfloor b/2 \rfloor$, r plays an interesting role. The r factor reduction is applied to the largest term that involves $\Delta_{\sigma(1)}^{-2}$. The terms involving arm $\sigma(2), \ldots, \sigma(\lfloor b/r \rfloor)$ disappear. Then, the rest of the terms enjoy b factor reduction.

Note the contribution of r also depends on the gaps $\{\Delta_i\}$. If the smallest gap $\Delta_{\sigma(1)}$ is much smaller than the other gaps, the term $(1/r')\overline{T}_{\sigma(1)}$ becomes dominant, thus making r important. On the other hand, if the gaps are all equal, one can show that the overall b fold reduction is achieved regardless of r using the fact $(1/r')\Delta_{\sigma(1)}^{-2}\log(n\log(\Delta_{\sigma(1)}^{-2})/\delta) \approx (1/b)\sum_{i=1}^{\lfloor b/r' \rfloor} \Delta_{\sigma(i)}^{-2}\log(n\log(\Delta_{\sigma(i)}^{-2})/\delta)$. We empirically verify this case with toy experiments in Section 4.4.

4.3 The Fixed Budget Setting

We consider the problem of identifying the top-k arms with as high a probability as possible within a given number of batches B under the (b, r)-batch setting. An algorithm must terminate after spending no more than B batches and output k arms that are believed to be the top-k. The guarantee is typically made on the misidentification probability $\mathbb{P}(A^* \neq \{1, \ldots, k\})$ where A^* is the output of the algorithm. However, it is often convenient to look at its batch complexity that can be derived from the misidentification probability.

Algorithm 3 BatchSAR

```
Input: $n$ arms, $k$: the target number of top arms, $b$: batch size, $r$: maximum repeated arm pull, $B$: batch budget
Output: $k$ arms.
Let $\tilde{n} = \max\{\lceil b/r \rceil, 2\}$ and $c_1 = b + \tilde{n} \min\{r, \lceil b/2 \rceil\} + n$
Define $m_s = \begin{cases} \left\lceil \frac{bB - \left(\sum_{i=\tilde{n}+1}^{n} \lceil b/i \rceil\right) - c_1}{\tilde{n}/2 + \sum_{i=\tilde{n}+1}^{n} (1/i)} \frac{1}{n-s+1} \right\rceil & \text{for } s \le n - \tilde{n} \\ \left\lceil \frac{bB - \left(\sum_{i=\tilde{n}+1}^{n} \lceil b/i \rceil\right) - c_1}{\tilde{n}/2 + \sum_{i=\tilde{n}+1}^{n} (1/i)} \frac{1}{2} \right\rceil & \text{for } s = n - \tilde{n} + 1 \end{cases}$
$t \leftarrow 1$, $S_1 \leftarrow [n]$, $A_1 \leftarrow \emptyset$, $T_i(0) \leftarrow 0, \forall i \in [n]$
for $s = 1, \ldots, (n - \tilde{n} + 1)$ do
  while $\min_{i \in S_s} T_i(t-1) < m_s$ do
    $I \leftarrow \mathbf{0} \in \mathbb{R}^n$
    for $\tau = 1 \ldots b$ do
      $j \leftarrow \arg\min_{\ell \in S_s : I_\ell \le r} T_\ell(t-1) + I_\ell$ (break ties arbitrarily)
      $I_j \leftarrow I_j + 1$
    end for
    Pull by $I$ (pull arm $j$ $I_j$ times, $\forall j$).
    $T_j(t) \leftarrow T_j(t-1) + I_j, \forall j$
    $t \leftarrow t + 1$
  end while
  Let $t_s = t - 1$, the last round in stage $s$, and $k' = k - |A_s|$, the remaining number of top arms to identify.
  if $s \le n - \tilde{n}$ then
    Let $\hat{\mu}_i = \hat{\mu}_{i,T_i(t_s)}$ and $\rho(i)$ be the arm with $i$-th largest empirical mean in $S_s$ so that $\hat{\mu}_{\rho(1)} \ge \ldots \ge \hat{\mu}_{\rho(n-s+1)}$. Let $\widehat{\Delta}_{\rho(1)} = \hat{\mu}_{\rho(1)} - \hat{\mu}_{\rho(k'+1)}$ and $\widehat{\Delta}_{\rho(n-s+1)} = \hat{\mu}_{\rho(k')} - \hat{\mu}_{\rho(n-s+1)}$.
    $j_s \leftarrow \arg\max_{i \in \{\rho(1), \rho(n-s+1)\}} \widehat{\Delta}_i$ (break ties arbitrarily).
    Remove arm $j_s$: $S_{s+1} \leftarrow S_s \backslash \{j_s\}$.
    If $j_s = \rho(1)$, accept it: $A_{s+1} \leftarrow A_s \cup \{j_s\}$. Otherwise, $A_{s+1} \leftarrow A_s$.
    Early exit cases: If $|S_{s+1}| = k - |A_{s+1}|$, accept all the remaining surviving arms $A^* \leftarrow A_{s+1} \cup S_{s+1}$ and exit the for loop. If $|A_{s+1}| = k$, set $A^* \leftarrow A_{s+1}$ and exit the for loop.
  else
    $A^* \leftarrow A_s \cup \left(\arg k' \max_{i \in S_s} \hat{\mu}_{i,T_i(t_s)}\right)$ (break ties arbitrarily)
  end if
end for
Output $A^*$.
```

Let $H_2 = \max_{i \in [n]} i \Delta_{\sigma(i)}^{-2}$. Bubeck et al. showed that the Successive Accepts and Rejects (SAR) algorithm for the top-k identification problem under the fixed budget setting has the sample complexity of $O(H_2 \log^2 n)$ [11]. One might try running SAR by supplying SAR with bB sample budget. However, such a modification breaks its theoretical guarantee. We propose BatchSAR that extends SAR to allow (b, r)-batch arm pull, and prove its performance guarantee.

The intuition behind BatchSAR is as follows. Imagine the $(b = n, r = 1)$-batch setting where one has no choice but to pull every arm once at every round. There is no room for adaptivity; eliminating an arm does not yield more arm pulls in return. Let $\tilde{r} = \min\{r, \lceil b/2 \rceil\}$ and $\tilde{n} = \lceil b/\tilde{r} \rceil$. Note $\tilde{n} \geq 2$ by definition. Similarly in the general (b, r)-batch setting, once the surviving number of arms becomes $\tilde{n}$ or less, eliminating an arm does not help. For example, if $r < \lceil b/2 \rceil$ and there are $\tilde{n} - 1$ surviving arms, the maximum number of arm pulls one can make in a batch is $r(\tilde{n} - 1) < b$, thus wasting $b - r(\tilde{n} - 1)$ arm pulls. Therefore, we design BatchSAR to have two phases: (i) the first $n - \tilde{n}$ stages where an arm is eliminated after each stage just as SAR and (ii) the last stage where uniform sampling is performed without any elimination.

BatchSAR is described in Algorithm 3. The algorithm starts with the surviving arm set $S_1 = [n]$ and performs $(n - \tilde{n} + 1)$ stages. In each stage s, surviving arms are pulled uniformly until every arm's pull count is at least m_s as defined in the algorithm. Then, we choose an arm j_s that is the safest to remove, meaning that the empirical gap $\widehat{\Delta}_i$ defined in the algorithm is the largest. The arm j_s is then removed from the surviving set and is accepted only if j_s is the arm with the largest empirical mean. The same is repeated until the final stage $s = (n - \tilde{n} + 1)$ where we choose the empirical top-k' arms and add to the final accept set.

We claim that the algorithm spends no more than B batches. Let t_s be the last round in stage s. It is easy to see that $\min_{i \in S_s} T_i(t_s) \leq m_s + \left\lceil \frac{b}{n-s+1} \right\rceil - 1$ due to the batch effect, and a surviving arm's pull count is either $\min_{i \in S_s} T_i(t_s)$ or $\min_{i \in S_s} T_i(t_s) + 1$. Then,

$$\forall i \in S_s, T_i(t_s) \leq m_s + \left\lceil \frac{b}{n - s + 1} \right\rceil. \tag{4.12}$$

We prove the claim by considering $t_{n-\tilde{n}+1} \leq \left\lceil \frac{1}{b} \left(\sum_{i \in S_{n-\tilde{n}+1}} T_i(t_{n-\tilde{n}+1}) + \sum_{s=1}^{n-\tilde{n}} T_{j_s}(t_s) \right) \right\rceil$, which can be shown to be bounded by B using (4.12) and $\lceil b/\tilde{n} \rceil \leq \tilde{r}$.

Theorem 4.5 states the misidentification probability of BatchSAR. Define

$$H_3^{<b,r>} = \frac{1}{b} \left(\frac{\tilde{n}}{2} + \sum_{i=\tilde{n}+1}^{n} \frac{1}{i} \right) \max_{i=2,\tilde{n}+1,\tilde{n}+2,\ldots,n} i \Delta_{\sigma(i)}^{-2}. \tag{4.13}$$

Theorem 4.5. *Let A^* be the output of BatchSAR. Given a batch budget B, the* *misidentification probability of BatchSAR under the (b,r)-batch setting satisfies*

$$\mathbb{P}(A^* \neq \{1,\ldots,k\}) \leq 2n^2 \exp\left(-\frac{1}{8}\frac{B - \left(\sum_{i=\tilde{n}+1}^{n} \lceil b/i \rceil\right)/b - c_1/b}{H_3^{<b,r>}}\right).$$

Proof. We define an event ξ_s where every arm's empirical mean is controlled as follows:

$$\xi_s = \begin{cases} \left\{\forall i \in [n], \left|\hat{\mu}_{i,T_i(t_s)} - \mu_i\right| \leq \frac{\Delta_{\sigma(n-s+1)}}{4}\right\} & \text{if } s \leq n-\tilde{n} \\ \left\{\forall i \in [n], \left|\hat{\mu}_{i,T_{n-\tilde{n}+1}(t_{n-\tilde{n}+1})} - \mu_i\right| \leq \frac{\Delta_{\sigma(2)}}{4}\right\} & \text{if } s = n-\tilde{n}+1 \end{cases}.$$

The proof consists of two parts. We first upper bound the probability that the empirical means are not controlled, which is $\mathbb{P}(\cup_{s=1}^{n-\tilde{n}+1}\overline{\xi_s})$. Next, we show that when the empirical means are controlled, i.e. $\cap_{s=1}^{n-\tilde{n}+1}\xi_s$, Algorithm 3 terminates with the correct top-k arms.

Upperbound the Probability $\mathbb{P}(\cup_{s=1}^{n-\tilde{n}+1}\xi_s)$ Define $Z = \tilde{n}/2 + \sum_{i=\tilde{n}+1}^{n} 1/i$. By Hoeffding's bound [33], for $s \leq n - \tilde{n}$,

$$\begin{aligned}\mathbb{P}(\overline{\xi_s}) \leq \sum_{i=1}^{n} 2\exp\left(-2T_i(t_s)\frac{\Delta_{\sigma(n-s+1)}^2}{16}\right) &\leq 2n\exp\left(-2m_s\frac{\Delta_{\sigma(n-s+1)}^2}{16}\right) \\ &\leq 2n\exp\left(-\frac{1}{8}Z^{-1}\frac{bB - \left(\sum_{i=\tilde{n}+1}^{n}\lceil b/i\rceil\right) - c_1}{(n-s+1)\Delta_{\sigma(n-s+1)}^{-2}}\right),\end{aligned}$$

and similarly $\mathbb{P}(\overline{\xi_{n-\tilde{n}+1}}) \leq 2n\exp\left(-\frac{1}{8}Z^{-1}\frac{bB-\left(\sum_{i=\tilde{n}+1}^{n}\lceil b/i\rceil\right)-c_1}{2\Delta_{\sigma(2)}^{-2}}\right)$. By union bound,

$$\begin{aligned}\mathbb{P}(\cup_{s=1}^{n-\tilde{n}+1}\overline{\xi_s}) &\leq (n-\tilde{n}+1)2n\exp\left(-\frac{1}{8}Z^{-1}\frac{bB - \left(\sum_{i=\tilde{n}+1}^{n}\lceil b/i\rceil\right) - c_1}{\max_{i=2,\tilde{n}+1,\ldots,n} i\Delta_{\sigma(i)}^{-2}}\right) \\ &\leq 2n^2\exp\left(-\frac{1}{8}\frac{B - \left(\sum_{i=\tilde{n}+1}^{n}\lceil b/i\rceil\right)/b - c_1/b}{H_3^{<b,r>}}\right).\end{aligned}$$

Correctness of the first stage We show that at every stage s the algorithm makes a correct decision given that ξ_s is true: the selected arm j_s is accepted if it is top-k and not accepted if it is not top-k. We prove it by induction. We first show that the decision at the first stage is correct. Then, assuming that the decision at the round s is correct, we show that the stage $s+1$ is also correct.

Consider the first stage. Assume that ξ_s is true. Suppose the decision on the selected arm j_1 at the first stage is incorrect. There are two cases: *(a)* j_1 is not top-k but is accepted and *(b)* j_1 is top-k but not accepted.

We here prove that *(a)* leads to a contradiction; one can show the same for *(b)* symmetrically. Define the empirical gap

$$\widehat{\Delta}_{\rho(i)} = \begin{cases} \hat{\mu}_{\rho(i),T_{\rho(i)}(t_s)} - \hat{\mu}_{\rho(k+1),T_{\rho(k+1)}(t_s)} & \text{if } i \leq k \\ \hat{\mu}_{\rho(k),T_{\rho(k)}(t_s)} - \hat{\mu}_{\rho(i),T_{\rho(i)}(t_s)} & \text{if } i > k \end{cases}.$$

Since j_1 is not top-k, i.e., $j_1 > k$, and satisfies *(a)*, j_1 has the largest empirical gap and the largest empirical mean:

$$\widehat{\Delta}_{j_1} \geq \widehat{\Delta}_{\ell}, \forall \ell \in S_1. \tag{4.14}$$

$$\hat{\mu}_{j_1} = \hat{\mu}_{\rho(1)}. \tag{4.15}$$

Note that $\Delta_{\sigma(n)}$ is either $\mu_1 - \mu_{k+1}$ or $\mu_k - \mu_n$.

Case (a1): $\Delta_{\sigma(n)} = \mu_1 - \mu_{k+1}$. By (4.15) and Lemma 4.2,

$$\begin{array}{rcccl} \mu_{j_1} + \Delta_{\sigma(n)}/4 & \geq & \hat{\mu}_{j_1} & = & \hat{\mu}_{\rho(1)} \;\geq\; \mu_1 - \Delta_{\sigma(n)}/4 \\ \Longrightarrow \qquad\qquad \mu_{k+1} & \geq & \mu_{j_1} & \geq & \mu_1 - \Delta_{\sigma(n)}/2 = (\mu_1 + \mu_{k+1})/2 \\ \Longrightarrow \qquad\qquad & & \mu_{k+1} & \geq & \mu_1, \end{array}$$

which contradicts $\mu_1 \geq \mu_k > \mu_{k+1}$.

Case (a2): $\Delta_{\sigma(n)} = \mu_k - \mu_n$

By (4.14),

$$\begin{aligned} \hat{\mu}_{j_1} - \hat{\mu}_{\rho(k+1)} &\geq \hat{\mu}_{\rho(k)} - \hat{\mu}_{\rho(n)} \\ &\geq \mu_k - \Delta_{\sigma(n)}/4 - (\mu_n + \Delta_{\sigma(n)}/4) = \Delta_{\sigma(n)}/2 \end{aligned}$$

Note that arm $1, \ldots, k$ have the following lower bound: $\forall i \leq k, \hat{\mu}_i \geq \mu_i - \Delta_{\sigma(n)}/4 \geq \mu_k - \Delta_{\sigma(n)}/4$. Using (4.15), $\hat{\mu}_{j_1} = \hat{\mu}_\rho(1) \geq \mu_1 - \Delta_{\sigma(n)}/4 \geq \mu_k - \Delta_{\sigma(n)}/4$. Thus, there are $k+1$ arms whose empirical mean is at least $\mu_k - \Delta_{\sigma(n)}/4$. This gives us a lower bound for the $(k+1)$-th largest empirical mean: $\hat{\mu}_{\rho(k+1)} \geq \mu_k - \Delta_{\sigma(n)}/4$. Then,

$$\begin{aligned} \hat{\mu}_{j_1} - \hat{\mu}_{\rho(k+1)} &\geq \Delta_{\sigma(n)}/2 \\ \mu_{j_1} + \Delta_{\sigma(n)}/4 - (\mu_k - \Delta_{\sigma(n)}/4) &\geq \\ \implies \qquad \mu_{j_1} &\geq \mu_k, \end{aligned}$$

which contradicts $j_1 > k$.

Correctness of the s-th stage where $s \leq n - \tilde{n}$ First, consider $s = 2$. After the first round, the arm j_1 is removed from the surviving set S_2. Let $k' = k - |A_2|$, the remaining number of top arms to identify at the second stage. The problem we are facing at the second stage is exactly the first stage with $[n] \setminus \{\sigma(n)\}$ as the surviving set S_1 and k replaced with k'. If the arm j_1 was $\sigma(n)$, the event ξ_2 where the empirical mean of each arm is controlled by $\Delta_{\sigma(n-1)}/4$ implies that the decision at the end of the second stage is correct. What happens if the arm j_1 was not $\sigma(n)$? Denote by $\sigma'(i)$ the arm with i-th smallest gap in S_2 and define the gap accordingly: $\Delta'_{\sigma'(1)} \leq \ldots \leq \Delta'_{\sigma'(n-1)}$. Assuming that the decision at the first stage was correct, it is easy to verify that the largest gap $\Delta'_{\sigma'(n-1)}$ in S_2 is no less than $\Delta_{\sigma(n-1)}$. Therefore, the event ξ_2 implies that the empirical means of arms are controlled by $\Delta'_{\sigma'(n-1)}/4$, which ensures that the decision at the second stage must be correct, too. In other words, removing arm $\sigma(n)$ at the first stage is the worst case; removing other arms can only make the gaps larger and make the problem "easier".

Correctness of the final stage It is easy to show that if arms are controlled by $\Delta_{\sigma(2)}/4$ then the empirical means of the top-k' will be greater than any non-top-k' in the surviving arms $S_{n-\tilde{n}+1}$. Thus, the output k' arms at the last stage is correct. □

One can derive the batch complexity of BatchSAR as

$$O\left(H_3^{<b,r>} \log(n) + \frac{1}{b} \sum_{i=\tilde{n}+1}^{n} \left\lceil \frac{b}{i} \right\rceil \right).$$

The term $\frac{1}{b}\sum_{i=\tilde{n}+1}^{n} \left\lceil \frac{b}{i} \right\rceil = O(\log(n) + n/b)$ is the price to pay for performing the (b, r)-batch where the algorithm might pull an arm more than necessary. The batch complexity is upper-bounded by $O\left(\left(\frac{1}{\tilde{r}} + \frac{\log n}{b}\right) H_2 \log n + \log n\right)$, where the additive $\log(n)$ is negligible unless $n/b \ll \log(n)$.

In the extreme case of b=r=1, the term $\frac{1}{b}\sum_{i=\tilde{n}+1}^{n}\left\lceil\frac{b}{i}\right\rceil$ is n, which can be ignored since the batch complexity is at least of order n when $b=1$. Thus, we achieve $O(H_2\log^2 n)$, which is equivalent to the sample complexity of SAR. If $r \geq \lceil b/2 \rceil$,

$$\begin{aligned} H_3^{<b,r>}\log n + \max\{\log n, n/b\} &\leq \frac{1}{b}(1+\log n)H_2\log n + \max\{\log n, n/b\} \\ &= O\left(\frac{1}{b}H_2\log^2(n) + \log n\right), \end{aligned}$$

which achieves an overall b factor reduction from the sample complexity of SAR except for the additive $\log n$. In the other extreme case where $b=n$ and $r=1$, BatchSAR reduces to uniform sampling and the batch complexity is $O(\Delta_{\sigma(2)}^{-2}\log n)$.

Practical Considerations In general, the algorithm finishes earlier than B batches. The reason is that m_s is defined to defend against the worst case. In practice, one might want to exhaust the budget B to minimize the error. Let $\overline{\log}(b||a)=\sum_{i=a+1}^{b}\frac{1}{i}$. We propose to replace m_s with

$$m'_s = \left\lceil \frac{bB - \left(\sum_{s'=1}^{s-1} T_{j_{s'}}(t_{s'})\right) - \left(\sum_{i=\tilde{n}+1}^{n-s+1}\lceil\frac{b}{i}\rceil\right) - c'_1}{\tilde{n}/2 + \overline{\log}(n-s+1||\tilde{n})}\frac{1}{n-s+1}\right\rceil, \tag{4.16}$$

where $c'_1 = b + \tilde{n}\tilde{r} + n - s + 1$. For the last stage $s = n - \tilde{n} + 1$, we sample the surviving arms until the budget B is exhausted. One can show that $m'_s \geq m_s$ after using (4.12) to verify $\sum_{s'=1}^{s-1}T_{j_{s'}}(t_{s'}) + \left(\sum_{i=\tilde{n}+1}^{n-s+1}\lceil\frac{b}{i}\rceil\right) + c'_1 \leq \left(\sum_{i=\tilde{n}+1}^{n}\lceil\frac{b}{i}\rceil\right) + c_1 + \frac{(bB-(\sum_{i=\tilde{n}+1}^{n}\lceil\frac{b}{i}\rceil)-c_1)}{(\tilde{n}/2)+\overline{\log}(n||\tilde{n})}\overline{\log}(n||n-s+1)$. Thus, Theorem 4.5 still holds true with m'_s.

4.4 Experiments

We verify our theoretical results with simulation experiments in the fixed confidence setting, then present two important applications of the (b,r)-batch MAB problem. For applications, we focus on the fixed budget setting due to its practicality.

Toy Experiments

The batch size b and repeated pull limit r contribute to the batch complexity in an intricate way in both the fixed confidence/budget settings. We would like to verify how b and r change the batch complexity empirically and compare it to our theory. Note that the batch complexity shown in the analysis considers the worst case; the number of batches suggested by the theory is often more than what is experienced in practice. However, if such slack is independent of b and r, we can remove it by monitoring the "speedup," see below.

We consider two bandit instances with n=100 arms: (i) Bernoulli arms with expected rewards that linearly decrease, i.e., $\mu_i = (99 - i + 1)/99$, and (ii) sparsely distributed Bernoulli arms where the expected rewards of 10 arms are 0.5 and the rest are 0.3. We call the former Linear and the latter Sparse. Note we focus on the fixed confidence setting here due to space constraints; the same result can be obtained in the fixed budget setting. We run BatchRacing 10 times with k=10, varying $b \in \{1, 4, 16, 64\}$ and $r \in \{1, 2, 4, 8, 16, 32\}$. For each pair (b', r'), we compute the empirical speedup by dividing the average number of batches in (b=1,r=1) by the average number of batches in (b', r'). Let $M_{b,r}$ be the batch complexity computed by (4.10). For each pair (b', r'), we also compute the theory-based speedup by

$$\left(\sum_{i=1}^{n} \overline{T}_{\sigma(i)} \right) / M_{b',r'}.$$

We present the speedup of each (b, r)-batch in Table 4.1. Overall, the theory-based speedup matches extremely well with the empirical speedup. Note that r plays an important role in Linear; it is only after $r = b/2$ that the speedup achieves b. This is because $\Delta_{\sigma(1)}^{-2} = (\mu_k - \mu_{k+1})^{-2}$ is dominantly large, which makes the budget complexity's reliance on the factor $1/r'$ significant. On the other hand, in Sparse the gaps $\{\Delta_i\}$ are uniformly 0.2, and we achieved an overall b fold speedup (almost) regardless of r as stated in Section 4.2.

Application 1: Virus Replication Experiments

Virologists are interested in identifying which genes are important in virus replication. For instance, Hao et al. tested 13k genes of drosophila to find which genes promote

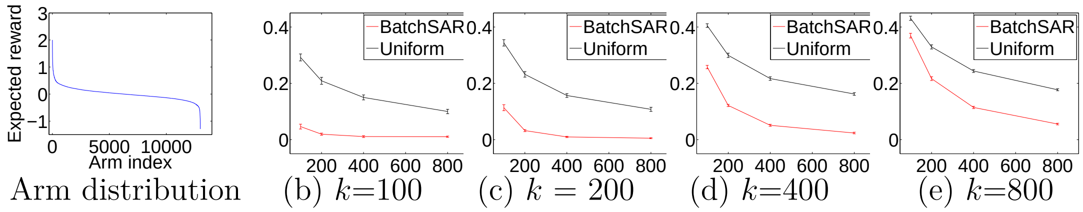
b	r	Linear		Sparse		b	r	Linear		Sparse	
		Theory	Actual	Theory	Actual			Theory	Actual	Theory	Actual
4	1	2.71	2.74	4.00	4.00	64	1	3.16	3.21	63.97	58.28
4	2	4.00	4.00	4.00	4.00	64	2	6.32	6.41	63.97	61.88
16	1	3.14	3.18	16.00	15.83	64	4	12.55	12.74	63.97	63.25
16	2	6.08	6.16	16.00	15.95	64	8	24.31	24.65	63.97	63.73
16	4	10.84	10.96	16.00	15.99	64	16	43.37	43.83	63.97	63.87
16	8	16.00	16.00	16.00	16.00	64	32	64.00	63.99	63.97	63.90

Table 4.1: Toy experiment result: the speedup in the number of batches in the fixed confidence setting.

influenza virus replication [28]. The authors perform a sequence of batch experiments using a plate with 384 microwells. Each well contains a single-gene knock-down cell strain to which the fluorescing virus is added. They measured the fluorescence levels after an incubation period and obtained an indication of how important the knock-down gene is in virus replication. Since the measurements are noisy, we formulate the problem as a top-k identification problem in MAB. Each arm is a gene, and the reward is its knock-down fluorescence level. The batch arm pull is performed with b=384 wells where the repeated pull limit r is equal to b since one can place the same gene knock-down in multiple wells.

Dataset We simulate the microwell experiments by assuming a Gaussian distribution on each arm, where the mean and variance are estimated from the actual data. Specifically, we use two measurements from each of the $n = 12,979$ [2] gene knock-downs from [28]. For the j-th measurement of arm/gene i, $x_{i,j}$, we remove the plate-specific noise by dividing it by the average control measurement from the same plate. Then, we take the logarithm which effectively makes the variance of each fluorescence measurement similar [28]. We also flip the sign to call it $z_{i,j}$ since we want the bottom-k rather than the top-k genes. We set the mean of arm i as $\mu_i = (z_{i,1} + z_{i,2})/2$. We compute the unbiased estimate of the variance $\widehat{V}_i$ of each gene and compute the average $(1/n)\sum_{i=1}^{n} \widehat{V}_i$ across all arms, which is an unbiased estimate of the variance if all arms have an equal variance. In the end, we simulate measurements of arm i using $\mathcal{N}(\mu_i, 0.1)$. We plot the arm distribution in Figure 4.1(a), where we see that relatively few genes have large rewards (that facilitate virus replication).

[2] The original dataset contains 13,071 genes. We found 92 genes whose control was measured incorrectly, so we removed them for the evaluation.



(a) Arm distribution (b) k=100 (c) k = 200 (d) k=400 (e) k=800

Figure 4.1: Microwell experiment result. (b-e) plots the false negative rate vs. budget B.

Result We use Uniform as a baseline method where the sampling strategy is to uniformly sample every arm with no eliminations until the batch budget runs out. For BatchSAR, we use m'_s defined in (4.16). We run Uniform and BatchSAR for each $k \in \{100, 200, 400, 800\}$ and batch budget $B \in \{100, 200, 400, 800\}$. We run each experiment 20 times and measure false negative rate (FNR). FNR is the ratio of arms that are not top-k in the output of the algorithm. It is of interest since high FNR was pointed out in [27] as the main issue in adaptive experiments. The result is shown in Figure 4.1 (b-e). We plot the FNR vs. batch budget B for each k, where the error bar is the 95% confidence interval. Overall, BatchSAR significantly outperforms Uniform. Uniform spends the vast majority of arm pulls on arms with small expected rewards and fails to distinguish the top-k arms from those near top-k.

Application 2: Twitter User Tracking

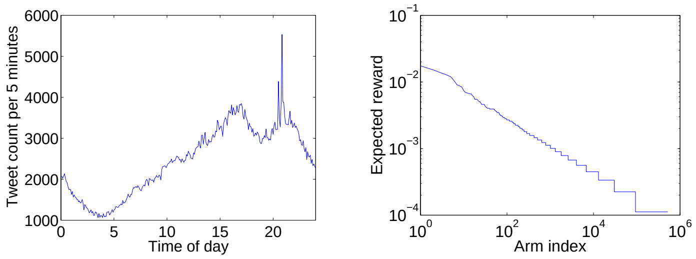
We are interested in identifying the top-k users in Twitter who talk about a given topic most frequently. For example, in the study of bullying researchers are interested in a longitudinal study on the long-term effects of bullying experience [47]. Such a study in Twitter requires first identifying the top-k (e.g., k=100) users with frequent bullying experience, then observing their posts for a long time (e.g., over a year). If one has an unlimited monitoring access to all users (e.g., firehose in Twitter), the first stage of identifying the top-k users would be trivial. However, researchers are often limited by the public API that has rate limits; for instance, user streaming API of Twitter allows one to follow at most 5000 users at a time. In MAB terminology, each user is an arm, one batch pull is to follow 5000 users at a time for a fixed time interval, and the reward is whether or not a user posted tweets about a bullying experience.

The latter is decided by a text classifier [57]. Even though it seems that $r = 1$ since we either follow a user or not, we discuss a procedure below where r can be larger.

We observe that the tweet rate of users changes significantly over time of day. Figure 4.2 (a) shows the daily trend. We plot the number of bullying tweets generated in each 5-minute interval where the count is accumulated over 31 days. To get around this diurnal issue, we define a pull of an arm to be following a user for 5 minutes every hour in a day, totaling two hours. Since there exist twelve 5-minute intervals in an hour, one can follow $b = 12 \cdot 5000 = 60,000$ users in a day. Note those 60,000 users need not be unique; one can follow the same user up to 12 times, each in a different 5-minute interval. Thus, r=12. Formally, we assume that the bullying tweet rate of a user i changes every hour of day h, but remains the same within. Let $Y_{i,d,h,c}$ indicate whether or not (1 or 0) the user i generates a bullying tweet at day d hour h interval $c \in [12]$. $Y_{i,d,h,c}$ follows a Bernoulli($p_{i,h}$) distribution, where $p_{i,h}$ is the tweet probability of user i during 5 minutes in hour h. We define the reward of an arm i on day d as $(1/24)\sum_{h=1}^{24} Y_{i,d,h,c(h)}$ where $c(h)$ indicates which interval to pick at each hour h. Then, the expected reward μ_i of an arm i is the average Bernoulli probability over a day: $\mu_i = (1/24)\sum_{h=1}^{24} p_{i,h}$.

Dataset The dataset consists of 707,776 bullying tweets collected over 31 days in January 2013. The number of distinct users n is 522,062. We compute the average reward for each user at each hour and take it as the true tweet probability $p_{i,h} \equiv (1/31)\sum_{d=1}^{31}(1/12)\sum_{c=1}^{12} Y_{i,d,h,c}$, from which we compute the expected reward μ_i as previously shown. We plot the expected reward in Figure 4.2(b), where both axes are in $\log_{10}$ scale. The plot nicely follows a power law.

Result We set k=100 and run both algorithms 10 times each where the randomness comes from breaking ties uniformly at random. We measure the error $\sum_{i=1}^{k} \mu_i - \mu_{J_i}$, where J_i is the arm with the i-th largest expected reward among the k arms selected by the algorithm. The result is summarized in Figure 4.2(c). The number in parentheses is the 95% confidence interval. BatchSAR outperforms Uniform, and the difference is statistically significant.



Algorithm	Error
BatchSAR	**0.299** (±0.01)
Uniform	0.372 (±0.01)

(a) (b) (c)

Figure 4.2: (a) Twitter daily pattern (b) Expected rewards of users (c) Experiment result

5 CONCLUSION

In this thesis, we have considered some problems that arise from sequential inputs.

In the first part of the thesis (Chapter 2 and 3), we proposed two models that are motivated by human-generated list. Human generated list has an interesting process of data creation that deserves attention from the machine learning community. In Chapter 2, we proposed SWIRL, a simple sampling process that can be explained as reducing ball sizes in the Pólya urn model. Our non-exchangeable model successfully learned from feature volunteering lists to improve classification accuracy and from verbal fluency responses to efficiently classify brain-damaged patients.

In Chapter 3, we study INVITE, a simple variant of the random walk, whose parameter estimation is not immediately clear with naive approaches. We proposed the first tractable way of computing the INVITE MLE and provided the consistency result. In the cognitive study of human memory search, we empirically proved the modeling power of INVITE.

In the second part (Chapter 4), we considered the top-k arms identification problem in MAB. We introduced a new setting where arms are pulled in batches, which we call the (b, r)-batch setting. We proposed two novel algorithms for the fixed confidence setting and the fixed budget setting respectively. Our analysis showed that when the repeated pull limit r is roughly $b/2$ we achieve b factor reduction in the batch complexity when compared to the sample complexity in the standard setting (ignoring an additive $\log(n)$ term). We showed that the algorithms work well in practice using two important applications: microwell biology experimentation and monitoring active users in social media.

Future Works

Our solutions to the challenges encountered in sequential inputs suggest numerous future research topics.

For SWIRL, recall that our parameter estimation from verbal fluency lists was performed at group level rather than individual level. We would like to develop a hierarchical version of SWIRL, so that each human has their own personalized parameters λ, $\mathbf{s}$, and α, while a group has summary parameters, too. This is particularly attractive for applications like verbal fluency, where we want to understand both individual and group behaviors.

For INVITE, first, more theoretical investigation is needed. For example, although we know the MLE of INVITE is consistent, the convergence rate is unknown. Second, one can improve the INVITE estimate when data is sparse by assuming certain cluster structures in the transition matrix $\mathbf{P}$, thereby reducing the degree of freedom. For instance, it is known that verbal fluency tends to exhibit "runs" of semantically related words. One can assume a stochastic block model $\mathbf{P}$ with parameter sharing at the block level, where the blocks represent semantic clusters of words. One then estimates the block structure and the shared parameters at the same time. Third, INVITE can be extended to allow repetitions in a list. The basic idea is as follows. In the k-th segment we previously used an absorbing random walk to compute $\mathbb{P}(a_{k+1} \mid a_{1:k})$, where $a_{1:k}$ were the nonabsorbing states. For each nonabsorbing state a_i, add a "dongle twin" absorbing state a'_i attached only to a_i. Allow a small transition probability from a_i to a'_i. If the walk is absorbed by a'_i, we output a_i in the censored list, which becomes a repeated item in the censored list. Note that the likelihood computation in this augmented model is still polynomial. Such a model with "relunctant repeatitions" will be an interesting interpolation between "no repetitions" and "repetitions as in a standard random walk." Fourth, in some applications an item in a list also carries its generation time stamp. For example, in verbal fluency, the generation time of each item is often recorded, too. The time stamp is a valuable information since there exists a longer time delay between items from different cluster than between items from the same cluster [26]. Incorporating the time stamps into INVITE would help revealing more information on the underlying Markov chain.

For the application of human-generated lists, first, we would like to design a "supervision pipeline" that starts with building a less accurate classifier with less supervision, then gradually increases the classification accuracy as more supervision is available. For example, one can start by feature volunteering, then combine with feature label querying, document label querying, and other forms of interactive learning to build better text classifiers more quickly. Second, it would be important to identify more applications that can benefit from models of human list generation. For example, creating lists of photo tags on Instagram, or hashtags on Twitter.com, can be viewed as a form of human list generation conditioned on a specific photo or tweet.

For the batch arm pull setting in the MAB problem, first, note that we analyzed the upper bound of the proposed algorithms. This does not tell us whether or not our batch complexity is order optimal. We would like to analyze the smallest number of batches that is required to solve the top-k identification problem under the batch setting. This lower bound result would explain the fundamental limit of the batch arm pulls. Second, LUCB algorithm [38] is known to work best in practice [35]. LUCB works by sorting arms in their empirical mean rewards then pulling the following two arms consecutively: the arm with the lowest LCB from the empirical top-k arms and the arm with the largest UCB from the bottom-$(n-k)$ arms. One can extend LUCB to choose $b/2$ arms from the top-k and $b/2$ arms from the bottom-$(n-k)$ (total b arms) in a batch so it works for the (b, r)-batch setting. It would be interesting to see if the batch LUCB algorithm outperforms our algorithms in both theory and practice. Third, no one has considered the (b, r)-batch setting in the regret setting, though some special cases of the (b, r)-batch was considered in prior works [3, 43], It would be intriguing to see how the batch size b and the repeated pull limit r come into play in the regret setting.

A.1 Results Required for Theorem 3.7

Throughout, assume $\boldsymbol{\theta} = (\boldsymbol{\pi}^\top, \mathbf{P}_{1\cdot}, \ldots, \mathbf{P}_{n\cdot})^\top$. Let $\mathrm{supp}(\boldsymbol{\theta})$ be the set of nonzero dimensions of $\boldsymbol{\theta}$: $\mathrm{supp}(\boldsymbol{\theta}) = \{i \mid \theta_i > 0\}$. Lemma A.1 shows conditions on which $\mathcal{Q}^*(\boldsymbol{\theta})$ and $\widehat{\mathcal{Q}}_m(\boldsymbol{\theta})$ are above $-\infty$.

Lemma A.1. *Assume A1. Then,*

$$\mathit{supp}(\boldsymbol{\theta}) \supseteq \mathit{supp}(\boldsymbol{\theta}^*) \iff \mathcal{Q}^*(\boldsymbol{\theta}) > -\infty \tag{A.1}$$

$$\mathit{supp}(\boldsymbol{\theta}) \supseteq \mathit{supp}(\boldsymbol{\theta}^*) \implies \widehat{\mathcal{Q}}_m(\boldsymbol{\theta}) > -\infty, \forall m. \tag{A.2}$$

Proof. Define two vectors of probabilities w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$: $\mathbf{q} = [q_{\mathbf{a}} = \mathbb{P}(\mathbf{a}; \boldsymbol{\theta})]_{\mathbf{a} \in \mathcal{D}}$ and $\mathbf{q}^* = [q^*_{\mathbf{a}} = \mathbb{P}(\mathbf{a}; \boldsymbol{\theta}^*)]_{\mathbf{a} \in \mathcal{D}}$. Note that

$$\mathrm{supp}(\mathbf{q}) \supseteq \mathrm{supp}(\mathbf{q}^*) \iff \mathcal{Q}^*(\boldsymbol{\theta}) > -\infty$$

by the definition of $\mathcal{Q}^*(\boldsymbol{\theta})$. Thus, for (A.1), it suffices to show that

$$\mathrm{supp}(\boldsymbol{\theta}) \supseteq \mathrm{supp}(\boldsymbol{\theta}^*) \iff \mathrm{supp}(\mathbf{q}) \supseteq \mathrm{supp}(\mathbf{q}^*).$$

($\implies$) The LHS implies that the directed graph induced by $\boldsymbol{\theta}$ includes the graph induced by $\boldsymbol{\theta}^*$; a path that is possible w.r.t. $\boldsymbol{\theta}^*$ is also possible w.r.t. $\boldsymbol{\theta}$. Recall that a list is generated by a random walk. Let $\mathbf{a} \in \mathrm{supp}(\mathbf{q}^*)$. There exists a random walk under $\boldsymbol{\theta}^*$ that generates $\mathbf{a}$. Then, the same random walk is also possible under $\boldsymbol{\theta}$, which implies $\mathbf{a} \in \mathrm{supp}(\mathbf{q})$.

($\impliedby$) Suppose the LHS is false. Then, there exists (i, j) s.t. $P_{ij} = 0$ and $P^*_{ij} > 0$. Consider a list $\mathbf{a}$ such that it has nonzero probability w.r.t. $\boldsymbol{\theta}^*$ (that is, $q^*_{\mathbf{a}} > 0$), and its first two items are i then j. Since $P_{ij} = 0$, $q_{\mathbf{a}} = 0$. However, the RHS implies that $q_{\mathbf{a}} > 0$ since $q^*_{\mathbf{a}} > 0$: a contradiction.

For (A.2),

$$\mathrm{supp}(\boldsymbol{\theta}) \supseteq \mathrm{supp}(\boldsymbol{\theta}^*) \implies \mathcal{Q}^*(\boldsymbol{\theta}) > -\infty \implies \widehat{\mathcal{Q}}_m(\boldsymbol{\theta}) > -\infty, \forall m,$$

where the last implication is due to the fact that a censored list $\mathbf{a}^{(i)}$ that appears in $\widehat{\mathcal{Q}}_m(\boldsymbol{\theta})$ is generated by $\boldsymbol{\theta}^*$, so the term $\log \mathbb{P}(\mathbf{a}^{(i)}; \boldsymbol{\theta})$ also appears in $\mathcal{Q}^*(\boldsymbol{\theta})$.

□

Lemma A.2. *Assume A1. Then, $\boldsymbol{\theta}^*$ is the unique maximizer of $\mathcal{Q}^*(\boldsymbol{\theta})$.*

Proof. If $\boldsymbol{\theta}$ satisfies $\mathrm{supp}(\boldsymbol{\theta}) \not\supseteq \mathrm{supp}(\boldsymbol{\theta}^*)$, then $\mathcal{Q}^*(\boldsymbol{\theta}) = -\infty$ by Lemma A.1, so such $\boldsymbol{\theta}$ cannot be a maximizer. Thus, it is safe to restrict our attention to $\boldsymbol{\theta}$'s whose support include that of $\boldsymbol{\theta}^*$: $\mathrm{supp}(\boldsymbol{\theta}) \supseteq \mathrm{supp}(\boldsymbol{\theta}^*)$.

Recall the definition of $\mathcal{Q}^*(\boldsymbol{\theta})$:

$$\mathcal{Q}^*(\boldsymbol{\theta}) = \sum_{\mathbf{a} \in \mathcal{D}} \mathbb{P}(\mathbf{a}; \boldsymbol{\theta}^*) \log \mathbb{P}(\mathbf{a}; \boldsymbol{\theta}) \propto -\mathrm{KL}(\boldsymbol{\theta}^* || \boldsymbol{\theta}),$$

where $\mathrm{KL}(\boldsymbol{\theta}^*||\boldsymbol{\theta})$ is well defined since $\mathrm{supp}(\boldsymbol{\theta}) \supseteq \mathrm{supp}(\boldsymbol{\theta}^*)$. Due to the identifiability of the model (Theorem 3.6) and the unique minimizer property of the KL-divergence, $\boldsymbol{\theta}^*$ is the unique maximizer. □

We denote by $\mathrm{decomp}(\boldsymbol{\theta}) = \{T, W_1, \ldots, W_K\}$ the decomposition induced by $\boldsymbol{\theta}$ as in Theorem 3.1.

Lemma A.3. *$supp(\widehat{\boldsymbol{\theta}}_m) \supseteq supp(\boldsymbol{\theta}^*)$ for large enough m. Furthermore, $decomp(\widehat{\boldsymbol{\theta}}_m) = decomp(\boldsymbol{\theta}^*)$ for large enough m.*

Proof. Note that due to the strong law of large numbers, a list $\mathbf{a}$ is valid in the true model $\boldsymbol{\theta}^*$ must appear in D_m for large enough m. Since the number of censored lists that can be generated by $\boldsymbol{\theta}^*$ is finite, one observes every valid censored list in the true model $\boldsymbol{\theta}^*$; that is, there exists m' such that

$$m \geq m' \implies \{\mathbf{a} \mid \mathbf{a} \in D_m\} = \{\mathbf{a} \mid \mathbb{P}(\mathbf{a}; \boldsymbol{\theta}^*) > 0\}.$$

For the first statement, assume that $m \geq m'$. Since we observe every valid list in $\boldsymbol{\theta}^*$, by the definition of $\widehat{\mathcal{Q}}_m(\boldsymbol{\theta})$, the following holds true:

$$\forall \boldsymbol{\theta} \in \boldsymbol{\Theta},\ \widehat{\mathcal{Q}}_m(\boldsymbol{\theta}) > -\infty \iff \mathcal{Q}^*(\boldsymbol{\theta}) > -\infty.$$

Then, using Lemma A.1,

$$\widehat{\mathcal{Q}}_m(\widehat{\boldsymbol{\theta}}_m) > -\infty \implies \mathcal{Q}^*(\widehat{\boldsymbol{\theta}}_m) > -\infty \implies \mathrm{supp}(\widehat{\boldsymbol{\theta}}_m) \supseteq \mathrm{supp}(\boldsymbol{\theta}^*).$$

For the second statement, assume $m \geq m'$. Let $\mathrm{decomp}(\widehat{\boldsymbol{\theta}}_m) = \{\widehat{T}, \widehat{W}_1, \ldots, \widehat{W}_{\widehat{K}}\}$ and $\mathrm{decomp}(\boldsymbol{\theta}^*) = \{T^*, W_1^*, \ldots, W_{K^*}^*\}$. Furthermore, define $\widehat{\tau}(i)$ to be the index of the closed irreducible set in $\mathrm{decomp}(\widehat{\boldsymbol{\theta}}_m)$ to which i belongs, and define $\tau^*(i)$ similary.

Suppose that the data D_m contains every valid list in $\boldsymbol{\theta}^*$, but $\mathrm{decomp}(\widehat{\boldsymbol{\theta}}_m) \neq \mathrm{decomp}(\boldsymbol{\theta}^*)$. There are four cases. In each case, we show that there exists a list that is valid in $\boldsymbol{\theta}^*$ but not in $\widehat{\boldsymbol{\theta}}_m$, which means that the log likelihood of $\widehat{\boldsymbol{\theta}}_m$ is $-\infty$. This is a contradiction in that $\widehat{\boldsymbol{\theta}}_m$ is the MLE.

Case 1 : $\exists s_1$ s.t. s_1 is transient in $\widehat{\boldsymbol{\theta}}$ but recurrent in $\boldsymbol{\theta}^*$.

Let W_k^* be the closed irreducible set to which s_1 belongs and $L = |W_k^*|$. Use $\boldsymbol{\theta}^*$ to start a random walk from s_1 and generate a censored list $\mathbf{a}$, which consists of all states in W_k^*: $\mathbf{a} = (s_1, s_2, \ldots, s_L)$. If $\mathbf{a}$ is invalid in $\widehat{\boldsymbol{\theta}}_m$, we have a contradiction. If not, s_L must be recurrent in $\widehat{\boldsymbol{\theta}}_m$ by Theorem 3.2. Use $\boldsymbol{\theta}^*$ to generate a censored list $\mathbf{a}'$ that starts from s_L. Then, s_1 must appear after s_L in $\mathbf{a}'$. However, this is impossible in $\widehat{\boldsymbol{\theta}}_m$ since s_1 is transient and s_L is recurrent: a contradiction.

Case 2 : $\exists t$ s.t. is transient in $\boldsymbol{\theta}^*$ but recurrent in $\widehat{\boldsymbol{\theta}}_m$.

For brevity, assume that t is the only transient state in $\boldsymbol{\theta}^*$; this can be easily relaxed. Use $\boldsymbol{\theta}^*$ to generate a censored list that starts with t, say $\mathbf{a} = (t, s_1, \ldots, s_L)$. By Theorem 3.2, $\{s_{1:L}\}$ is a closed irreducible set in $\boldsymbol{\theta}^*$. Define $\mathbf{a}' = (s_{1:L})$, which is also valid in $\boldsymbol{\theta}^*$. Now, $\mathbf{a}$ may or may not be valid in $\widehat{\boldsymbol{\theta}}_m$. Assume that $\mathbf{a}$ is valid in $\widehat{\boldsymbol{\theta}}_m$ since otherwise we have a contradiction. Then, in $\widehat{\boldsymbol{\theta}}_m$, $\{t, s_{1:L}\}$ must be a closed irreducible set since t is recurrent. Then, $\mathbf{a}' = (s_{1:L})$ is invalid in $\widehat{\boldsymbol{\theta}}_m$ since t must be visited as well: a contradiction.

Case 3. $\exists (i,j)$ s.t. $\widehat{\tau}(i) = \widehat{\tau}(j)$, but $\tau^*(i) \neq \tau^*(j)$.

Start a random walk from the state i w.r.t. $\boldsymbol{\theta}^*$ and generate a censored list $\mathbf{a}$. By Theorem 3.2, the censored list $\mathbf{a}$ does not contain j. In $\widehat{\boldsymbol{\theta}}_m$, however, a censored list starting from i must also output j since i and j are in the same closed irreducible set. Thus, $\mathbf{a}$ is invalid in $\widehat{\boldsymbol{\theta}}_m$: a contradiction.

Case 4. $\exists (i,j)$ s.t. $\tau^*(i) = \tau^*(j)$, but $\widehat{\tau}(i) \neq \widehat{\tau}(j)$.

Start a random walk from the state i w.r.t. $\boldsymbol{\theta}^*$ and generate a censored list $\mathbf{a}$. By

Theorem 3.2, the censored list $\mathbf{a}'$ must also contain j. In $\widehat{\boldsymbol{\theta}}_m$, however, a censored list starting from i cannot output j since j is in a different closed irreducible set. Thus, $\mathbf{a}'$ is invalid in $\widehat{\boldsymbol{\theta}}_m$: a contradiction.

□

Lemma A.4. *Assume A1. Let* $\{\widehat{\boldsymbol{\theta}}_{m_j}\}$ *be a convergent subsequence of* $\{\widehat{\boldsymbol{\theta}}_m\}$ *and* $\boldsymbol{\theta}'$ *be its limit point:* $\boldsymbol{\theta}' = \lim_{j\to\infty} \widehat{\boldsymbol{\theta}}_{m_j}$. *Then,* $\lim_{j\to\infty} \mathbb{P}(\mathbf{a}; \widehat{\boldsymbol{\theta}}_{m_j}) = \mathbb{P}(\mathbf{a}; \boldsymbol{\theta}')$ *for all* $\mathbf{a}$ *that is valid in* $\boldsymbol{\theta}^*$.

Proof. There are exactly two case-by-case operators that causes the likelihood function to be discontinuous. The operators appear in (3.3) and (3.1), which respectively rely on the following conditions w.r.t. a list $\mathbf{a} = (a_{1:M})$:

$$(\mathbf{I} - \mathbf{Q}^{(k)})^{-1} \text{ exists}, \forall k \in [M-1] \tag{A.3}$$

$$\mathbb{P}(s \mid a_{1:M}; \boldsymbol{\theta}) = 0, \forall s \in S \setminus \{a_{1:M}\}. \tag{A.4}$$

Step 1: claim that $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}$,

$$\operatorname{supp}(\boldsymbol{\theta}) \supseteq \operatorname{supp}(\boldsymbol{\theta}^*) \text{ and } \operatorname{decomp}(\boldsymbol{\theta}) = \operatorname{decomp}(\boldsymbol{\theta}^*) \implies \forall \mathbf{a} \text{ valid in } \boldsymbol{\theta}^* \text{ (A.3) and (A.4)}$$

To show (A.3), suppose it is false for some $k \in [M-1]$ and some censored list $\mathbf{a} = (a_{1:M})$ valid in $\boldsymbol{\theta}^*$. The nonexistence of $(\mathbf{I} - \mathbf{Q}^{(k)})^{-1}$ implies that there is no path from a_k to a state that is outside of $\{a_{1:k}\}$ whereas there is such a path w.r.t. $\boldsymbol{\theta}^*$. This contradicts $\operatorname{supp}(\boldsymbol{\theta}) \supseteq \operatorname{supp}(\boldsymbol{\theta}^*)$.

To show (A.4), consider a censored list $\mathbf{a} = (a_{1:M})$ that is valid in $\boldsymbol{\theta}^*$. By Theorem 3.2, the last state a_M must be a recurrent state in a closed irreducible set W w.r.t. $\boldsymbol{\theta}^*$. Since $\boldsymbol{\theta}$ has the same decomposition as $\boldsymbol{\theta}^*$ and every state in W must be present in $\mathbf{a}$, no other state can appear after a_M. This implies (A.4).

Define

$$\boldsymbol{\Theta}' = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} \mid \ ||\boldsymbol{\theta} - \boldsymbol{\theta}'||_\infty < \min_i \theta'_i, \operatorname{decomp}(\boldsymbol{\theta}) = \operatorname{decomp}(\boldsymbol{\theta}')\}.$$

Step 2: claim that $\mathbb{P}(\mathbf{a}; \boldsymbol{\theta})$ is a continuous function of $\boldsymbol{\theta}$ in the subspace $\boldsymbol{\Theta}'$, $\forall \mathbf{a}$ valid in $\boldsymbol{\theta}^*$.

Note that $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}'$,

$$\begin{array}{rcccl} \mathrm{supp}(\boldsymbol{\theta}) & \supseteq & \mathrm{supp}(\boldsymbol{\theta}') & \supseteq & \mathrm{supp}(\boldsymbol{\theta}^*) \\ \mathrm{decomp}(\boldsymbol{\theta}) & = & \mathrm{decomp}(\boldsymbol{\theta}') & = & \mathrm{decomp}(\boldsymbol{\theta}^*), \end{array}$$

where the first subset relation is due to the ∞-norm in the definition of $\boldsymbol{\Theta}'$, the second subset relation and the last equality is due to Lemma A.3 and $\boldsymbol{\theta}' = \lim_{j\to\infty} \widehat{\boldsymbol{\theta}}_{m_j}$.

This implies, together with step 1, that $\forall \boldsymbol{\theta} \in \boldsymbol{\Theta}'$, (A.3) and (A.4) are satisfied, which effectively gets rid of the case-by-case operators in $\boldsymbol{\Theta}'$. This concludes the claim.

Step 3: $\lim_{j\to\infty} \mathbb{P}(\mathbf{a}; \widehat{\boldsymbol{\theta}}_{m_j}) = \mathbb{P}(\mathbf{a}; \boldsymbol{\theta}')$ for all $\mathbf{a}$ that is valid in $\boldsymbol{\theta}^*$.

Since $\widehat{\boldsymbol{\theta}}_{m_j} \to \boldsymbol{\theta}'$, there exists J such that

$$j \geq J \implies ||\widehat{\boldsymbol{\theta}}_{m_j} - \boldsymbol{\theta}'||_\infty < \min_i \theta'_i \,.$$

Thus, after J, the sequence enters the subspace $\boldsymbol{\Theta}'$ in which $\mathbb{P}(\mathbf{a}; \boldsymbol{\theta})$ is continuous $\forall \mathbf{a}$ valid in $\boldsymbol{\theta}^*$, which concludes the claim.

□

Lemma A.5. *Assume A1. Let $\{\widehat{\boldsymbol{\theta}}_{m_j}\}$ be a convergent subsequence of $\{\widehat{\boldsymbol{\theta}}_m\}$ and $\boldsymbol{\theta}'$ be its limit point: $\boldsymbol{\theta}' = \lim_{j\to\infty} \widehat{\boldsymbol{\theta}}_{m_j}$. Then, $\mathcal{Q}^*(\boldsymbol{\theta}') > -\infty$.*

Proof. Suppose not: $\mathcal{Q}^*(\boldsymbol{\theta}') = -\infty$. Then, there exists a list $\mathbf{a}'$ that is valid in $\boldsymbol{\theta}^*$ whose likelihood w.r.t. $\boldsymbol{\theta}'$ converges to 0:

$$\exists \mathbf{a}' \text{ s.t. } \mathbb{P}(\mathbf{a}'; \boldsymbol{\theta}^*) > 0 \text{ and } \mathbb{P}(\mathbf{a}'; \boldsymbol{\theta}') = 0,$$

By Lemma A.4, $\mathbb{P}(\mathbf{a}'; \boldsymbol{\theta}') = 0$ implies that $\lim_{j\to\infty} \mathbb{P}(\mathbf{a}'; \widehat{\boldsymbol{\theta}}_{m_j}) = 0$.

Let $0 < \epsilon < \mathbb{P}(\mathbf{a}'; \boldsymbol{\theta}^*)$. Denote by $\#\{\mathbf{a}'\}$ the number of occurrences of the list $\mathbf{a}'$ in $\{\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(m_j)}\}$. Then, the following statements hold:

$$\exists J_1 \text{ s.t. } j > J_1 \implies \left| \frac{\#\{\mathbf{a}'\}}{m_j} - \mathbb{P}(\mathbf{a}'; \boldsymbol{\theta}^*) \right| < \epsilon \tag{A.5}$$

$$\exists J_2 \text{ s.t. } j < J_2 \implies \left| \widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta}^*) - \mathcal{Q}^*(\boldsymbol{\theta}^*) \right| < \epsilon \tag{A.6}$$

$$\exists J_3 \text{ s.t. } j > J_3 \implies \log \mathbb{P}(\mathbf{a}'; \widehat{\boldsymbol{\theta}}_{m_j}) < \frac{\mathcal{Q}^*(\boldsymbol{\theta}^*) - \epsilon}{\mathbb{P}(\mathbf{a}'; \boldsymbol{\theta}^*) - \epsilon}. \tag{A.7}$$

The first two statements are due to the law of large numbers, and the last statement is due to the convergence of $\mathbb{P}(\mathbf{a}';\widehat{\boldsymbol{\theta}}_{m_j})$ to 0. Note that $\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta}^*) \leq \widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})$ since $\widehat{\boldsymbol{\theta}}_{m_j}$ is the maximizer of the function $\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta})$. Then, if $j > \max\{J_1, J_2, J_3\}$,

$$\begin{aligned}\mathcal{Q}^*(\boldsymbol{\theta}^*) - \epsilon &\leq \widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta}^*)\\ &\leq \widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})\\ &= \left(\sum_{\mathbf{a}\neq\mathbf{a}'} \frac{\#\{\mathbf{a}\}}{m_j}\log\mathbb{P}(\mathbf{a};\widehat{\boldsymbol{\theta}}_{m_j})\right) + \frac{\#\{\mathbf{a}'\}}{m_j}\log\mathbb{P}(\mathbf{a}';\widehat{\boldsymbol{\theta}}_{m_j})\\ &< (\mathbb{P}(\mathbf{a}';\boldsymbol{\theta}^*) - \epsilon)\log\mathbb{P}(\mathbf{a}';\widehat{\boldsymbol{\theta}}_{m_j})\\ &< \mathcal{Q}^*(\boldsymbol{\theta}^*) - \epsilon,\end{aligned}$$

where the last inequality is due to (A.7). This is a contradiction. □

Lemma A.6. *Assume A1. Let $\{\widehat{\boldsymbol{\theta}}_{m_j}\}$ be a convergent subsequence of $\{\widehat{\boldsymbol{\theta}}_m\}$. Let $\boldsymbol{\theta}' = \lim_{j\to\infty}\widehat{\boldsymbol{\theta}}_{m_j}$. Then, $\lim_{j\to\infty}\widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j}) = \mathcal{Q}^*(\boldsymbol{\theta}')$.*

Proof. The idea is that we can have a compact ball around the limit point $\boldsymbol{\theta}'$ and show that the log likelihood $\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta})$ converges uniformly on the ball. Then, after the sequence $\widehat{\boldsymbol{\theta}}_{m_j}$ gets in the ball, we can use the uniform convergence of the log likelihood.

Let $B_{\boldsymbol{\theta}'}(r) = \{\boldsymbol{\theta} \in \boldsymbol{\Theta} \mid ||\boldsymbol{\theta} - \boldsymbol{\theta}'||_\infty \leq r\}$ be an ∞-norm ball around $\boldsymbol{\theta}'$. Choose $\epsilon' < \min_{i\in\text{supp}(\boldsymbol{\theta})}\theta_i$. We claim that

$$\forall\boldsymbol{\theta} \in B_{\boldsymbol{\theta}'}(\epsilon'),\ \mathcal{Q}^*(\boldsymbol{\theta}) > -\infty \text{ and } \widehat{\mathcal{Q}}_m(\boldsymbol{\theta}) > -\infty, \forall m. \tag{A.8}$$

Let $\boldsymbol{\theta} \in B_{\boldsymbol{\theta}'}(\epsilon')$. By the definition of the ball $B_{\boldsymbol{\theta}'}(\epsilon')$, $\text{supp}(\boldsymbol{\theta}) \supseteq \text{supp}(\boldsymbol{\theta}')$. Note that $\mathcal{Q}^*(\boldsymbol{\theta}') > -\infty$ by Lemma A.5. By Lemma A.1, $\text{supp}(\boldsymbol{\theta}') \supseteq \text{supp}(\boldsymbol{\theta}^*)$:

$$\text{supp}(\boldsymbol{\theta}) \supseteq \text{supp}(\boldsymbol{\theta}') \supseteq \text{supp}(\boldsymbol{\theta}^*).$$

This then, again by Lemma A.1, implies the claim. Now, $\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta})$ converges to $\mathcal{Q}^*(\boldsymbol{\theta})$ uniformly on the ball $B_{\boldsymbol{\theta}'}(\epsilon')$ since the function is continuous on the ball that is compact.

Let $0 < \epsilon < 2\epsilon'$. Note

$$P\left(\|\widehat{\boldsymbol{\theta}}_{m_j} - \boldsymbol{\theta}'\|_\infty > \epsilon/2\right) \to 0 \tag{A.9}$$

$$P\left(\sup_{\boldsymbol{\theta}\in B_{\boldsymbol{\theta}'}(\epsilon')}\left|\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta})-\mathcal{Q}^*(\boldsymbol{\theta})\right|>\epsilon/2\right)\to 0 \tag{A.10}$$

$$P\left(|\mathcal{Q}^*(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\boldsymbol{\theta}')|>\epsilon/2\right)\to 0. \tag{A.11}$$

(A.9) is due to the convergence of $\{\widehat{\boldsymbol{\theta}}_{m_j}\}$. (A.10) holds because of the uniform convergence on the ball $B_{\boldsymbol{\theta}'}(\epsilon')$. (A.11) holds because $\mathcal{Q}^*(\boldsymbol{\theta})$ is continuous at $\boldsymbol{\theta}$.

Recall we want to show $\mathbb{P}(|\widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\boldsymbol{\theta}')|>\epsilon)\to 0$. Note:

$$\begin{aligned}&\mathbb{P}(|\widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\boldsymbol{\theta}')|>\epsilon)\\ \leq&\mathbb{P}(|\widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\widehat{\boldsymbol{\theta}}_{m_j})|>\epsilon/2)+\mathbb{P}(|\mathcal{Q}^*(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\boldsymbol{\theta}')|>\epsilon/2).\end{aligned}$$

The second term goes to zero by (A.11). It remains to show that the first term goes to 0:

$$\begin{aligned}&\mathbb{P}(|\widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\widehat{\boldsymbol{\theta}}_{m_j})|>\epsilon/2)\\ \leq&\mathbb{P}\left(|\widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\widehat{\boldsymbol{\theta}}_{m_j})|>\epsilon/2\,\middle|\,||\widehat{\boldsymbol{\theta}}_{m_j}-\boldsymbol{\theta}'||_\infty>\epsilon/2\right)\mathbb{P}\left(||\widehat{\boldsymbol{\theta}}_{m_j}-\boldsymbol{\theta}'||_\infty>\epsilon/2\right)+\\ &\mathbb{P}\Big(\left\{|\widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\widehat{\boldsymbol{\theta}}_{m_j})|>\epsilon/2\right\}\bigcap\left\{||\widehat{\boldsymbol{\theta}}_{m_j}-\boldsymbol{\theta}'||_\infty\leq\epsilon/2\right\}\Big).\end{aligned}$$

The first term goes to zero by (A.9). The second term also goes to zero as follows, which completes the proof:

$$\begin{aligned}&\mathbb{P}\Big(\left\{|\widehat{\mathcal{Q}}_{m_j}(\widehat{\boldsymbol{\theta}}_{m_j})-\mathcal{Q}^*(\widehat{\boldsymbol{\theta}}_{m_j})|>\epsilon/2\right\}\bigcap\left\{||\widehat{\boldsymbol{\theta}}_{m_j}-\boldsymbol{\theta}'||_\infty\leq\epsilon/2\right\}\Big)\\ \leq&\mathbb{P}\left(\left\{\sup_{\boldsymbol{\theta}\in B_{\boldsymbol{\theta}'}(\epsilon/2)}|\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta})-\mathcal{Q}^*(\boldsymbol{\theta})|>\epsilon/2\right\}\bigcap\left\{||\widehat{\boldsymbol{\theta}}_{m_j}-\boldsymbol{\theta}'||_\infty\leq\epsilon/2\right\}\right)\\ \leq&\mathbb{P}\left(\sup_{\boldsymbol{\theta}\in B_{\boldsymbol{\theta}'}(\epsilon/2)}|\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta})-\mathcal{Q}^*(\boldsymbol{\theta})|>\epsilon/2\right)\\ \leq&\mathbb{P}\left(\sup_{\boldsymbol{\theta}\in B_{\boldsymbol{\theta}'}(\epsilon')}|\widehat{\mathcal{Q}}_{m_j}(\boldsymbol{\theta})-\mathcal{Q}^*(\boldsymbol{\theta})|>\epsilon/2\right)\to 0,\end{aligned}$$

where the last line is due to (A.10). □

A.2 Proof of (4.9)

Lemma A.7. *Let t be a positive integer and $c > 0$, $\omega in (0, 1)$. Then,*

$$\tau \leq c \log \left(\frac{\log_2 2\tau}{\omega} \right) \implies \tau \leq c \log \left(\frac{2}{\omega} \log_2 \left(\frac{3c}{\omega} \right) \right) \tag{A.12}$$

Proof. Using $\log_2(2x) = 1 + \log_2 x \leq x$ for positive integer x and $\log y \leq 2\sqrt{y}$,

$$\begin{aligned} \tau &\leq c \log \left(\frac{\log_2(2\tau)}{\omega} \right) \leq c \log \left(\frac{\tau}{\omega} \right) \leq c2\sqrt{\frac{\tau}{\omega}} \\ \tau^2 &\leq 4c^2 \frac{\tau}{\omega} \\ \tau &\leq \frac{4c^2}{\omega} \end{aligned}$$

Then,

$$\begin{aligned} \tau &\leq c \log \left(\frac{\log_2(2\tau)}{\omega} \right) = c \log \left(\frac{\log_2 \left(2 \left(\frac{4c^2}{\omega} \right) \right)}{\omega} \right) \\ &\leq c \log \left(\frac{2}{\omega} \log_2 \left(\frac{\sqrt{8}c}{\sqrt{\omega}} \right) \right) \\ &\leq c \log \left(\frac{2}{\omega} \log_2 \left(\frac{3c}{\omega} \right) \right). \end{aligned}$$

□

BIBLIOGRAPHY

[1] Joshua T. Abbott, Joseph L. Austerweil, and Thomas L. Griffiths. Human memory search as a random walk in a semantic network. In *NIPS*, pages 3050–3058, 2012.

[2] Bruno D. Abrahao, Flavio Chierichetti, Robert Kleinberg, and Alessandro Panconesi. Trace complexity of network inference. *CoRR*, abs/1308.2954, 2013.

[3] V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: I.i.d. rewards. *Automatic Control, IEEE Transactions on*, 32(11):968–976, 1987.

[4] Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2014.

[5] J. V. Baldo and A. P. Shimamura. Letter and category fluency in patients with frontal lobe lesions. *Neuropsychology*, 12(2):259–267, 1998.

[6] Robert E. Bechhofer. A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs. *Biometrics*, 14(3):408–429, 1958.

[7] Léon Bottou. Stochastic gradient tricks. In Grégoire Montavon, Genevieve B. Orr, and Klaus-Robert Müller, editors, *Neural Networks, Tricks of the Trade, Reloaded*, Lecture Notes in Computer Science (LNCS 7700), pages 430–445. Springer, 2012.

[8] S. Boyd and L. Vandenberge. *Convex Optimization.* Cambridge University Press, Cambridge UK, 2004.

[9] T. Brants, A.C. Popat, P. Xu, F.J. Och, and J. Dean. Large language models in machine translation. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[10] Andrei Z. Broder. Generating random spanning trees. In *FOCS*, pages 442–447. IEEE Computer Society, 1989.

[11] Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 258–265, 2013.

[12] A. S. Chan, N. Butters, J. S. Paulsen, D. P. Salmon, M. R. Swenson, and L. T. Maloney. An assessment of the semantic network in patients with alzheimer's disease. *Journal of Cognitive Neuroscience*, 5(2):254–261, 1993.

[13] J. Chesson. A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation. *Journal of Applied Probability*, 13(4):795–797, 1976.

[14] J. R. Cockrell and M. F. Folstein. Mini-mental state examination. *Principles and practice of geriatric psychiatry*, pages 140–141, 2002.

[15] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 509–516. AAAI Press, 1998.

[16] Peter G. Doyle and J. Laurie Snell. *Random Walks and Electric Networks*. Mathematical Association of America, Washington, DC, 1984.

[17] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the International ACM SIGIR Conference on Research and Development In Information Retrieval*, pages 595–602, 2008.

[18] Rick Durrett. *Essentials of stochastic processes*. Springer texts in statistics. Springer, New York, second edition, 2012.

[19] Eyal Even-dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 255–270, 2002.

[20] Paul Flory. *Principles of polymer chemistry*. Cornell University Press, 1953.

[21] A. Fog. Calculation methods for Wallenius' noncentral hypergeometric distribution. *Communications in Statistics - Simulation and Computation*, 37(2):258–273, 2008.

[22] J. A. Glasdjo, C. C. Schuman, J. D. Evans, G. M. Peavy, S. W. Miller, and R. K. Heaton. Normas for letter and category fluency: Demographic corrections for age, education, and ethnicity. *Assessment*, 6(2):147–178, 1999.

[23] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. pages 1019–1028, New York, NY, USA, July 2010. Max-Planck-Gesellschaft, ACM Press.

[24] JoaquÃ-n GoÃ±i, Gonzalo Arrondo, Jorge Sepulcre, IÃ±igo Martincorena, Nieves VÃ©lez de MendizÃ¡bal, Bernat Corominas-Murtra, BartolomÃ© Bejarano, Sergio Ardanza-Trevijano, Herminia Peraita, Dennis P. Wall, and Pablo Villoslada. The semantic organization of the animal category: evidence from semantic verbal fluency and network theory. *Cognitive Processing*, 12(2):183–196, 2011.

[25] Thomas L. Griffiths, Mark Steyvers, and Alana Firl. Google and the mind: Predicting fluency with pagerank. *Psychological Science*, 18(12):1069–1076, 2007.

[26] P. J. Gruenewald and G. R. Lockhead. The free recall of category examples. *Journal of Experimental Psychology: Human Learning and Memory*, 6(3):225–241, 1980.

[27] Linhui Hao, Quiling He, Zhishi Wang, Mark Craven, Michael A. Newton, and Paul and Ahlquist. Limited agreement of independent rnai screens for virus-required host genes owes more to false-negative than false-positive factors. *PLoS Computational Biology*, 9(9):e1003235, 2013.

[28] Linhui Hao, Akira Sakurai, Tokiko Watanabe, Ericka Sorensen, Chairul A Nidom, Michael A Newton, Paul Ahlquist, and Yoshihiro Kawaoka. Drosophila rnai screen identifies host genes important for influenza virus replication. *Nature*, 454(7206):890–893, 2008.

[29] V. Heidrich-Meisner and Christian Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 401–408, 2009.

[30] Nancy M. Henley. A psychological study of the semantics of animal terms. *Journal of Verbal Learning and Verbal Behavior*, 8(2):176–184, April 1969.

[31] Thomas T. Hills, Peter M. Todd, and Michael N. Jones. Optimal foraging in semantic memory. *Psychological Review*, pages 431–440, 2012.

[32] J. R. Hodges, P. Garrard, R. Perry, K. Patterson, T. Bak, and C. Gregory. The differentiation of semantic dementia and frontal lobe dementia from early alzheimer's disease: a comparative neuropsychological study. *Neuropsychology*, 13:31–40, 1999.

[33] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[34] Kevin G. Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil' ucb : An optimal exploration algorithm for multi-armed bandits. In *Proceedings of The 27th Conference on Learning Theory*, pages 423–439, 2014.

[35] Kevin G. Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *48th Annual Conference on Information Sciences and Systems, CISS 2014, Princeton, NJ, USA, March 19-21, 2014*, pages 1–6, 2014.

[36] Pooria Joulani, AndrÃ¡s GyÃ¶rgy, and Csaba SzepesvÃ¡ri. Online learning under delayed feedback. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1453–1461, 2013.

[37] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 511–518, 2010.

[38] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 655–662, 2012.

[39] Emilie Kaufmann, Olivier CappÃ©, and AurÃ©lien Garivier. On the complexity of best arm identiï¬'cation in multi-armed bandit models. Dec. 2014.

[40] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[41] B. Liu, X. Li, W.S. Lee, and P.S. Yu. Text classification by labeling words. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 425–430. AAAI Press, 2004.

[42] A. U. Monsch, M. W. Bondi, N. Butters, D. P. Salmon, R. Katzman, and L. J. Thal. Comparisons of verbal fluency tasks in the detection of dementia of the alzheimer type. *Archives of Neurology*, 49(12):1253–1258, 1992.

[43] Alexandru Niculescu-Mizil. Multi-armed bandits with betting. In *COLT'09 Workshop on On-line Learning with Limited Feedback*, 2009.

[44] Andrew Moore Oded Maron. Hoeffding races: Accelerating model selection search for classification and function approximation. In Jack D. Cowan, G. Tesauro & J. Alspector, editor, *Advances in Neural Information Processing Systems (NIPS)*, pages 59–66, 340 Pine Street, 6th Fl., San Francisco, CA 94104, April 1994. Morgan Kaufmann.

[45] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up: Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86. ACL, 2002.

[46] F. Pasquier, F. Lebert, L. Grymonprez, and H. Petit. Verbal fluency in dementia of frontal lobe type and dementia of Alzheimer type. *Journal of Neurology*, 58(1):81–84, 1995.

[47] A. D. Pellegrini and Jeffrey D. Long. A longitudinal study of bullying, dominance, and victimization during the transition from primary school through secondary school. *British Journal of Developmental Psychology*, 20(2):259–280, 2002.

[48] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July 1992.

[49] T. T. Rogers, A. Ivanoiu, K. Patterson, and J. R. Hodges. Semantic memory in Alzheimer's disease and the frontotemporal dementias: a longitudinal study of 236 patients. *Neuropsychology*, 20(3):319–335, 2006.

[50] T.T. Rogers, A. Ivanoiu, K. Patterson, and J.R. Hodges. Semantic memory in alheimer's disease and the fronto-temporal dementias: A longitudinal study of 236 patients. *Neuropsychology*, 20(3):319–335, 2006.

[51] A. Rosser and J.R. Hodges. Initial letter and semantic category fluency in alzheimer's disease, huntington's disease, and progressive supranuclear palsy. *journal of Neurology, Neurosurgery, and Psychiatry*, 57:1389–1394, 1994.

[52] D. Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

[53] B. Settles. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1467–1478. ACL, 2011.

[54] A.K. Troyer, M. Moscovitch, G. Winocur, M. Alexander, and D. Stuss. Clustering and switching on verbal fluency: The effects of focal fronal- and temporal-lobe lesions. *Neuropsychologia*, 36(6), 1998.

[55] K.T. Wallenius. *Biased Sampling: The Non-Central Hypergeometric Probability Distribution.* PhD thesis, Department of Statistics, Stanford University, 1963.

[56] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, KDD, pages 326–333, New York, NY, USA, 2004. ACM.

[57] Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. Learning from bullying traces in social media. In *Proceedings of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), pages 656–666. The Association for Computational Linguistics, 2012.