Methods for Reducing the Spread of Misinformation on the Web

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

MSc. Seyed Behzad Tabibian

aus Teheran/Iran

Tübingen

2022

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	12.09.2022
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Bernhard Schölkopf
2. Berichterstatter:	Prof. Dr. Philipp Hennig

Acknowledgments

First and foremost, I want to thank my advisors, Prof. Dr. Bernhard Schölkopf and Dr. Manuel Gomez-Rodrgiuez. They have been incredibly supportive of my work and enabled me to pursue my passion in the field of Machine Learning. Prof. Schölkopf inspired me to pursue research in Machine Learning during his lectures at Machine Learning Summer School in 2012. His lectures presented the field of Machine Learning research as a rigorous tool to understand and explain our observations in the world instead of merely a predictive tool. Dr. Gomez-Rodrgiuez enabled me to build my research skills in the particular area of Point Processes and control which I found fascinating.

My Ph.D. work would have been impossible without collaboration from incredible colleagues. I want to thank Prof. Isabel Valera, who helped me find my path in the early stages of my Ph.D. Similarly, Dr. Mehrdad Farajtabar and Prof. Le Song were kind to host me at Georgia Tech and support me in the early stages of my research. I am also indebted to Dr. Utkarsh Upadhyay, Dr. Vicenç Gomez, Dr. Abir De, Dr. Ali Zarezade, and Dr. Adish Singla for the many stimulating conversations and collaborations. Working with these people helped me become a better scientist, communicator, collaborator, and friend.

During my Ph.D., I also had the unique opportunity to assist in organizing the Neurips 2016 conference with Prof. Isabelle Guyon and Prof. Ulrike Von Luxburg. This opportunity enabled me to understand better the unique challenges of organizing Neurips, the flagship conference on Machine Learning, during the surge of interest in this field.

I would also like to thank a group of people I can proudly call friends for their help and support over the years: Dr. Shervin Safavi, Dr. Krikamol Muandet, Dr. Mijung Park, and many others. Completing a Ph.D. is an endeavor filled with many failures before reaching success. For me, this journey would have been impossible without the immense support of my family, and in particular, my wife.

Dedicated to the angel in my life, Fereshteh.

Therefore the seeker after the truth is not one who studies the writings of the ancients and, following his natural disposition, puts his trust on them, but rather the one who suspects his faith in them and questions what he gathers from them, the one who submits to argument and demonstration, and not to the sayings of a human being whose nature is fraught with all kinds of imperfection and deficiency. Thus the duty of man who investigates the writings of scientists, if learning the truth is his goal, is to make himself an enemy of all that he reads, and, applying his mind to the core and margins of its content, attack it from every side. He should also suspect himself as he performs his critical examination of it, so that he may avoid falling into either prejudice or leniency.

Ibn al Haytham 965-1040

Summary

The significant growth of the internet over the past thirty years reduced the cost of access to information for anyone who has unfettered access to the internet. During this period, internet users were also empowered to create new content that could instantly reach millions of people via social media platforms like Facebook and Twitter. This transformation broke down the traditional ways mass-consumed content was distributed and ultimately ushered in the era of citizen journalism and freeform content. The unrestricted ability to create and distribute information was considered a significant triumph of freedom and liberty. However, the new modes of information exchange posed new challenges for modern societies, namely trust, integrity, and the spread of misinformation.

Before the emergence of the Internet, newsrooms and editorial procedures required minimum standards for the published information; today, such requirements are not necessary when posting content on social media platforms. This change led to the proliferation of information that attracts attention but lacks integrity and reliability.

There are currently two broad approaches to solving the problem of information integrity on the internet; first, the revival of trusted and reliable sources of information; second, creating new mechanisms for increasing the quality of information published and spread on major social media platforms. These approaches are still in their infancy, each having its pros and cons. In this thesis, we explore the latter and develop modern machine learning methods that can help identify (un)reliable information and their sources, efficiently prioritize content requiring human fact-checking at scale, and ultimately minimize their harm to the end-users by improving the quality of the news-feeds that users access.

The first chapter of this thesis leverages the collaborative dynamics of content creation on *Wikipedia* to extract a grounded measure of information and source reliability. Later in the same chapter, we develop a method capable of modifying ranking algorithms used widely on social media platforms such as Facebook and Twitter to minimize the long-term harm posed by the spread of misinformation.

In the second chapter, we study the problem of reliability in the context of the reviewing process of Neurips conference. This chapter provides an extensive analysis of how the design of reviewing process can impact the quality of reviews posted for the conference submissions.

In the final chapter, we study the problem of human learning via spaced repetition. Spaced-repetition algorithms are often used when individuals try to learn new languages or memorize content for examination. In this chapter, we develop a method to schedule practice sessions efficiently. In a comprehensive series of experiments, we show that learners on *Duolingo*[1] that have practice patterns similar to our proposed solution tend to learn faster and with less effort.

[1] Duolingo is a language-learning website where users practice vocabulary, grammar, and pronunciation using spaced-repetition method.

Contents

Chapter 1

Problem and Motivation

We begin by giving a motivation for the thesis and a brief outline of the subsequent chapters.

1.1 Motivation

Over the past thirty years, the internet has become the cornerstone of modern society. Among many tools invented, social media platforms such as Facebook and Twitter entirely transformed how individuals connect and communicate.

Social media started as a tool to connect friends and family members to share photos and their significant moments of life. Soon after, these tools replaced almost every traditional outlet for collecting news and information and became the primary source of news. This change shifted the primary source of news from conventional newspapers to individual internet users who could now share news and opinions on social media and reach millions of people [34]. This transformation enabled internet users to reach millions of individuals, coordinate social movements, run more effective election campaigns, and express or hear opinions that were not heard until then.

However, after the wide adoption of social platforms such as Facebook and Twitter, internet users faced a challenge, i.e., reliability and trust in the published content on the web. Unlike traditional newsrooms where editorial standards are often observed, posting content on social media does not require following any standards. This led to the emergence of a large volume of unreliable information, which has posed a significant problem in many domains like elections [12] or public health[74].

Tackling misinformation on social media became a priority, as the operators of platforms such as Facebook and Twitter were heavily criticized for the consequences of the spread of misinformation on their platforms [22, 96]. The increasing volume of misinformation on social platforms also degraded the trust of their users and reduced engagement. Almost all major platforms, such as Twitter and Facebook, created Trust and Safety teams within their organizations responsible for tackling the problem of misinformation.

The primary approach for tackling misinformation broadly falls into the following two steps:

- **Identification:** identifying instances of misinformation in posted links, posts, and photos.
- **Enforcement:** taking action on a piece of content, such as removing a link from the platform, based on available information.

We briefly explore both steps below.

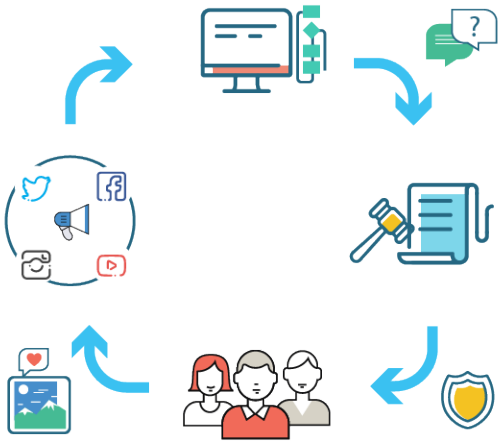


Figure 1.1: Trust and Safety cycle in reducing the impact of misinformation in social media. Users (bottom) post content on social media platforms such as Facebook. These platforms utilize a variety of data-driven algorithmic solutions to identify suspicious content that might be misinformation (top). Then they pass them on to third party fact-checkers who would provide details about the content and check if it is misinformation or not. The platform uses this feedback to enforce various policies, including banning the users or removing specific content from the platform (left).

Identifying Misinformation

The first step to reduce the spread of misinformation on social platforms is to identify instances of misinformation. Given the significant volume of content circulating on social media platforms, any solution that will tackle this problem needs to be highly scalable and data-driven. Machine Learning has proven to meet these criteria; however, labeled training data is required, similar to any other Machine Learning-based solution. Since the emergence of misinformation on the web, several independent organizations, namely *Snopes*[1], *ProPublica*[2], and associations such as *Poynter Network*[3] have started to fact-check online content. These websites regularly publish fact-checked articles about various content posted on social media. The social platforms also work with third-party fact-checkers to fact-check content that may be labeled as misinformation.

Social platforms use fact-checked articles to create large-scale datasets and Machine Learning algorithms to identify other instances of misinformation on their platform.

Enforcement

The ultimate objective of social platforms is not only to identify misinformation but to reduce its impact as early as possible. They achieve this by enforcing specific policies such as removing content that violates their terms of service, locking accounts that produce such content, or

[1] https://www.snopes.com/
[2] https://www.propublica.org/
[3] https://www.poynter.org/ifcn/

limiting distribution of violating content by modifying the ranking algorithms.

There are many trade-offs involved in designing and enforcing policies related to misinformation. For example, a piece of content can be removed only if a human fact-checker has identified the content as misinformation. This would require identifying potential instances of misinformation as early as possible so the content can be fact-checked before it is widely circulated.

In some cases, the platform may reduce the distribution of the content that is flagged as likely misinformation until the fact-checking is completed. In this case, modifying the ranking algorithms that distribute the content to the users is necessary.

This thesis explores methods that address the problem in both steps; we discuss how to formalize the issues, describe the trade-offs involved, develop algorithms, and eventually evaluate such algorithms using data from social platforms.

Information Quality in Other Domains

The question of the integrity of online platforms is not limited to social media. For many years, the Machine Learning research community has grappled with similar challenges in their conference reviewing processes.

In 2016, the flagship conference on Machine Learning, Neural Information Processing Systems received 2,425 paper submissions; only two years earlier, this number was 1,678 papers, equal to a 40% growth in the number of submissions. The rapid growth of publications in Machine Learning research presents significant challenges for the conference organizers that seek to publish high-quality research.

One way to address this concern is to develop scalable statistical methods to monitor the integrity of every step involved in the reviewing processes. This thesis presents a comprehensive analysis of the reviewing process with this aim in mind. We show how to use a wide range of statistical methods to ensure every paper submitted to the conference has equal opportunity in the reviewing process.

Human Learning and Spaced-repetition Algorithms

We conclude this thesis by studying a concept known as learning through spaced repetition. This method underpins many software platforms that assist their users with learning new topics such as languages. This chapter does not relate directly to the theme of earlier chapters. However, we used the methodology developed in this chapter in a different research project on efficient scheduling of potential misinformation content for third party fact-checkers. This research project is listed at the end of this chapter.

Spaced repetition is a theory suggesting that repetitive training with long intervals between sessions helps to form long-term memory. Hermann Ebbinghaus demonstrated in his seminal work [33] that spaced repetition is superior to training that includes short inter-trial intervals (massed training or massed learning) in terms of its ability to promote memory formation. It is a learning technique that is performed with flashcards. Through the process of repeating the flash cards, newly introduced and more difficult flashcards turn up more frequently, while older and less difficult flashcards are shown less regularly to exploit the psychological spacing effect. The

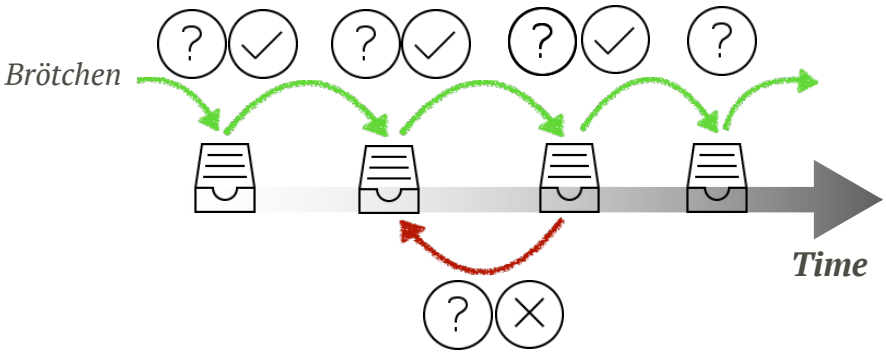


Figure 1.2: Depiction of the spaced-repetition learning process for acquiring new vocabulary. A learner with the aim of memorizing the translation of the word "Brötchen" practices the word in different time intervals associated with how well that word has been recalled. Every time the learner recalls the word correctly, the word is practiced at in the interval associated with a stronger recall pattern, and every time he does not recall the word correctly, the word is practiced at an interval associated with a weaker recall pattern.

use of spaced repetition has been proven to increase the rate of learning. A simplified version of the learning process through spaced-repetition is presented in Figure 1.2.

Although the principle is helpful in many contexts, spaced repetition is commonly applied in contexts where a learner must acquire many items and retain them indefinitely in memory. It is, therefore, well suited for the problem of vocabulary acquisition in second language learning. Several spaced repetition software have been developed to aid the learning process.

Recently, researchers were able to build predictive models of the difficulty of items that learners study. However, it remained an open problem whether scheduling the reviewing sessions based on user performance, and the available data was also possible. We present a novel method to solve this problem. The solution enables users to learn more rapidly and efficiently.

1.2 Conclusion and Future works

Access to accurate and reliable information is critical for a well-organized society. With the emergence of social media as the primary medium for exchanging news, trust and reliability of the content circulating on platforms such as Facebook and Twitter have become critical. In this thesis, we present several Machine-Learning methods to identify and reduce the spread of misinformation on the web. However, there are yet many challenges that future works could explore:

- **Deep Fakes** Deep Fakes are synthetically created images or videos that replace a person with someone else's likeness. Deep Fakes use Machine Learning to manipulate or generate visual and audio content that can easily deceive the audience. This new type of misinformation can give the impression that a notable person has said or done something that is not real. The rapid spread of this kind of misinformation on social media (going viral) can potentially mislead millions of people in a short span of time. To tackle this

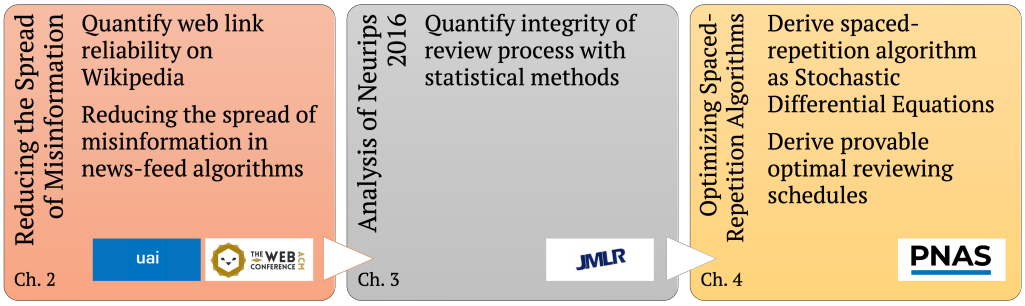


Figure 1.3: The outline of this thesis, each box corresponds to one chapter, the content of every chapter is summarized in the corresponding box.

problem, platforms such as Facebook have released new datasets[4].

- **Enforcement** There is a growing concern about how social media platforms enforce various policies related to misinformation. This is further complicated by different countries requiring platforms to enforce local laws related to misinformation as well [5]. Algorithmic and manually enforced methods for removing content and penalizing users that spread misinformation have also come under scrutiny and require more transparency[6].

1.3 Outline of the Thesis

The following section presents the outline of this thesis, the overview of the thesis is also depicted in Figure 1.3.

1. **Chapter 2** This chapter addresses the problem of information quality and integrity on the web. The first part develops a novel method to quantify information reliability using traces of user activities on platforms such as Wikipedia. The second part of this chapter addresses how to reduce the spread of misinformation by modifying ranking algorithms.

   The contents of this chapter are published in the following two publications:

   - **B. Tabibian**, I Valera, M Farajtabar, L Song, B Schölkopf, Manuel Gomez-Rodriguez Distilling information reliability and source trustworthiness from digital traces. *Proceedings of the 26th International Conference on World Wide Web, 2017*
   - **B Tabibian**, V Gomez, A De, B Schölkopf, M Gomez Rodriguez On the design of consequential ranking algorithms. *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR 124:171-180, 2020.*

2. **Chapter 3** This chapter looks at the quality of reviewing process at the Flagship Machine Learning conference, i.e., Neural Information Processing Systems. It explores how the

[4] https://ai.facebook.com/datasets/dfdc/

[5] https://www.poynter.org/ifcn/anti-misinformation-actions

[6] https://www.washingtonpost.com/technology/2022/06/22/facebook-oversight-board-annual-report-transparency/

process can impact the quality of reviewing procedures and suggests practical steps and tools for improving the conference reviewing process in the future.

- N B Shah*, **B Tabibian***, K Muandet, I Guyon, U Von Luxburg. Design and Analysis of the NIPS 2016 Review Process. *Journal of machine learning research (2018)* *equal contribution.

3. **Chapter 4** This chapter looks at the problem of optimal scheduling in spaced repetition algorithms. It develops a simple algorithm for scheduling items under study by a user. It also presents provable optimality guarantees, an extensive experimental analysis of real data, and explores how the same algorithm can be used with different cognitive memory models.

    - **B Tabibian**, U Upadhyay, A De, A Zarezade, B Schölkopf, M Gomez-Rodriguez. Enhancing human learning via spaced repetition optimization. *Proceedings of the National Academy of Sciences 116.10 (2019): 3988-3993.*

In addition to the materials covered in this thesis, I have been fortunate to be involved in other projects during my Ph.D. These publications include:

- J Kim, **B Tabibian**, A Oh, B Schölkopf, M Gomez-Rodriguez. Leveraging the crowd to detect and reduce the spread of fake news and misinformation, *Proceedings of the eleventh ACM international conference on web search and data mining. 2018.*

- M Khajehnejad *, **B Tabibian**, B Schölkopf, A Singla, M Gomez-Rodriguez.(*Work done during M Khajehnejad's internship at the Max Planck Institute for Software Systems.) *Under Review arXiv preprint arXiv:1905.09239*

- A Aghaeifar, J Zhou, R Heule, **B Tabibian**, B Schölkopf, F Jia, M Zaitsev, K Scheffle. A 32-channel multi-coil setup optimized for human brain shimming at 9.4 T *Magnetic Resonance in Medicine 83.2 (2020): 749-764.*

Chapter 2

Misinformation on the Web and Reducing its Impact in Social Media

Over the years, the Web has become a vast repository of information and knowledge about a rich variety of topics and real-world events – one much larger than anything we could hope to accumulate in conventional textbooks and traditional news media outlets. Unfortunately, due to its immediate nature, it also contains an ever-growing number of opinionated, inaccurate or false facts, urban legends and unverified stories of unknown or questionable origin, which are often refuted over time.[1][2] To overcome this problem, online knowledge repositories, such as *Wikipedia*, *Stack Overflow* and *Quora*, put in place different evaluation mechanisms to increase the reliability of their content. These mechanisms can be typically classified as:

I. **Refutation:** A user refutes, challenges or questions a statement contributed by another user or a piece of content originated from an external web source. For example, in *Wikipedia*, an editor can refute a questionable, false or incomplete statement in an article by removing it.

II. **Verification:** A user verifies, accepts or supports a statement contributed by another user or a piece of content originated from an external web source. For example, in *Stack Overflow*, a user can accept or up-vote the answers provided by other users.

However, these evaluation mechanisms only provide noisy measurements of the reliability of information and the trustworthiness of the information sources. Can we leverage these noisy measurements, often biased, to distill a robust, unbiased and interpretable measure of both notions?

In this section, we argue that the *temporal traces* left by these noisy evaluations give cues on the reliability of the information and the trustworthiness of the sources. For example, while statements provided by an untrustworthy user will be often spotted by other users as unreliable and refuted quickly, statements provided by trustworthy users will be refuted less often. However, at a particular point in time, a statement about a complex, controversial or time evolving topic, story, or more generally, *knowledge item*, may be refuted by other users independently of the source. In this case, quick refutations will not reflect the trustworthiness of the source but the intrinsic unreliability of the knowledge item the statement refers to.

To explore this hypothesis, we propose a temporal point process modeling framework of refutation and verification in online knowledge repositories, which leverages the above mentioned temporal traces to obtain a meaningful measure of both information reliability and source

[1] http://www.snopes.com

[2] http://www.factcheck.org

trustworthiness. The key idea is to disentangle to what extent the temporal information in a statement evaluation (verification or refutation) is due to the intrinsic unreliability of the involved knowledge item, or to the trustworthiness of the source providing the statement. To this aim, we model the times at which statements are added to a knowledge item as a counting process, whose intensity captures the temporal evolution of the reliability of the item—as a knowledge item becomes more reliable, it is less likely to be modified. Moreover, each added statement is supported by an information source and evaluated by the users in the knowledge repository at some point after its addition time. Here, we model the evaluation time of each statement as a survival process, which starts at the addition time of the statement and whose intensity captures both the trustworthiness of the associated source and the unreliability of the involved knowledge item.

For the proposed model, we develop an efficient method to find the optimal model parameters that jointly maximize the likelihood of an observed set of statement addition and evaluation times. This efficient algorithm allows us to apply our framework to $\sim$19 million addition and refutation events in $\sim$100 thousand *Wikipedia* articles and $\sim$1 million addition and verification events in $\sim$378 thousand questions in *Stack Overflow*. Our experiments show that our model accurately predicts whether a statement in a *Wikipedia* article (an answer in *Stack Overflow*) will be refuted (verified), it provides interpretable measures of source trustworthiness and information reliability, and yields interesting insights:[3]

I. Most active sources are generally more trustworthy, however, trustworthy sources can be also found among less active ones.

II. Changes on the reliability of a *Wikipedia* article over time, as inferred by our framework, match external noteworthy events.

III. Questions and answers in *Stack Overflow* cluster into groups with similar levels of difficulty and popularity.

Related work. The research area most closely related to ours is on truth discovery and source trustworthiness. The former aims at resolving conflicts among noisy information published by different sources and the latter assesses the quality of a source by means of its ability to provide correct factual information. Most previous works have studied both problems together and measure the trustworthiness of a source using link-based measures [14, 44], information retrieval based measures [122], accuracy-based measures [27, 28, 123], content-based measures [2], and graphical model analysis [85, 124, 129, 130]. A recent line of work [68, 71, 83, 117] also considers scenarios in which the truth may change over time. However, previous work typically shares one or more of the following limitations, which we address in this work: (i) they only support knowledge triplets (subject, predicate, object) or structured knowledge; (ii) they assume there is a *truth*, however, a statement may be under discussion when a source writes about it; and, (iii) they do not distinguish between the unreliability of the knowledge item to which the statement refers and the trustworthiness of the source.

Temporal point processes have been previously used to model information cascades [41, 31, 23], social activity [37, 55, 35], badges [30], network evolution [50, 36], opinion dynamics [24], or product competition [115]. However, to the best of our knowledge, the present work is the

[3] We will release code for our inference method, datasets and a web interface for exploring results at http://btabibian.com/projects/reliability.

first that leverages temporal point processes in the context of information reliability and source trustworthiness.

Background on Temporal Point Processes

A temporal point process is a stochastic process whose realization consists of a list of discrete events localized in time, $\{t_i\}$ with $t_i \in \mathbb{R}^+$ and $i \in \mathbb{Z}^+$. Many different types of data produced in social media and the Web can be represented as temporal point processes [24, 36, 115]. A temporal point process can be equivalently represented as a counting process, $N(t)$, which records the number of events up to time t, and can be characterized via its conditional intensity function — a stochastic model for the time of the next event given all the times of previous events. More formally, the conditional intensity function $\lambda^*(t)$ (intensity, for short) is given by

$$\lambda^*(t)dt := \mathbb{P}\{\text{event in } [t, t+dt) | \mathcal{H}(t)\} = \mathbb{E}[dN(t)|\mathcal{H}(t)],$$

where $dN(t) \in \{0,1\}$ denotes the increment of the process, $\mathcal{H}(t)$ denotes the history of event times $\{t_1, t_2, \ldots, t_n\}$ up to but not including time t, and the sign * indicates that the intensity may depend on the history. Then, given a time $t_i \geqslant t_{i-1}$, we can also characterize the conditional probability that no event happens during $[t_{i-1}, t_i)$ and the conditional density that an event occurs at time t_i as $S^*(t_i) = \exp(-\int_{t_{i-1}}^{t_i} \lambda^*(\tau)\, d\tau)$ and $f^*(t_i) = \lambda^*(t_i) S^*(t_i)$, respectively. Furthermore, we can express the log-likelihood of a list of events $\{t_1, t_2, \ldots, t_n\}$ in an observation window $[0, T)$ as [1]

$$\mathcal{L} = \sum_{i=1}^{n} \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau)\, d\tau. \tag{2.1}$$

This simple log-likelihood will later enable us to learn the parameters of our model from observed data. Finally, the functional form of the intensity $\lambda^*(t)$ is often designed to capture the phenomena of interests. Some useful functional forms we will use later are [1]:

I. **Poisson process.** The intensity is assumed to be independent of the history $\mathcal{H}(t)$, but it can be a time-varying function, *i.e.*, $\lambda^*(t) = g(t) \geqslant 0$;

II. **Hawkes Process.** The intensity models a mutual excitation between events, *i.e.*,

$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} \kappa_\omega(t - t_i), \tag{2.2}$$

where $\kappa_\omega(t)$ is the triggering kernel, $\mu \geqslant 0$ is a baseline intensity independent of history. Here, the occurrence of each historical event increases the intensity by a certain amount determined by the kernel and the weight $\alpha \geqslant 0$, making the intensity history dependent and a stochastic process by itself; and,

III. **Survival process.** There is only one event for an instantiation of the process, *i.e.*,

$$\lambda^*(t) = g(t)(1 - N(t)), \tag{2.3}$$

where $\lambda^*(t)$ becomes 0 if an event already happened before t and $g(t) \geqslant 0$.

2.1 Proposed Model of Information Reliability on the Web

In this section, we formulate our modeling framework of verification and refutation in knowledge repositories, starting with the data representation it uses.

Data representation. The digital traces generated during the construction of a knowledge repository can be represented using the following three entities: the *statements*, which are associated to particular *knowledge items*, and the *information sources*, which support each of the statements. More specifically:

— *An information source* is an entity that supports a statement in a knowledge repository, *i.e.*, the web source an editor uses to support a paragraph in *Wikipedia*, the user who posts an answer on a Q&A site, or the software developer who contributes a piece of code in *Github*. We denote the set of information sources in a knowledge repository as $\mathcal{S}$.

— *A statement* is a piece of information contributed to a knowledge repository, which is characterized by its addition time t, its evaluation time τ, and the information source $s \in \mathcal{S}$ that supports it. Here, we represent each statement as the triplet

$$e = (\overset{\text{source}}{\overset{\downarrow}{s}}, \overset{\text{addition time}}{\overset{\uparrow}{t}}, \overset{\text{evaluation time}}{\overset{\downarrow}{\tau}}), \tag{2.4}$$

where an evaluation may correspond either to a verification or refutation.[4] Moreover, if a statement is *never* refuted or verified, then we set $\tau = \infty$.

— *A knowledge item* is a collection of statements. For example, a knowledge item corresponds to an article in *Wikipedia*; to a question and its answer(s) in a Q&A site; or to a software project on *Github*. Here, we gather the history of the d-th knowledge item, $\mathcal{H}_d(t)$, as the set of statements added to the knowledge item d up to but not including time t, *i.e.*,

$$\mathcal{H}_d(t) = \{e_i | t_i < t\}. \tag{2.5}$$

In most knowledge repositories, one can recover the source, addition time, and evaluation time of each statement added to a knowledge item. For example, in *Wikipedia*, there is an edit history for each *Wikipedia* article; on Q&A sites, all answers to a question are recorded; and, in *Github*, there is a version control mechanism to keep track of all changes.

Generative process for knowledge evolution. Our hypothesis is that the temporal information related to statement additions and evaluations reflects both the reliability of knowledge items and the trustworthiness of information sources. More specifically, our intuition is as follows:

I. A reliable knowledge item should be stable in the sense that new statement addition will be rare, and it is less likely to be changed compared to unreliable items. Such notion of reliability should be reflected in the statement addition process—as a knowledge item becomes more reliable, the number of statement addition events within a unit of time should be smaller.

II. A trustworthy information source should result in statements which are verified quickly and refuted rarely. Such notion of trustworthiness should be therefore reflected in its

[4] For clarity, we assume that a knowledge repository either uses refutation or verification. However, our model can be readily extended to knowledge repositories using both.

statement evaluation time—the more trustworthy an information source is, the shorter (longer) the time it will take to verify (refute) its statements.

In our temporal point process modeling framework, we build on the above intuition to account for both information reliability and source trustworthiness. In particular, for each knowledge item, we model the statement addition times $\{t_i\}$ as a counting process whose intensity directly relates to the reliability of the item—as a knowledge item becomes more reliable, it is less likely to be changed. Moreover, each addition time t_i is marked by its information source s_i and its evaluation time τ_i, which in turn depends on the source trustworthiness and also have an impact on the overall reliability of the knowledge item—the verification (refutation) of statements supported by trustworthy sources result in an increase (decrease) of the reliability of the knowledge item.

More in detail, for each knowledge item d, we represent the statement addition times $\{t_i\}$ as a counting process $N_d(t)$, which counts the number of statements that have been added up to but not including time t. Thus, we characterize the statement addition process using its corresponding intensity $\lambda_d^*(t)$ as

$$\mathbb{E}[dN_d(t)|\mathcal{H}_d(t)] = \lambda_d^*(t)dt, \tag{2.6}$$

which captures the evolution of the reliability of the knowledge item over time. Here, the smaller the intensity $\lambda^*(t)$, the more reliable the knowledge item at time t. Moreover, since a knowledge item consists of a collection of statements, the overall reliability of the knowledge item will also depend on the individual reliability of its added statements through their evaluations—the verification (refutation) of statements may result in an increase (decrease) of the reliability of the knowledge item, leading to an inhibition (increase) in the intensity of the statement additions to the knowledge item.

Additionally, every time a statement i is added to the knowledge item d, the corresponding information source $s_i \in \mathcal{S}$ is sampled from a distribution $p(s|d)$ and the evaluation time τ_i is sampled from a survival process, which we represent as a binary counting process $N_i(t) \in \{0,1\}$, in which $t = 0$ corresponds to the time in which the statement is added and becomes one when $t = \tau_i - t_i$. Here, we characterize this survival process using its corresponding intensity $\mu_i^*(t)$ as

$$\mathbb{E}[N_i(t)|\mathcal{H}_d(t)] = \mu_i^*(t)dt, \tag{2.7}$$

which captures the temporal evolution of the reliability of the i-th statement added to the knowledge item. Here, the smaller the intensity $\mu_i^*(t)$, the shorter (longer) time it will take to verify (refute) it. This intensity will depend, on the one hand, on the current intrinsic reliability of the corresponding knowledge item and, on the other hand, on the trustworthiness of the source supporting the statement.

Next, we formally define the functional form of the intensities $\lambda_d^*(t)$ and $\mu_i^*(t)$, and the source distribution $p(s|d)$.

Knowledge item reliability. For each knowledge item d, we consider the following form for

its reliability function, or equivalently, its statement addition intensity:

$$\lambda_d(t) = \underbrace{\sum_j \phi_{d,j} k(t - t_j)}_{\text{item intrinsic reliability}} + \underbrace{\sum_{e_i \in \mathcal{H}_d(t)} \mathbf{w}_d^\top \boldsymbol{\gamma}_{s_i} g(t - \tau_i)}_{\text{effect of past evaluations}}. \tag{2.8}$$

In the above expression, the first term is a mixture of kernels $k(t)$ accounting for the temporal evolution of the intrinsic reliability of a knowledge item over time, and the second term accounts for the effect that previous statement evaluations have on the overall reliability of the knowledge item. Here, $\mathbf{w}_d$ and $\boldsymbol{\gamma}_{s_i}$ are L-length vectors whose elements indicate, respectively, the weight (presence) of each topic in the knowledge item and the per-topic influence of past evaluations of statements backed by source s_i. Finally, the function $g(t)$ is a nonnegative triggering kernel, which models the decay of the influence of past evaluations over time. If the evaluation is a refutation then we assume $\boldsymbol{\gamma}_{s_i} \geq 0$, since a refuted statement typically decreases the reliability of the knowledge item and thus triggers the arrival of new statements to replace it. If the evaluation is a verification, we assume $\boldsymbol{\gamma}_{s_i} \leq 0$, since a verified statement typically increases the reliability of the knowledge item and thus inhibits the arrival of new statements to the knowledge item. As a consequence, the above design results in an "evaluation aware" process, which captures the effect that previous statement evaluations exert on the reliability of a knowledge item.

Statement reliability. As discussed above, every statement addition event e_i is *marked* with an evaluation time τ_i, which we model using a survival process. The process is "statement driven" since it starts at the time when the statement addition event occurs and, within the process, $t = 0$ corresponds to the addition time of the statement. For each statement i, we adopt the following form for the statement reliability or, equivalently, for the intensity associated with its survival process:

$$\mu_i(t) = (1 - N_i(t)) \Big[\underbrace{\sum_j \beta_{d,j} k(t + t_i - t_j)}_{\text{item intrinsic reliability}} + \underbrace{\mathbf{w}_d^\top \boldsymbol{\alpha}_{s_i}}_{\substack{\text{source} \\ \text{trustworthiness}}} \Big]. \tag{2.9}$$

In the above expression, the first term is a mixture of kernels $k(t)$ accounting for the temporal evolution of the intrinsic reliability of the corresponding knowledge item d and the second term captures the trustworthiness of the source that supports the statement. Here, $\mathbf{w}_d$ and $\boldsymbol{\alpha}_{s_i}$ are L-length nonnegative vectors whose elements indicate, respectively, the weight (presence) of each topic in the knowledge item d and the trustworthiness of source s_i in each topic. Since the elements $\mathbf{w}_d$ sum up to one, the product $\mathbf{w}_d \boldsymbol{\alpha}_{s_i}$ can be seen as the average trustworthiness of the source s_i in the knowledge item d. With this modeling choice, the higher the parameter $\boldsymbol{\alpha}_{s_i}$, the quicker the evaluation of the statement. Then, if the evaluation is a refutation, a high value of $\boldsymbol{\alpha}_{s_i}$ implies low trustworthiness of the source s_i. In contrast, if it is a verification, a high value of $\boldsymbol{\alpha}_{s_i}$ implies high trustworthiness.

Finally, note that the reliability of a statement, as defined in Eq. 2.9, reflects how quickly (slowly) it will be refuted or verified, and the reliability of a knowledge item, as defined in Eq. 2.8, reflects how quickly (slowly) new statements are added to the knowledge item.

Selection of source. The source popularity $p(s|d)$ typically depends on the topics contained in

the knowledge item d. Therefore, we consider the following form for the source distribution:

$$p(s|d) = \sum_{\ell=1}^{L} w_{d,\ell} p(s|\ell), \tag{2.10}$$

where $w_{d,\ell}$ denotes the weight of topic ℓ in knowledge item d and $p(s|\ell) \propto Multinomial(\boldsymbol{\pi}_\ell)$ is the distribution of the sources for topic ℓ, *i.e.*, the vector $\boldsymbol{\pi}_\ell$ contains the probability of each source to be assigned to a topic ℓ.

2.1.1 Parameter Estimation

In this section, we show how to efficiently learn the parameters of our model, as defined by Eqs. 2.8 and 2.9, from a set of statement addition and evaluation events. Here, we assume that the topic weight vectors $\mathbf{w}_d$ are given[5]. More specifically, given a set of sources $\mathcal{S}$ and a set of knowledge items $\mathcal{D}$ with histories $\{\mathcal{H}_1(T), \ldots, \mathcal{H}_{|\mathcal{D}|}(T)\}$, spanning a time period $[0, T)$, we find the model parameters $\{\boldsymbol{\pi}_\ell\}_{\ell=1}^{L}$, $\{\boldsymbol{\beta}_d\}_{d=1}^{|\mathcal{D}|}$, $\{\boldsymbol{\phi}_d\}_{d=1}^{|\mathcal{D}|}$, $\{\boldsymbol{\alpha}_s\}_{s=1}^{|\mathcal{S}|}$ and $\{\boldsymbol{\gamma}_s\}_{s=1}^{|\mathcal{S}|}$, by solving the following maximum likelihood estimation (MLE) problem

$$\begin{aligned} &\text{maximize } \mathcal{L}(\{\boldsymbol{\pi}_\ell\}, \{\boldsymbol{\beta}_d\}, \{\boldsymbol{\phi}_d\}, \{\boldsymbol{\alpha}_s\}, \{\boldsymbol{\gamma}_s\}) \\ &\text{subject to } \boldsymbol{\pi}_\ell \geqslant 0, \boldsymbol{\beta}_d \geqslant 0, \boldsymbol{\phi}_d \geqslant 0, \boldsymbol{\alpha}_s \geqslant 0, \mathbf{1}^T \boldsymbol{\pi}_\ell = 1 \end{aligned}$$

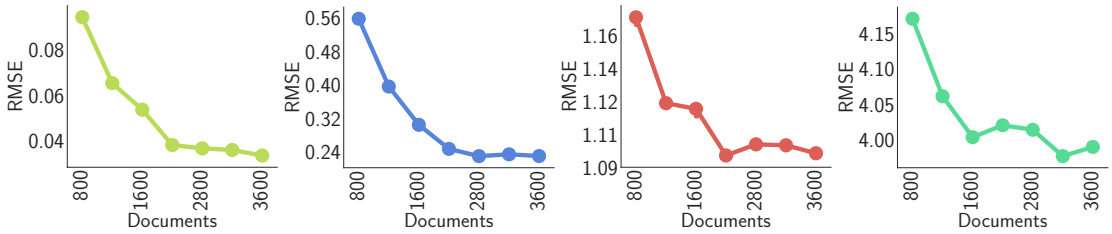
where the log-likelihood is given by

$$\begin{aligned} \mathcal{L} = &\sum_{d=1}^{|\mathcal{D}|} \sum_{i:e_i \in \mathcal{H}_d(T)} \log \underbrace{p(t_i|\mathcal{H}_d(t_i), \boldsymbol{\phi}_d, \{\boldsymbol{\gamma}_s\}, \mathbf{w}_d)}_{\text{statements additions}} + \sum_{d=1}^{|\mathcal{D}|} \sum_{i:e_i \in \mathcal{H}_d(T)} \log \underbrace{p(\Delta_i|t_i, \boldsymbol{\beta}_d, \{\boldsymbol{\alpha}_s\}, \mathbf{w}_d)}_{\text{statements evaluations}} \\ &+ \sum_{d=1}^{|\mathcal{D}|} \sum_{i:e_i \in \mathcal{H}_d(T)} \log \underbrace{p(s_i|\{\boldsymbol{\pi}_\ell\}, \mathbf{w}_d)}_{\text{sources popularity}}. \end{aligned} \tag{2.11}$$

In the above likelihood, the first term accounts for the times at which statements are added to the knowledge item, the second term accounts for the times at which statements are evaluated, and the third term accounts for the probability that source s_i is assigned to the statement addition event e_i. Since the first two terms correspond to likelihoods of temporal point processes, they can be computed using Eq. 2.1. The third term is simply given by $p(s_i|\{\boldsymbol{\pi}_\ell\}_{\ell=1}^{L}, \mathbf{w}_d) = \sum_{\ell=1}^{L} w_{d,\ell} \boldsymbol{\pi}_\ell(s_i)$, where $\boldsymbol{\pi}_\ell(s_i)$ denotes the s_i-th element of $\boldsymbol{\pi}_\ell$.

Remarkably, the above terms can be expressed as linear combinations of logarithms and linear functions or compositions of linear functions with logarithms and thus easily follow that the above optimization problem is jointly convex in all the parameters. Moreover, the problem can be decomposed into three independent problems, which can be solved in parallel obtaining local solutions that are in turn globally optimal. For knowledge repositories using refutation, *i.e.*, $\boldsymbol{\gamma}_s \geqslant 0$, we solve both the first and second problem by adapting the algorithm by Zhou et

[5] There are many topic modeling tools to learn the topic weight vectors $\mathbf{w}_d$.



(a) α (b) γ (c) β (d) ϕ

Figure 2.1: Performance of our model parameter estimation method on synthetic data in terms of root mean squared error (RMSE). The estimation becomes more accurate as we feed more knowledge items into our estimation procedure. However, since each new knowledge item increases the number of β and ϕ parameters, once the source parameter estimation becomes accurate enough, the estimation error for β and ϕ flattens.

al. [133]. For knowledge repositories using verification, *i.e.*, $\boldsymbol{\gamma}_s \leqslant 0$, we solve the first problem using cvxpy [26] and the second problem by adapting the algorithm by Zhou et al. [133]. In both cases, the third problem can be computed analytically as

$$\boldsymbol{\pi}_\ell(s) = \frac{\sum_{d=1}^{|\mathcal{D}|} \mathbf{w}_d(\ell)\hat{\boldsymbol{\pi}}_d(s)}{\sum_{d=1}^{|\mathcal{D}|}\sum_{s'=1}^{|\mathcal{S}|} \mathbf{w}_d(\ell)\hat{\boldsymbol{\pi}}_d(s')}, \tag{2.12}$$

where $\mathbf{w}_d(\ell)$ denotes the ℓ-th element of $\mathbf{w}_d$, and $\hat{\boldsymbol{\pi}}_d(s)$ is the probability that source s is assigned to a statement in knowledge item d. In particular, $\hat{\boldsymbol{\pi}}_d(s)$ can be computed as

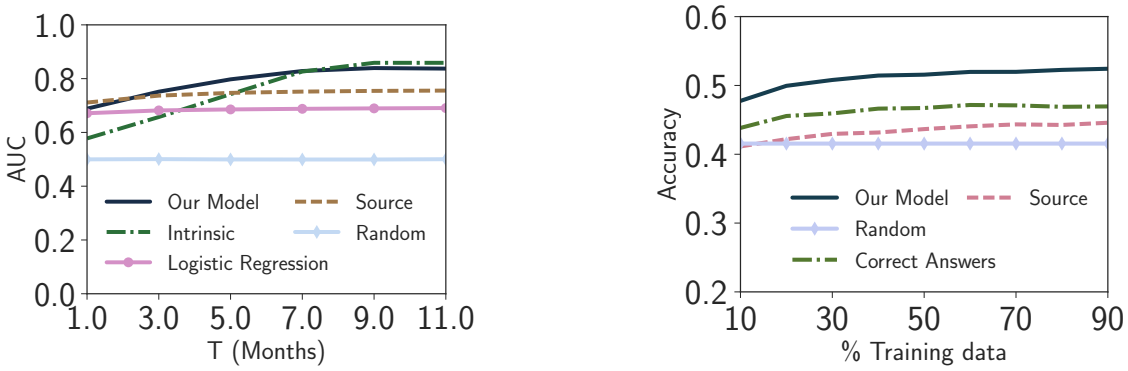
$$\hat{\boldsymbol{\pi}}_d(s) = \frac{n_{d,s}}{\sum_{s'=1}^{|\mathcal{S}|} n_{d,s'}}, \tag{2.13}$$

where $n_{d,s}$ is the number of statement addition events in the history of the knowledge item that are backed by source s, *i.e.*, $|\{e_i \in \mathcal{H}_d(T) | s_i = s\}|$. In practice, we found that adding a ℓ-1 penalty term on the parameters $\{\boldsymbol{\beta}_d\}$, *i.e.*, $\eta \sum_d ||\boldsymbol{\beta}_d||_1$, which we set by cross-validation, avoids overfitting and improves the predictive performance of our model.

2.1.2 Experiments on Synthetic Data

Our goal in this section is to investigate if our parameter estimation method can accurately recover the true model parameters from statement addition and evaluation events. We examine this question using a synthetically generated dataset from our probabilistic model.

Experimental setup. We set the number of sources to $|\mathcal{S}| = 400$, the total number of knowledge items to $|\mathcal{D}| = 3{,}600$, and assume the evaluation mechanism is refutation. We assume there is only one topic and then, for each source, we sample its trustworthiness α_s from the Beta distribution $Beta(2.0, 5.0)$ and its parameter γ_s from the uniform distribution $U(0, b)$, where $b = 0.03 \times \max(\{\alpha_s\}_{s\in\mathcal{S}})$. For the temporal evolution of the intrinsic reliability in the addition and evaluation processes, we consider a mixture of three radial basis (RBF) kernels located at times $t_j = 0, 6, 12$, with standard deviations of 2 and 0.5, respectively. Then, for each knowledge



(a) *Wikipedia* (b) *Stack Overflow*

Figure 2.2: Prediction performance. Panel (a) shows the AUC achieved by our model and three baselines (Intrinsic, Source and Logistic Regression) for predicting whether a statement will be removed (refuted) from a *Wikipedia* article within a time period of T after it is posted; for different values of T. Panel (b) shows the success probability achieved by our model and two baseline (Source and Correct Answer) at predicting which answer to a question, among several answers, will be eventually verified in *Stack Overflow*.

item, we first pick one of the kernel locations j uniformly at random, which determines the only *active* kernel for both the addition and the evaluation processes in the knowledge item, and sample their associated parameters, $\phi_{d,j}$ and $\beta_{d,j}$, from the log-normal distribution $\ln\mathcal{N}(3.5, 0.1)$ and the uniform distribution $U(0, 0.2\phi_d)$, respectively. Moreover, we assume that only up to five (different) sources are active in each knowledge item, which we pick at random, and then draw a source probability vector for these five active sources in the knowledge item from a Dirichlet distribution with parameter 0.5. The choice of prior distributions for the model parameters ensures enough variability across knowledge items and sources, so that the model parameters can be recovered. Finally, we generate addition and refutation samples from the resulting addition and evaluation processes during the time interval $(0, 15]$.

Results. We evaluate the accuracy of our model estimation procedure by means of the root mean square error (RMSE) between the true (x) and the estimated $(\hat{x})$ parameters, *i.e.*, $\text{RMSE}(x) = \sqrt{\mathbb{E}[(x-\hat{x})^2]}$. Figure 2.1 shows the parameter estimation error with respect to the number of knowledge items used to estimate the model parameters. Since the source parameters α and γ are shared across knowledge items, the estimation becomes more accurate as we feed more knowledge items into our estimation procedure. However, every time we observe a new knowledge item, the number of parameters increases with an additional β_d and ϕ_d. Therefore, the knowledge item parameter estimation only becomes more accurate as a consequence of a better estimation of the source parameters. As soon as the source parameter estimation becomes *good enough*, the estimation does not improve further and the estimation error flattens.

2.1.3 Experiments on Real Data

In this section, we apply our model estimation method to large-scale data gathered from two knowledge repositories: *Wikipedia*, which uses refutation as evaluation mechanism (*i.e.*, deleted

statements), and *Stack Overflow*, which uses verification (*i.e.*, accepted answers). First, we show that our model can accurately predict whether a particular statement in a *Wikipedia* article will be refuted after a certain period of time, as well as which of the answers to a question in *Stack Overflow* will be accepted. Then, we show that it provides meaningful measures of web source trustworthiness in *Wikipedia* and user trustworthiness in *Stack Overflow*. Finally, we demonstrate that our model can be used to: (i) pinpoint the changes on the intrinsic reliability of a *Wikipedia* article over time and these changes match external noteworthy controversial events; and, (ii) find questions and answers in *Stack Overflow* with similar levels of intrinsic reliability, which in this case correspond to popularity and difficulty.

Data description and methodology. To build our *Wikipedia* dataset, we gather complete edit history, up to July 8, 2014, for 1 million *Wikipedia* English articles and track all the references (or links) to sources within each of the edits. Then, for each article d, we record for each added statement, its associated source s_i, its addition time t_i, and its refutation (deletion) time τ_i, if any. Such recorded data allows us to reconstruct the history of each article (or knowledge item), as given by Eq. 2.5. Moreover, since we can only expect our model estimation method to provide reliable and accurate results for articles and web sources with enough number of events, we only consider articles with at least 20 link additions and web sources that are used in at least 10 references. After these preprocessing steps, our dataset consists of $\sim$50 thousand web sources that appeared in $\sim$100 thousand articles, by means of $\sim$10.4 million addition events and $\sim$9 million refutation (deletion) events. The significant drop in the number of articles can be attributed to the large number of incomplete articles on Wikipedia, which lack reasonable number of citations. Finally, we run the (Python library) Gensim [93] on the latest revision of all documents in the dataset, with 10 topics and default parameters, to obtain the topic weight vectors $\mathbf{w}_d$, and apply our model estimation method, described in Section 2.1.1. Both in Eqs. 2.8 and 2.9, we used 19 RBF kernels, spaced every 9 months with standard deviation of 3 months. In Eq. 2.8, we used exponential triggering kernels with $\omega = 0.5$ hours^{-1}.

To build our *Stack Overflow* dataset, we gathered history of answers from January 1, 2011 up to June 30, 2011, for $\sim$500 thousand questions [6]. Then, for each answer, we record the question d it belongs to, the user s_i who posted the answer, its addition time t_i, and its verification (acceptance) time τ_i, if any. Similarly as in the *Wikipedia* dataset, such recorded data allows us to reconstruct the history of each question (or knowledge item), as given by Eq. 2.5. Again, since our model estimation method can only provide reliable and accurate results for questions and users with enough number of events, we only consider questions with an accepted answer (if any exist) within 4 days of publication time and users who posted at least 4 accepted answers. After these preprocessing steps, our data consists of $\sim$378 thousand questions which accumulate $\sim$724 thousand addition events (answers) and $\sim$224 thousand verification events (accepted answers). In this case, we assume a single topic and therefore the weight vector $\mathbf{w}_d$ becomes a scalar value of 1. Finally, we apply our model estimation method, described in Section 2.1.1. In this case, in Eqs. 2.8 and 2.9, we used single constant kernels β_d and ϕ_d, respectively, since the intrinsic reliability of questions in *Stack Overflow* does not typically change over time. In Eq. 2.8, we used step functions as triggering kernels, since the inhibiting effect of an accepted answer does not decay over time.

In both datasets, our parameter estimation method runs in $\sim$4 hours using a single machine

[6] Dataset available at https://archive.org/details/stackexchange.

	Music		Politics	
Rank	domain	Pr. rm. in 6 months	domain	Pr rm. in 6 months
1	guardian.co.uk	0.15	nytimes.com	0.18
2	rollingstone.com	0.17	guardian.co.uk	0.19
3	nytimes.com	0.17	google.com	0.20
6	billboard.com	0.26	usatoday.com	0.24
13	mtv.com	0.32	whitehouse.gov	0.29
Last	twitter.com	0.56	cia.com	0.45

Table 2.1: Top 20 most popular web sources from *Wikipedia* in each topic ranked by the probability that a link from them is removed within 6 months (Most reliable on top).

with 10 cores and 64 GB RAM.

Can we predict if a statement will be removed from Wikipedia? Our model can answer this question by solving a binary classification problem: predict whether a statement will be removed (refuted) within a time period of T after it is posted.

— *Experimental setup:* We first split all addition events into a training set (90% of the data) and a test set (the remaining 10%) at random, then fit the parameters of the information survival processes given by Eq. 2.9 using only the evaluation times of the addition events from the training set, and finally predict whether particular statements in the test set will be removed within a time period of T after it is posted. We compare the performance of our model with three baselines: “Intrinsic", “Source" and “Logistic Regression." “Intrinsic" attributes all changes in an article to the intrinsic (un)reliability of that document. We can capture this assumption in our model by assuming that the parameter $\boldsymbol{\alpha}_s$ in Eq. 2.9 is set to zero. Inspired by the model proposed by Adler and De Alfaro [2], we implement the baseline “Source”, which only accounts for the trustworthiness of the source that supports a statement, *i.e.*, it assumes that the intrinsic reliability of the article, parametrized by $\boldsymbol{\beta}_d$ in Eq. 2.9, is set to zero. Finally, “Logistic Regression" is a logistic regression model that uses the source identity (in one-hot representation), the document topic vector and the addition time of links as features. Here, we train a different logistic regression model per time window.

— *Results:* Since the dataset is highly unbalanced (only 25% of statements in the test set survive longer than 6 months), we evaluate the classification accuracy in terms of the area under the ROC curve (AUC), a standard metric for quantifying classification performance on unbalanced data. Figure 2.2(a) shows the AUC achieved by our model and the baselines for different values of T. Our model always achieves AUC values over 0.69, it improves its performance as T increases, and outperforms all baselines across the full spectrum of values of T. The “Source" baseline exhibits a comparable performance to our method for low values of T, however, its performance barely improves as T increases, in contrast, the “Intrinsic" baseline performs poorly for low values of T but exhibits a comparable performance to our method for high values of T. Finally, “Logistic Regression" achieves an AUC lower than our method across the full spectrum of values of T.

The above results suggest that refutations that occur quickly after a statement is posted, are mainly due to the untrustworthiness of the source; while refutations that occur later in time are

Stack Overflow			
Rank	user-id	ranking	P accept in 4 days
1	318425	top 0.30%	0.93
2	405015	top 0.07%	0.81
3	224671	top 0.01%	0.81
138	246342	top 0.12%	0.53
139	616700	top 0.36%	0.53
Last	344491	top 0.97%	0.53

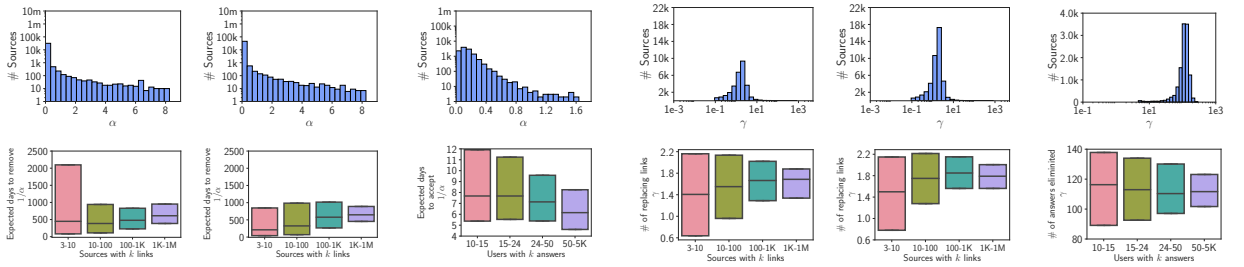
Table 2.2: *Stack Overflow* users with more than 100 answers (140 users) ranked by the probability that answer they provide is verified within 4 days (Most reliable on top). The table also shows the ranking provided by *Stack Overflow*.

due to the intrinsic unreliability of the article. As a consequence, our model, by accounting for both source trustworthiness and intrinsic reliability of information, can predict both quick and slow refutations more accurately than models based only on one of these two factors.

Can we predict which of the answers to a question in Stack Overflow will be accepted? Unlike Wikipedia where each article receives multiple evaluations (*i.e.*, deleted links), we have only one evaluation (*i.e.*, accepted answer) for every question in Stack Overflow. This property prevents us from estimating question difficulty in the test set and subsequently making predictions similar to that of *Wikipedia*. However, we can estimate users' reliability from all the questions in the training set and predict which of several competing answers to a question will be most likely verified.

— *Experimental setup:* We first split all questions (and corresponding answers) into a training set (90% of the questions) and test set (the remaining 10%) at random, then fit the parameters of the evaluation process given by Eq. 2.9 using only the evaluation times of the answers in the training set, and finally predict which answers will be accepted in the test set by computing the expected verification time for all answers to a question using the fitted model and selecting the earliest estimated verification time. We compare the performance of our model with two baselines: "Source" and "Correct Answers". "Source" only accounts for the trustworthiness of the sources (users) and ignores the intrinsic reliability (difficulty) of the questions. Thus, it computes the expected verification time of an answer in the test set as the average verification time of all the answers provided by its associated source user in the training set. Then, for each question in the test set, this baseline selects the answer with the lowest expected verification time. "Correct Answers" ranks sources (users) according to the number of accepted answers posted by each user in the training set. Then, for each question in the test set, it selects the answer with the highest ranked associated source.

— *Results:* Figure 2.2(b) summarizes the results by means of success rate for different training set sizes. Note that, unlike in the *Wikipedia* experiment, this prediction task does not correspond to a binary classification problem and therefore AUC is not a suitable metric in this case. Our model always achieves a rate of success over 0.47, consistently beats both baselines and, as expected, it becomes more accurate as we feed more events into the estimation procedure. Note that, for most questions, there are more than two answers and the success rate



(a) Analysis of parameter $\boldsymbol{\alpha}$

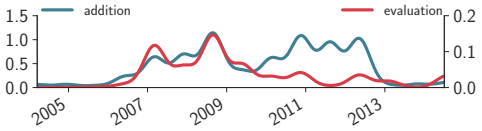
(b) Analysis of parameter $\boldsymbol{\gamma}$

Figure 2.3: Source Trustworthiness. Panels (a) and (b) show the distributions of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ for the Web sources in *Wikipedia* for the topics "music" and "politics" and for the *Stack Overflow* users, respectively. In both panels, the top row shows the distributions across all sources, while the bottom row shows the distributions for four set of sources, grouped by their popularity in the case of *Wikipedia* and by the number of answered questions in the case of *Stack Overflow* users. In *Wikipedia*, the evaluation mechanism is refutation and thus larger values of $1/\boldsymbol{\alpha}$ correspond to more trustworthy users whose contributed content is refuted more rarely. In *Stack Overflow*, the evaluation mechanism is verification and thus smaller values of $1/\boldsymbol{\alpha}$ correspond to more trustworthy users whose contributed content is verified quicker. In both cases, higher values of $\boldsymbol{\gamma}$ imply a larger impact on the overall reliability of the knowledge item (*i.e.*, article and question) after an evaluation.

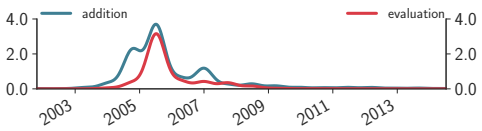
of a random baseline is 0.41. The above results suggest that one needs to account for both the users' trustworthiness and the difficulty of the questions to be able to accurately predict which answer will be accepted, in agreement with previous work [3].

Do our model parameters provide a meaningful and interpretable measure of source trustworthiness? We answer this question by analyzing the source parameters $\boldsymbol{\gamma}_s$ and $\boldsymbol{\alpha}_s$ estimated by our parameter estimation method, both in *Wikipedia* and *Stack Overflow*.

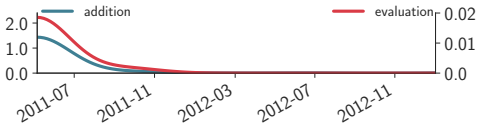
First, we pay attention to the 20 most used web sources in *Wikipedia* for two topics, *i.e.*, politics and music, and active users in *Stack Overflow* with over 100 answers, and rank them in terms of source trustworthiness (*i.e.*, in *Wikipedia*, higher trustworthiness means lower α_s, while in *Stack Overflow* higher trustworthiness means higher α_s). Then, we compute the probability that a statement supported by each source is refuted in less than 6 months in *Wikipedia* or verified in less than 4 days in *Stack Overflow* due to only the source trustworthiness (*i.e.*, setting $\boldsymbol{\beta} = 0$). Table 2.1 and 2.2 summarize the results, which reveal several interesting patterns. For example, our model identifies social networking sites such as Twitter, which often accumulate questionable facts and opinionated information, as untrustworthy sources for music in *Wikipedia*. Similarly, for articles related to politics, some notable news agencies close to the left of the political spectrum are considered to be more trustworthy, in agreement with previous studies on political bias in Wikipedia [43]. Moreover, users with high reputation, as computed by *Stack Overflow* itself, are indeed identified in our framework as trustworthy. However, the ranking among these users in terms of reputation does not always match our measure of trustworthiness since it also takes into account other factors such as number of up-votes on questions and



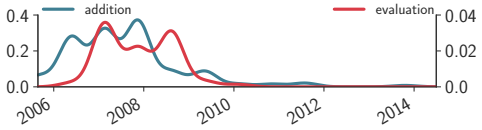
(a) Barack Obama's biography



(b) George W. Bush's biography



(c) "2011 military intervention" in Libya



(d) TV show Prison Break

Figure 2.4: Temporal evolution of the article intrinsic reliability for four *Wikipedia* articles. The blue (red) line shows intensity of statement addition (evaluation) process. Changes on the intrinsic reliability closely match external noteworthy events, often controversial, related to the corresponding article.

answers.

Next, we look at the source parameters at an aggregate level by means of their empirical distribution across users. Figure 2.3 summarizes the results, which show that: (i) the distributions are remarkably alike across both topics in *Wikipedia* and (ii) γ values are distributed similarly both for *Stack Overflow* and *Wikipedia*, however, α values are distributed differently since they capture a different mechanism, verification instead of refutation. Finally, we group web sources in *Wikipedia* by popularity and users of *Stack Overflow* by number of contributed answers, and analyze the source parameters. We summarize the results in Figure 2.3, which show that: (i) more popular web sources in *Wikipedia* and more active users in *Stack Overflow* tend to be more trustworthy, *i.e.*, lower (higher) α in *Wikipedia* (*Stack Overflow*); (ii) popular sources in *Wikipedia* have a larger impact on the reliability of the article, triggering a larger number of new statements additions (*i.e.*, larger values of γ) after a refutation; and, (iii) there is ample variation across sources in terms of trustworthiness within all groups.

What do the temporal evolution of the intrinsic reliability of Wikipedia articles tell us? In this section, we show that changes on the intrinsic reliability of a *Wikipedia* article closely match external noteworthy events, often controversial, related to the article.

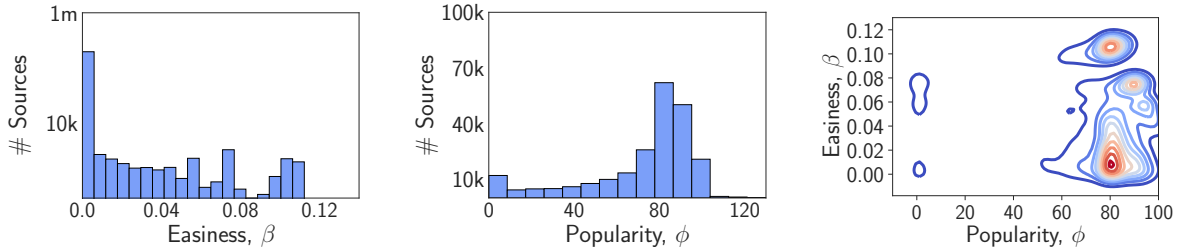
Figure 2.4 shows the intrinsic reliability both in the statement addition process (first term in Eq. 2.8), which captures the arrival of new information, and the verification process (first term in Eq. 2.9), which captures the controversy of the article, for four different articles – Barack Obama's biography,[7] George W. Bush's biography,[8] an article on 2011 military intervention in Libya,[9] and an article on the TV show Prison Break.[10] Each of the articles exhibits different characteristic temporal patterns. In the two biographical articles and the article on the TV show, we find several peaks in the arrival of new information and controversy over time, which typically match remarkable real-world events. For example, in Barack Obama's article, the peaks in early 2007 and mid-2008 coincide with the time in which he won the Democratic

[7] https://en.wikipedia.org/wiki/Barack_Obama

[8] https://en.wikipedia.org/wiki/George_W._Bush

[9] https://en.wikipedia.org/wiki/2011_military_intervention_in_Libya

[10] https://en.wikipedia.org/wiki/Prison_Break



(a) $P(\beta)$ (b) $P(\phi)$ (c) $P(\phi, \beta)$

Figure 2.5: Difficulty vs. popularity in *Stack Overflow* questions. Panels (a) and (b) show the distribution of the parameters β and ϕ, which represent respectively the difficulty and the popularity of *Stack Overflow* questions. Panel (c) shows the joint distribution of both parameters β and ϕ. Higher value of β (ϕ) implies easier (more popular) questions.

nomination and the 2008 US election campaign; and, in the Prison Break's article, the peaks coincide with the broadcasting of the four seasons. In contrast, in the article about 2011 military intervention in Libya, we only find one peak, localized at the beginning of the article life cycle, which is followed by a steady decline in which the controversy lasts for a few months longer than the arrival of new information. A comparison of the temporal patterns of new information arrivals and controversy within an article reveals a more subtle phenomenon: while sometimes a peak in the arrival of new information also results in a peak of controversy, there are peaks in the arrival that do not trigger controversy and vice-versa.

What do the intrinsic reliability of Stack Overflow questions tell us? We answer this question by analyzing the parameters β_d and ϕ_d estimated by our parameter estimation method for questions in *Stack Overflow*. For each question, such parameters are unidimensional since, unlike *Wikipedia*, the reliability of questions in *Stack Overflow* does not typically change over time. Moreover, the parameters have natural interpretation: β reflects the easiness of a question and ϕ reflects its popularity.

Figures 2.5(a-b) show the empirical marginal distribution of the parameters across questions and Figure 2.5(c) shows the joint distribution for questions with $\beta > 0$. The results reveal four clusters: questions which are popular and easy, questions which are popular but difficult, questions that are not popular and difficult, and questions that are not popular but easy.

2.1.4 Conclusion

In this section, we proposed a temporal point process modeling framework of refutation and verification in online knowledge repositories and developed an efficient convex optimization procedure to fit the parameters of our framework from historical traces of the refutations and verifications provided by the users of a knowledge repository. Then, we experimented with real-world data gathered from *Wikipedia* and *Stack Overflow* and showed that our framework accurately predicts refutation and verification events, provides an interpretable measure of information reliability and source trustworthiness, and yields interesting insights about real-world events.

Our work also opens many interesting directions for future work. For example, natural

follow-ups to potentially improve the expressiveness of our modeling framework include:

1. Consider sources can change their trustworthiness over time due to, *e.g.*, increasing their expertise [113].
2. Allow for non-binary refutation and verification events, *e.g.*, partial refutations, ratings.
3. Augment our model to consider the trustworthiness of the user who refutes or verifies a statement.
4. Reduce number of parameters in the model by clustering sources and knowledge items.

Moreover, we experimented with data gathered from *Wikipedia* and *Stack Overflow*, however, it would be interesting to apply our model (or augmented versions of our model) to other knowledge repositories (*e.g.*, *Quora*), other types of online collaborative platforms (*e.g.*, *Github*), and the Web at large. Finally, one can think of using our measure of trustworthiness, as inferred by our estimation method, to perform credit assignment in online collaborative platforms—in *Wikipedia*, one could use our model to identify trustworthy users (or dedicated editors) who can potentially make an article more reliable and stable.

2.2 Proposed Model for Reducing the Spread of Misinformation in News-feed Algorithms

Rankings are ubiquitous across a large variety of online services, from search engines, online shops and recommender systems to social media and online dating. They have undoubtedly increased the utility users obtain from online services. However, rankings have also been blamed for negative developments, particularly in the context of social and information systems, from fueling the spread of misinformation [116], increasing polarization [47] and degrading social discourse [121], to undermining democracy [114]. As the decisions taken by ranking models become more consequential to individuals and society, one must ask: what went wrong in these cases?

Current ranking models are typically designed to optimize immediate measures of utility, which often reward instant gratification. For example, one of the guiding technical principles behind the optimization of ranking models in the information retrieval literature, the *probability ranking principle* [94], states that the optimal ranking should order items in terms of probability of relevance to the user. However, such measures of immediate utility do not account for long-term consequences. As a result, ranking models often have an unexpected cost to the long-term welfare. In this work, our goal is to design consequential ranking models which anticipate the long-term consequences of their proposed rankings.

More specifically, we focus on a problem setting that fits a variety of real-world applications, including those mentioned previously: at every time step, an existing ranking model receives a set of items and ranks these items on the basis of a measure of immediate (possibly unknown) utility[11] and a set of features. Items may appear over time and be present at several time steps. Moreover, their corresponding features may also change over time and these changes may be due to the influence of previous rankings. For example, the number of likes, votes, or comments—the

[11] Our methodology does not need to observe the immediate utility the ranking model based their rankings on.

features—that a post—the item—published by a user receives in social media depends largely on its ranking position [48, 42, 64, 53]. Moreover, for every sequence of rankings, there is an associated long-term (cost to the) welfare, whose specific definition is application dependent. For example, in information integrity, the welfare may be defined based on the number of posts including misinformation at the top of the rankings, averaged over time. Our goal is then to construct consequential ranking models that optimally trade off fidelity to the original ranking model maximizing immediate utility and long-term welfare[12].

Contributions. In this part of the thesis, we first introduce a joint representation of ranking models and user dynamics using Markov decisions processes (MDPs), which is particularly well-fitted to faithfully characterize the above problem setting[13]. Then, we show that this representation greatly simplifies the construction of consequential ranking models that trade off fidelity to the rankings provided by a ranking model maximizing immediate utility and the long-term welfare. More specifically, we apply Bellman's principle of optimality and show that it is possible to derive an analytical expression for the optimal consequential ranking model in terms of the original ranking model and the cost to the welfare. This means that we can obtain optimal consequential rankings by applying weighted sampling on the rankings provided by the original ranking model using the (exponentiated) cost to welfare. However, in practice, such a naive sampling will be inefficient, especially in the presence of high-dimensional features. Therefore, we design a practical and efficient gradient-based algorithm to learn parameterized consequential ranking models that effectively approximate optimal ones[14].

Finally, we evaluate our methodology using synthetic and real data gathered from Reddit. The results show that our consequential ranking models provide rankings that may mitigate the spread of misinformation and improve the civility of online discussions without significant deviations from the original rankings provided by models maximizing immediate utility measures.

Related work. Our work relates to several lines of research: (i) ranking algorithms; (ii) delayed impact of machine learning algorithms; (iii) optimal control and reinforcement learning; and, (iv) reducing the spread of misinformation and polarization.

— *Ranking algorithms:* the work most closely related to ours is devoted to construct either fair rankings [101, 102, 103, 128] or diverse rankings [17, 21]. However, this line of research defines fairness and diversity in terms of exposure allocation on an individual ranking rather than in a sequence of rankings. In contrast, we consider sequences of rankings, we characterize the consequences of these rankings on the user dynamics, and focus on improving the welfare in the long-term.

— *Delayed impact of ML algorithms:* the delayed impact of machine learning algorithms has not been studied until very recently [49, 70, 80**?**]. However, most of these recent approaches have focused on classification tasks and have considered simple one-step feedback models. In contrast, in this work, we focus on rankings and consider a multiple step feedback model based on Markov decision processes (MDPs).

— *Optimal control and reinforcement learning:* the work most closely related to ours within the extensive literature on optimal control and reinforcement learning is devoted to improving the

[12] In practice, one can only measure a welfare *proxy*, however, for brevity, we will refer to welfare proxy as welfare. Moreover, the effectiveness of our methodology will depend on the quality of the welfare proxies at our disposal.

[13] In this work, for ease of exposition, we assume all users are exposed to the same rankings, as in, *e.g.*, Reddit. However, our methodology can be readily extended to the scenario in which each user is exposed to a different ranking, as in, *e.g.*, Twitter.

[14] We will release an open-source implementation of our algorithm with the final version of the paper.

functioning of social and information systems [118, 127]. However, this line of work has mainly focused on representations based on temporal point processes and have not considered rankings. Recently, a framework based on survival process has been proposed to optimize click through rate using reinforcement learning [132].

— *Reducing the spread of misinformation and polarization:* the literature on algorithms for reducing the spread of misinformation [6, 58, 112] and reducing polarization [39, 40] is expanding very rapidly (see [61] for an excellent review of recent work). However, to the best of our knowledge, previous work has not approached the problem from the perspective of ranking algorithms.

Rankings and User Dynamics

In this section, we first introduce our joint representation of rankings and user dynamics, starting from the problem setting it is designed for. Then, we formally define consequential rankings as the solution to a particular reinforcement learning problem.

Problem setting. Let p be a particular ranking model (or, equivalently, ranking algorithm). At each time step $t \in \{1, \dots, T\}$, the ranking model receives a set of n items and these items are characterized by a feature matrix $\boldsymbol{X}(t) \in \mathbb{R}^{n \times m}$, where the i-th row $\boldsymbol{X}_i(t)$ contains the feature values for item $i \in [n]$ and m is the number of features per item. Here, we assume that items may appear over time and be present at several time steps. Moreover, their corresponding feature values may also change over time. For example, think of the number of likes, votes or comments that a post receives in social media—they are often used as features to decide the ranking of the post and they change over time.

Then, the ranking model provides a ranking $\boldsymbol{y}(t)$ of the items on the basis of their set of features and a (hidden) measure of immediate utility. A ranking $\boldsymbol{y}(t) = (y_1(t), \dots, y_n(t))$ is defined as a permutation of the n rank indices, *i.e.*, the model ranks item i in position $y_i(t)$, where highest rank is position 1. In addition, we also define the ordering $\boldsymbol{\omega}(t) = (\omega_1(t), \dots, \omega_n(t))$ of a ranking as a permutation of the n item indices, *i.e.*, the model ranks item $\boldsymbol{\omega}_i(t)$ in position i. The ranking and orderings are related by $w_{y_i(t)}(t) = i$ and $y_{w_i(t)}(t) = i$. Here, we assume that the provided ranking at time step t may influence the feature matrix at time step $t+1$. This is in agreement with recent empirical studies [42, 48, 53, 64], which have shown that the posts (the items) that are ranked highly receive a higher number of likes, comments or shares (the features).

Finally, given a trajectory of feature matrices and rankings $\tau = \{(\boldsymbol{X}(t), \boldsymbol{y}(t))\}_{t=0}^{T}$ there is an additive cost to the welfare, $c(\tau) = \sum_{t=0}^{T} c(\boldsymbol{X}(t), \boldsymbol{y}(t))$, where $c(\boldsymbol{X}(t), \boldsymbol{y}(t))$ is an arbitrary immediate cost whose specific definition is application dependent. For example, in information integrity, the cost may be defined as the average number of posts including misinformation at the top of the rankings over time. In the remainder, we will say that a trajectory τ is *induced* by a ranking model p.

Joint representation of rankings and user dynamics. The above problem setting naturally fits the following joint representation of rankings and user dynamics using Markov decision

processes (MDPs) [107], which also has an intuitive causal interpretation:

$$
\begin{aligned}
& p(\tau \,|\, \boldsymbol{X}(t_0), \boldsymbol{y}(t_0)) \\
&= \prod_{t=1}^{T} p(\boldsymbol{X}(t), \boldsymbol{y}(t) \,|\, \boldsymbol{X}(t-1), \boldsymbol{y}(t-1)) \\
&= \prod_{t=1}^{T} \underbrace{p(\boldsymbol{y}(t) \,|\, \boldsymbol{X}(t))}_{\text{ranking model}} \underbrace{p(\boldsymbol{X}(t) \,|\, \boldsymbol{X}(t-1), \boldsymbol{y}(t-1))}_{\text{user dynamics}},
\end{aligned}
\tag{2.14}
$$

where the first term represents the particular choice of ranking model[15], the second term represents the distribution for the user dynamics, which determines the feature matrix at any given time step, and the initial feature matrix $\boldsymbol{X}(t_0)$ and ranking $\boldsymbol{y}(t_0)$ are given. Moreover, the

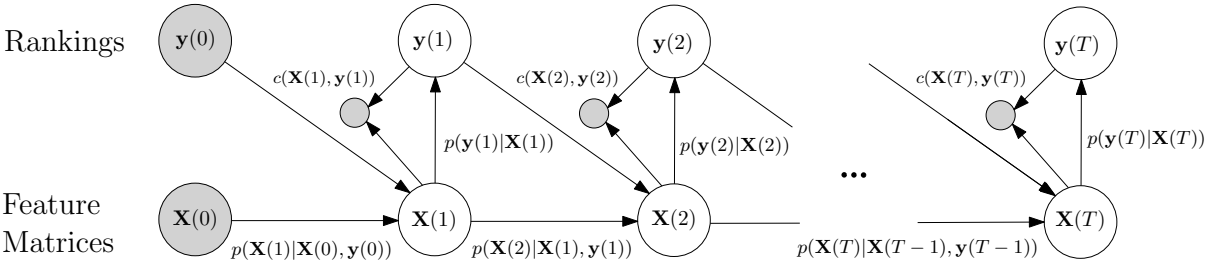


Figure 2.6: Our joint representation of rankings and user dynamics using Markov decision processes (MDPs). The ranking model $p_\theta(\boldsymbol{y}(t) \,|\, \boldsymbol{X}(t))$ provides a ranking $\boldsymbol{y}(t)$ for a set of items on the basis of the feature matrix $\boldsymbol{X}(t)$ of the items and both the feature matrix and the provided ranking result in a cost to welfare $c(\boldsymbol{X}(t), \boldsymbol{y}(t))$. The distribution of user dynamics $p(\boldsymbol{X}(t+1) \,|\, \boldsymbol{X}(t), \boldsymbol{y}(t))$ determines the feature matrix $\boldsymbol{X}(t+1)$ on the basis of the previous feature matrix $\boldsymbol{X}(t)$ and ranking $\boldsymbol{y}(t)$.

above representation makes two major assumptions, which are also illustrated in Figure 2.6.

(i) To provide a ranking for a set of items at time step t, the ranking model only uses the feature matrix corresponding to that set of items. More formally, given the feature matrix $\boldsymbol{X}(t)$, the ranking $\boldsymbol{y}(t)$ provided by the ranking model is conditionally independent of previous feature matrices $\boldsymbol{X}(t')$, $t' < t-1$.

(ii) The dynamics of the feature matrices, which characterize the user dynamics, are Markovian. That means, given the feature matrix $\boldsymbol{X}(t-1)$ and ranking $\boldsymbol{y}(t-1)$, the feature matrix $\boldsymbol{X}(t)$ is conditionally independent of previous feature matrices $\boldsymbol{X}(t')$ and rankings $\boldsymbol{y}(t')$, $t' < t-1$.

We would like to highlight that, in most practical scenarios, ranking models optimizing for immediate utility satisfy the first assumption. However, depending on the choice of features, the second assumption may be violated and thus the representation of the user dynamics becomes an approximation. It would be very interesting, albeit challenging, to lift the second assumption in future work.

Next, we elaborate further on the specifics of the ranking model and the distribution of the user dynamics.

[15] In our work, we consider probabilistic ranking models, which assign a probability to each ranking. Extending the methodology to deterministic ranking models is left for future work.

— *Ranking model:* Our approach is agnostic to the particular choice of ranking model—it provides a methodology to derive consequential rankings that are optimal under a ranking model. In our experiments, we showcase our methodology for one well-known model, Plackett-Luce (P-L) ranking [73, 88], which is best described in terms of the orderings of the rankings. Under the P-L model, at each time step t, the ranking $\boldsymbol{y}(t)$ with ordering $\boldsymbol{\omega}(t)$ is sampled from a distribution

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}(t)\,|\,\boldsymbol{X}(t)) = \prod_{k=1}^{n} f_k(\boldsymbol{X}(t)), \tag{2.15}$$

with

$$f_k(\boldsymbol{X}(t)) = \frac{\exp\left(\boldsymbol{\theta}^T \boldsymbol{X}_{\omega_k}(t)\right)}{\sum_{k'=k}^{N} \exp\left(\boldsymbol{\theta}^T \boldsymbol{X}_{\omega_{k'}}(t)\right)}, \tag{2.16}$$

where $\boldsymbol{\theta}$ is a given parameter. In the above, we can think of $\boldsymbol{\theta}^T \boldsymbol{X}_{\omega_k}(t)$ as a *quality score* associated to the item ω_k, which controls the probability that this item is ranked at the top—the higher the quality score, the higher the probability that the item is ranked first. In practice, the quality score of the above P-L ranking model may be computed using a complex nonlinear function [111], *e.g.*, a neural network.

— *User dynamics:* Our approach only requires to be able to sample $\boldsymbol{X}(t)$ from any arbitrary model for the transition probability $p(\boldsymbol{X}(t)\,|\,\boldsymbol{X}(t-1), \boldsymbol{y}(t-1))$, which may be estimated using historical ranking and user data. Here, in contrast with the ranking model, the user dynamics are not something that one can decide upon—they are given.

Consequential rankings. Let p_0 be an existing ranking model[16] that optimizes some hidden immediate utility and $c(\cdot)$ be a given cost to the welfare. Then, we construct a consequential ranking model p^*, which optimally trades off the fidelity to the original ranking model and the cost to the long-term welfare, by solving the following optimization problem:

$$\underset{p}{\text{minimize}} \quad \mathbb{E}_{\tau\sim p}\left[S(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0))\right], \tag{2.17}$$

with

$$S(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0)) = c(\tau) + \lambda \log \frac{p(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0))}{p_0(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0))}, \tag{2.18}$$

where the expectation is taken over all the trajectories τ of feature matrices and rankings of length T induced by the ranking model p_0. The choice of trajectory length T will depend on the definition of long-term—accounting for longer-term consequences to the welfare will require larger trajectory lengths T. In Eq. 2.18, the parameter $\lambda \geq 0$ controls the trade off between the fidelity to the original ranking model and the long-term cost to the welfare. Note that, for $\lambda \to \infty$, the optimal ranking p^* coincides with the original ranking p_0. Moreover, the first term penalizes trajectories that achieve a large cost to the welfare and the second term penalizes ranking models whose induced trajectories differ more from those induced by the original model, since the terms

[16] In our experiments, we will approximate the existing ranking model using a P-L ranking model. We will fit the parameters of this P-L ranking model from historical rankings provided the original ranking model via regularized maximum likelihood estimation (MLE) [51].

associated to the user dynamics $p(\boldsymbol{X}(t)\,|\,\boldsymbol{X}(t-1),\boldsymbol{y}(t-1))$ cancel.

Finally, note that, from the perspective of reinforcement learning, we are solving a *forward problem*, where the cost is given, rather than an *inverse problem*, where the cost is inferred. Moreover, our measure of fidelity has a natural interpretation in terms of the Kullback-Leibler (KL) divergence [60], which has been extensively used as a distance measure between distributions, leading to a formulation of reinforcement learning as probabilistic inference [66, 54, 134]. More specifically, we can write the expectation of the second term as the KL divergence between the original and the consequential ranking model, *i.e.*,

$$\begin{aligned} &KL[p(\cdot\,|\,\boldsymbol{X}(0),\boldsymbol{y}(0))\,||\,p_0(\cdot\,|\,\boldsymbol{X}(0),\boldsymbol{y}(0))] \\ &\qquad = \mathbb{E}_{\tau\sim p}\left[\log\frac{p(\tau\,|\,\boldsymbol{X}(0),\boldsymbol{y}(0))}{p_0(\tau\,|\,\boldsymbol{X}(0),\boldsymbol{y}(0))}\right]. \end{aligned}$$

In the next section, we will exploit this interpretation to greatly simplify the construction of consequential rankings.

2.2.1 Building Consequential Rankings

In this section, we tackle the optimization problem defined by Eq. 2.17 from the perspective of reinforcement learning and show that the optimal consequential ranking model p^* can be expressed in terms of the original ranking model.

We can first break the above problem into small recursive subproblems using Bellman's principle of optimality [10]. This readily follows from the fact that, under the representation introduced in Section 2.2, the ranking model and the user dynamics are a Markov decision process (MDP). More specifically, Bellman's principle tells us that the optimal ranking model should satisfy the following recursive equation, which is called the Bellman optimality equation:

$$\begin{aligned} V_t(\boldsymbol{X},\boldsymbol{y}) = &\min_{p}\ \ell(\boldsymbol{X},\boldsymbol{y}) \\ &+\lambda\mathbb{E}_{(\boldsymbol{X}',\boldsymbol{y}')\sim p(\cdot,\cdot\,|\,\boldsymbol{X},\boldsymbol{y})}\left[V_{t+1}(\boldsymbol{X}',\boldsymbol{y}')\right] \end{aligned} \tag{2.19}$$

with $V_T(\boldsymbol{X},\boldsymbol{y}) = \ell(\boldsymbol{X},\boldsymbol{y})$. The function $V_t(\boldsymbol{X},\boldsymbol{y})$ is called the value function and the function $\ell(\boldsymbol{X},\boldsymbol{y})$ is called immediate loss. Moreover, in our problem, it can be readily shown that the immediate loss adopts the following form:

$$\begin{aligned} \ell(\boldsymbol{X},\boldsymbol{y}) &= c(\boldsymbol{X},\boldsymbol{y}) \\ &\quad +\lambda\mathbb{E}_{(\boldsymbol{X}',\boldsymbol{y}')\sim p(\cdot,\cdot\,|\,\boldsymbol{X},\boldsymbol{y})}\left[\log\frac{p(\boldsymbol{X}',\boldsymbol{y}'\,|\,\boldsymbol{X},\boldsymbol{y})}{p_0(\boldsymbol{X}',\boldsymbol{y}'\,|\,\boldsymbol{X},\boldsymbol{y})}\right] \\ &= c(\boldsymbol{X},\boldsymbol{y}) + \lambda KL(p(\cdot,\cdot\,|\,\boldsymbol{X},\boldsymbol{y})\,||\,p_0(\cdot,\cdot\,|\,\boldsymbol{X},\boldsymbol{y})). \end{aligned}$$

Within the loss function, the first term penalizes the immediate cost to the welfare and the second term penalizes consequential ranking models whose induced transition probability differs from that induced by the original ranking model.

In general, Bellman optimality equations are difficult to solve. However, the structure of our problem will help us find an analytical solution. Inspired by [108], we proceed as follows. Let

$Z_t(\boldsymbol{X},\boldsymbol{y}) = \exp(-V_t(\boldsymbol{X},\boldsymbol{y}))$. Then, we can rewrite the minimization in the right hand side of Eq. 4.9 as

$$\min_p \mathop{\mathbb{E}}_{(\boldsymbol{X}',\boldsymbol{y}')\sim p(\cdot,\cdot\,|\,\boldsymbol{X},\boldsymbol{y})} \left[\log \frac{p(\boldsymbol{X}',\boldsymbol{y}'\,|\,\cdot)}{p_0(\boldsymbol{X}',\boldsymbol{y}'\,|\,\cdot)Z_{t+1}(\boldsymbol{X}',\boldsymbol{y}')}\right],$$

where we have dropped λ and $c(\boldsymbol{X},\boldsymbol{y})$ because they do not depend on p and, for brevity, we have replaced the conditionals (X,Y) inside the logarithm with $\cdot$. Then, we can use Eq. 2.14 to factorize both transition probabilities in the numerator and the denominator within the logarithm and, as a result, the terms $p(\boldsymbol{X}'\,|\,\boldsymbol{X},\boldsymbol{y})$ cancel and we obtain:

$$\min_p \mathop{\mathbb{E}}_{(\boldsymbol{X}',\boldsymbol{y}')\sim p(\cdot,\cdot\,|\,\boldsymbol{X},\boldsymbol{y})} \left[\log \frac{p(\boldsymbol{y}'\,|\,\boldsymbol{X}')}{p_0(\boldsymbol{y}'\,|\,\boldsymbol{X}')Z_{t+1}(\boldsymbol{X}',\boldsymbol{y}')}\right].$$

The above equation resembles a KL divergence, however, note that the fraction within the logarithm does not depend on $(\boldsymbol{X},\boldsymbol{y})$ and the denominator $p_0(\boldsymbol{y}'\,|\,\boldsymbol{X}')Z_{t+1}(\boldsymbol{X}',\boldsymbol{y}')$ is not normalized to one. If we multiply and divide the fraction by the following normalization term:

$$G[Z_{t+1}](\boldsymbol{X}') = \mathbb{E}_{\boldsymbol{y}'\sim p_0(\boldsymbol{y}'|\boldsymbol{X}')}[Z_{t+1}(\boldsymbol{X}',\boldsymbol{y}')], \tag{2.20}$$

we obtain:

$$\begin{aligned}\min_p & - \mathop{\mathbb{E}}_{\boldsymbol{X}'\sim p(\cdot\,|\,\boldsymbol{X},\boldsymbol{y})} \left[\log G[Z_{t+1}](\boldsymbol{X}')\right] \\ & + \mathop{\mathbb{E}}_{(\boldsymbol{X}',\boldsymbol{y}')\sim p(\cdot,\cdot\,|\,\boldsymbol{X},\boldsymbol{y})} \left[\log \frac{p(\boldsymbol{y}'\,|\,\boldsymbol{X}')G[Z_{t+1}](\boldsymbol{X}')}{p_0(\boldsymbol{y}'\,|\,\boldsymbol{X}')Z_{t+1}(\boldsymbol{X}',\boldsymbol{y}')}\right].\end{aligned}$$

Here, note that the first term does not depend on p and the second term achieves its global minimum of zero if the numerator and the denominator are equal. Thus, the optimal consequential ranking model is just given by:

$$p^*(\boldsymbol{y}\,|\,\boldsymbol{X}) = \frac{p_0(\boldsymbol{y}\,|\,\boldsymbol{X})Z_{t+1}(\boldsymbol{X},\boldsymbol{y})}{G[Z_{t+1}](\boldsymbol{X})}. \tag{2.21}$$

The above equation reveals that the optimal consequential ranking model $p^*(\boldsymbol{y}\,|\,\boldsymbol{X})$ does implicitly depend on time due to Z_{t+1}. Finally, if we substitute back the above expression into the Bellman equation, given by Eq. 4.9, we can also find the function Z_t using the following recursive expression:

$$\begin{aligned}Z_t(\boldsymbol{X},\boldsymbol{y}) = & \exp\big(-c(\boldsymbol{X},\boldsymbol{y}) \\ & + \lambda \mathbb{E}_{\boldsymbol{X}'\sim p(\boldsymbol{X}'\,|\,\boldsymbol{X},\boldsymbol{y})} \left[\log G[Z_{t+1}](\boldsymbol{X}')\right]\big),\end{aligned}$$

with $Z_T(\boldsymbol{X},\boldsymbol{y}) = -\log c(\boldsymbol{X},\boldsymbol{y})$. This result has an important implication. It means that we can use sampling methods to obtain (unbiased) samples from the optimal consequential ranking, *e.g.*, stratified sampling [29], as shown in Algorithm 1, where $\textsc{Sample}(p_0,\kappa)$ samples κ trajectories from $p_0(\tau)$ and $\textsc{StratifiedSampler}(\mathcal{D}',W)$ generates $|\mathcal{D}'|$ samples weighted by W using stratified sampling.

Algorithm 1: It samples from an optimal consequential ranking model given p_0.

Input: Cost to welfare $c(\cdot)$, parameter λ, original ranking model p_0, $(\boldsymbol{X}(0), \boldsymbol{y}(0))$, # of samples B, # of samples κ to compute $G[Z_T]$.

$\mathcal{D} \leftarrow \textsc{Sample}(p_0, \kappa)$ ▷ samples for estimating $G[Z_T]$.
$\Lambda[Z_T] \leftarrow 0$
for $c(\tau_i) \in \mathcal{D}$ **do**
    $\Lambda[Z_T] \leftarrow \Lambda[Z_T] + \exp\left(-\lambda^{-1} c(\tau_i)\right)/\kappa$
end
$\mathcal{D}' \leftarrow \textsc{Sample}(p_0, B)$ ▷ unweighted samples.
$W \leftarrow []$ ▷ array of weights.
for $c(\tau_i) \in \mathcal{D}'$ **do**
    $W[i] \leftarrow \exp\left(-\lambda^{-1} c(\tau_i)\right)/\kappa/G[Z_T]$
end
$W \leftarrow W/\textsc{Sum}(W)$
return $\textsc{StratifiedSampler}(\mathcal{D}', W)$

Unfortunately, in practice, these sampling methods may be inefficient and have high variance if the original ranking model p_0 produces rankings that have very low probability under the optimal consequential ranking model. This will be specially problematic in the presence of high-dimensional feature vectors due to the curse of dimensionality. In the next section, we will present a practical method for approximating $p^*(\boldsymbol{y}\,|\,\boldsymbol{X})$, which iteratively adapts a parameterized consequential ranking model $p^*_\theta(\boldsymbol{y}\,|\,\boldsymbol{X})$ using a stochastic gradient-based algorithm.

2.2.2 A Stochastic Gradient-Based Algorithm

In this section, our goal is to find a parameterized consequential ranking model p^*_θ within a class of parameterized ranking models $\mathcal{P}(\Theta)$ (*e.g.* PL models in Eq. 2.15) that approximates well the optimal consequential ranking model p^*, given by Eq. 2.21, *i.e.* $p^*_\theta \approx p^*$. To this aim, we minimize the parameterized version of the objective function in Eq. 2.17, *i.e.*,

$$\mathbb{E}_{\tau \sim p_\theta}\left[S_\theta(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0))\right]. \tag{2.22}$$

where,

$$S_\theta(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0)) = c(\tau) + \lambda \log \frac{p_\theta(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0))}{p_0(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0))}$$

More specifically, we introduce a general gradient-based algorithm, which only requires the class of parameterized ranking models $\mathcal{P}(\Theta)$ to be differentiable. In particular, we resort to stochastic gradient descent (SGD) [56], *i.e.*,

$$\theta^{(j+1)} = \theta^{(j)} + \gamma_j \nabla_\theta \, \mathbb{E}_{\tau \sim p_\theta}\left[S_\theta(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0))\right]\Big|_{\theta=\theta^{(j)}},$$

where $\gamma_j > 0$ is the learning rate at step $j \in \mathbb{N}$. Here, it may seem challenging to compute a finite sample estimate of the gradient of the objective function $\mathbb{E}_{\tau \sim p_\theta}\left[S_\theta(\tau\,|\,\boldsymbol{X}(0), \boldsymbol{y}(0))\right]$ since the derivative is taken with respect to the parameters of the ranking model p_θ, which we are trying

Algorithm 2: Training a parameterized consequential ranking model.

Input: Cost to welfare $c(\cdot)$, parameter λ, original ranking model p_0, $(\boldsymbol{X}(0), \boldsymbol{y}(0))$, # of iterations M, mini batch size B, and learning rate γ

$\theta^{(0)} \leftarrow$ INITIALIZERANKINGMODEL()

while $j = 1, \ldots, M$ **do**

  $\mathcal{D} \leftarrow$ MINIBATCH(p_θ, B) ▷ sample mini batch

  $\nabla \leftarrow 0$

  while $\tau^{(i)} \in \mathcal{D}$ **do**

    $S \leftarrow c(\tau^{(i)}) + \lambda \log \frac{p_{\theta^{(j)}}(\tau^{(i)} \mid \boldsymbol{X}(0), \boldsymbol{y}(0))}{p_0(\tau^{(i)} \mid \boldsymbol{X}(0), \boldsymbol{y}(0))}$

    $\widetilde{\nabla} \leftarrow \nabla_\theta \log p_{\theta^{(j)}}(\tau^{(i)} \mid \boldsymbol{X}(0), \boldsymbol{y}(0))$ $\nabla \leftarrow \nabla + (S + \lambda)\widetilde{\nabla}$

  end

  $\theta^{(j+1)} \leftarrow \theta^{(j)} + \gamma \frac{\nabla}{B}$

end

return $\theta^{(M)}$

to learn. However, we can overcome this challenge using the log-derivative trick as in [119], *i.e.*,

$$\begin{aligned} &\nabla_\theta \mathbb{E}_{\tau \sim p_\theta} \left[S_\theta(\tau \mid \boldsymbol{X}(0), \boldsymbol{y}(0)) \right] \\ &\quad = \mathbb{E}_{\tau \sim p_\theta} \left[(S_\theta(\tau \mid \boldsymbol{X}(0), \boldsymbol{y}(0)) + \lambda) \times \right. \\ &\qquad \left. \nabla_\theta \log p_\theta(\tau \mid \boldsymbol{X}(0), \boldsymbol{y}(0)) \right], \end{aligned} \tag{2.23}$$

where $\nabla_\theta \log p_\theta(\tau \mid \boldsymbol{X}(0), \boldsymbol{y}(0))$ is often referred as the score function [52]. This yields the following unbiased finite sample Monte-carlo estimator for the gradient:

$$\begin{aligned} &\nabla_\theta \mathbb{E}_{\tau \sim p_\theta} \left[S_\theta(\tau \mid \boldsymbol{X}(0), \boldsymbol{y}(0)) \right] \approx \\ &\quad \sum_{i=1}^{B} \left(S_\theta(\tau^{(i)} \mid \boldsymbol{X}(0), \boldsymbol{y}(0)) + \lambda \right) \times \\ &\qquad \nabla_\theta \log p_\theta(\tau^{(i)} \mid \boldsymbol{X}(0), \boldsymbol{y}(0)), \end{aligned} \tag{2.24}$$

where B is the number of sampled trajectories from the joint distribution $p_\theta(\tau \mid \boldsymbol{X}(0), \boldsymbol{y}(0))$ induced by the ranking model p_θ. The overall procedure is summarized in Algorithm **??**, where MINIBATCH(p_θ, B) samples a minibatch of size B from $p_\theta(\tau)$ and INITIALIZERANKINGMODEL() initializes the parameters of the ranking model.

Remarks. Note that, to compute an empirical estimate of the gradient in Eq. 2.23, we only need to be able to sample from the user dynamics $p(\boldsymbol{X}(t) \mid \boldsymbol{X}(t-1), \boldsymbol{y}(t-1))$, since the explicit dependence cancels out within $S_\theta(\tau \mid \boldsymbol{X}(0), \boldsymbol{y}(0))$, as pointed out in Section 2.2. Moreover, depending on the choice of parameterized family of ranking models, one may be able to compute the score functions analytically. In our experiments, the class of Plackett-Luce (P-L) ranking

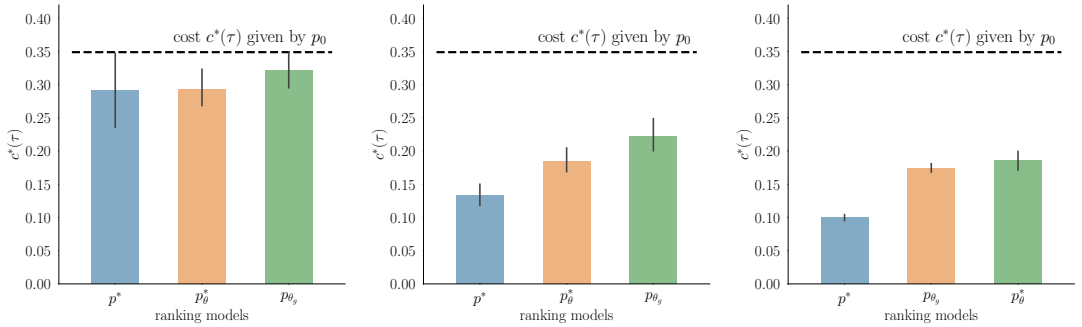
models allows for that. More specifically, it readily follows from Eq. 2.15 that

$$\begin{aligned}\nabla_\theta \log p_\theta(\tau \,|\, \boldsymbol{X}(0), \boldsymbol{y}(0)) &= \nabla_\theta \sum_{t=1}^{T} \sum_{k=1}^{n} \log f_k(\boldsymbol{X}(t)) \\ &= \nabla_\theta \sum_{t=1}^{T} \sum_{k=1}^{n} \left(\theta^T \boldsymbol{X} \omega_k(t) - \log \sum_{k'=k}^{n} \exp(\theta^T \boldsymbol{X} \omega_{k'}(t)) \right) \\ &= \sum_{t=1}^{T} \sum_{k=1}^{n} \left(\theta^T - \nabla_\theta \log \sum_{k'=k}^{n} \exp(\theta^T \boldsymbol{X} \omega_{k'}(t)) \right),\end{aligned}$$

where the second term within the logarithm in the last equation is the derivative of the log-sum-exp function, whose analytical expression can be found elsewhere. Finally, if we think of the parameterized ranking model p_θ as a policy, our algorithm resembles policy gradient algorithms used in the reinforcement learning literature [107]. This connection opens up the possibility of using variance reduction techniques used in policy gradient to improve the empirical estimation of the gradient [131].

2.2.3 Experiments on Synthetic Data

In this section, we compare the performance achieved by the original ranking models, which maximize an immediate measure of utility, the optimal consequential rankings models, implemented using Algorithm 1, the P-L consequential ranking model learned using Algorithm **??**, and a non-trivial greedy baseline, which downranks items with high values of cost to welfare in an heuristic manner, using synthetic data.



(a) $\bar{u} \in (0.852, 0.862)$ (b) $\bar{u} \in (0.862, 0.872)$ (c) $\bar{u} \in (0.872, 0.882)$

Figure 2.7: Performance of the original ranking model p_0, the optimal consequential ranking model p^*, implemented using Algorithm 1, the P-L consequential ranking model p^*_θ, learned using Algorithm **??**, and the greedy baseline p_{θ_g} on synthetic data. It shows the true cost to welfare $c^*(\tau)$ for three different ranges of average utility $\bar{u} = \sum_{t=1}^{T} u(t)/T$ for all models. Here, we tuned over the parameters λ (for p^* and p^*_θ) and d (for p_{θ_g}) to obtain the corresponding range for the average utility. The results show that the consequential ranking models p^* and p^*_θ outperform the greedy baseline p_{θ_g} in terms of the cost to welfare $c^*(\tau)$ and that the optimal consequential ranking model p^* performs best.

Experimental setup. Each trajectory has length $T = 20$ and, at each time step $t \in \{1, \ldots, T\}$, the ranking model receives a set $\mathcal{I}(t)$ of $n = 4$ posts and ranks them. Given a set of items $\mathcal{I}(t)$ and a ranking $\boldsymbol{y}(t)$, we assume that the set of items $\mathcal{I}(t+1)$ is just a copy of $\mathcal{I}(t)$ where the $d \sim \text{Poisson}(1)$ posts at the bottom of the ranking $\boldsymbol{y}(t)$ are replaced by new posts.

Each post i has two features $\boldsymbol{X}_i(t) = [r_i, a_i(t)]$, where r_i is the (static) probability that the post is misinformation and $a_i(t)$ is the (dynamic) rate of shares at time t, initialized with $a_i(0) = 0$. There are high-risk posts ($r_i = 0.6$) and low risk posts ($r_i = 0.1$) and a post is either high-risk or low-risk uniformly at random. Thus, whether the actual post is misinformation or not is a latent variable $m_i \sim \text{Bernoulli}(r_i)$, which is unobserved by the ranking model. The instantaneous rate of shares for each item i is given by:

$$a_i(t+1) = \exp(-2(t - s_i)) \times \tag{2.25}$$

$$(a_i(t) + \alpha_i + 0.02(5.0 - y_i(t))), \tag{2.26}$$

where s_i is the time when the post was first ranked by the ranking model, α_i is the virality, and a post is either viral ($\alpha_i = 10$) or non-viral ($\alpha_i = 0.1$) uniformly at random. Here, note that rate of shares of an item increases if the item is ranked at the top, as observed in previous empirical studies.

The original ranking model p_0 aims to rank posts according to the number of shares $a(t)$ at each time t, *i.e.*, its immediate utility $u(t)$ is defined as

$$u(t) = \zeta(t) \tag{2.27}$$

where $\zeta(t)$ is the Kendall-Tau correlation between the ordering induced by the ranking $\boldsymbol{y}(t)$ and the ordering induced by the sorted items according to $a(t)$. To this aim, it uses a Plackett-Luce (P-L) model, given by Eq. 2.15, with $\theta = [0, 20]$.

The cost to welfare measures the long-term presence of misinformation on the top position of the rankings. More specifically, it is defined as

$$c(\tau) = \frac{1}{T} \sum_{t=1}^{T} r_{\omega_k(t)}. \tag{2.28}$$

Moreover, we compare the original ranking model with three ranking models, which aim to trade off fidelity to the original model and the cost to welfare:

(i) An optimal consequential ranking model p^*, which is implemented using Algorithm 1.

(ii) A Plackett-Luce (P-L) consequential ranking model p^*_θ, which is learned using Algorithm **??** with $M = 100$ iterations and $B = 50$ as batch size.

(iii) A greedy baseline p_{θ_g}, which is a P-L ranking model with parameters $\theta = [-d,\ 20]$, which downranks items i with nonzero misinformation probability, *i.e.*, $r_i > 0$. Here, d is a given parameter that controls how much we downrank such items.

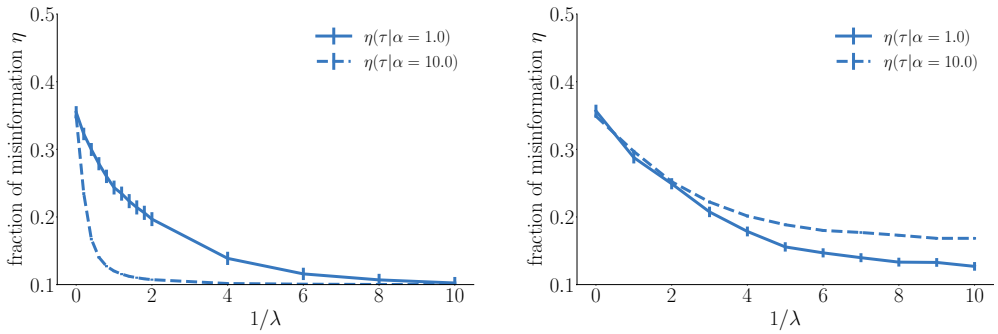
For the P-L consequential ranking model and the greedy baseline, we experiment with different values of the parameters λ and d, respectively. Finally, for each experiment, we perform 8,000 repetitions.

Quality of the rankings. We compare the original ranking model p_0, the optimal consequential ranking model p^*, the P-L consequential ranking model p^*_θ and the greedy baseline p_{θ_g} in terms

of two quality metrics: (i) the immediate utility $u(t)$, given by Eq. 2.27; and (ii) the true cost to welfare $c^*(\tau)$, defined as

$$c^*(\tau) = \frac{1}{T}\sum_{t=1}^{T} m_{\omega_1(t)}. \tag{2.29}$$

Figure 2.7 summarizes the results, which show that: (i) the (optimal and P-L) consequential ranking models outperform the greedy baseline in terms of the cost to welfare, for three different ranges of utility; (ii) the optimal consequential ranking achieves a significantly better tradeoff between the fidelity to the original ranking model and the cost to welfare, than the P-L ranking model, as one may have expected; and, (iii) the optimal consequential ranking model reduces the (true) cost to welfare without decreasing its fidelity to the original ranking model.



(a) Optimal consequential ranking model p^*_θ (b) Greedy baseline p_{θ_g}

Figure 2.8: Variation of % of misinformation in the top position for the optimal consequential ranking model p^*_θ, implemented using Algorithm 1, and the greedy baseline p_{θ_g} across both viral and non-viral posts. For the optimal consequential ranking model (panel (a)), as we increase $1/\lambda$, the fraction of misinformation for viral posts on the top 3 positions is significantly lower than the fraction of misinformation for non-viral posts. In contrast, for the greedy baseline (panel (b)), as we increase d, the fraction of misinformation η on the top position does not change significantly with the virality of the posts.

Viral vs. non-viral high risk posts. We investigate whether the optimal consequential ranking model and the greedy baseline treat viral and non-viral posts differently. Intuitively, the ranking model should be more willing to change the rank of high risk viral posts than that of high risk non-viral posts. To confirm this intuition, we compute the fraction of estimated and true misinformation, $\eta(\tau)$ and $\eta^*(\tau)$, in the top position of the rankings over time for both viral ($\alpha = 10$) and non-viral ($\alpha = 0.1$) posts, *i.e.*,

$$\eta(\tau|\alpha) = \frac{\sum_{t=1}^{T} p_{\omega_1}(t)\mathbb{I}(\alpha_{\omega_1(t)} = \alpha)}{\sum_{t=1}^{T} \mathbb{I}(\alpha_{\omega_1(t)} = \alpha)}$$

Figure 2.8 summarizes the results. For the optimal consequential ranking model, as we increase $1/\lambda$, the fraction of misinformation η for the viral posts on the top position is significantly lower than the fraction of misinformation for non-viral posts. In contrast, for the greedy baseline, as we increase d, the fraction of misinformation η on the top position does not change significantly

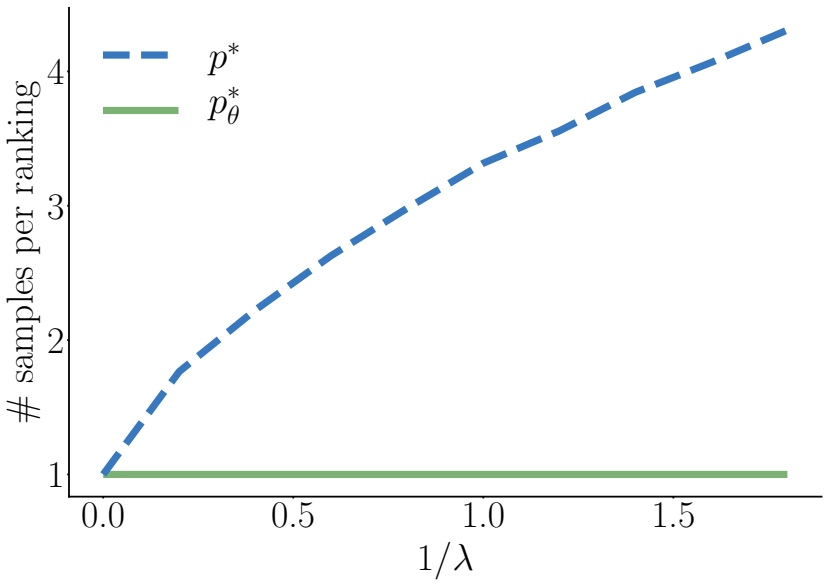


Figure 2.9: Sample complexity of the optimal consequential ranking model p^*_θ and the P-L consequential ranking model p_θ. The optimal consequential ranking model, implemented using Algorithm 1, becomes computationally prohibitive in terms of # samples needed per ranking as $1/\lambda$ increases and the difference between the original ranking model. This is in contrast with the P-L consequential ranking model, learned using Algorithm **??**, which exhibits much better sampling complexity with respect to $1/\lambda$.

with the virality of the posts.

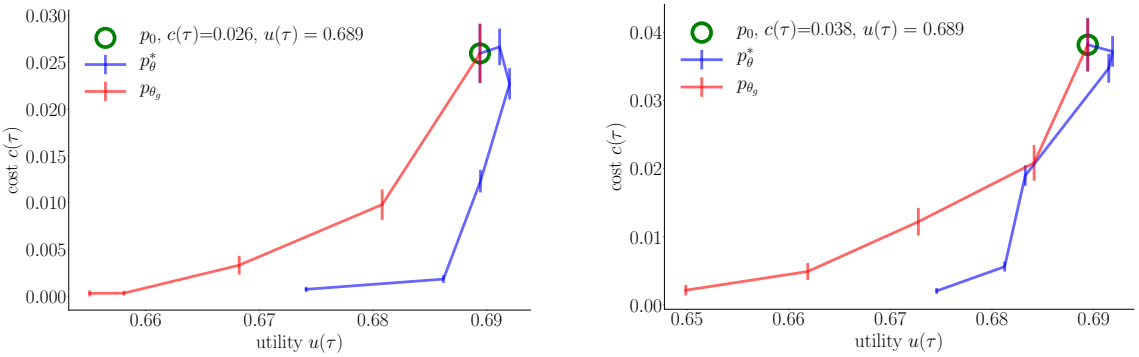
Sampling complexity. We compare the sampling complexity of the optimal consequential ranking model, implemented using Algorithm 1, and the P-L consequential ranking model, learned using Algorithm **??**. Figure 2.9 summarizes the results, which shows that, as $1/\lambda$ grows, it becomes computationally prohibitive to generate optimal consequential rankings using Algorithm 1 due to the growing difference between p^* and p_0. In contrast, the complexity of learning P-L consequential ranking model, using Algorithm **??**, stays constant as $1/\lambda$ changes.

2.2.4 Experiments on Real Data

In this section, we compare the performance achieved by the original ranking models, which maximize an immediate measure of utility, the P-L consequential ranking model learned using Algorithm **??**, and the same greedy baseline introduced in Section 2.2.3 using Reddit data[17]. Before we proceed further, we would like to acknowledge that:

(i) Since we do not have access to the ranking algorithm used by Reddit (or any other social media platform), our experiments are a proof of concept, which demonstrate the practical potential of our methodology on real data using a simple P-L ranking model. Evaluating the efficacy of our methodology across a wide range of deployed ranking algorithms is left as future work.

[17] Due to the size of the dataset, we were unable to run Algorithm 1 and thus we could not experiment with the optimal consequential rankings model.



(a) Uncivility

(b) Misinformation

Figure 2.10: Cost to welfare $c(\tau)$ vs. utility $u(\tau)$ achieved by the original ranking model p_0, the consequential PL-ranking model p^*_θ and the greedy baseline p_{θ_g} on Reddit data. The consequential PL-ranking model p^*_θ achieves a better trade off between the fidelity to the original ranking model ranking models optimizing immediate utility and the long-term welfare than the greedy baseline p_{θ_g}.

(ii) We consider a batch reinforcement learning setting. As a result, the rankings only influence the immediate utility and the cost of welfare but not the user dynamics. However, our evaluation is likely to be conservative—consequential rankings may achieve a greater reduction of the cost to welfare in an interventional experiment.

Dataset description. We used a publicly available Reddit dataset[18], which contains (nearly) all publicly available comments to link submissions posted by Reddit users from October 2007 to May 2015. In our experiments, we focused on the links submissions to the subreddit Politics and selected the set of submissions with more than 10 and less than 60 comments. After these preprocessing steps, our dataset comprised 3,173 submissions and 68,016 comments. The average length of a comment thread in our dataset is 21, with median of 17 and maximum length of 60. In a first set of experiments, we focus on the civility of the comments in each submission, as measured by an incivility score ϕ. In a second set of experiments, we focus on the misinformation spread by the comments of each submission, as measured by an unreliability score γ. In both sets of experiments, we use 1,973 submissions as training set for learning the parameterized consequential ranking models and the remaining 1,200 submissions as test set for evaluation, and we repeat our experiments for three different random sets of training and test sets.

Data preprocessing. In the first set of experiments using Reddit data, we focus on the civility of the comments in each submission. To this aim, we apply sentiment analysis on the text of the comments using the software package *Pattern*[19] and, for each comment, obtain two quantities: mood and polarity. The mood of a comment can take one of the following four values: indicative, imperative, conditional and subjunctive. The polarity of a comment is a real number in $[-1, 1]$, where lower (higher) values indicate more negative (positive) words in the text. Then, we define the incivility score ϕ of a comment as the absolute value of the polarity of the comment if the

[18] https://archive.org/details/2015_reddit_comments_corpus

[19] https://www.clips.uantwerpen.be/pages/pattern-en

polarity is negative and the mood of the comment is indicative or imperative and zero otherwise. Finally we apply a uniformly distributed quantile transformer to map the incivility scores to a value in $[0,1]$ with largest values always mapped to 1. Table 2.3 provides a few examples of sentences with a high value of ϕ.

Comment	Incivility (ϕ)
If you once tell a lie, the truth is ever after your enemy.	0.0
I dream of a world where your bigoted stupid ideas don't have the popular shield of faith.	0.1
Shut the f**k up and die already you POS warmongering profiteer.	0.4
Crap? Or pap. Take your pick.	0.8
i blame the evil KOCH BROTHERS!	1.0

Table 2.3: Examples of sentences with different levels of incivility, as estimated by the feature ϕ. Comments with higher levels of incivility typically correspond to those that use foul language.

In the second set of experiments, we focus on the misinformation spread by the comments of each submission. To this aim, we estimate the unreliability score γ for each comment by estimating the average unreliability score of the domains that appeared in each of them, as estimated by aggregating publicly available data from Politifact and Snopes[20]. More specifically, our combined dataset contains fact checking information for 17,804 unique urls from 4,540 unique domains. For each url, it assigns a label that indicates the reliability of its content. We used these labels to assign a numerical unreliability score for each url. More specifically, if the url is labeled as "false", "pants-fire", "mfalse" or "legend", we set the unreliability score to 1. If the url is labeled as "true", "mtrue" or "mostly-true", then we set the unreliability score to -1. And, if the url is labeled using some other label value, we set the unreliability score to 0. We computed an unreliability score for each domain, which measures its level of (un)trustworthiness, by taking the average of the unreliability scores of individual urls from the domain. Then, we define the unreliability score γ of a comment as the average unreliability score of the domain(s) of the link(s) used in the comment if the average is negative and zero otherwise. Here, also note that, if a comment does not contain any links or the domain(s) of the link(s) does not appear in our dataset, we set the unreliability score for that comment to 0. Finally we apply a uniformly distributed quantile transformer to map the unreliability scores to a value in $[0,1]$ with largest values always mapped to 1. Table 2.4 provides a few domains with a different values of γ.

Experimental setup. Each submission corresponds to one trajectory whose length T is just the number of comments in the submission, *i.e.*, each time step corresponds to the time at which a new comment was created. Then, at each time step $t \in \{0,\ldots,T\}$, the ranking model ranks the latest set of $n = 5$ comments $\mathcal{I}(t)$[21].

Each comment i has three features $\boldsymbol{X}_i(t) = [l_i, \phi_i, \gamma_i]$, where l_i is the number of comments posted until time step i, ϕ_i is the incivility score and γ_i is the unreliability score. At each time t, the original ranking model p_0 aims to promote the most recent comment to the top of the

[20] https://www.kaggle.com/arminehn/rumor-citation/version/3

[21] Experiments with $n > 5$ give qualitatively the same results because comments with high score of incivility/unreliability are rare in the dataset.

Url	Misinformation (γ)
aids.gov	0.0
pbs.org	0.26
breitbart.com	0.56
lifeisajoke.com	1.0

Table 2.4: Examples of domains that spread different amounts of misinformation, as estimated by the feature γ.

ranking, *i.e.*, its immediate utility $u(t)$ is defined as

$$u(t) = \zeta(t) \tag{2.30}$$

where $\zeta(t)$ is the Kendall Tau correlation between the ordering induced by the ranking $\boldsymbol{y}(t)$ and the inverse chronological ordering. To this aim, it uses a Plackett-Luce (P-L) model, fitted by maximizing the likelihood function over traces with reverse chronological order.

In the first set of experiments, the cost to welfare measures the long-term presence of uncivil comments on the top position of the rankings. More specifically, it is defined as

$$c(\tau) = \frac{1}{T}\sum_{t=1}^{T} \phi_{\omega_1(t)}. \tag{2.31}$$

In the second set of experiments, the cost to welfare measures the long-term presence of unreliable comments on the top position of the rankings. More specifically, it is defined as

$$c(\tau) = \frac{1}{T}\sum_{t=1}^{T} \gamma_{\omega_1(t)}. \tag{2.32}$$

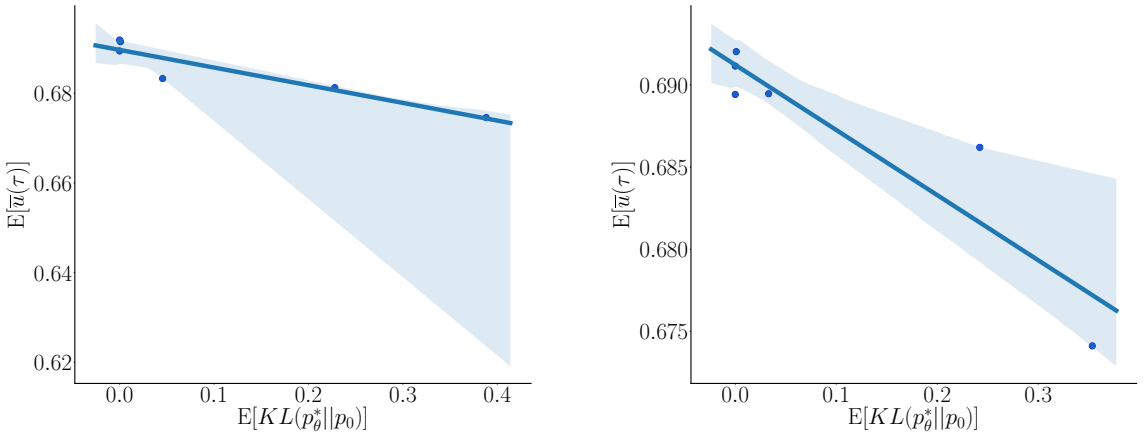
Similarly as in Section 2.2.3, we compare the original ranking model with two ranking models, which aim to trade off fidelity to the original model and the cost to welfare:

(i) A Plackett-Luce (P-L) consequential ranking model p^*_θ, which is learned using Algorithm **??** with $M = 20$ iterations and $B = 100$ as batch size; and,

(ii) A greedy baseline p_{θ_g} with parameters $\theta = [1,\ -d,\ 0]$ for the first set of experiments and $\theta = [1,\ 0,\ -d]$ for the second set of experiments. Here, the greedy baseline downranks items i with nonzero incivility (or unreliability) score *i.e.*, $\phi_i > 0$ (or $\gamma_i > 0$).

For both ranking models (i-ii), we experiment with different values of the parameters λ and d, respectively. Finally, for each experiment, we perform 8,000 repetitions.

Results. We first compare the original ranking model p_0, the consequential P-L ranking models p^*_θ and the greedy baseline p_{θ_g} in terms of the tradeoff between cost to welfare $c(\tau)$ and the immediate utility given by Eq. 2.30. Here, note that, in the first set of experiments, the cost to welfare measures the degree of incivility of the top ranking positions (Eq. 2.31) while, in the second set of experiments, it measures the amount of misinformation (Eq. 2.32). Figure 2.10 summarizes the results, which shows that (i) our consequential PL-ranking model



(a) Misinformation

(b) Incivility

Figure 2.11: Average immediate utility $\mathbb{E}[\bar{u}(\tau)]$ achieved by the consequential ranking model vs the average Kullback-Leibler (KL) divergence $\mathbb{E}[KL(p^*_\theta \,||\, p_0)]$ between its induced probability $p^*_\theta(\tau)$ and the probability $p_0(\tau)$ induced by the original ranking model. It shows that there is a negative correlation between the immediate utility and the KL divergence for both the experiments.

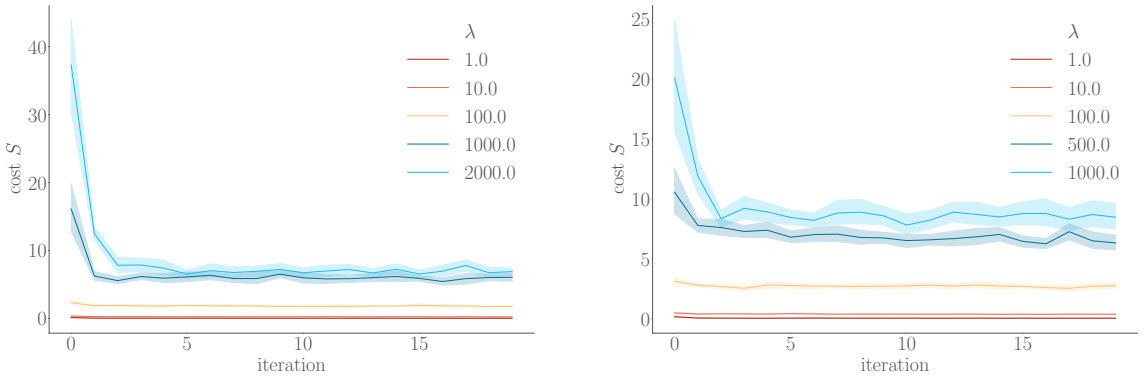
p^*_θ can trade off between the fidelity to ranking models optimizing immediate utility and the long-term welfare more effectively than the greedy baseline p_{θ_g}, and (ii) the PL-ranking model p^*_θ is able to reduce the degree of incivility and the amount of misinformation at the top ranking positions without significant changes to the original reverse chronological ranking.

So far, we have assumed that there is a negative correlation between the immediate utility achieved by the consequential ranking model and the Kullback-Leibler (KL) divergence between its induced probability $p^*_\theta(\tau)$ and the probability $p_0(\tau)$ induced by the original ranking model. Here, we verify this assumption by looking into the variation of $\mathbb{E}[\bar{u}(\tau)]$ with $\mathbb{E}[KL(p^*_\theta \,||\, p_0)]$. Figure 2.11 demonstrates the results, which show that there is indeed a negative correlation between the immediate utility and the KL divergence for both the experiments.

Finally, we also compute the average cost S per iteration during training. Figure 2.12 summarizes the results, which show that, as λ increases, the model takes longer to converge.

2.2.5 Conclusions

We have initiated the design of (parameterized) consequential ranking models that optimally trade off between (1) the fidelity to ranking models optimizing for immediate utility and (2) long-term welfare. More specifically, we have first introduced a joint representation of rankings and user dynamics using Markov decisions processes. Exploiting this representation, we have shown that we can obtain optimal consequential rankings just by applying weighted sampling on the rankings provided by the model optimizing for immediate utility. However, in practice, such a strategy may be inefficient and impractical, specially in high dimensional scenarios. To overcome this, we introduced an efficient gradient-based algorithm to learn parameterized consequential ranking models that effectively approximate the optimal ones. Finally, we have



(a) Uncivility

(b) Misinformation

Figure 2.12: Average cost S per iteration during training. The results show that, as λ increases, the model takes longer to converge.

experimented on synthetic and real data to show the efficacy of our parameterized consequential ranking models.

Our work opens up several venues for future work. For example, we have considered probabilistic ranking models and a fidelity measure based on KL divergence. A natural next step is to augment our methodology to allow for deterministic ranking models and consider other fidelity measures between rankings. Finally, we have evaluated our algorithm using observational real data, however, it would be very valuable to perform interventional experiments.

Chapter 3

Analysis of the Reliability of Neurips 2016 Conference Reviewing Process

Introduction

The review process for NIPS 2016 involved 2,425 papers submitted by 5,756 authors, 100 area chairs, and 3,242 active reviewers submitting 13,674 reviews in total. Designing a review process as fair as possible at this scale was a challenge. In order to scale, all parts of the process have to be as decentralized as possible. Just to get a feeling, if the two program chairs were supposed to take final decisions just for the 5% most challenging submissions, which means that they would have to read and decide on 150 papers — this is the scale of a whole conference such as COLT. Furthermore, the complexity of the logistics and software to manage the review process is rather high already. A controlled experiment [62] from NIPS 2014 has shown that there is a high disagreement in the reviews. Hence the primary goal must be to keep bias and variance of the decisions as small as possible.

In this section, we present an analysis of many aspects of the data collected throughout the review phase of the NIPS 2016 conference, performed subsequent to the completion of the review process. Our goal in this analysis is to examine various aspects of the data collected from the peer review process to check for any systematic issues. Before delving into the details, the reader should importantly note the following limitations of this analysis:

- There is no ground truth ranking of the papers or knowledge of the set of papers which should ideally have been accepted.
- The analysis is post hoc, unlike the controlled experiment from NIPS 2014 [62].
- The analysis primarily evaluates the ratings and rankings provided by reviewers, and does not study the textual comments provided by the reviewers.

The analysis is used to obtain insights into the peer-review process, usable suggestions for subsequent conferences, and important open problems towards improving peer-review in academia.

Here is a summary of our findings:

(i) there are very few positive bids by reviewers and area chairs (Section 3.2),

(ii) graph-theoretic techniques can be used to ensure a good reviewer assignment (Section 3.2),

(iii) there is significant miscalibration with respect to the rating scale (Section 3.2),

(iv) review scores provided by invited and volunteer reviewers have comparable biases and variance; junior reviewers report a lower confidence (Section 3.2),

(v) there is little change in reviewer scores after rebuttals (Section 3.2),

(vi) there is no observable bias towards any research area in accepted papers (Section 3.2),

(vii) there is lower disagreement among reviewers in NIPS 2016 as compared to NIPS 2015 (Section 3.2),

(viii) significant fraction of scores provided by the reviewers are tied and ordinal rankings can ameliorate this issue (Section 3.2),

(ix) there are some inconsistencies in the reviews and these can be identified in an automated manner using ordinal rankings (Section 3.2).

We describe the review procedure followed at NIPS 2016 in Section 3.1. We present an elaborate description of the analysis and the results in Section 3.2. Alongside each analysis, we present a set of key observations, action items for future conferences, and some open problems that arise out of the analysis. We conclude the paper with a discussion in Section 3.3.

3.1 Review Procedure

In this section, we present an overview of the design of the review process at NIPS 2016.

Selecting area chairs and reviewers

Area Chairs (ACs) are the backbone of the NIPS reviewing process. Their role is similar to that of an associate editor for a journal. Each AC typically handles 20-30 submissions, so with an estimated number of submissions between 2000 and 3000, we needed to recruit about 100 area chairs. As it is impossible to intimately know all the diverse research areas covered by NIPS, we came up with the following procedure. We asked the NIPS Board and all the ACs of NIPS from the past two years to nominate potential ACs for this year. In this manner, we covered the entire variety of NIPS topics and obtained qualified suggestions. We obtained around 350 suggestions. We asked the NIPS Board to go through the list of suggested ACs and vote in favor of suggested ACs. We also accounted for the distribution of subject areas of submitted papers of the previous year's NIPS conference. Combining all these inputs, we compiled a final list of ACs: by the end of January we had recruited exactly 100 ACs. In a subsequent step, we formed "buddy pairs" among the ACs. Based on the ACs preferences, each AC got assigned a buddy AC. We revisit the role of buddy pairs in more detail later.

The process of **recruiting reviewers** is time consuming, it essentially went on from January until the submission deadline at end of May. A significant departure from the review processes of NIPS from earlier years, this time we had two kinds of reviewers, "invited senior reviewers" (Pool 1) and "volunteer reviewers" (Pool 2):

- **Pool 1, invited senior reviewers:** We asked all ACs to suggest at least 30 reviewers who have completed their PhDs (however, this requirement was not strictly observed by all ACs). We obtained 2500 suggested experienced reviewers. We invited all of them, and 1100 accepted. We then asked all confirmed reviewers to "clone themselves" by inviting at least one researcher with a similar research background and with at least as good a qualification as themselves. This resulted in an additional 500 experienced reviewers.

- **Pool 2, volunteer author-reviewers:** The rapid growth in the number of submissions at NIPS poses the formidable challenge of accordingly scaling the number of reviewers. An obvious mean to achieve this objective is to ask authors to become reviewers as well. This idea has been used in the past, for example, to evaluate NSF grant proposals [77] or to allocate telescope time [76]. In order to implement this idea, without constraining unwilling authors, we requested authors to volunteer during the submission process by naming at least one author per paper as volunteer reviewers. We invited all of them and about 2000 of the volunteers accepted the invitation.

The area chairs were aware of the respective pools to which each of their reviewers belonged. The number of reviewers that we eventually ended up with are as follows:

	Senior researchers / faculty	Junior researchers / postdocs	PhD students
Pool 1: Invited reviewers	1236	566	255
Pool 2: Volunteer reviewers	143	206	827

Assignment of papers to reviewers and area chairs

The assignment of papers to area chairs was made in the following manner. Prior to the review process, the ACs (and reviewers) were allowed to see the list of submitted papers and "bid" whether they were interested or disinterested in handling (or reviewing) any paper. For any paper, an AC (or reviewer) could either indicate "Not Willing" or "In-a-pinch" – which we count as negative bids, or indicate "Willing" or "Eager" – which we count as positive bids, or choose to not bid for that paper. The Toronto paper matching system or TPMS was then employed to compute an affinity score for every AC (and reviewer) with every submitted paper based on the content of the paper and the academic profile of the AC or reviewer. In addition, every AC (and reviewer) as well as the submitter of every paper was asked to select a set of most relevant subject areas, and these subject areas were also employed to compute a similarity between each AC (and reviewer) and paper.

Based on the similarity scores and bids, an overall similarity score is computed for every {paper, AC} and every {paper, reviewer} pair: score $= b(s_{\text{affinity}} + s_{\text{subject}})$, where $s_{\text{affinity}} \in [0,1]$ is the affinity score obtained from TPMS, $s_{\text{subject}} \in [0,1]$ is the score obtained by comparing the subject areas of the paper and the subject areas selected by the AC or reviewer, and $b \in [0.25,1]$ is the bidding score provided by the AC or reviewer. Based on these overall similarity scores, a preliminary paper assignment to ACs was then produced in an automated manner using the TPMS assignment algorithm [20]. The ACs were given a provision to decline handling certain

papers for various reasons such as conflicts of interest. These papers were re-assigned manually by the program chairs.

The AC of each paper was responsible to first assign one senior, highly qualified reviewer manually. Two more invited reviewers from pool 1 and three volunteer reviewers from pool 2 were then assigned automatically to each paper using the same procedure as described above. The ACs were asked to verify whether each of their assigned papers had at least 3 highly competent reviewers; the ACs could manually change reviewer assignments to ensure that this is the case. During the decision process, additional emergency reviewers were invited to provide complementary reviews if some of the reviewers had defected or if no consensus was reached among the selected reviewers.

Review criteria and scores

We completely changed the scoring method this year. In previous years, NIPS papers were rated using a single score between 1 and 10. A single score alone did not allow reviewers to give a differentiated quantitative appreciation on various aspect of paper quality. Furthermore, the role of the ACs was implicitly to combine the decisions of the reviewers (late integration) rather than combining the reviews to make the final decision (early integration). Introducing multiple scores allowed us to better separate the roles: the reviewers were in charge of evaluating the papers; the ACs were in charge of making decisions based on all the evaluations. Furthermore the multiple specialized scores allowed the ACs to guide reviewers to focus discussions on "facts" rather than "opinion" in the discussion phase. We asked reviewers to provide a separate score for each of the following four features:

- Technical quality,
- Novelty/originality,
- Potential impact or usefulness,
- Clarity and presentation.

The scores were on a scale of 1 to 5, with the following rubric provided to the reviewers:

5 = Award level (1/1000 submissions),

4 = Oral level (top 3% submissions),

3 = Poster level (top 30% submissions),

2 = Sub-standard for NIPS,

1 = Low or very low.

The scoring guidelines also reflect the hierarchy of the papers: the conference selects the top few papers for awards, the next best accepted papers are presented as oral presentations, and the remaining accepted papers are presented as posters at the conference. The scores provided by reviewers had to be complemented by justifications in designated text boxes. We also asked the reviewers to flag "fatal flaws" in the papers they reviewed. For each paper, we also asked the reviewers to declare their overall "level of confidence":

3 = Expert (read the paper in detail, know the area, quite certain of opinion),

2 = Confident (read it all, understood it all reasonably well),

1 = Less confident (might not have understood significant parts).

Discussions and rebuttals

Once most reviews were in, authors had the opportunity to look at the reviews and write a rebuttal. One section of the rebuttal was revealed to all the reviewers of the paper, and a second section was private and visible only to the ACs. Some reviews were still missing at this point, but it would not have helped to delay the rebuttal deadline as the missing reviews trickled in only slowly. Subsequently, ACs and reviewers engaged in discussions about the pros and cons of the submitted papers. To support the ACs, we sent individual reports to all area chairs to flag papers whose reviews were of too low confidence, too high variance or where reviews were still missing. In many cases, area chairs recruited additional emergency reviewers to increase the overall quality of the decisions.

Decision procedure

The decision procedure involved making an acceptance or rejection decision for each paper, and furthermore, to select a subset of (the best) accepted papers for oral presentation.

We introduced a decentralized decision process based on pairs of ACs ("buddy pairs"). Each AC got assigned one buddy AC. Each pair of buddy ACs was responsible for all papers in their joint bag and made the accept/reject decisions jointly, following guidelines given by the program chairs. Difficult cases were taken to the program chairs, which included cases involving conflicts of interest and plagiarism. In order to harmonize decisions across buddy pairs, all area chairs had access to various statistics and histograms over the set of their papers and the set of all submitted papers. To decide which accepted paper would get an oral presentation, each buddy pair was asked to champion one or two papers from their joint bag as a candidate for an oral presentation. The final selection was then made by the program chairs, with the goals of exhibiting the diversity of NIPS papers and exposing the community with novel and thought provoking ideas. In the end, 568 papers got accepted to the conference, and 45 of these papers were selected for oral presentations.

Like previous years, we adopted a "double blind" review policy. That is, the author(s) of each paper did not get to know the identity of the reviewers and vice versa throughout the review process. ACs got to know the identity of the reviewers and the author(s) for the papers under their responsibility. During the discussion phase, reviewers who reviewed the same papers got to know each others' identity. Lastly, PCs and program managers had access to all information about the submissions, the ACs, the reviewers, and the authors.

Experimental ordinal reviews

In the main NIPS 2016 review process, we elicited only cardinal scores from the reviewers – one score in 1 to 5 for each of four features. Subsequent to the review process, we then requested each reviewer to also provide a total ranking of the papers that they reviewed. We received rankings from a total of 2189 reviewers. Note that the collection of ordinal data was performed subsequent to the normal review submission but before release of the final decisions. The ordinal data was not used as a part of the decision procedure in the conference.

3.2 Detailed Analysis

In this section, we present details of our analyses of the review data and the associated results. Each subsection contains one analysis and concludes with a summary that highlights the key observations, concrete action items for future conferences, and open problems that arise from the analysis.

The results are computed for a snapshot of reviews at the end of the review process when the acceptance decisions were made. This choice does not affect the results since there was very little change in the scores provided by reviewers across different time instants. All t-tests conducted correspond to two-sample t-tests with unequal variances. All mentions of p-values correspond to two-sided tail probabilities. All mentions of statistical significance correspond to a p-value threshold of 0.01 (we also provide the exact p-values alongside). Multiple testing is accounted for using the Bonferroni correction. The effect sizes refer to Cohen's d. Wherever applicable, the error bars in the figures represent 95% confidence intervals.

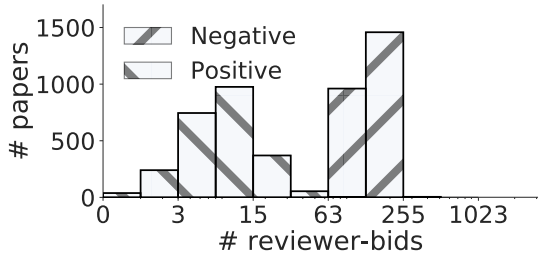
Wherever applicable, we also perform our analyses on a subset of the submitted papers which we term as the top 2k papers. The top 2k papers comprise all of the 568 accepted papers, and an equal number (568) of the rejected papers. The 568 rejected papers are chosen as those with maximum scores averaged across all reviewers and all features.

Reviewer bids

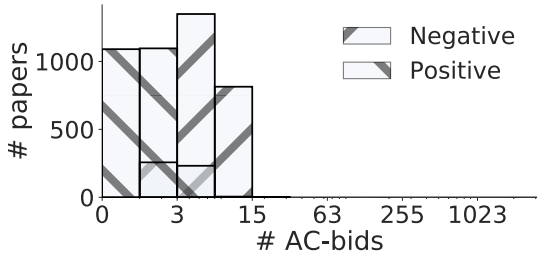
A large number of conferences in computer science ask area chairs and/or reviewers to bid which papers they would like or not like to review, in order to obtain a better understanding of the expertise and the preferences of reviewers. Such an improved understanding is desirable as it leads to a more informed assignment of reviewers to papers, thereby improving the overall quality of the review process.

Figure 3.1 depicts the distribution of number of bids on papers submitted by area chairs and reviewers in NIPS 2016. Panels (a) and (b) of the figure depict the distribution of counts per paper for reviewers and area chairs respectively; panels (c) and (d) depict the distribution per area chairs and reviewers. From the data, we observe that there are very few positive bids, but a considerably higher number of negative bids.

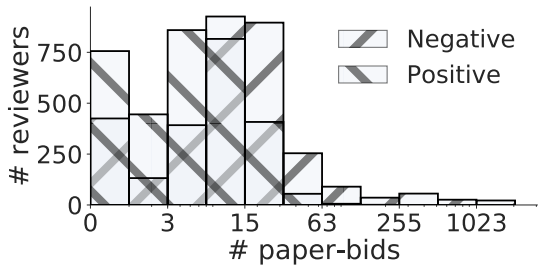
The distribution of number of bids by reviewers is skewed by few reviewers who bid (positive and negative) on too many papers: 27% of reviewers make 90% of all bids, and 50% of reviewers make 90% of all positive bids. Moreover, there are 148 reviewers with no (positive or negative) bids and 1201 reviewers with at most 2 positive bids. In comparison, NIPS 2016 assigned at least 3 papers to most reviewers and many conferences do likewise. We thus observe that a large number of reviewers do not even provide positive bids amounting to the number of papers they would review. As a consequence of the low number of bids by reviewers, we are left with 278 papers with at most 2 positive bids and 816 papers with at most 5 positive bids. In contrast, NIPS 2016 assigned 6 reviewers to most papers. There is thus a significant fraction of papers with fewer positive bids than the number of requisite reviewers. Finally there are 1090 papers with no positive bids by any AC.



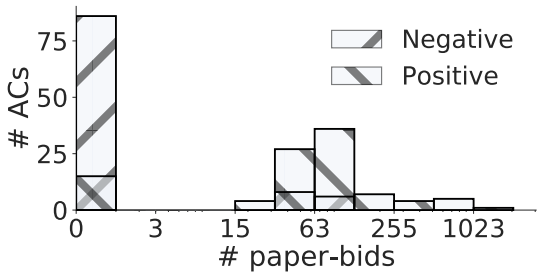
(a) total number of bids by reviewers per paper



(b) total number of bids by ACs per paper

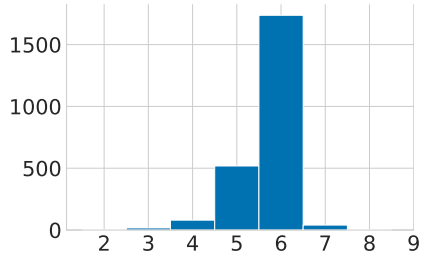


(c) number of bids per reviewer

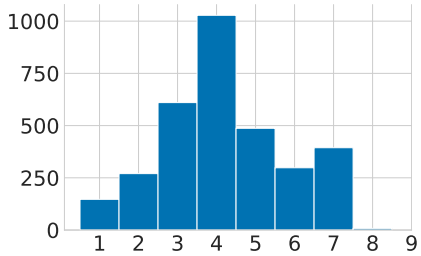


(d) number of bids per AC

Figure 3.1: Histogram of number of positive and negative bids (x-axis; on a logarithmic scale) per entity (counts on y-axis) for various entities. The "not willing" and "in-a-pinch" bids were considered negative bids, whereas "willing" and "eager" bids were considered positive bids. The first column in each histogram represents number of entities with 0 bids. For example, the first column of panel (c) depicts that 756 reviewers made zero positive bids and 425 reviewers made zero negative bids.



(a) number of reviewers per paper



(b) number of papers per reviewer

Figure 3.2: Histogram of number of reviews.

Reviewer assignment

Figure 3.2 depicts the histograms of the number of reviewers assigned per paper, and the number of papers handled by each reviewer.

In order to ensure that the information about each paper "spreads" across the entire system, it is important that there is no set of reviewers or papers that has only a small overlap with the remaining reviewers and papers [82, 99]. To analyze whether this was the case, we considered two graphs. We built a **reviewer graph** that has reviewers as vertices, and an edge between any two reviewers if there exists at least one paper that has been reviewed by both of them.

Analogously we built a **paper graph**, where vertices represent papers, and we connect two papers by an edge if there exists a reviewer who has reviewed both papers. Note that the graph structure is in part dictated by a constraint on the maximum number of papers per reviewer, as well as the specified number of reviewers per paper.

Our objective is to examine the structure of the graphs and determine if there were any separated communities of nodes. In order to do so, we employ a method based on spectral clustering. Formally, denote any graph as $G = (V, E)$ where V is set of nodes, and E is the set of (undirected) edges between nodes, and $|V|$ is number of nodes in the graph. We can denote graph connectivity by its associated adjacency matrix A which is a $(|V| \times |V|)$ matrix; we have $A_{ij} = 1$ if there is an edge between nodes i and j and $A_{ij} = 0$ otherwise. With this notation, a quantity known as the "conductance" Φ of any set of nodes $S \subset V$ is then defined as:

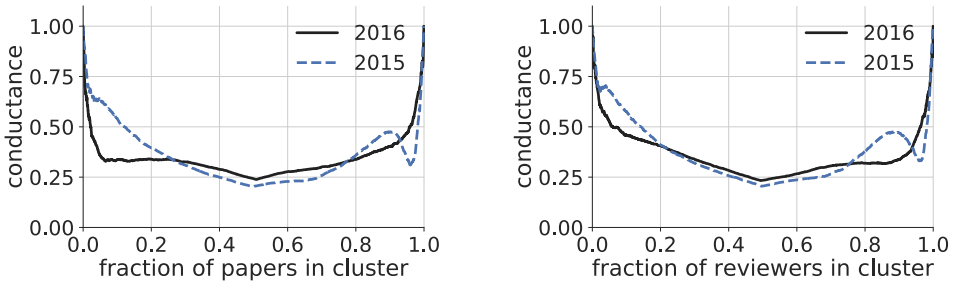
$$\Phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\max\{|S|, |V \backslash S|\}},$$

where $V \backslash S$ is the complement of set S. A lower value of the conductance indicates that the nodes in the cut are less connected to the remaining graph. Next, with a minor abuse of notation, the conductance of a graph as function of cluster sizes is defined as:

$$\Phi(k) = \min_{S \in V, |S| = k} \Phi(S),$$

for every $k \in \{1, \ldots, |V| - 1\}$. The plot of k versus $\Phi(k)$ is called a Network Community Profile or NCP plot [65]. The NCP plot measures the quality of the least connected community (lowest conductance) in a large network, as a function of the size of the community. Although computing the function $\Phi(k)$ exactly may be computationally hard, an approximate value can be computed using a simple "second left eigenvector" procedure (Section 2.3 of 8). A well connected graph would have a smooth plot of $\Phi(k)$ with a minima at around $k = |V|/2$.

Figure 3.3 shows the NCP plot for an increasing number of papers (respectively reviewers) in the paper graph (respectively reviewer graph). For reference we also plot the same curve for graphs associated with NIPS 2015 conference. Both plots for NIPS 2015 have local minima at around $k = 0.96|V|$, indicating that there is a densely connected community of reviewers and



(a) paper graph (b) reviewer graph

Figure 3.3: Conductance value as function of varying cluster size. The x-axes in these plots is the normalized cluster size $k/|V|$.

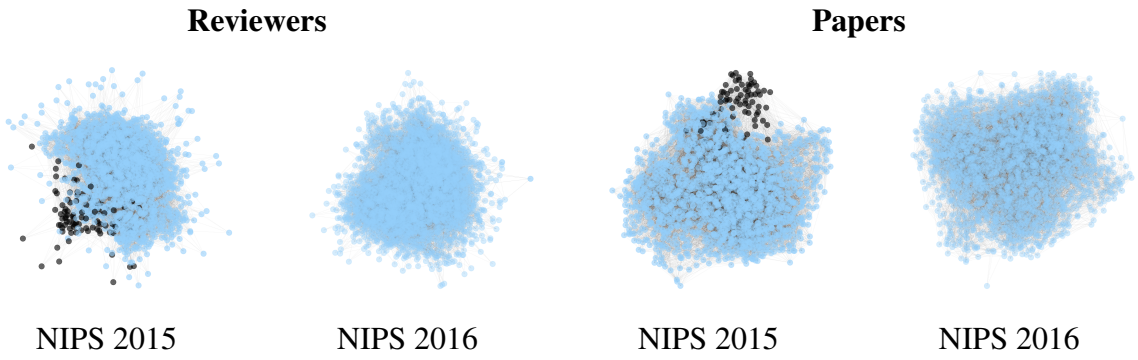


Figure 3.4: Graphs depicting connectivity of reviewers and that of papers for NIPS 2015 and NIPS 2016. The nodes in black (dark) show set of nodes identified by the local minima in the conductance plots (Figure 3.3) for NIPS 2015, and the remaining nodes are plotted in blue (light).

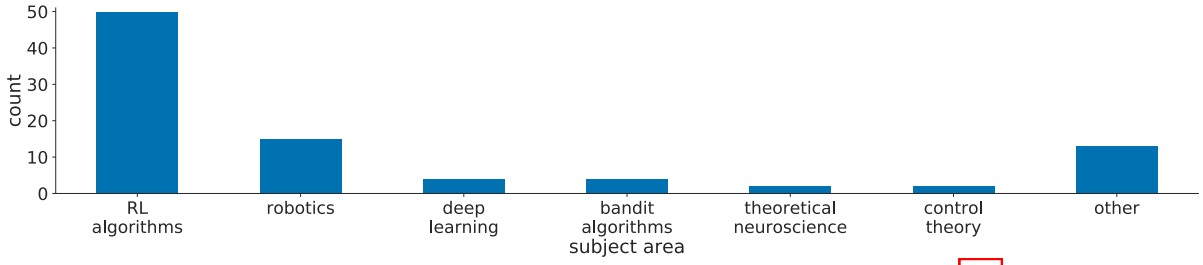


Figure 3.5: Histogram of subject areas in the identified cluster (from Figure 3.4) of reviewers in NIPS 2015 which is not well connected with the set of remaining reviewers.

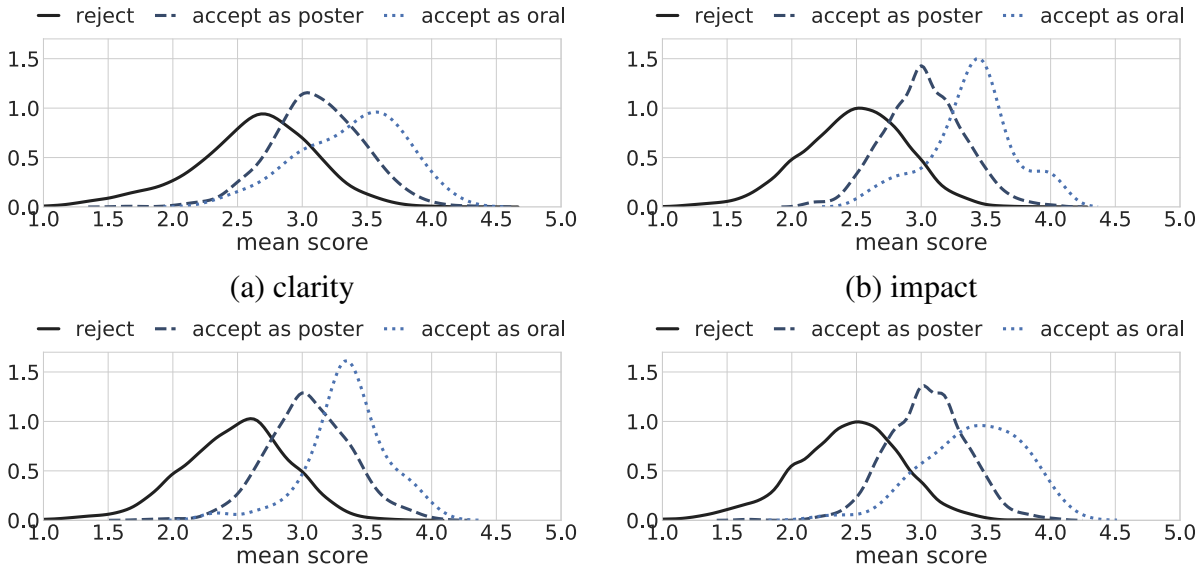
papers that are not well connected with the rest of the graph. In contrast, the plot associated with NIPS 2016 decreases smoothly and reaches its global minimum when half of the nodes are in one cluster and the other half in another cluster, indicating an absence of such a fragmentation.

In Figure 3.4, we plot the graph of reviewers and papers using the algorithm of [38]. In these figures we identify the set of nodes that are identified using the aforementioned NCP method; these nodes are colored black (dark) in the figure in contrast to the blue (light) color of the remaining nodes. We can see from the Figure 3.4 that these nodes are on the periphery of the network with lower connectivity compared to the rest of the graph.

We further examine the cluster of reviewers in NIPS 2015 which is not well connected with the rest. In Figure 3.5, we plot the decomposition of this set in terms of the primary subject areas indicated by the reviewers. Our analysis reveals that a bulk of this cluster comprises a single subject area—reinforcement learning. Conversely, 50 out of 78 reviewers who identified their primary subject area as reinforcement learning lie in this cluster. All in all, graph connectivity issues of this form can lead to increased noise or bias in the overall decisions. Our main message for future conferences is to employ such methods of graph analysis in order to catch issues of this form *at a global level* (not just local to individual ACs) before the reviews are assigned.

Review-score distribution and mismatches in calibration

Recall from Section 3.1 that in the review process, for each feature, the reviewers were asked to provide a score on a scale of 1 to 5. Specifically, they were asked to provide a score of 5



(a) clarity (b) impact

(c) novelty (d) quality

Figure 3.6: Distribution of the mean value (across reviewers) of the score per paper for different features, separated according to the final decisions.

for submissions they considered as being in the top 0.1%, a score of 4 for submissions that they deemed to be in the top 3%, and a score of 3 for submissions they deemed to be in the top 30%. In this section, we compare the actual empirical distribution of reviewer scores with the distribution prescribed in the guidelines to reviewers.

We begin by computing the distribution of the mean value (across reviewers) of the score per paper for different features, separated according to the final decisions. We plot these distributions in Figure 3.6 for each of the four features of clarity, impact, novelty, and quality separately.

At first glance, these histograms and numbers look quite reasonable. However, what was surprising to us was the percentage of papers that received any particular score – see Table 3.1. Even though the reviewers were asked to give a paper a score of 3 (poster level) or higher only if they think the paper lies in the top 30% of all papers, nearly 60% of the scores were 3 or higher. Similar effects occurred for scores 4 and 5.

	1 (low or very low)	2 (sub-standard)	3 (poster level: top 30%)	4 (oral level: top 3%)	5 (award level: top 0.1%)
Impact	6.6 %	36.4 %	45.9 %	10.7 %	0.4 %
Quality	6.7 %	38.3 %	45.0 %	9.6 %	0.4 %
Novelty	6.4 %	35.0 %	48.4 %	9.8 %	0.4 %
Clarity	7.1 %	28.1 %	48.9 %	14.7 %	1.2 %

Table 3.1: Distribution of the reviews according to the provided scores for each of the four features. The column headings indicate the guidelines that were provided to the reviewers. Observe that the percentage of reviews providing scores of 3, 4 or 5 is considerably higher than the requested values.

One possible explanation for this phenomenon is that there were a large number of high-quality submissions to NIPS 2016. Such an improvement in quality has obvious upsides such as uplifting the overall experience of the conference. The downside is that the burden on selecting the accepted papers among all those good submissions is with the area chairs, who now still had to reduce the 60% good papers to 23% accepted papers. A second possible explanation is that the reviewers were not calibrated that well with respect to the paper quality. In either case, we understand that this obviously led to the frustration of many authors, whose papers received good scores but were rejected.

In addition to scores for the four features, the reviewer could also indicate whether the paper had a "fatal flaw". We observe that 32% of all papers were flagged to have a "fatal flaw" by at least one reviewer.

Behavior of different pools of reviewers

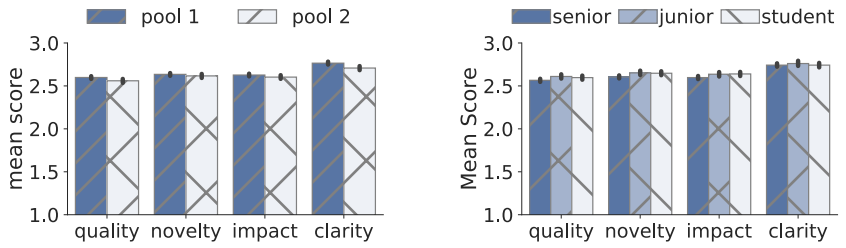
In this section, we compare the reviews provided by the volunteer (pool 2) reviewers to those provided by the invited (pool 1) reviewers. The inclusion of volunteer reviewers has two important benefits: (a) It increases the transparency of the review process. (b) Volunteer reviewers may be new today but in 2 years down the line they will gain experience and become useful to accommodate the massive growth of the conference. Given these benefits of including volunteer reviewers, this analysis looks for any systematic differences between the review scores provided by the two pools of reviewers.

Mean scores. Junior reviewers are often perceived to be more critical than senior reviewers [109, 110]. As [109] notes, *"You submit your manuscript and then just pray it doesn't get sent to a junior faculty member – young faculty are merciless!"* In this section, we examine this hypothesis in the NIPS 2016 reviews. In Figure 3.7, we plot the mean score provided by each group of reviewers for each individual feature. We apply a t-test on observed scores and compute the effect size to examine if there is a statistically significant difference in the underlying means of the scores provided by different categories of reviewers. For Pool 1 vs Pool 2, this analysis shows only clarity to have a statistically significant difference between the two pools after accounting for multiple testing. Specifically, the p-values (before accounting for multiple testing) and effect sizes for the four features are: novelty p=0.2143, d= 0.0264, quality p=0.0061, d= 0.0581, impact p=0.0961, d= 0.0353, and clarity p=1.91×10^{-04}, d= 0.0788. Sample sizes for Pool 1 and Pool 2 reviews are 9244 and 4430 respectively.

A similar analysis between senior researchers (e.g., faculty), junior researchers (e.g., postdocs), and PhD students reveals no significant difference between these categories. Specifically, the p-values (before accounting for multiple testing) and effect sizes for senior researcher vs. junior researchers for the four features are: quality p=0.0071, d= -0.0662, novelty p=0.0037, d= -0.0704, impact p=0.0199, d= -0.0569, and clarity p=0.3064, d= -0.0253; for junior researcher vs. students: quality p=0.4662, d= 0.0164, novelty p=0.8247, d= 0.0049, impact p=0.8733,d= -0.0036, and clarity p=0.3529, d= 0.0209; for senior researcher vs. students: quality p=0.0440, d= -0.0454, novelty p=0.0499, d= -0.0629, impact p=0.0076, d= -0.0601 and clarity p=0.9968, d= 0.00009. The sample sizes for senior, junior and student reviews are: 6335, 3938, and 3354 respectively. This analysis excludes 47 reviews by reviewers who did not identify themselves as any of the above categories.

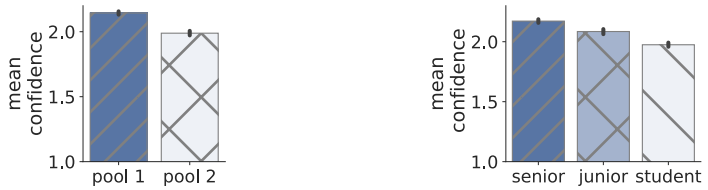
Self-reported confidence. We next study the difference in the self-reported confidence among different groups of reviewers. The mean value of reported confidence is plotted in Figure 3.8. In this case, we see a statistically significant correlation between seniority and self-reported confidence. Following are p-values (before accounting for multiple testing) and corresponding effect sizes: senior vs. junior researcher: p=4.1683×10^{-11}, d= 0.1604, senior researcher vs. PhD student: p=3.308×10^{-57}, d= 0.3577 and junior researcher vs. PhD student: p=8.074×10^{-15}, d= 0.1758. We observe a similar difference in confidence score and effect size between pool 1 and pool 2 reviewers: p=3.9679×10^{-44}, d= 0.2943.

Consistency. We now study the consistency within reviewers of pool 1 (invited), and within reviewers of pool 2 (volunteer). The consistency captures the amount of variance or disagreements in the reviews provided by that pool. As noted by [90], *"the disagreement among reviewers is a useful metric to check and monitor during the review process. Having a high disagreement means, in some way, that the judgment of the involved peers is not sufficient to state the value of the contribution itself. This metric can be useful to improve the quality of the review process..."*



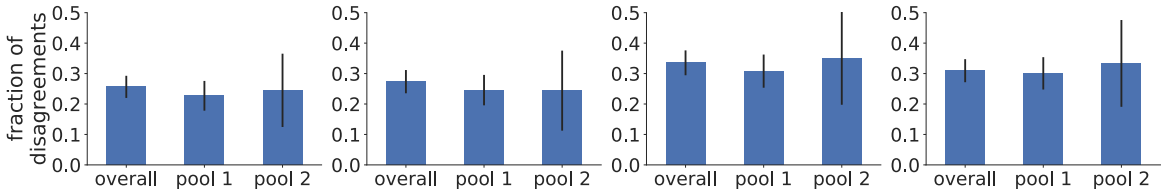
(a) invited and volunteer reviewers (b) senior, junior, and student reviewers

Figure 3.7: Mean of scores provided for different features grouped by different reviewer types.



(a) invited and volunteer reviewers (b) senior, junior, and student reviewers

Figure 3.8: Self-reported confidence grouped by different reviewer types.



(a) novelty (b) quality (c) impact (d) clarity

Figure 3.9: Proportions of inter-reviewer disagreements on each feature.

Concretely, consider any pair of reviewers within a given pool, any pair of papers that is reviewed by both the reviewers, and any feature. We say that this pair of reviewers agrees on this pair of papers (for this feature) if both reviewers rate the same paper higher than the other; we say that this pair disagrees if the paper rated higher by one reviewer is rated lower by the other. Ties are discarded. We count the total number of such agreements and disagreements within each of the two pools.

Figure 3.9 plots the fraction of disagreements within each of the two pools for the cardinal scores. At this aggregate level, we do not see enough difference to conclusively rate any one pool's intra-pool agreement above the other, and this conclusion is also confirmed by an absence of a statistically significant difference in the proportion of agreements within pool 1 from the proportion of agreements within pool 2. Specifically, for the Pearson's chi-squared test and effect sizes of pool 1 vs. pool 2, the results for the four features (before accounting for multiple testing) are: novelty p=0.9269 d= -0.0426, quality p=0.8648, d= 0.0039, impact p=0.7296, d= -0.0936, and clarity p=0.8029, d= -0.0709. The total sample sizes for the three categories of overall, pool 1 and pool 2 respectively across the four features are: novelty 554, 282 and 49; quality 523, 285 and 41; impact 513, 276 and 37; and clarity 572, 286 and 42. Section 3.2 presents similar consistency results for the two pools in the ordinal rankings. (We also attempted to run this analysis restricted to the top 2k papers, but this restriction results in a very low sample complexity and hence underpowered tests.)

Participation in discussions. One fact that caught our attention was the amount of participation in the discussion by the different reviewer groups: senior reviewers take much more active roles in the discussions than junior researchers. Please see Section 3.2 for details, where we provide a more detailed study of the discussion phase.

Rebuttals and discussions

This section is devoted to the analysis of the rebuttal stage and the participation of reviewers in discussions. We begin with some summary statistics. The authors of 2188 papers submitted a rebuttal.

There were a total of 12154 reviews that came in before the rebuttals started, and with some more reviews received after the rebuttal round, the total number of {reviewer, paper} pairs eventually ended up being 13674. Among the 12154 reviews that were submitted before the rebuttal, the scores of only 1193 of them changed subsequent to the rebuttal round. These changed review scores were distributed among 886 papers.

There were 842 papers for which no reviewer participated in the discussions, 339 papers for which exactly one reviewer participated, and 436, 376, 218, 135 and 49 papers for which 2, 3, 4, 5 and 6 reviewers participated respectively. There were a total of 5255 discussion posts, and 4180 of the 13674 {reviewer, paper} pairs participated in the discussions.

Who participates in discussions?

We compare the amount of participation of various groups of reviewers in the discussion phase of the review process.

Pool 1 (invited) versus pool 2 (volunteer) reviewers. We compare the participation of the reviewers in two pools in the discussions as follows, and plot the results in Figure 3.10(a). In

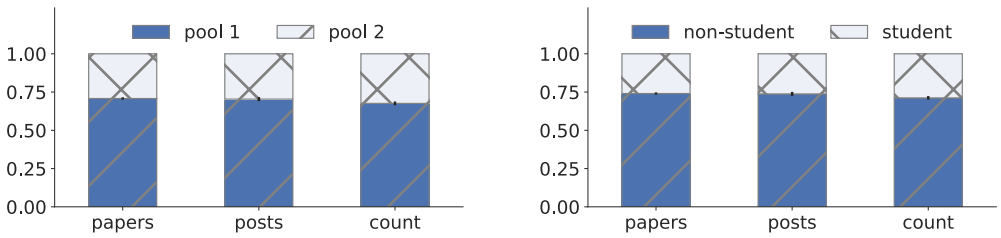
order to set a baseline, we first compute the total number of {pool 1 reviewer, paper} pairs and the total number of {pool 2 reviewer, paper} pairs – these counts are computed irrespective of whether the reviewer participated in the discussions or not. We plot the proportions of these counts as the "count" bar in the figure. Next we compute the total number of posts that were made by pool 1 reviewers and the total number of posts that were made by pool 2 reviewers – the resulting proportions are plotted as the "posts" bar in the figure. Finally, we compute the total number of {pool 1 reviewer, paper} pairs in which that reviewer put at least one post in the discussion for that paper, and the total number of {pool 2 reviewer, paper} pairs in which that reviewer put at least one post in the discussion for that paper. We plot the two proportions in the "papers" bar. The total sample sizes for the categories of counts, posts and papers are 13674, 5255 and 4180 respectively.

We tested whether the mean number of posts per {reviewer, paper} pair is identical for the two pools of reviewers. For the null hypothesis that the means are identical for the two pools of reviewers, the t-test yielded a p-value of $p = 1.36 \times 10^{-4}$. We also conducted this analysis for the restriction of papers to the top 2k, and for this subset, the t-test yielded a p-value of $p = 9.458 \times 10^{-4}$. We see a statistically significantly higher participation by the pool 1 reviewers as compared to the pool 2 reviewers in the discussions. However, the absolute amount of participation by either group is moderate at best, and the effect sizes are small with d= 0.0704 and d= 0.0894 for analysis of all papers and top 2k papers respectively.

Student versus non-student reviewers. We calculated the above three sets of quantities for student and non-student reviewers. Figure 3.10(b) depicts the results. We tested whether the mean number of posts per {reviewer, paper} pair for the student reviewers is identical to the non-student reviewers. For the null hypothesis that the means are identical, the t-test yielded a p-value of $p = 3.016 \times 10^{-4}$. We also conducted this analysis for the restriction of papers to the top 2k, and for this subset, the t-test yielded a p-value of $p = 8.932 \times 10^{-4}$. We see a statistically significantly higher participation by the non-student reviewers as compared to the student reviewers in the discussions. However, the total amount of participation by either group is not too large, and the effect sizes are small with d= 0.0695 and d= 0.0929 respectively.

How do discussions change the scores?

A total of 1193 out of 12154 reviews that were submitted before rebuttals changed subsequently. These changed reviews were distributed among 886 papers. As a result, the amount of change in



(a) invited and volunteer reviewers

(b) student and non-student reviewers

Figure 3.10: Proportions of contributions from different types of reviewers in discussions ("posts" and "papers") and the total number of such reviewers ("count").

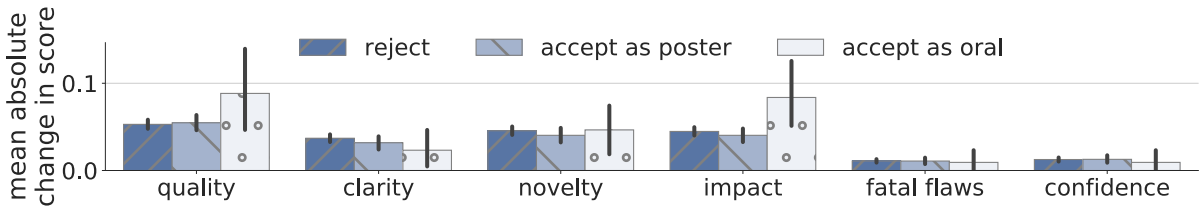


Figure 3.11: Mean absolute value of the change in the scores from before the rebuttal round to the end of the discussion phase.

review scores is quite small. Figure 3.11 depicts the score change – in absolute value – averaged across all reviewers and all papers. While the allowed range of the scores is 1 to 5, the change in mean score is less than 0.1.

From the point of view of reviewers, we see a significant correlation between participation in the discussions and the final decisions. Specifically, for each paper we computed the average of scores given by all reviewers who participated in the discussions and the average of scores given by all reviewers who did not participate (when there was at least one reviewer of each type). We discarded this paper if both types of reviewers provided an identical average score. Now, if the participating reviewers gave a higher score than the non-participating reviewers and if the paper was accepted, we counted it as an agreement of the final decision with the participating reviewers. If the participating reviewers gave a lower score than the non-participating reviewers and if the paper was not accepted, then also we count it as an agreement of the final decision with the participating reviewers. Otherwise, we counted the paper as having a disagreement between the final decisions and the participating reviewers. From the data, we observe a statistically significant agreement of the final decisions and the participating reviewers with $p = 1.6 \times 10^{-6}$ with d= 0.13. We continue to observe a statistically significant correlation when this analysis is performed restricted to pool 1 reviewers ($p = 7.7 \times 10^{-4}$) or to pool 2 reviewers ($p = 1.3 \times 10^{-4}$) alone. Of course, we cannot tell the causality from this correlation, as to whether the discussions actually influenced the decisions or not.

All in all, we observe that only a small fraction of the reviews change scores following the rebuttals. Moreover the magnitude of this change in scores is very small. This observation suggests that this rebuttal process may not be very useful. That said, there are various qualitative aspects that are not accommodated in this quantitative aggregate statistic. First, it may be possible that more reviews changed with respect to the text comments but the reviewers just did not bother to change the scores – we are unable to check this property as there is no snapshot of the text comments before the rebuttal. Second, there are a reasonable number of discussion posts, however, we do not know what fraction of these posts where reviewers shifted from their earlier opinion. Third, the final decisions are correlated positively with the reviewers who participated in discussions. Taking these factors into account, we think that the present rebuttal system should be put under the microscope regarding its value for the time and effort of such a large number of people. It may also be worth trying alternative systems of recourse for authors, such as a formal appeals process, that help to put more focus on the actual borderline cases.

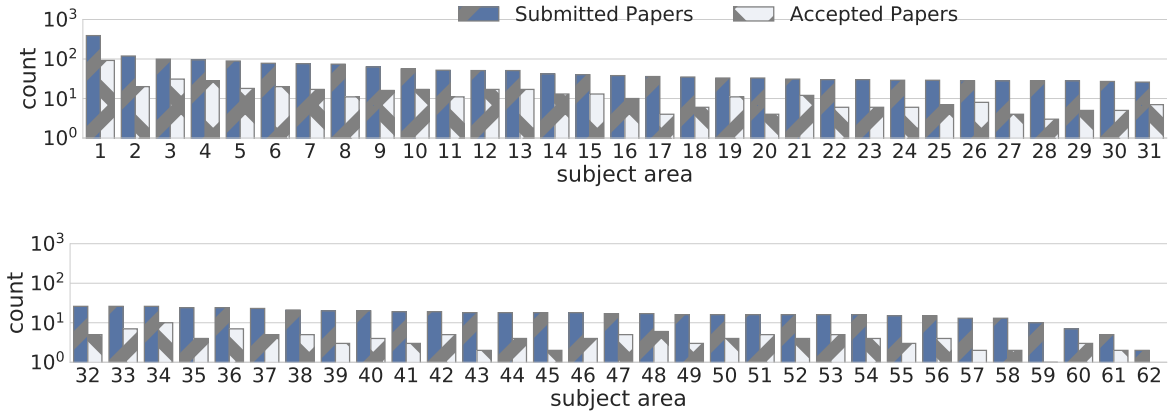


Figure 3.12: The number of submitted and accepted papers per (primary) subject area. The names of the subject areas corresponding to each of the numbers on the x-axis are provided in Table 3.2.

Distribution across subject areas

Figure 3.12 plots the distribution per subject area (primary subject area), for the submitted papers and for the accepted papers. Of course the proportions are not identical, but the plots do not show any systematic bias either towards or against any particular areas. The heavy tail in the distributions below also corroborates the significant diversity of topics in the NIPS community. A chi-square test of homogeneity of the two distributions failed to detect any significant difference between the two distributions: p=0.6029, $\chi^2(dof = 62, \#samples = 2425) = 57.51$.

Quantifying the randomness

Quantifying the extent to which the outcome of a peer-review process is different from a random selection of papers is one of the most pressing questions for the scientific community [104]. In this section we conduct two analyses to quantify the randomness in the review scores in NIPS 2016.

Messy middle model

The NIPS 2014 experiment [62] led to the proposal of an interesting "messy middle" model [89]. The messy middle model postulates that the best and the worst papers are clear accepts and clear rejects, respectively, whereas the papers in the middle suffer from random decisions that are independent of the content of the papers. The messy middle model is obviously a stylized model, but it nevertheless suggests an interesting investigation into the randomness in the reviews and decisions of the papers that lie in the middle. In this section, we describe such an investigation using the NIPS 2016 data.

The messy middle model assumes random judgments for the middle papers. If the messy middle model were correct then for any pair of papers in the middle, and any pair of common reviewers, the probability of an agreement on the relative ranking of the two papers must be

1. Deep learning/Neural networks	32 Causality
2. (Application) Computer Vision	33 Bayesian nonparametrics
3. Learning theory	34 Variational inference
4. Convex opt. and big data	35 Similarity and Distance Learning
5. Sparsity and feature selection	36 (Other) Statistics
6. Clustering	37 Spectral methods
7. Reinforcement learning	38 Active Learning
8. Large scale learning	39 Graph-based Learning
9. Graphical models	40 (Other) Bayesian Inference
10. Bandit algorithms	41 (Application) Collab. Filtering / Recommender Systems
11. Matrix factorization	42 Information Theory
12. Online learning	43 (Application) Signal and Speech Processing
13. (Other) Optimization	44 (Application) Social Networks
14. (Other) Neuroscience	45 (Other) Robotics and Control
15. Kernel methods	46 Nonlin. dim. reduction
16. Gaussian process	47 Model selection and structure learning
17. Multitask/Transfer learning	48 Ensemble methods and Boosting
18. Component Analysis (ICA, PCA, . . .)	49 Stochastic methods
19. Combinatorial optimization	50 (Other) Cognitive Science
20. Time series analysis	51 Structured prediction
21. (Other) Probabilistic Models and Methods	52 Ranking and Preference Learning
22. (Other) Applications	53 Game Theory and Econometrics
23. (Other) Machine Learning Topics	54 (Application) Privacy, Anonymity, Security
24. (Cognitive/Neuro) Theoretical Neuroscience	55 (Cognitive/Neuro) Perception
25. (Other) Unsupervised Learning Methods	56 (Application) Bioinfo. and Systems Bio.
26. MCMC	57 Regularization and Large Margin Methods
27. Semi-supervised	58 (Other) Regression
28. (Other) Classification	59 (Application) Information Retrieval
29. (Application) Natural Language and Text	60 (Application) Web App. and Internet
30. (Application) Object and Pattern Recognition	61 (Cognitive/Neuro) Reinforcement Learning
31. (Cognitive/Neuro) Neural Coding	62 (Cognitive/Neuro) Language

Table 3.2: List of subject areas associated to the subject area numbers in Figure 3.12

identical to the probability of disagreement. With this model in mind, we restrict attention to the papers in the middle, and then measure how far the agreements of the reviewers are from equiprobable agreements and disagreements. An analysis of this quantity for various notions of the "middle" papers yields insight into the messiness in the reviews for papers in the middle.

Procedure: We now describe the procedure employed for the analysis. Here we let n denote the total number of papers submitted to the conference and β denote the fraction of papers accepted to the conference (we have $n = 2425$ and $\beta = 0.237$ in NIPS 2016). The procedure is associated to two parameters: μ is the minimum number of samples required and α is a threshold of messiness. We choose $\mu = 100$ and $\alpha = 0.01$ in our subsequent analysis, noting that importantly, our overall conclusions are robust to these choices.

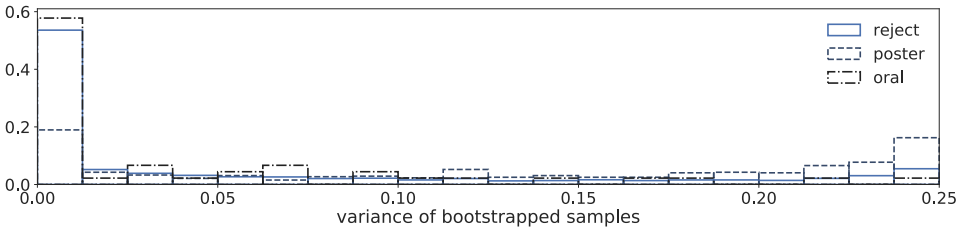
1. Rank order all papers with respect to their mean scores. Call this ordering as O.
2. For every $t \in [0,1]$ and $b \in [0,1]$ (up to some granularity), do the following.
   - 2.1 Initialize variables $n_{\text{agree}}[t,b] = n_{\text{disagree}}[t,b] = 0$.
   - 2.2 Consider the set of papers obtained by removing the top t fraction of papers and bottom b fraction of papers from O. Call this (unordered) set of "middle papers" as M.
   - 2.3 If $(\beta - t)n < \mu$ or $((1-\beta) - b)n < \mu$ then continue to the next values of (t,b) in Step 2.
   - 2.4 Consider any pair of reviewers and any pair of papers in M that is reviewed by both the reviewers. We say that this pair of reviewers agrees on this pair of papers if both reviewers provide a higher mean score (taken across the features) to the same paper as compared to the other paper. We say that this pair disagrees if the paper rated higher by one reviewer (in terms of the mean score across the features) is rated lower by the other reviewer. Ties are discarded. We count the total number of such agreements (denoted as $n_{\text{agree}}[t,b]$) and disagreements (denoted as $n_{\text{disagree}}[t,b]$) within each of the two pools.
3. Find the largest value of $(1-t-b)$ such that $(n_{\text{agree}}[t,b] + n_{\text{disagree}}[t,b]) \geq \mu$ and $\frac{n_{\text{agree}}[t,b]}{n_{\text{agree}}[t,b]+n_{\text{disagree}}[t,b]} < 0.5+\alpha$. This largest value of $(1-t-b)$ is defined as the size of the messy middle.

Let us spend a moment interpreting some steps of the procedure. Step 2.3 as well as the μ-condition in Step 3 ensures that there are a sufficient number of samples for any computation on the messy middle region. Specifically, the conditions $(\beta - t)n < \mu$ and $((1-\beta) - b)n < \mu$ ensure existence of a sufficient number of papers above and below the acceptance threshold. Under this constraint, Step 3 then finds the largest window of papers in the middle such that the fraction of reviewer-agreements is at most $(0.5 + \alpha)$. Thus a *smaller* size of the window is a desirable property.

We can now use this analysis to compare messy middle window sizes for two or more conferences. When making such a comparison, we make one adjustment. In the last step (Step 3), we consider only those values of (t,b) such that $n_{\text{agree}}[t,b] + n_{\text{disagree}}[t,b] \geq \mu$ for both datasets. Then compare the sizes of the messy middle.

Conference	Size of messy middle
NIPS 2015	45%
NIPS 2016	30%

(a) Size of the messy middle windows.



(b) Histogram of the variance of acceptance decisions (according to mean scores) of the papers in a bootstrapped analysis.

Figure 3.13: Amount of randomness in the reviews.

Results: We used this procedure to compute the size of the messy middle in NIPS 2016 and also in NIPS 2015. The granularity we used is $1/20$, that is, $t,b \in \{0,1/20,2/20,\ldots,1\}$. NIPS 2015 had a marginally higher average number of reviews per paper as compared to NIPS 2016. We set $\mu = 100$ and $\alpha = 0.01$ (note that the conclusions drawn below are robust to these choices). The results of the analysis are tabulated in Figure 3.13a.

In the NIPS 2016 data, we observe that the size of the messy middle is 30%. Specifically, if we remove the bottom 70% of papers (and none of the top papers) then we see that the inter-reviewer agreements are near-random, but farther from random otherwise. On the other hand, we observe that the size of the messy middle is 45% in the NIPS 2015 data, which occurs when removing 15% of the top papers and 40% of the bottom papers. We thus see that in terms of this metric of the size of the messy middle, the NIPS 2016 review data is an improvement over the previous edition of the conference.

Such an analysis is useful in comparing the noise in the review data across conferences. It can particularly be useful to evaluate the effects of any changes made in the peer-review process. The ease of doing this post hoc analysis, without necessitating any controlled experiment, is a significant benefit to this approach of analysis. In order to enable comparisons of the size of the messy middle of NIPS 2016 with other conferences, we provide the values of $\frac{n_{\text{agree}}[t,b]}{n_{\text{agree}}[t,b]+n_{\text{disagree}}[t,b]}$ and $(n_{\text{agree}}[t,b]+n_{\text{disagree}}[t,b])$ for the NIPS 2016 data for all values of (t,b) in Figure 3.14.

In Figure 3.14 we provide the values of the fraction of agreements $r := \frac{n_{\text{agree}}[t,b]}{n_{\text{agree}}[t,b]+n_{\text{disagree}}[t,b]}$ at the top of the corresponding cell and number of pairs $m := (n_{\text{agree}}[t,b]+n_{\text{disagree}}[t,b])$ for every value of (t,b) at the bottom of the corresponding cell. Note that the values are computed for all values of (t,b) ignoring the sample size restriction imposed by Step 2.3 of the procedure outlined in Section 3.2. Each cell in the table is color-coded by the size of the 95% confidence interval (on a log-scale) computed as $(2\times 1.96)\sqrt{\frac{r(1-r)}{m}}$.

It is important to note that this post hoc analysis is not strictly comparable to the NIPS 2014 controlled experiment because we do not have access to a true ranking or a counterfactual. That said, since such an analysis can easily be performed post hoc using the data from reviews and

does not require any special arrangement in the review process, it would be useful to see how these results compare to the data from other conferences.

A bootstrapped analysis

In this section we conduct an analysis to measure the randomness in the reviews in the NIPS 2016 data compared to that of random selection. In our analysis, we first conduct 1000 iterations of the following procedure. For each paper, we consider the set of reviewers who reviewed this paper. We then choose the same number of reviewers uniformly at random with replacement from the set of original reviewers for this paper. We then take the mean of the scores across all features and across all the sampled reviewers for that paper. Next we rank order all papers in

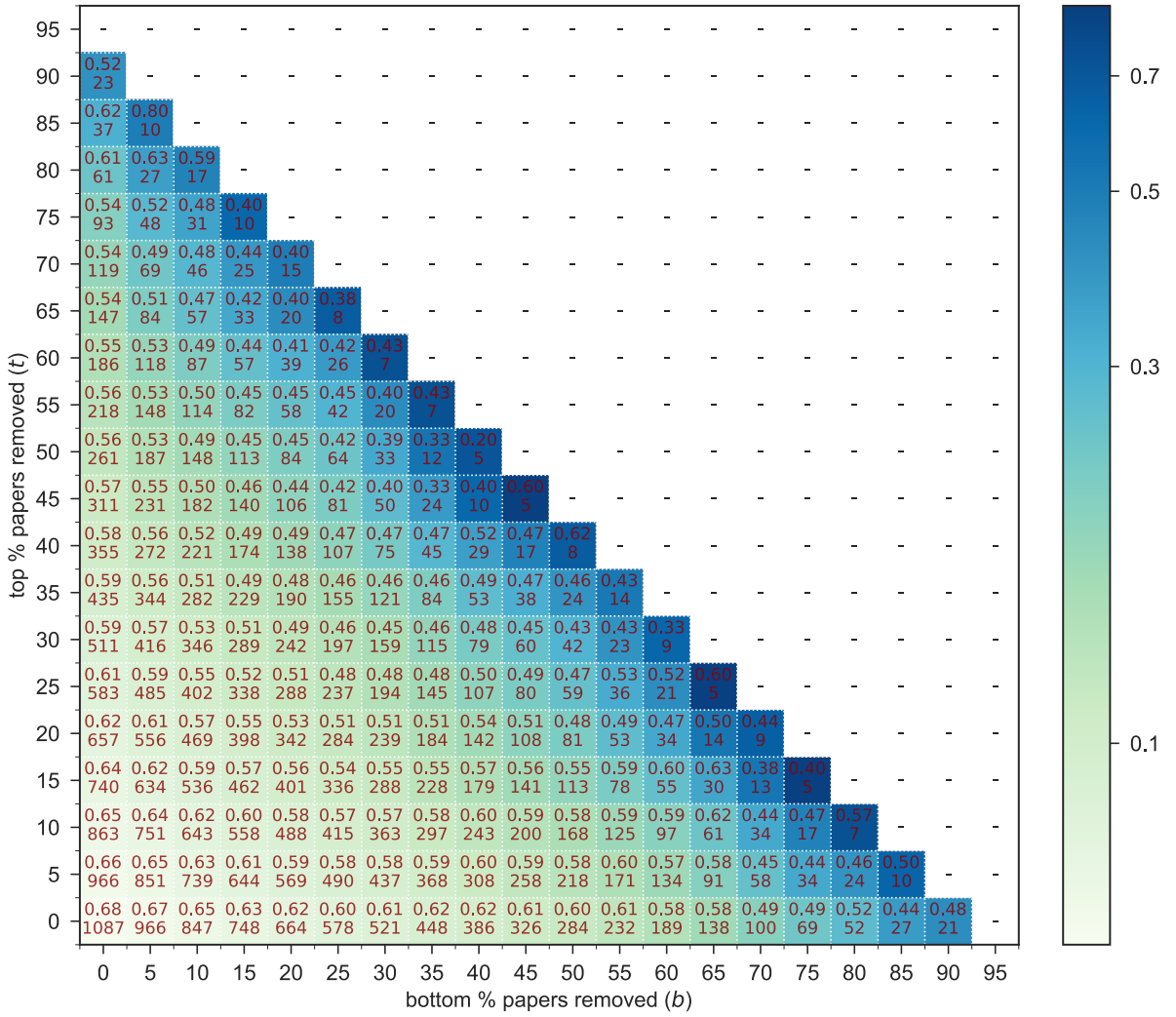


Figure 3.14: The inter-reviewer agreement ratios in the messy middle. For each value of t and b, we report two numbers: The agreement ratio $r := n_{\text{agree}}/(n_{\text{agree}} + n_{\text{disagree}})$ and the number of overlapping paper-reviewer pairs $m := n_{\text{agree}} + n_{\text{disagree}}$. Each cell is color-coded by the size of the 95% confidence interval (on a log scale).

terms of these mean scores and choose the top 23.7% of the papers as "accepted" in this iteration and the others as rejected.

Our analysis focuses on the variance of the acceptance decisions for each paper. At the end of all iterations, for each paper, we compute the fraction of iterations in which the paper was accepted. Letting $\beta_i \in [0,1]$ denote this fraction for any paper i, the variance in the acceptance decisions for this paper equals $\beta_i(1-\beta_i)$. We plot a histogram of the computed variances (for every paper) in Figure 3.13b. For comparison, note that in an ideal world, the variance of the decisions for each paper would be zero. Observe that a large fraction of rejected papers as well as a large fraction of papers that were accepted as oral presentations have a near-zero variance. On the other hand, a notable fraction of papers accepted as posters as well as those rejected have a variance close to its largest possible value of $\frac{1}{4}$.

Ordinal data collection

The data collected from the reviewers in the NIPS 2016 review process comprises cardinal ratings (in addition to the free-form text-based reviews) where reviewers score each paper on four features on a scale of 1 to 5. A second form of data collection that is popular in many applications, although not as much in conference reviews, is ordinal or comparative ranked data. The ordinal data collection procedure that we consider asks each reviewer to provide a total ordering of all papers that the reviewer reviewed.

There are various tradeoffs between collecting cardinal ratings and ordinal rankings. In the context of paper reviews, cardinal ratings make reviewers read each individual paper more carefully (and not make snap judgments), and can elicit more than a just one bit of information. On the other hand, ordinal rankings allow for nuanced comparative feedback, help avoid ties, and are free of various biases and calibration issues that otherwise arise in cardinal scores [46, 59, 97, 91, 16]. We refer the reader [7, 105, 99, 100] and references therein for more details on ordinal data collection and processing. In the present paper, we present three sets of analyses with the ordinal rankings collected from reviewers.

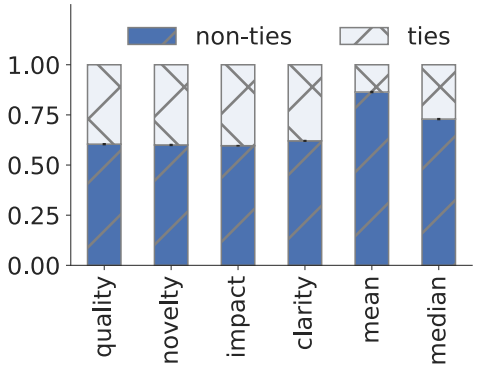


Figure 3.15: Proportion of ties in reviewer scores. The bars titled "mean" and "median" represent the mean and median scores across all four features.

Tie breaks

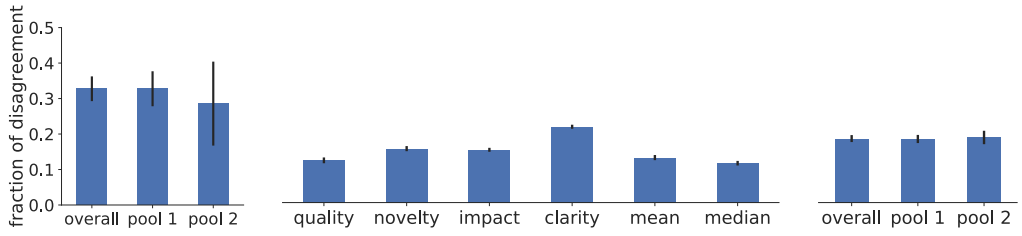
An ordinal ranking of the papers provided by a reviewer ensures that there are no ties in the reviewer's evaluations. On the other hand, asking cardinal scores can result in scores that are tied, thereby preventing an opportunity for the AC to discern a difference between the two papers from the provided scores.

In order to evaluate the prevalence of ties under cardinal scores, we performed the following computation. For every {`paper`, `paper`, `reviewer`} triplet such that the reviewer reviewed both papers, and for any chosen feature (i.e., quality, novelty, impact, and clarity), we computed whether the reviewer provided the same score to both papers or not. We totaled such ties and non-ties across all such triplets.

Figure 3.15 depicts the proportion of ties computed across all submitted papers. The total sample size is 26106. Observe that a significant fraction – exceeding 30% for each of the four features – of pairs of reviewer scores are tied. When only the top 2k papers were used in the calculation, the fraction of ties in each feature is even higher, by approximately 10% – 15% of the respective value in the setting of all papers. In conclusion, these results reveal a significant proportion of ties in the cardinal scoring scheme and also confirm that, by design, ties are inevitable in this scoring scheme. The use of ordinal rankings, on the other hand, does not suffer from such a drawback.

Consistency of ordinal ranking data

While there is substantial literature on benefits of collecting data in an ordinal ranking form, several past works also recommend verifying if the application setting under consideration is appropriate for ordinal rankings. For instance, [97] states the benefits of ranking for settings "where the items are highly discriminable"; [87] asks respondents to rank 18 values in order of importance but observe unstable and inconsistent results; [46] argues that ranking generally requires a higher level of attention than rating and that asking respondents to rank more than a handful of statements puts a very high demand on their cognitive abilities. Accordingly, this section is devoted to performing sanity checks on the ordinal ranking data obtained subsequent to the NIPS 2016 review process. We do so by comparing certain measures of consistency of



(a) (b) (c)

Figure 3.16: Fraction of disagreements (a) within ordinal rankings between different pairs of reviewer types; (b) between ordinal rankings and cardinal ratings ("mean" and "median" refer to the mean and median of the cardinal scores for the four features); and (c) between ordinal rankings and final acceptance decisions.

the ordinal data with the cardinal data obtained in the main review process.

Agreements within ordinal rankings. For every pair of papers that have two reviewers in common, we compute whether these two reviewers agree on the relative ordinal ranking of the two papers or if they disagree. In more detail, we say that this pair of reviewers agrees on this pair of papers if both reviewers rank the same paper higher than the other in their respective ordinal rankings; we say that this pair disagrees if the paper ranked higher by one reviewer is rated lower by the other. Figure 3.16a depicts the proportion of disagreements for the ordinal rankings in the entire set of papers, as well as broken down by the type of reviewer. First, observe that the ordinal rankings have a similar level of consistency as that observed in the cardinal scores in Figure 3.9. Second, we observe no statistically significant difference between the two pools: p=0.9849 for Pearson's chi-squared test and effect size d= 0.0018. The sample sizes are 696, 348 and 56 for all reviewers, pool 1 and pool 2 respectively.

Agreement of ordinal rankings with cardinal ratings. Let us now evaluate how well the overall ordinal rankings associate with the cardinal scores given for the individual features. For every pair of papers that have a common reviewer, we compare whether the relative ordering of the cardinal scores for a given feature agree with the ordinal ranking given by the reviewer for the pair of papers. We report the proportion of disagreements in Figure 3.16b. We observe the high amount of agreement of the ordinal rankings with the cardinal scores – for instance, the median cardinal score agrees in about 90% of cases with the overall ordinal rankings provided by the reviewers.

Agreement of ordinal rankings with final decisions. We finally compute the amount of agreement between the ordinal rankings provided by the reviewers and the final decisions of acceptance. We consider all {paper, paper, reviewer} triplets where the reviewer reviewed both papers, and one of these papers was eventually accepted and the other was rejected. For every such triplet, we evaluate whether the reviewer had ranked the accepted paper higher than the rejected paper ("agreement") or vice versa ("disagreement"). We report the proportion of agreements and disagreements in Figure 3.16c. We see that the agreement of the overall rankings with the eventual decisions is quite high – there are roughly five agreements for every disagreement.

When restricted to the top 2k papers, we observe that the disagreements of ordinal rankings with final decisions increase to 27-28% in all three categories (overall, pool 1 and pool 2) from 16-17% in the case of all papers. Note that the experiments on inter-reviewer agreements do not permit an effective analysis when restricted to top 2k papers as the sample size reduces quadratically (that is, reduces to a fraction $.47^2 \approx .2$ of the sample size with all papers).

Detecting anomalies

Ordinal rankings can be used to detect anomalies in reviews. We discuss this aspect in the Section 3.2.

Checking inconsistencies

In this section, we propose an automated technique to help reduce some human errors and inconsistencies in the review process. In particular, we propose to automatically check for inconsistencies in the review ratings provided by the reviewers. On finding any such inconsistency, we

propose to then have the area chairs either manually investigate this inconsistency or to manually or automatically contact the reviewer requesting an explanation. In what follows, we propose two notions of inconsistencies in regards to the NIPS 2016 review process and quantify their presence in the NIPS 2016 review data.

Anomalies in feature ratings. We investigate whether any reviewer indicated that paper "A" is strictly better than paper "B" in all four features, but rank paper "A" lower than paper "B" in the ordinal ranking. We find that there are 55 such pairs of reviews provided by 44 distinct reviewers. If we restrict attention to the top 2k papers, we find that that there are 10 such pairs of reviews provided by 10 distinct reviewers.[1]

Anomalies in fatal flaws. We now investigate if there are cases when a reviewer indicated a fatal flaw in a paper, but that reviewer ranked it above another paper that did not have a fatal flaw according to the reviewer. We found 349 such cases across 176 such reviewers. The proportion of such cases is similar among volunteer and invited reviewers. Among the top 2k papers, there are 55 such pairs across 33 reviewers.

One may think that the number of such cases is large because ordinal survey was done after the review process, so people may not have remembered the papers well or may not have done a thorough job as they knew it would not count towards the reviews. However, the ordinal data actually is quite consistent with the cardinal data (Section 3.2). Hence we do not think such a large discrepancy with fatal flaws can be explained solely due to such a delay-related noise.

Two possible explanations for such anomalies are as follows. Either the reviewer may not have done an adequate job of the review, or the set of provided features are grossly inadequate to express reviewers' opinions. In either case, we suggest automatically checking for such glaring inconsistencies (irrespective of whether ordinal or cardinal final ratings are used) during the review process, and contacting the respective reviewers to understand their reasoning.[2] We hope that such a checkpoint will be useful in improving the overall quality of the review process.

3.3 Discussion and Conclusions

NIPS has historically been the terrain of much experimentation to improve the review process and this paper is our contribution to advance the state of the art in review process design. In this paper, we reported a post hoc analysis of the NIPS 2016 review process. Our analysis yielded useful insights into the peer-review process, suggested action items for future conferences, and resulted in several open problems towards improving the academic peer-review process, as enumerated throughout this paper.

Our tools include several means of detecting potential artifacts or biases, and statistical tests to validate hypotheses made: Comparing the distribution of topics in submitted papers and accepted papers; creating a graph of proximity of reviewers (according to commonly reviewed papers) and papers (according to common reviewers) to detect potential disconnected communities; test to compare two pools of reviewers; quantifying the noise in the review scores. We also observed that the histogram of scores obtained included a significantly larger fraction of papers than the

[1] Note that the total number of pairs of papers reduces more than 4-fold when moving from the set of all papers to the top 2k set.

[2] This analysis was performed after completion of the review process, and hence reviewers were not contacted for these inconsistencies.

guidelines suggested. This observation suggests a more careful design of the elicitation interface and the type of feedback provided to authors.

Selection biases that arise when recruiting reviewers and ACs in a review process of this scale are difficult to deal with. Some designs in the selection of reviewers lend themselves more to bias than others. In NIPS2016, we made some design choices of the review process with the intention of reducing these biases. For instance, the recruitment of volunteer author-reviewers helped increase the diversity of the reviewer pool. They were less prone to selection bias compared to selecting reviewers by invitation only, primarily based on AC recommendations. With respect to reducing bias across AC decisions, we introduced the "AC buddy system" in which pairs of ACs had to make decisions jointly about all their papers. This method scales well with the increase in number of papers, but is sub-optimal to calibrate well decisions since buddy pairs form disjoint decision units (no paper overlap between buddy pairs). However, decision processes based on a conference between several or all ACs, as done in earlier editions of the conference, are also not perfect because decisions are sometimes dominated by self-confident and/or opinionated ACs. Although the evidence we gathered from our analyses did not reveal any "obvious" bias, it does not mean that there is none. We hope that some designs of our review process will shed some lights on ways of improving bias-immune or bias-avoidance procedures for future conferences.

The reviews themselves were of mixed quality, but recruiting more reviewers (between 4 and 6 per paper) ensured that each paper had a better chance to get a few competent reviews. We gave a strong role to the ACs who arbitrated between good and bad reviews and made the final decision, which was not just based on an average score. To recruit more reviewers (and possibly a more diverse and less biased set of reviewers) we introduced the new idea to invite *volunteer* author reviewers, which we think is a good contribution. In particular, next to many PhD students, this brought a considerable amount of senior reviewers in the system as well. Some of the ACs systematically disregarded volunteer reviews, judging that they could not be trusted. But, our analysis did not reveal that reviewers from that pool made decisions significantly different from the pool of reviewers invited by recommendation. However, more senior reviewers seem to put more effort into providing detailed reviews, and participating to rebuttals and discussions. Hence we need to find means of encouraging and possibly educating more junior reviewers to participate in these aspects. As a means of self-assessment and encouragement, reviewers could receive statistics about review length, amount of agreement between reviewers, and participation to rebuttals and discussions, as well as figures concerning their own participation. Naturally, the participation of junior reviewers in the review process is a form of education. It would be nice to track from year to year whether individual reviewers ramp up their review length, level of agreement with other reviewers, and participation in discussions and rebuttals. Note that we believe that such statistics should not be used as a means of selecting reviewers because this could bias the selection.

It is an on-going debate to which extent the decision process should be automated and what means could be used to automate it. We provide some elements to fuel this discussion. We evaluated how rebuttals and discussions change the scores. Although this concerns only a minority of papers, we believe that ACs have a key role in arbitrating decisions when there is a controversy and that this is not easy to monitor merely with scores. Since scores do not seem to be consistently updated by reviewers after rebuttal/discussions, maybe the review process should include a score confirmation to make sure that absence of change in score is not due to

negligence. Mixing ordinal and cardinal scores may reduce the problems of reviewer calibration, tie breaking, and identifying anomalies possibly due to human error.

All in all, it is important to realize that in a review process of this scale, there is not a single person who really controls what is going on at all levels. Program chairs spend a lot of time on quality control, but definitely cannot control the decisions on all individual papers or the quality of individual reviewers. In the end, we have to trust the area chairs and reviewers: the better reviews *all of us* provide, the better the outcome of the review process. We as a community must also continue to strive improving the peer-review process itself, via experiments, analysis, and open discussions. This topic in itself is a fertile ground for future research with many useful open problems including those enumerated throughout the paper.

Chapter 4

Enhancing Human Learning via Spaced repetition Optimization

Our ability to remember a piece of information depends critically on the number of times we have reviewed it, the temporal distribution of the reviews, and the time elapsed since the last review, as first shown by a seminal study by Ebbinghaus [32]. The effect of these two factors has been extensively investigated in the experimental psychology literature [75, 25], particularly in second language acquisition research [4, 13, 18, 86]. Moreover, these empirical studies have motivated the use of *flashcards*, small pieces of information a learner repeatedly reviews following a schedule determined by a spaced repetition algorithm [15], whose goal is to ensure that learners spend more (less) time working on forgotten (recalled) information.

The task of designing spaced repetition algorithms has a rich history, starting with the Leitner system [63]. More recently, several works [79, 69] have proposed heuristic algorithms that schedule reviews just as the learner is about to forget an item, *i.e.*, when the probability of recall, as given by a memory model of choice [32, 84], falls below a threshold. An orthogonal line of research [86, 78] has pursued locally optimal scheduling by identifying which item would benefit the most from a review given a fixed reviewing time. In doing so, they have also proposed heuristic algorithms that decide which item to review by greedily selecting the item which is closest to its maximum *learning rate*.

In recent years, spaced repetition software and online platforms such as Mnemosyne[1], Synap[2], or Duolingo[3] have become increasingly popular, often replacing the use of physical flashcards. The promise of these pieces of software and online platforms is that automated fine-grained monitoring and greater degree of control will result in more effective spaced repetition algorithms. However, most of the above spaced repetition algorithms are simple rule-based heuristics with a few hard-coded parameters [15], which are unable to fulfil this promise—adaptive, data-driven algorithms with provable guarantees have been largely missing until very recently [81, 92]. Among these recent notable exceptions, the work most closely related to ours is by Reddy *et al.* [92], who proposed a queueing network model for a particular spaced repetition method—the Leitner system [63] for reviewing flashcards—and then developed a heuristic approximation algorithm for scheduling reviews. However, their heuristic does not have provable guarantees, it does not adapt to the learner's performance over time, and it is specifically designed for the Leitner systems.

[1] http://mnemosyne-proj.org/

[2] http://www.synap.ac

[3] http://www.duolingo.com

In this work, we develop a computational framework to derive optimal spaced repetition algorithms, specially designed to adapt to the learner's performance, as continuously monitored by modern spaced repetition software and online learning platforms. More specifically, we first introduce a novel, flexible representation of spaced repetition using the framework of marked temporal point processes [1]. For several well-known human memory models [32, 72, 120, 84, 5], we use this presentation to express the dynamics of a learner's forgetting rates and recall probabilities for the content to be learned by means of a set of stochastic differential equations (SDEs) with jumps. Then, we can find the optimal reviewing schedule for spaced repetition by solving a stochastic optimal control problem for SDEs with jumps [45, 125, 126, 57]. In doing so, we need to introduce a new proof technique to find a solution to the so-called HJB equation (refer to 5), which is of independent interest.

For two well-known memory models, we show that, if the learner aims to maximize recall probability of the content to be learned subject to a cost on the reviewing frequency, the solution uncovers a linear relationship with a negative slope between the optimal rate of reviewing, or reviewing intensity, and the recall probability of the content to be learned. As a consequence, we can develop a simple, scalable online spaced repetition algorithm, which we name MEMORIZE, to determine the optimal reviewing times. Finally, we perform a large-scale natural experiment using data from Duolingo, a popular language-learning online platform, and show that learners who follow a reviewing schedule determined by our algorithm memorize more effectively than learners who follow alternative schedules determined by several heuristics. To facilitate research in this area, we are releasing an open source implementation of our algorithm[4].

4.1 Modeling Framework of Spaced Repetition

Our framework is agnostic to the particular choice of memory model—it provides a set of techniques to find reviewing schedules that are optimal under a memory model. Here, for ease of exposition, we showcase our framework for one well-known memory model from the psychology literature, the exponential forgetting curve model with binary recalls [32, 72], and use (a variant of) a recent machine learning method, half-life regression [98], to estimate the effect of the reviews on the parameters of such model[5].

More specifically, given a learner who wants to memorize a set of items $\mathcal{I}$ using spaced repetition, *i.e.*, repeated, spaced review of the items, we represent each reviewing event as a triplet

$$e := (\underset{\text{item}}{i}, \overset{\text{time}}{t}, \underset{\text{recall}}{r}),$$

which means that the learner reviewed item $i \in \mathcal{I}$ at time t and either recalled it ($r = 1$) or forgot it ($r = 0$). Here, note that each reviewing event includes the outcome of a test (*i.e.*, a recall) since, in most spaced repetition software and online platforms such as Mnemosyne, Synap, or Duolingo, the learner is tested in each review, following the seminal work of Reidiger and Karpicke [95].

[4] http://learning.mpi-sws.org/memorize/

[5] In, we apply our framework to other two popular memory models, the power-law forgetting curve model [120, 5] and the multiscale context model (MCM) [84].

Given the above representation, we model the probability that the learner recalls (forgets) item i at time t using the exponential forgetting curve model, *i.e.*,

$$m_i(t) := \mathbb{P}(r) = \exp\left(-n_i(t)(t-t_r)\right), \tag{4.1}$$

where t_r is the time of the last review and $n_i(t) \in \mathbb{R}^+$ is the forgetting rate[6] at time t, which may depend on many factors, e.g., item difficulty, number of previous (un)successful recalls of the item. Then, we keep track of the reviewing times using a multidimensional counting process $\boldsymbol{N}(t)$, in which the i-th entry, $N_i(t)$, counts the number of times the learner has reviewed item i up to time t. Following the literature on temporal point processes [1], we characterize these counting processes using their corresponding intensities $\boldsymbol{u}(t)$, *i.e.*, $\mathbb{E}[d\boldsymbol{N}(t)] = \boldsymbol{u}(t)dt$, and think of the recall r as their binary *marks*. Moreover, every time that a learner reviews an item, the recall r has been experimentally shown to have an effect on the forgetting rate of the item [25, 92, 98]. Here, we estimate such an effect using half-life regression [98], which implicitly assumes that recalls of an item i during a review have a multiplicative effect on the forgetting rate $n_i(t)$—a successful recall at time t_r changes the forgetting rate by $(1-\alpha_i)$, *i.e.*, $n_i(t) = (1-\alpha_i)n_i(t_r)$, $\alpha_i \leq 1$, while an unsuccessful recall changes the forgetting rate by $(1+\beta_i)$, *i.e.*, $n_i(t) = (1+\beta_i)n_i(t_r)$, $\beta_i \geq 0$. In this context, the initial forgetting rate, $n_i(0)$, captures the difficulty of the item, with more difficult items having higher initial forgetting rates compared to easier items, and the parameters α_i, β_i and $n_i(0)$ are estimated using real data (refer to 4.6 for more details).

Before we proceed further, we would like to acknowledge that several laboratory studies [18, 19] have provided empirical evidence that the retention rate follows an *inverted U-shape*, *i.e.*, *mass practice* does not improve the forgetting rate—if an item is in a learner's short term memory when the review happens, the long term retention does not improve. Thus, one could argue for time-varying parameters $\alpha_i(t)$ and $\beta_i(t)$ in our framework. However, there are several reasons that prevent us from that: (i) the derivation of an optimal reviewing schedule under time-varying parameters becomes very challenging; (ii) for the reviewing sequences in our Duolingo dataset, allowing for time-varying α_i and β_i in our modeling framework does not lead to more accurate recall predictions (refer to); and, (iii) several popular spaced repetition heuristics, such as the Leitner system with exponential spacing[7] and SuperMemo, have achieved reasonable success in practice despite implicitly assuming constant α_i and β_i. That being said, it would be an interesting venue for future work to derive optimal reviewing schedules under time-varying parameters.

Next, we express the dynamics of the forgetting rate $n_i(t)$ and the recall probability $m_i(t)$ for each item $i \in \mathcal{I}$ using stochastic differential equations (SDEs) with jumps. This will be very useful for the design of our spaced repetition algorithm using stochastic optimal control. More specifically, the dynamics of the forgetting rate $n_i(t)$ are readily given by:

$$dn_i(t) = -\alpha_i n_i(t) r_i(t) dN_i(t) + \beta_i n_i(t)(1-r_i(t))dN_i(t), \tag{4.2}$$

[6] Previous work often uses the inverse of the forgetting rate, referred as memory strength or half-life, $s(t) = n^{-1}(t)$ [92, 98]. However, it will be more tractable for us to work in terms of forgetting rates.

[7] The Leitner system with exponential spacing can be explicitly cast using our formulation with particular choices of α_i and β_i and the same initial forgetting rate, $n_i(0) = n(0)$, for all items (refer to).

where $N_i(t)$ is the corresponding counting process and $r_i(t) \in \{0,1\}$ indicates whether item i has been successfully recalled at time t. Similarly, the dynamics of the recall probability $m_i(t)$ are given by the following Proposition:

Proposition 1 *Given an item $i \in \mathcal{I}$ with reviewing intensity $u_i(t)$, the recall probability $m_i(t)$, defined by Eq. 4.1, is a Markov process whose dynamics can be defined by the following SDE with jumps:*

$$dm_i(t) = -n_i(t)m_i(t)dt + (1 - m_i(t))dN_i(t), \tag{4.3}$$

where $N_i(t)$ is the counting process associated to the reviewing intensity $u_i(t)$[8]*.*

Proof According to Eq. 4.1, the recall probability $m(t)$ depends on the forgetting rate, $n(t)$, and the time elapsed since the last review, $D(t) := t - t_r$. Moreover, we can readily write the differential of $D(t)$ as $dD(t) = dt - D(t)dN(t)$.

We define the vector $\boldsymbol{X}(t) = [n(t), D(t)]^T$. Then, we use Eq. 4.2 and Itö's calculus [45] to compute its differential:

$$d\boldsymbol{X}(t) \overset{\text{dt}}{\underset{zol}{=}} \boldsymbol{f}(\boldsymbol{X}(t),t)dt + \boldsymbol{h}(\boldsymbol{X}(t),t)dN(t) \tag{4.4}$$

$$\boldsymbol{f}(\boldsymbol{X}(t),t) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{4.5}$$

$$\boldsymbol{h}(\boldsymbol{X}(t),t) = \begin{bmatrix} -\alpha n(t) r(t) + \beta n(t)(1 - r(t)) \\ -D(t) \end{bmatrix} \tag{4.6}$$

Finally, using again Itö's calculus and the above differential, we can compute the differential of the recall probability $m(t) = e^{-n(t)D(t)} := F(\boldsymbol{X}(t))$ as follows:

$$\begin{aligned}
dF(\boldsymbol{X}(t)) &= F(\boldsymbol{X}(t+dt)) - F(\boldsymbol{X}(t)) \\
&= F(\boldsymbol{X}(t) + d\boldsymbol{X}(t)) - F(\boldsymbol{X}(t)) \\
&\overset{\text{dt}}{\underset{zol}{=}} (f^T F_{\boldsymbol{X}}(\boldsymbol{X}(t)))dt + F(\boldsymbol{X}(t) + \boldsymbol{h}(\boldsymbol{X}(t),t)dN(t)) - F(\boldsymbol{X}(t)) \\
&\overset{\text{dt}}{\underset{zol}{=}} (f^T F_{\boldsymbol{X}}(\boldsymbol{X}(t)))dt + (F(\boldsymbol{X}(t) + h(\boldsymbol{X}(t),t)) - F(\boldsymbol{X}(t)))dN(t) \\
&= (e^{-(D(t)-D(t))n(t)(1+\alpha r_i(t) - \beta(1-r_i(t)))} - e^{-D(t)n(t)})dN(t) - n(t)e^{-D(t)n(t)}dt \\
&= -n(t)e^{-D(t)n(t)}dt + (1 - e^{-D(t)n(t)})dN(t) \\
&= -n(t)F(X(t))dt + (1 - F(X(t)))dN(t) \\
&= -n(t)m(t)dt + (1 - m(t))dN(t).
\end{aligned}$$

■

Finally, given a set of items $\mathcal{I}$, we cast the design of a spaced repetition algorithm as the search of the optimal item reviewing intensities $\boldsymbol{u}(t) = [u_i(t)]_{i\in\mathcal{I}}$ that minimize the expected value of a particular (convex) loss function $\ell(\boldsymbol{m}(t), \boldsymbol{n}(t), \boldsymbol{u}(t))$ of the recall probability of the items, $\boldsymbol{m}(t) = [m_i(t)]_{i\in\mathcal{I}}$, the forgetting rates, $\boldsymbol{n}(t) = [n_i(t)]_{i\in\mathcal{I}}$, and the intensities themselves,

[8] To derive Eq. 4.3, we assume that the recall probability $m_i(t)$ is set to 1 every time item i is reviewed. Here, one may also account for item difficulty by considering that, for more difficult items, the recall probability is set to $o_i \in [0,1)$ every time item i is reviewed.

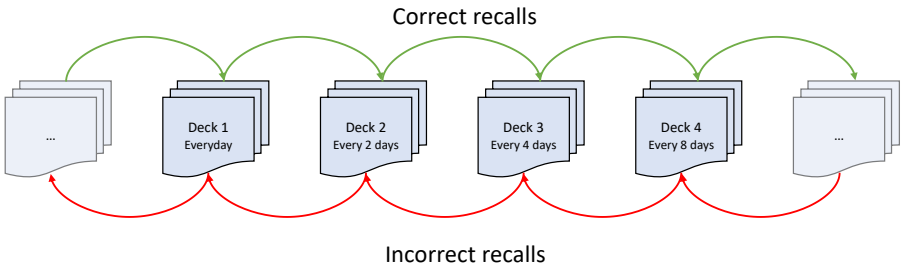


Figure 4.1: The Leitner system. The learner picks flashcards for review from several decks and flashcards are moved from one deck to another based on the recall outcome after a review. The higher the index of the deck, the lower the rate at which cards are picked for review from that deck, *e.g.*, cards in deck 1 may be reviewed once per day, cards in deck 2 once every two days, and so on.

$\boldsymbol{u}(t)$, over a time window $(t_0, t_f]$, *i.e.*,

$$\begin{aligned} &\underset{\boldsymbol{u}(t_0,t_f]}{\text{minimize}} \;\; \mathbb{E}\left[\phi(\boldsymbol{m}(t_f),\boldsymbol{n}(t_f)) + \int_{t_0}^{t_f} \ell(\boldsymbol{m}(\tau),\boldsymbol{n}(\tau),\boldsymbol{u}(\tau))d\tau\right] \\ &\text{subject to} \;\; \boldsymbol{u}(t) \geq 0 \; \forall t \in (t_0,t_f), \end{aligned} \tag{4.7}$$

where $\boldsymbol{u}(t_0,t_f]$ denotes the item reviewing intensities from t_0 to t_f, the expectation is taken over all possible realizations of the associated counting processes and (item) recalls, the loss function is nonincreasing (nondecreasing) with respect to the recall probabilities (forgetting rates and intensities) so that it rewards long-lasting learning while limiting the number of item reviews, and $\phi(\boldsymbol{m}(t_f),\boldsymbol{n}(t_f))$ is an arbitrary penalty function.[9] Here, note that the forgetting rates $\boldsymbol{n}(t)$ and recall probabilities $\boldsymbol{m}(t)$, as defined by Eq. 4.2 and Eq. 4.3, depend on the reviewing intensities $\boldsymbol{u}(t)$ we aim to optimize since $\mathbb{E}[d\boldsymbol{N}(t)] = \boldsymbol{u}(t)dt$.

Leitner system: a case study

In this section, we first describe the Leitner system and then show that it can be explicitly cast using our modeling framework with particular choices of α, β and $n_i(0)$.

Leitner system. More than 40 years ago, Sebastian Leitner introduced the Leitner System as a method used to memorize flash cards [63]. Since then, several variants of the system have been introduced and some of them are still in active use. Next, for the sake of brevity, we describe one of these variants, which has been recently studied by Reddy *et al.* [92] and Settles *et al.* [98].

The learner maintains several *decks* of flashcards, labelled $j \in \mathbb{Z}$, each of which is reviewed at exponentially decreasing frequency $\lambda_j = \lambda_0 c^j$, for some constants c and λ_0. Whenever a card i from deck j is reviewed, it is moved to deck $j+1$ if it is recalled correctly (*i.e.*, if $r_i(t) = 1$), or else (if $r_i(t) = 0$) it is moved to deck $j-1$, as shown in Figure 4.1. The intuition behind the Leitner system is that cards which belong to a deck with a large index j have been learned (or were easy to learn), *i.e.*, they have a low forgetting rate, and cards which are in lower decks have

[9] The penalty function $\phi(\boldsymbol{m}(t_f),\boldsymbol{n}(t_f))$ is necessary to derive the optimal reviewing intensities $\boldsymbol{u}^*(t)$.

not been learned yet (or were difficult to learn), *i.e.*, they have a high forgetting rate. Then, the learning strategy of the learner is to select flashcards at random from any deck as long as the reviewing rate for flashcards in each deck j remains λ_j, *i.e.*, the expected number of flashcards selected for review from deck j in any time interval Δt is $\lambda_j \times \Delta t$.

Modeling the Leitner system. For ease of exposition, we assume that the number of decks is unbounded both from above and below[10], *i.e.*, there are always decks with higher (or lower) rate of review than the current deck. Under this assumption, we can faithfully represent the Leitner system under our modeling framework as follows.

First, we assign a fixed forgetting rate $\tilde{n}_j$ to all flashcards in deck j, *i.e.*, if at time t, a flashcard i is in deck j then $n_i(t) = \tilde{n}_j$ and, at the beginning, all flashcards are placed in the first deck, *i.e.*, $\forall i.\, n_i(0) = \tilde{n}_0$. Then, every time a card i moves from deck j to $j+1$, we change its forgetting rate $n_i(t)$ by a factor of $(1-\alpha) = \frac{\tilde{n}_{j+1}}{\tilde{n}_j}$ and, similarly, every time it moves from j to $j-1$, we change it by a factor of $(1+\beta) = \frac{\tilde{n}_{j-1}}{\tilde{n}_j}$. Finally, we set $\lambda_j = \tilde{n}_j$, *i.e.*, the reviewing intensity of a card is proportional to the rate of forgetting associated to its deck, where the constant of proportionality has been absorbed into $\tilde{n}_j$. Now we can uncover α and β by solving the equations $(1-\alpha) = \frac{\tilde{n}_{j+1}}{\tilde{n}_j} = c$ and $(1+\beta) = \frac{\tilde{n}_{j-1}}{\tilde{n}_j} = \frac{1}{c}$. It is easy to see that, with minor modifications, our framework can be also used to represent many other variants of the Leitner system with, *e.g.*, bounded number of decks.

4.2 The MEMORIZE Algorithm

The spaced repetition problem, as defined by Eq. 4.7, can be tackled from the perspective of stochastic optimal control of jump SDEs [45]. Here, we first derive a solution to the problem considering only one item, provide an efficient practical implementation of the solution, and then generalize it to the case of multiple items.

Given an item i, we can write the spaced repetition problem, *i.e.*, Eq. 4.7, for it with reviewing intensity $u_i(t) = u(t)$ and associated counting process $N_i(t) = N(t)$, recall outcome $r_i(t) = r(t)$, recall probability $m_i(t) = m(t)$ and forgetting rate $n_i(t) = n(t)$. Further, using Eq. 4.2 and Eq. 4.3, we can define the forgetting rate $n(t)$ and recall probability $m(t)$ by the following two coupled jump SDEs:

$$\begin{aligned} dn(t) &= -\alpha n(t) r(t) dN(t) + \beta n(t)(1 - r(t)) dN(t) \\ dm(t) &= -n(t) m(t) dt + (1 - m(t)) dN(t) \end{aligned}$$

with initial conditions $n(t_0) = n_0$ and $m(t_0) = m_0$.

Next, we will define an optimal cost-to-go function J for the above problem, use Bellman's principle of optimality to derive the corresponding Hamilton-Jacobi-Bellman (HJB) equation [9], and exploit the unique structure of the HJB equation to find the optimal solution to the problem.

Definition 2 *The optimal cost-to-go $J(m(t), n(t), t)$ is defined as the minimum of the expected*

[10] In a real deployment of the system, the bounds on the number of decks will be self imposed, *e.g.*, flashcards which move to a high enough deck will effectively never be studied, *i.e.*, they will be *retired*.

value of the cost of going from state $(m(t), n(t))$ at time t to the final state at time t_f.

$$J = \min_{u(t,t_f]} \mathbb{E}_{(N(s),r(s))|_{s=t}^{s=t_f}} \Big[\phi(m(t_f), n(t_f)) + \int_t^{t_f} \ell(m(\tau), u(\tau)) d\tau\Big] \tag{4.8}$$

Now, we use Bellman's principle of optimality, which the above definition allows[11], to break the problem into smaller subproblems. With $dJ(m(t), n(t), t) = J(m(t+dt), n(t+dt), t+dt) - J(m(t), n(t), t)$, we can, hence, rewrite Eq. 4.8 as:

$$\begin{aligned} J(m(t), n(t), t) &= \\ &\min_{u(t,t+dt]} \mathbb{E}[J(m(t+dt), n(t+dt), t+dt)] + \ell(m(t), n(t), u(t))dt \\ 0 &= \min_{u(t,t+dt]} \mathbb{E}[dJ(m(t), n(t), t)] + \ell(m(t), n(t), u(t))dt. \end{aligned} \tag{4.9}$$

Then, to derive the HJB equation, we differentiate J with respect to time t, $m(t)$ and $n(t)$ using the following Lemma:

Lemma 3 *Let $x(t)$ and $y(t)$ be two jump-diffusion processes defined by the following jump SDEs:*

$$\begin{aligned} dx(t) =& f(x(t), y(t), t)dt + g(x(t), y(t), t)z(t)dN(t) + h(x(t), y(t), t)(1 - z(t))dN(t) \\ dy(t) =& p(x(t), y(t), t)dt + q(x(t), y(t), t)dN(t) \end{aligned}$$

where $N(t)$ is a jump process and $z(t) \in \{0, 1\}$. If function $F(x(t), y(t), t)$ is once continuously differentiable in $x(t)$, $y(t)$ and t, then,

$$\begin{aligned} dF(x, y, t) = (F_t + fF_x + pF_y)(x, y, t)dt + [F(x + g, y + q, t)z(t) + F(x + h, y + q, t)(1 - z(t)) - \\ F(x, y, t)]dN(t), \end{aligned}$$

where for notational simplicity we dropped the arguments of the functions f, g, h, p, q and argument of state variables.

Proof According to the definition of differential,

$$\begin{aligned} dF := dF(x(t), y(t), t) &= F(x(t + dt), y(t + dt), t + dt) - F(x(t), y(t), t) \\ &= F(x(t) + dx(t), y(t) + dy(t), t + dt) - F(x(t), y(t), t). \end{aligned}$$

Then, using Itö's calculus, we can write

$$\begin{aligned} dF \overset{\text{dt}}{=}_{zol} F(x + fdt + g, y + pdt + q, t + dt)dN(t)z + F(x + fdt + h, y + pdt + q, t + dt)dN(t)(1 - z) \\ + F(x + fdt, y + pdt, t + dt)(1 - dN(t)) - F(x, y, t) \end{aligned} \tag{4.10}$$

[11] Bellman's principle of optimality readily follows using the Markov property of the recall probability $m(t)$ and forgetting rate $n(t)$.

where for notational simplicity we drop arguments of all functions except F and dN. Then, we expand the first three terms:

$$\begin{aligned}
F(x+fdt+g,y+pdt+q,t+dt) &= F(x+g,y+q,t)+F_x(x+g,y+q,t)fdt \\
&\quad +F_y(x+g,y+q,t)pdt+F_t(x+g,y+q,t)dt \\
F(x+fdt+h,y+pdt+q,t+dt) &= F(x+h,y+q,t)+F_x(x+h,y+q,t)fdt \\
&\quad +F_y(x+h,y+q,t)pdt+F_t(x+h,y+q,t)dt \\
F(x+fdt,y+pdt,t+dt) &= F(x,y,t)+F_x(x,y,t)fdt+F_y(x,y,t)pdt \\
&\quad +F_t(x,y,t)dt
\end{aligned}$$

using that the bilinear differential form $dt\,dN(t)=0$. Finally, by substituting the above three equations into Eq. 4.10, we conclude that:

$$\begin{aligned}
dF(x(t),y(t),t) &= (F_t+fF_x+pF_y)(x(t),y(t),t)dt+ \\
&\quad \big[F(x+g,y+q,t)z(t)-F(x,y,t)\big]dN(t),
\end{aligned}$$

■

Specifically, consider $x(t)=n(t)$, $y(t)=m(t)$, $z(t)=r(t)$ and $J=F$ from equation 4.9 in the above equation, then,

$$\begin{aligned}
dJ(m,n,t) &= J_t(m,n,t)-nmJ_m(m,n,t)+[J(1,(1-\alpha)n,t)r(t)+J(1,(1+\beta)n,t)(1-r) \\
&\quad -J(m,n,t)]dN(t).
\end{aligned}$$

substituting the above in equation 4.9 we get:

$$\begin{aligned}
0 &= J_t(m,n,t)-nmJ_m(m,n,t)+\min_{u(t,t+dt]}\big\{\ell(m,n,u)[J(1,(1-\alpha)n,t)m \\
&\quad +J(1,(1+\beta)n,t)(1-m)-J(m,n,t)]u(t)\big\}
\end{aligned} \tag{4.11}$$

To solve the above equation, we need to define the loss ℓ. Following the literature on stochastic optimal control [9], we consider the following quadratic form, which is nonincreasing (nondecreasing) with respect to the recall probabilities (intensities) so that it rewards learning while limiting the number of item reviews:

$$\ell(m(t),n(t),u(t)) = \frac{1}{2}(1-m(t))^2+\frac{1}{2}qu^2(t), \tag{4.12}$$

where q is a given parameter, which trade-offs recall probability and number of item reviews—the higher its value, the lower the number of reviews. Note that this particular choice of loss function does not directly place a hard constraint on number of reviews, instead, it limits the number of reviews by penalizing high reviewing intensities[12].

Under these definitions, we can find the relationship between the optimal intensity and the

[12] Given a desired level of practice, the value of the parameter q can be easily found by simulation since the average number of reviews decreases monotonically with respect to q.

optimal cost by taking the derivative with respect to $u(t)$ in Eq. 4.11:

$$u^*(t) = q^{-1}\big[J(m(t),n(t),t) - J(1,(1-\alpha)n(t),t)m(t) \\ - J(1,(1+\beta)n(t),t)(1-m(t))\big]_+.$$

Finally, we plug in the above equation in Eq. 4.11 and find that the optimal cost-to-go J needs to satisfy the following nonlinear differential equation:

$$0 = J_t(m(t),n(t),t) - n(t)m(t)J_m(m(t),n(t),t) + \\ \frac{1}{2}(1-m(t))^2 - \frac{1}{2}q^{-1}\big[J(m(t),n(t),t) - \\ J(1,(1-\alpha)n(t),t)m(t) - J(1,(1+\beta)n(t),t)(1-m(t))\big]_+^2.$$

To continue further, we rely on the following technical Lemma which derives the optimal cost-to-go J for a parametrized family of losses ℓ.

Lemma 4 *Consider the following family of losses with parameter $d > 0$,*

$$\begin{aligned} \ell_d(m(t),n(t),u(t)) &= h_d(m(t),n(t)) + g_d^2(m(t),n(t)) + \frac{1}{2}qu(t)^2, \\ g_d(m(t),n(t)) &= 2^{-1/2}\Big[c_2\frac{\log(d)}{-m(t)^2+2m(t)-d} - c_2\frac{\log(d)}{1-d} + c_1 m(t)\log\left(\frac{1+\beta}{1-\alpha}\right) - \\ &\quad c_1\log(1+\beta)\Big]_+, \\ h_d(m(t),n(t)) &= -\sqrt{q}m(t)n(t)c_2\frac{(-2m(t)+2)\log(d)}{(-m(t)^2+2m(t)-d)^2}. \end{aligned} \tag{4.13}$$

where $c_1, c_2 \in \mathbb{R}$ are arbitrary constants. Then, the cost-to-go $J_d(m(t),n(t),t)$ that satisfies the HJB equation, defined by Eq. 4.11, is given by:

$$J_d(m(t),n(t),t) = \sqrt{q}\left(c_1\log(n(t)) + c_2\frac{\log(d)}{-m(t)^2+2m(t)-d}\right) \tag{4.14}$$

and the optimal intensity is given by:

$$u_d^*(t) = q^{-1/2}[c_2\frac{\log(d)}{-m(t)^2+2m(t)-d} - c_2\frac{\log(d)}{1-d} + c_1 m(t)\log\left(\frac{1+\beta}{1-\alpha}\right) - c_1\log(1+\beta)]_+.$$

Proof *Consider the family of losses defined by Eq. 4.13 and the functional form for the cost-to-go defined by Eq. 4.14. Then, for any parameter value $d > 0$, the optimal intensity $u_d^*(t)$ is given by*

$$\begin{aligned} u_d^*(t) &= q^{-1}\left[J_d(m(t),n(t),t) - J_d(1,(1-\alpha)n(t),t)m(t) - J_d(1,(1+\beta)n(t),t)(1-m(t))\right]_+ \\ &= q^{-1/2}\left[c_2\frac{\log(d)}{-m^2+2m-d} - c_2\frac{\log(d)}{1-d} + c_1 m(t)\log\left(\frac{1+\beta}{1-\alpha}\right) - c_1\log(1+\beta)\right]_+, \end{aligned}$$

and the HJB equation is satisfied:

$$\begin{aligned}
&\frac{\partial J_d(m,n,t)}{\partial t} - mn\frac{\partial J_d(m,n,t)}{\partial m} + h_d(m,n) + g_d^2(m,n) - \frac{1}{2}q^{-1}\big(J_d(m,n,t) \\
&- J_d(1,(1-\alpha)n,t)m - J_d(1,(1+\beta)n,t)(1-m)\big)_+^2 \\
&= \sqrt{q}mnc_2\frac{(-2m+2)\log(d)}{(-m^2+2m-d)^2} + h_d(m,n) + g_d^2(m,n) - \frac{1}{2}\left[c_1\log(n) + c_2\frac{\log(d)}{-m^2+2m-d}\right. \\
&\quad \left. -m\left(c_1\log(n(1-\alpha)) + c_2\frac{\log(d)}{1-d}\right) - (1-m)\left(c_1\log(n(1+\beta)) + c_2\frac{\log(d)}{1-d}\right)\right]_+^2 \\
&= \sqrt{q}mnc_2\frac{(-2m+2)\log(d)}{(-m^2+2m-d)^2} \underbrace{-\sqrt{q}mnc_2\frac{(-2m+2)\log(d)}{(-m^2+2m-d)^2}}_{h_d(m,n)} \\
&\quad -\frac{1}{2}\Big[c_2\frac{\log(d)}{-m^2+2m-d} - c_2\frac{\log(d)}{1-d} + c_1 m\log(\frac{1+\beta}{1-\alpha}) - c_1\log(1+\beta)\Big]_+^2 \\
&\quad +\frac{1}{2}\Big[c_2\frac{\log(d)}{-m^2+2m-d} - c_2\frac{\log(d)}{1-d} + c_1 m\log(\frac{1+\beta}{1-\alpha}) - c_1\log(1+\beta)\Big]_+^2
\end{aligned}$$

where for notational simplicity $m = m(t)$, $n = n(t)$ and $u = u(t)$. ■

Using this Lemma, the optimal reviewing intensity is readily given by following Theorem:

Theorem 5 *Given a single item, the optimal reviewing intensity for the spaced repetition problem, defined by Eq. 4.7, under quadratic loss, defined by Eq. 4.12, is given by*

$$u^*(t) = q^{-1/2}(1-m(t)). \tag{4.15}$$

Proof Consider the family of losses defined by Eq. 4.13 in Lemma 4 whose optimal intensity is given by:

$$u_d^*(t) = q^{-1/2}\left[c_2\frac{\log(d)}{-m^2+2m-d} - c_2\frac{\log(d)}{1-d} + c_1 m(t)\log\left(\frac{1+\beta}{1-\alpha}\right) - c_1\log(1+\beta)\right]_+.$$

Now, set the constants $c_1, c_2 \in \mathbb{R}$ to the following values:

$$c_1 = \frac{-1}{\log\left(\frac{1+\beta}{1-\alpha}\right)} \quad c_2 = \frac{-\log(1-\alpha)}{\log\left(\frac{1+\beta}{1-\alpha}\right)}.$$

Since the HJB equation is satisfied for any value of $d > 0$, we can recover the quadratic loss $l(m,n,u)$ and derive its corresponding optimal intensity $u^*(t)$ using point wise convergence:

$$\begin{aligned}
l(m(t),n(t),u(t)) &= \lim_{d\to 1} l_d(m(t),n(t),u(t)) = \frac{1}{2}(1-m(t))^2 + \frac{1}{2}qu^2(t), \\
u^*(t) &= \lim_{d\to 1} u_d^*(t) = q^{-1/2}(1-m(t)),
\end{aligned}$$

Algorithm 3: The MEMORIZE Algorithm

Input: Parameters q, α, β, last reviewing time s and forgetting rate $n(s)$

Output: Next reviewing time t' and forgetting rate $n(t')$

$\forall t > s.\ n(t) \leftarrow n(s)$ — compute forgetting rate.
$\forall t > s.\ m(t) \leftarrow \exp(-(t-s)n(t))$ — compute recall probability.
$\forall t > s.\ u(t) \leftarrow q^{-1/2}(1-m(t))$ — compute reviewing intensity.
$t' \leftarrow \mathit{Sample}(u(t))$ — sample next reviewing time
$r \leftarrow \mathit{ReviewItem}(t')$ — review item, $r \in \{0,1\}$
$n(t') \leftarrow (1-\alpha)n(t)r + (1+\beta)n(t)(1-r)$ — update forgetting rate
return $t', n(t')$

where we used that $\lim_{d\to 1} \frac{\log(d)}{1-d} = -1$ (L'Hospital's rule). This concludes the proof. ■

Note that the optimal intensity only depends on the recall probability, whose dynamics are given by Eqs. 4.2 and 4.3, and thus allows for a very efficient procedure to sample reviewing times, which we name MEMORIZE. Algorithm 3 provides a pseudocode implementation of MEMORIZE. Within the algorithm, $\mathit{Sample}(u(t))$ samples from an inhomogeneous poisson process with intensity $u(t)$ and it returns the sampled time and $\mathit{ReviewItem}(t')$ returns the recall outcome r of an item reviewed at time t', where $r = 1$ indicates the item was recalled successfully and $r = 0$ indicates it was not recalled. Moreover, note that t denotes a (time) parameter, s and t' denote specific (time) values, and we sample from an inhomogeneous poisson process using a standard thinning algorithm [67][13].

Given a set of multiple items $\mathcal{I}$ with reviewing intensities $\boldsymbol{u}(t)$ and associated counting processes $\boldsymbol{N}(t)$, recall outcomes $\boldsymbol{r}(t)$, recall probabilities $\boldsymbol{m}(t)$ and forgetting rates $\boldsymbol{n}(t)$, we can solve the spaced repetition problem defined by Eq. 4.7 similarly as in the case of a single item. More specifically, consider the following quadratic form for the loss ℓ:

$$\ell(m(t), n(t), u(t)) = \frac{1}{2}\sum_{i\in\mathcal{I}} (1-m_i(t))^2 + \frac{1}{2}\sum_{i\in\mathcal{I}} q_i u_i^2(t),$$

where $\{q_i\}_{i\in\mathcal{I}}$ are given parameters, which trade-off recall probability and number of item reviews and may *favor* the learning of one item over another. Then, one can exploit the independence among items assumption to derive the optimal reviewing intensity for each item, proceeding similarly as in the case of a single item:

Theorem 6 *Given a set of items $\mathcal{I}$, the optimal reviewing intensity for each item $i \in \mathcal{I}$ in the spaced repetition problem, defined by Eq. 4.7, under quadratic loss is given by*

$$u_i^*(t) = q_i^{-1/2}(1-m_i(t)). \tag{4.16}$$

Finally, note that we can easily sample item reviewing times simply by running $|\mathcal{I}|$ instances of MEMORIZE, one per item.

[13] In some practical deployments, one may like to discretize the optimal intensity $u(t)$ and, *e.g.*, "at top of each hour, decide whether to do a review or not".

4.3 Power-Law Forgetting Curve Model

In this section we examine our proposed model under the power-law forgetting curve model. The probability of recalling an item i at time t under the power-law forgetting curve model is given by [120]:

$$m_i(t) := \mathbb{P}(r_i(t)) = (1 + \omega(t - t_r))^{-n_i(t)}, \tag{4.17}$$

where t_r is the time of the last review, $n_i(t) \in RR^+$ is the forgetting rate and ω is a time scale parameter.

Similarly as in Proposition 1 for the exponential forgetting curve model, we can express the dynamics of the recall probability $m_i(t)$ by means of a SDE with jumps:

$$dm_i(t) = -\frac{n_i(t)m_i(t)\omega dt}{(1 + \omega D_i(t))} + (1 - m_i(t))dN_i(t) \tag{4.18}$$

where $D_i(t) := t - t_r$ and thus the differential of $D_i(t)$ is readily given by $dD_i(t) = dt - D_i(t)dN_i(t)$.

Next, similarly as in the case of the exponential forgetting curve model in the main paper, we consider a single item with $n_i(t) = n(t)$, $m_i(t) = m(t)$, $D_i(t) = D(t)$ and $r_i(t) = r(t)$, and adapt Lemma 3 to the power-law forgetting curve model as follows:

Lemma 7 *Let $x(t)$ and $y(t)$, $k(t)$ be three jump-diffusion processes defined by the following jump SDEs:*

$$\begin{aligned}
dx(t) =& f(x(t),y(t),t)dt + g(x(t),y(t),t)z(t)dN(t) + h(x(t),y(t),t)(1 - z(t))dN(t) \\
dy(t) =& p(x(t),y(t),t)dt + q(x(t),y(t),t)dN(t) \\
dk(t) =& s(x(t),y(t),k(t),t)dt + v(x(t),y(t),k(t),t)dN(t)
\end{aligned}$$

where $N(t)$ is a jump process and $z(t) \in \{0,1\}$. If function $F(x(t),y(t),k(t),t)$ is once continuously differentiable in $x(t)$, $y(t)$, $z(t)$ and t, then,

$$\begin{aligned}
dF(x,y,k,t) = (F_t + fF_x + pF_y + sF_s)(x,y,k,t)dt + [F(x + g, y + q, k + v, t)z(t) \\
+ F(x + h, y + q, k + v, t)(1 - z(t)) - F(x,y,t)]dN(t),
\end{aligned}$$

where for notational simplicity we dropped the arguments of the functions f, g, h, p, q, s, v and argument of state variables.

Then, if we consider $x(t) = n(t)$, $y(t) = m(t)$, $k(t) = D(t)$, $z(t) = r(t)$ and $J = F$ in the above Lemma, the differential of the optimal cost-to-go is readily given by

$$\begin{aligned}
dJ(m,n,t) = J_t(m,n,t) - \frac{\omega nm}{1 + \omega D}J_m(m,n,D,t) + J_D(m,n,D,t) + [J(1,(1 - \alpha)n,0,t)r(t) \\
+ J(1,(1 + \omega)n,0,t)(1 - r) - J(m,n,D,t)]dN(t).
\end{aligned}$$

Moreover, under the same loss function $\ell(m(t),n(t),u(t))$ as in Eq. 4.12, it is easy to show

that the optimal cost-to-go J needs to satisfy the following nonlinear partial differential equation:

$$0 = J_t(m(t),n(t),t) - \frac{\omega n(t) m(t)}{1+\omega D} J_m(m(t),n(t),t) + \quad J_D(m(t),n(t),t) + \frac{1}{2}(1-m(t))^2 \\ - \frac{1}{2} q^{-1} \left(J(m(t),n(t),t) - J(1,(1-\alpha)n(t),t) m(t) - J(1,(1+\beta)n(t),t)(1-m(t)) \right)_+^2. \tag{4.19}$$

Then, we can adapt Lemma 4 to derive the optimal scheduling policy for a single item under the power-law forgetting curve model:

Lemma 8 *Consider the following family of losses with parameter $d > 0$,*

$$\begin{aligned} \ell_d(m(t),n(t),D(t),u(t)) &= h_d(m(t),n(t),D(t)) + g_d^2(m(t),n(t)) + \frac{1}{2} q u(t)^2, \\ g_d(m(t),n(t)) &= 2^{-1/2} \left[c_2 \frac{\log(d)}{-m(t)^2 + 2m(t) - d} - c_2 \frac{\log(d)}{1-d} + c_1 m(t) \log\left(\frac{1+\beta}{1-\alpha}\right) - \right. \\ &\qquad \left. c_1 \log(1+\beta) \right]_+, \\ h_d(m(t),n(t)) &= -\sqrt{q} \frac{\omega n(t) m(t)}{1+\omega D(t)} c_2 \frac{(-2m(t)+2)\log(d)}{(-m(t)^2+2m(t)-d)^2}. \end{aligned} \tag{4.20}$$

where $c_1, c_2 \in \mathbb{R}$ are arbitrary constants. Then, the cost-to-go $J_d(m(t),n(t),t)$ that satisfies the HJB equation, defined by Eq. 4.19, is given by:

$$J_d(m(t),n(t),D(t),t) = \sqrt{q} \left(c_1 \log(n(t)) + c_2 \frac{\log(d)}{-m(t)^2 + 2m(t) - d} \right) \tag{4.21}$$

which is independent of $D(t)$, and the optimal intensity is given by:

$$u_d^*(t) = q^{-1/2} [c_2 \frac{\log(d)}{-m(t)^2 + 2m(t) - d} - c_2 \frac{\log(d)}{1-d} + c_1 m(t) \log\left(\frac{1+\beta}{1-\alpha}\right) - c_1 \log(1+\beta)]_+.$$

Proof *Consider the family of losses defined by Eq. 4.20 and the functional form for the cost-to-go defined by Eq. 4.21. Then, for any parameter value $d > 0$, the optimal intensity $u_d^*(t)$ is given by*

$$\begin{aligned} u_d^*(t) &= q^{-1} \left[J_d(m(t),n(t),t) - J_d(1,(1-\alpha)n(t),t) m(t) - J_d(1,(1+\beta)n(t),t)(1-m(t)) \right]_+ \\ &= q^{-1/2} \left[c_2 \frac{\log(d)}{-m^2 + 2m - d} - c_2 \frac{\log(d)}{1-d} + c_1 m(t) \log\left(\frac{1+\beta}{1-\alpha}\right) - c_1 \log(1+\beta) \right]_+, \end{aligned}$$

and the HJB equation is satisfied:

$$
\begin{aligned}
&\frac{\partial J_d(m,n,t)}{\partial t} - \frac{\omega n m}{1+\omega D}\frac{\partial J_d(m,n,t)}{\partial m} + h_d(m,n) + g_d^2(m,n) - \frac{1}{2}q^{-1}\big(J_d(m,n,t) \\
&- J_d(1,(1-\alpha)n,t)m - J_d(1,(1+\beta)n,t)(1-m)\big)_+^2 \\
&= \sqrt{q}\frac{\beta c_2 n m}{1+\omega D}\frac{(-2m+2)\log(d)}{(-m^2+2m-d)^2} + h_d(m,n) + g_d^2(m,n) - \frac{1}{2}\left[c_1\log(n) + c_2\frac{\log(d)}{-m^2+2m-d}\right. \\
&\quad \left. -m\left(c_1\log(n(1-\alpha)) + c_2\frac{\log(d)}{1-d}\right) - (1-m)\left(c_1\log(n(1+\beta)) + c_2\frac{\log(d)}{1-d}\right)\right]_+^2 \\
&= \sqrt{q}\frac{\omega n m c_2}{1+\omega D(t)}\frac{(-2m+2)\log(d)}{(-m^2+2m-d)^2} \underbrace{-\sqrt{q}\frac{\omega n m c_2}{1+\omega D}\frac{(-2m+2)\log(d)}{(-m^2+2m-d)^2}}_{h_d(m,n)} \\
&\quad - \frac{1}{2}\Big[c_2\frac{\log(d)}{-m^2+2m-d} - c_2\frac{\log(d)}{1-d} + c_1 m\log(\frac{1+\beta}{1-\alpha}) - c_1\log(1+\beta)\Big]_+^2 \\
&\quad + \frac{1}{2}\Big[c_2\frac{\log(d)}{-m^2+2m-d} - c_2\frac{\log(d)}{1-d} + c_1 m\log(\frac{1+\beta}{1-\alpha}) - c_1\log(1+\beta)\Big]_+^2
\end{aligned}
$$

where for notational simplicity $m = m(t)$, $n = n(t)$, $D = D(t)$ *and* $u = u(t)$. ■

Finally, reusing Theorem 5, the optimal reviewing intensity for a single item under the power-law forgetting curve model is given by

$$u^*(t) = \lim_{d\to 1} u_d^*(t) = q^{-1/2}(1-m(t)).$$

It is then straightforward to derive the optimal reviewing intensity for a set of items, which adopts the same form as in Theorem 6.

Multiscale Context Memory Model

In this section, we will briefly describe the Multiscale Context Model (MCM) of memory [84] and sketch how to derive the optimal reviewing schedule for this model.

MCM models memory as *activation* of M time-varying *context* units with the values $\{x_i(t)\}_{[M]}$. The units are *leaky* and lose their activation at different decay rates $\{\tau_i\}_{[M]}; \tau_{i+1} > \tau_i$, such that $\Delta x_i(t+\Delta t) = x_i(t)\exp(-\Delta t/\tau_i)$. The accumulated activity of the units, each scaled by $\{\gamma_i\}_{[M]}$, represents the total *trace strength* of the item at time t, *i.e.*, $s_M(t) = \frac{\sum_{i=0}^M \gamma_i x_i(t)}{\sum_{i=0}^M \gamma_i}$. The probability of recall of the item at time t is then calculated as $m_{MCM}(t) = \min\{1, s_M(t)\}$.

Each time a review happens, the activation of the context pools *jumps* by an amount which depends on how much activation at timescale i, *i.e.*, $x_i(t)$ had *contributed* to the recall. So if a review happens at time t_r, $\Delta x_i(t_r^+) = \varepsilon(1 - s_i(t_r^-))$, where $s_i(t)$ represents the strength of all activations which decay *faster* than the current trace, *i.e.*, $s_i(t) = \frac{\sum_{j=1}^i \gamma_j x_j(t)}{\sum_{j=1}^i \gamma_j}$. The value of ε is

	HLR	Our Model	Our Model
	Exponential	Exponential	Power-law
MAE↓	0.128	0.129	**0.105**
AUC↑	0.538	**0.542**	0.533
COR_h ↑	**0.201**	0.165	0.123

Table 4.1: Predictive performance of the exponential and power-law forgetting curve models in comparison with the results reported by Settles et al. [98]. The arrows indicate whether a higher value of the metric is better (↑) or a lower value (↓).

set to ε_α if the recall was successful and to ε_β if the recall was unsuccessful. For a more detailed description of MCM, we refer the readers to [84].

The dynamics of the context pools $\{x_i(t)\}_{[M]}$ can be converted to the corresponding SDEs readily as follows.

$$\begin{aligned} s_i(t) &= \frac{\sum_{j=1}^{i} \gamma_j x_j(t)}{\sum_{j=1}^{i} \gamma_j} \\ dx_i(t) &= \varepsilon_\alpha r(t)(1-s_i(t))dN(t) + \varepsilon_\beta(1-r(t))(1-s_i(t))dN(t) - \frac{x_i(t)}{\tau_i}dt \\ ds_M(t) &= \frac{\sum_{j=1}^{M} \gamma_j dx_j(t)}{\sum_{j=1}^{M} \gamma_j}. \end{aligned} \tag{4.22}$$

For modeling the probability of recall $m_{MCM}(t)$, we can use a differentiable approximation to the $\min\{1, s_M(t)\}$ function. For example, we can use hyperbolic-tan, and approximate $m_{MCM}(t)$ via $\tilde{m}_{MCM}(t)$:

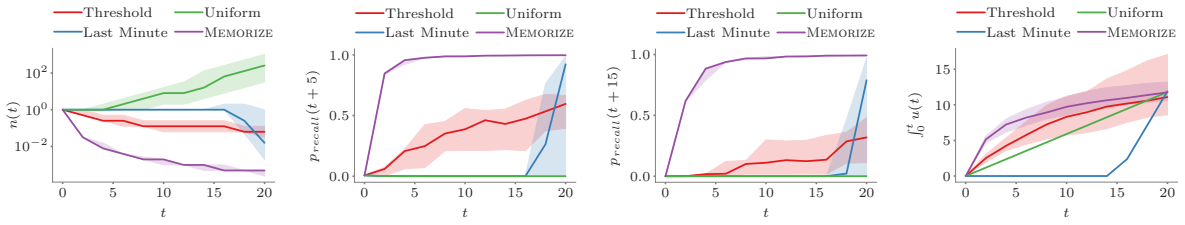
$$\begin{aligned} \tilde{m}_{MCM}(t) &= \tanh s_M(t) \\ \implies d\tilde{m}_{MCM}(t) &= (1-\tilde{m}^2_{MCM}(t))ds_M(t). \end{aligned} \tag{4.23}$$

One can contrast Eq. 4.22 and Eq. 4.23 with Eq. 4.2 and Eq. 4.3 (or Eq. 4.18) respectively to compare the derivations for the exponential forgetting curve model and the MCM.

Extension of Lemma 4 for Eq. 4.23 is straight-forward and the nonlinear partial differential equation corresponding to Eq. 4.11 (or Eq. 4.19) can be solved to arrive at the optimal scheduling for the MCM model. The resulting equation, however, does not readily admit to an analytical solution as was the case for the exponential and power-law forgetting curve models.

4.4 Synthetic Experiments

In this section, our goal is analyzing the performance of MEMORIZE under a controlled setting using metrics and baselines that we cannot compute in the real data we have access to. We evaluate the performance of MEMORIZE using two quality metrics: recall probability $m(t+\tau)$ at a given time in the future $t+\tau$ and forgetting rate $n(t)$. Here, by considering high (low) values of τ, we can assess long-term (short-term) retention. Moreover, we compare the performance of our



(a) Forgetting rate (b) Short-term recall prob. (c) Long-term recall prob. (d) Reviewing intensity

Figure 4.2: Performance of MEMORIZE in comparison with several baselines. The solid lines are median values and the shadowed regions are 30% confidence intervals. Short-term recall probability corresponds to $m(t+5)$ and long-term recall probability to $m(t+15)$. In all cases, we use $\alpha = 0.5$, $\beta = 1$, $n(0) = 1$ and $t_f - t_0 = 20$. Moreover, we set $q = 3 \cdot 10^{-4}$ for MEMORIZE, $\mu = 0.6$ for the uniform reviewing schedule, $t_{lm} = 5$ and $\mu = 2.38$ for the last minute reviewing schedule, and $m^{th} = 0.7$ and $c = \zeta = 5$ for the threshold based reviewing schedule. Under these parameter values, the total number of reviewing events for all algorithms are equal (with a tolerance of 5%).

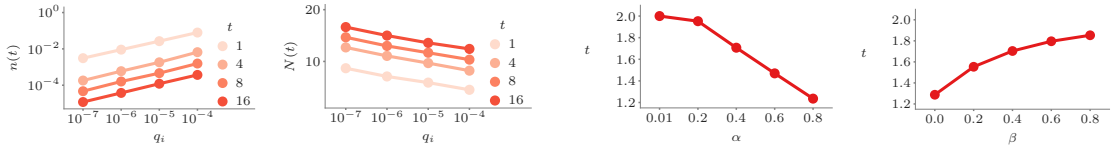
algorithm with three baselines: (i) a *uniform* reviewing schedule, which sends item(s) for review at a constant rate μ; (ii) a *threshold* based reviewing schedule, which increases the reviewing intensity of an item by $c \exp\left((t-s)/\zeta\right)$ at time s, when its recall probability reaches a threshold m^{th}; and, (iii) a *last minute* reviewing schedule, which only sends item(s) for review during a period $[t_{lm}, t_f]$, at a constant rate μ therein.[14] Unless otherwise stated, we set the parameters of the baselines and our algorithm such that the total number of reviewing events during $(t_0, t_f]$ are equal.

First, we run 100 independent simulations and compute the above quality metrics over time. Figure 4.2 summarizes the results, which show that our model: (i) consistently outperforms all the baselines in terms of both quality metrics; (ii) is more robust across runs both in terms of quality metrics and reviewing schedule; and (iii) reduces the reviewing intensity as times goes by and the recall probability improves, as one could have expected.

Second, we experiment with different values for the parameter q, which controls the learning effort required by MEMORIZE—the lower its value, the higher the number of reviewing events. Intuitively, one may also expect the learning effort to influence how quickly a learner memorizes a given item—the lower its value, the quicker a learner will memorize it. Figure 4.3a confirms this intuition by showing the average forgetting rate $n(t)$ and number of reviewing events $N(t)$ at several times t for different q values.

Finally, we experiment with different values for the parameters α and β, which capture the aptitude of a learner and the difficulty of the item to be learned—the higher (lower) the value of α (β), the quicker a learner will memorize the item. In Figure 4.3b, we evaluate quantitatively this effect by means of the average time the learner takes to reach a forgetting rate of $n(t) = \frac{1}{2}n(0)$ using MEMORIZE for different parameter values.

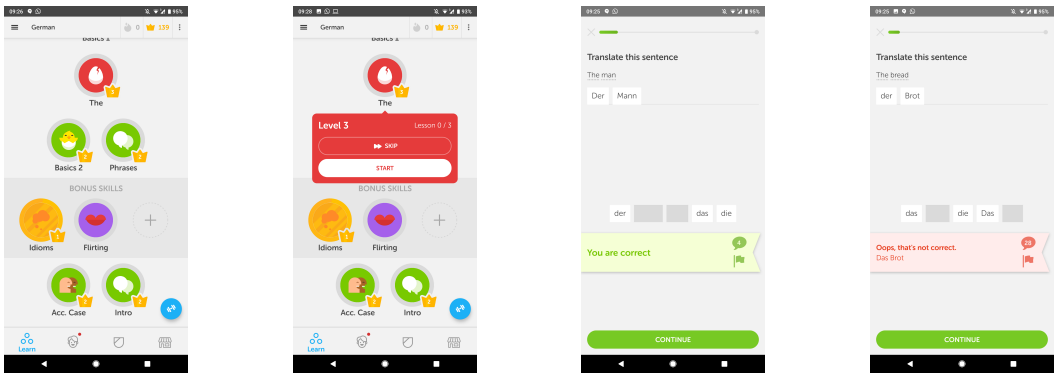
[14] The last minute reviewing schedule is only introduced here and was not used in empirical evaluations since in Duolingo there is no terminal time t_f which users target. Additionally, in many (user, item) pairs, the first review takes place close to $t = 0$ and thus the last minute baseline is equivalent to the uniform reviewing schedule.



(a) Learning effort

(b) Item difficulty and aptitude of the learner

Figure 4.3: Learning effort, aptitude of the learner and item difficulty. Panel (a) shows the average forgetting rate $n(t)$ and number of reviewing events $N(t)$ for different values of the parameter q, which controls the learning effort. Panel (b) shows the average time the learner takes to reach a forgetting rate $n(t) = \frac{1}{2}n(0)$ for different values of the parameters α and β, which capture the aptitude of the learner and the item difficulty. In Panel (a), we use $\alpha = 0.5$, $\beta = 1$, $n(0) = 1$ and $t_f - t_0 = 20$. In Panel (b), we use $n(0) = 20$ and $q = 0.02$. In both panels, error bars are too small to be seen.



(a) Skill selection

(b) Session start

(c) Correct response

(d) Incorrect response

Figure 4.4: The Duolingo learning app.

4.5 Natural Experimental Design

We use data gathered from Duolingo, a popular language-learning online platform,[15] to validate our algorithm MEMORIZE.

Learners on Duolingo have a source language (which they already know) and a target language (which they wish to learn). Upon log-in, they are greeted with a screen to select the skill they wish to train/learn, shown in Figure 4.4a. As soon as they select a skill, a session begins (see Figure 4.4b). In each session, the learner is asked to translate ∼10 *phrases* from the source language to the target language or vice-versa. A typical session may last for ∼2–5 minutes but if it is interrupted in the middle due to any reason (loss of connectivity, student logging-out, *etc.*), the session (and its associated data) is discarded. Figures 4.4c and 4.4d show a correct translation and an incorrect translation, respectively, for one such phrases for the language pair (English, French). For each word that appears at least once in a session, our dataset contains identity of the learner, a timestamp (in UTC) indicating when the session started, the total number of times

[15] The dataset is available at https://github.com/duolingo/halflife-regression.

the word appears, and how many times the learner correctly/incorrectly translated the phrases containing that word. We consider a session to be a single point in time (localized at the start of the session), when the student practiced all the words appearing in it. A learner may do several sessions in a single sitting: the sessions in our dataset are separated by a median of $\sim$7 minutes, *including* the time the learner spent in the session.

As discussed in Section 4.6, our experimental design differs from that of Settles *et al.* [98] in some respects: we consider a word i to be recalled correctly at time t, *i.e.*, $r_i(t) = 1$, if the student answered *all* the questions containing that word correctly. Otherwise, we assume that the word was not recalled correctly. The dataset consists of $\sim$12 million *sessions* of study, involving $\sim$5.3 million unique (user, word) pairs, which we denote by $\mathcal{D}$, collected over the period of two weeks. In a single session, a user answers multiple questions, each of which contains multiple words[16]. Each word maps to an item i and the fraction of correct recalls of sentences containing a word i in the session is used as an empirical estimate of its recall probability $\hat{m}(t)$ at the time of the session t, as in previous work [98]. If a word is recalled perfectly during a session then it is considered as a successful recall, *i.e.*, $r_i(t) = 1$, and otherwise it is considered as an unsuccessful recall, *i.e.*, $r_i(t) = 0$. Since we can only expect the estimation of the model parameters to be accurate for users and items with enough number of reviewing events, we only consider users with at least 30 reviewing events and words that were reviewed at least 30 times. After this preprocessing step, our dataset consists of $\sim$5.2 million unique (user, word) pairs.

We compare the performance of our method with two baselines: (i) a *uniform* reviewing schedule, which sends item(s) for review at a constant rate μ; and (ii) a *threshold* based reviewing schedule, which increases the reviewing intensity of an item by $c\exp\left((t-s)/\zeta\right)$ at time s, when its recall probability reaches a threshold m^{th}. The threshold baseline is similar to the heuristics proposed by previous work [79, 69, 11], which schedule reviews just as the learner is about to forget an item. We do not compare with the algorithm proposed by Reddy *et al.* [92] because, as it is specially designed for Leitner system, it assumes a discrete set of forgetting rate values and, as a consequence, is not applicable to our (more general) setting.

Although we cannot make actual interventions to evaluate the performance of each method, the following insight allows for a large-scale natural experiment: Duolingo uses hand-tuned spaced repetition algorithms, which propose reviewing times to the users, however, users often do not perform reviews exactly at the recommended times, and thus schedules for some (user, item) pairs will be *closer* to uniform than threshold or MEMORIZE and vice versa, as shown in Figure 4.5. As a consequence, we are able to assign each (user, item) pair to a treatment group (*i.e.*, MEMORIZE) or a control group (*i.e.*, uniform or threshold). More in detail, we leverage this insight to design a robust evaluation procedure which relies on: (i) likelihood comparisons to determine *how closely* a user followed a particular reviewing schedule during all reviews but the last in a reviewing sequence, *i.e.*, $e_1, \ldots, e_{n-1}$ in a sequence with n reviews; (ii) a quality metric, empirical forgetting rate $\hat{n}$, which can be estimated using only the last review e_n (and the *retention interval* $t_n - t_{n-1}$) of each reviewing sequence and does not depend on the particular choice of memory model. Refer to section 4.5.1 for more details on our evaluation procedure[17].

[16] Refer to for additional details on the Duolingo dataset.

[17] Note that our goal is to evaluate how *well* different reviewing schedule spaces the reviews—our objective is not to evaluate the predictive power of the underlying memory models, we are relying on previous work for that [98, 120]. However, for completeness, we provide a series of benchmarks and evaluations later for the memory models we used in this work in .

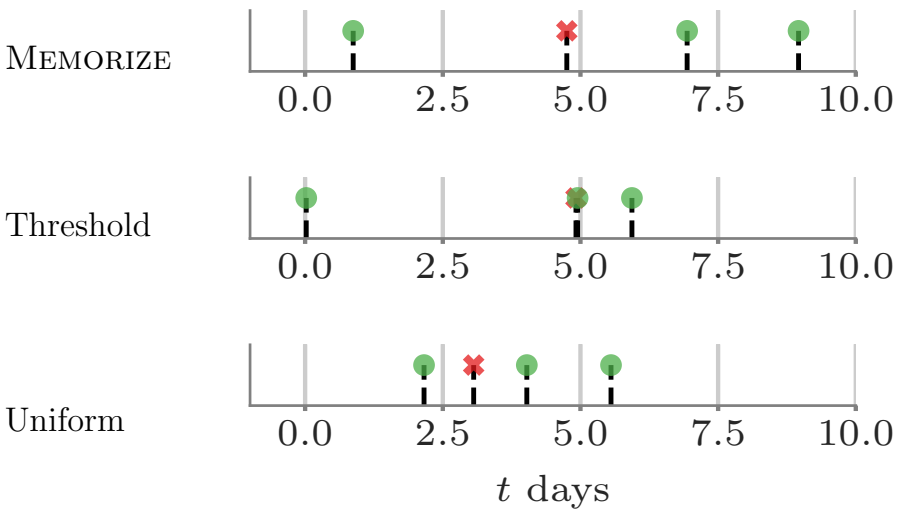
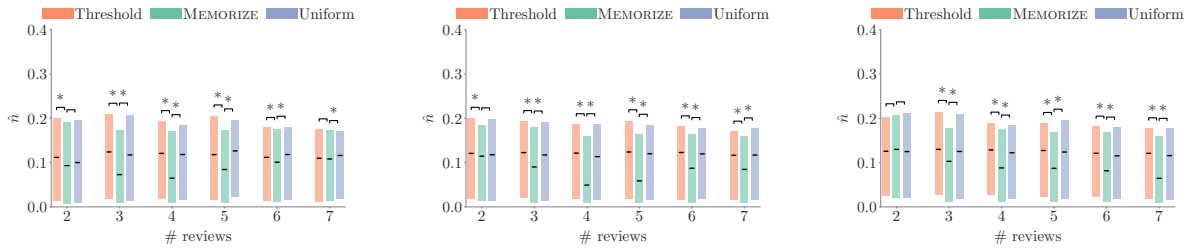


Figure 4.5: Examples of (user, item) pairs whose corresponding reviewing times have high likelihood under MEMORIZE (top), threshold based reviewing schedule (middle) and uniform reviewing schedule (bottom). In every figure, each candle stick corresponds to a reviewing event with a green circle (red cross) if the recall was (un)successful, and time $t = 0$ corresponds to the first time the user is exposed to the item in our dataset, which may or may not correspond with the first reviewing event. The pairs whose reviewing times follow more closely MEMORIZE or the threshold based schedule tend to increase the time interval between reviews every time a recall is successful while, in contrast, the uniform reviewing schedule does not. MEMORIZE tends to space the reviews more than the threshold based schedule, achieving the same recall pattern with less effort.

4.5.1 Evaluation Procedure

In order to evaluate performance of our proposed algorithm, we rely on the following evaluation procedure. For each (user, item) reviewing sequence, we first perform a likelihood-based comparison to determine how closely it follows a specific reviewing schedule (be it MEMORIZE, uniform or threshold) during the first $n-1$ reviews, the *training reviews*, where n is the number of reviews in the reviewing sequence. Second, we compute a quality metric, empirical forgetting rate $\hat{n}(t_n)$, using the last review, the n-th review or *test review*, and the retention interval $t_n - t_{n-1}$. Third, for each reviewing sequence, we record the value of the quality metric, the *training period* (*i.e.*, $T = t_{n-1} - t_1$) and the likelihood under each reviewing schedule. Finally, we control for the training period and the number of reviewing events and create the treatment and control groups by picking the top 25% pairs in terms of likelihood for each method, where we skip any sequence lying in the top 25% for more than one method. Later we also show that our evaluation procedure satisfies the random assignment assumption for the item difficulties between treatment and control groups [106].

In the above procedure, to do the likelihood-based comparison, we first estimate the parameters α and β and the initial forgetting rate $n_i(0)$ using half-life regression on the Duolingo dataset. Here, note that we fit a single set of parameters α and β for all items and a different initial



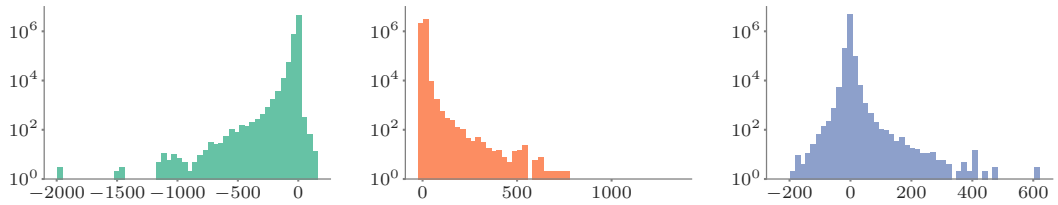
(a) $T = 3 \pm 0.3$ days **(b)** $T = 5 \pm 0.5$ days **(c)** $T = 7 \pm 0.7$ days

Figure 4.6: Average empirical forgetting rate for the top 25% pairs in terms of likelihood for MEMORIZE, the uniform reviewing schedule and the threshold based reviewing schedule for sequences with different number of reviews n and different training periods $T = t_{n-1} - t_1$. Boxes indicate 25% and 75% quantiles and solid lines indicate median values, where lower values indicate better performance. MEMORIZE offers a competitive advantage with respect to the uniform and the threshold based baselines and, as the training period increases, the number of reviews under which MEMORIZE achieves the greatest competitive advantage increases. For each distinct number of reviews and training periods, $*$ indicates a statistically significant difference (Mann–Whitney U test; p-value<0.05) between MEMORIZE vs. threshold and MEMORIZE vs. uniform scheduling.

forgetting rate $n_i(0)$ per item i as discussed later. Then, for each user, we use maximum likelihood estimation to fit the parameter q in MEMORIZE and the parameter μ in the uniform reviewing schedule. For the threshold based schedule, we fit one set of parameters c and ζ for each sequence of review events, using maximum likelihood estimation for the parameter c and grid search for the parameter ζ, and we fit one parameter m^{th} for each user using grid search. Finally, we compute the likelihood of the times of the $n-1$ reviewing events for each (user, item) pair under the intensity given by MEMORIZE, *i.e.*, $u(t) = q^{-1/2}(1-m(t))$, the intensity given by the uniform schedule, *i.e.*, $u(t) = \mu$, and the intensity given by the threshold based schedule, *i.e.*, $u(t) = c\exp((t-s)/\zeta)$. The likelihood $LL(\{t_i\})$ of a set of reviewing events $\{t_i\}$ given an intensity function $u(t)$ can be computed as follows [1]:

$$LL(\{t_i\}) = \sum_i \log u(t_i) - \int_0^T u(t)\,dt.$$

We compute the likelihood of each sequence of review events in our dataset under different reviewing schedules. Figure 4.7 summarizes the results by showing the empirical distribution of estimated likelihood values for MEMORIZE, threshold and uniform schedules. Since Duolingo uses a near-optimal hand-tuned reviewing schedule, the peak of the distribution for MEMORIZE corresponds to the highest likelihood values, *i.e.*, there are many (user, item) pairs who follow MEMORIZE closely. .



(a) MEMORIZE (b) Threshold (c) Uniform

Figure 4.7: Empirical distribution of log-likelihood values for all (user, item) pairs under different reviewing schedules. Since Duolingo uses a near-optimal hand-tuned reviewing schedule, the peak of the distribution for MEMORIZE corresponds to the highest likelihood values, *i.e.*, there are many (user, item) pairs who follow MEMORIZE closely.

4.5.2 Quality Metric: Empirical Forgetting Rate

For each (user, item), the empirical forgetting rate is an empirical estimate of the forgetting rate by the time t_n of the last reviewing event, *i.e.*,

$$\hat{n} = -\log(\hat{m}(t_n))/(t_n - t_{n-1}),$$

where $\hat{m}(t_n)$ is the empirical recall probability, which consists of the fraction of correct recalls of sentences containing word (item) i in the session at time t_n. Note that this empirical estimate does not depend on the particular choice of memory model and, given a sequence of reviews, the lower the empirical forgetting rate, the more effective the reviewing schedule.

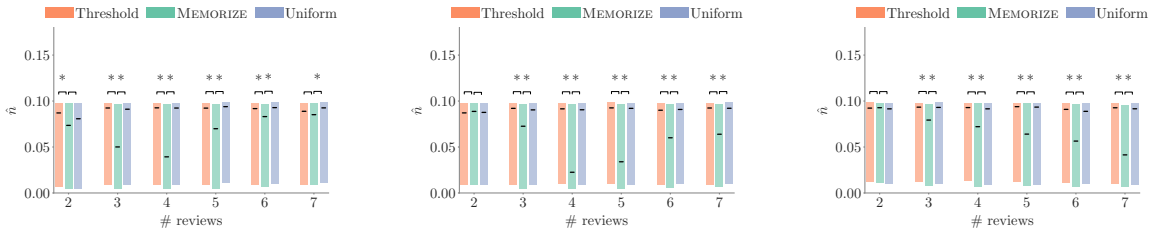
Moreover, for a more fair comparison across items, we normalize each empirical forgetting rate using the average empirical initial forgetting rate of the corresponding item at the beginning of the observation window $\hat{n}_0$, *i.e.*, for an item i,

$$\hat{n}_0 = \frac{1}{|\mathcal{D}_i|} \sum_{(u,i)\in\mathcal{D}_i} \hat{n}_{0,(u,i)},$$

where $\mathcal{D}_i \subseteq \mathcal{D}$ is the subset of (user, item) pairs in which item i was reviewed. Furthermore, $\hat{n}_{0,(u,i)} = -\log(\hat{m}(t_{(u,i),1}))/(t_{(u,i),1} - t_{(u,i),0})$, where $t_{(u,i),k}$ is the k-th review in the reviewing sequence associated to the (u,i) pair. However, our results are not sensitive to this normalization step. More specifically, we re-run our analysis using unnormalized values for the empirical forgetting rates. Figure 4.8 summarizes the results, which still show a competitive advantage of MEMORIZE with respect to the uniform and threshold based baselines. The primary difference between using normalization (Figure 4.6) or not using normalization (Figure 4.8) is just a scaling factor.

4.6 Results and Discussion

We first group (user, item) pairs by their number of reviews n and their training period, *i.e.*, $t_{n-1} - t_1$. Then, for each recall pattern, we create the treatment (MEMORIZE) and control



(a) $T = 3 \pm 0.3$ days (b) $T = 5 \pm 0.5$ days (c) $T = 7 \pm 0.7$ days

Figure 4.8: Average empirical forgetting rate without normalization for the top 25% pairs in terms of likelihood for MEMORIZE, the uniform reviewing schedule and the threshold based reviewing schedule for sequences with different number of reviews n and different training periods $T = t_{n-1} - t_1$. Boxes indicate 25% and 75% quantiles and solid lines indicate median values, where lower values indicate better performance. MEMORIZE offers a competitive advantage with respect to the uniform and the threshold based baselines and, as the training period increases, the number of reviews under which MEMORIZE achieves the greatest competitive advantage increases. For each distinct number of reviews and training periods, $*$ indicates a statistically significant difference (Mann–Whitney U test; p-value<0.05) between MEMORIZE vs. threshold and between MEMORIZE and uniform scheduling.

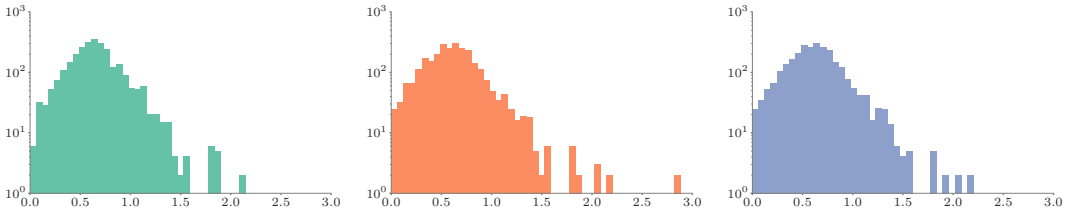
(uniform and threshold) groups and, for every reviewing sequence in each group, compute its empirical forgetting rate. Figure 4.6 summarizes the results for sequences with up to seven reviews since the beginning of the observation window for three distinct training periods. The results show that MEMORIZE offers a competitive advantage with respect to the uniform and the threshold based baselines and, as the training period increases, the number of reviews under which MEMORIZE achieves the greatest competitive advantage increases. To rule out that the competitive advantage that MEMORIZE offers with respect to the uniform and the threshold based baselines is a consequence of selection bias due to the item difficulty, we compute the empirical distributions of item difficulties for the treatment (MEMORIZE) and control (uniform and threshold) groups and check whether the allocation of items across the treatment and control groups resemble random assignment.

Figure 4.9 summarizes the results for reviewing sequences with a training period $T = 5 \pm 0.5$ days, which show a striking similarity between distributions across groups. In fact, the treatment group is *indistinguishable* from both the control groups in terms of item difficulties because their values are within a standardize mean difference (SMD)[18] of 0.25 standard deviations [106]. Similar results are obtained for sequences with a training period $T = 3 \pm 0.3$ and with $T = 7 \pm 0.7$.

Finally, we would like to acknowledge that there may be other covariates that influence the performance of a learner such as, *e.g.*, time of the day, amount of stress, amount of sleep or degree of concentration. Unfortunately, we did not have access to measurement about them.

Next, we go a step further and verify that, whenever a specific learner follows MEMORIZE more closely, her performance is superior. More specifically, for each learner with at least 70 reviewing sequences with a training period $T = 8 \pm 3.2$ days, we select their top and

[18] The standardized mean difference (SDM) is defined as the difference in means of the treatment and control group divided by the pool standard devision of both groups.



(a) MEMORIZE (b) Threshold (c) Uniform

Figure 4.9: Empirical distribution of n_0 for (user, item) pairs with a training period $T = 5 \pm 0.5$ days in the treatment (MEMORIZE) and control groups (threshold based and uniform).

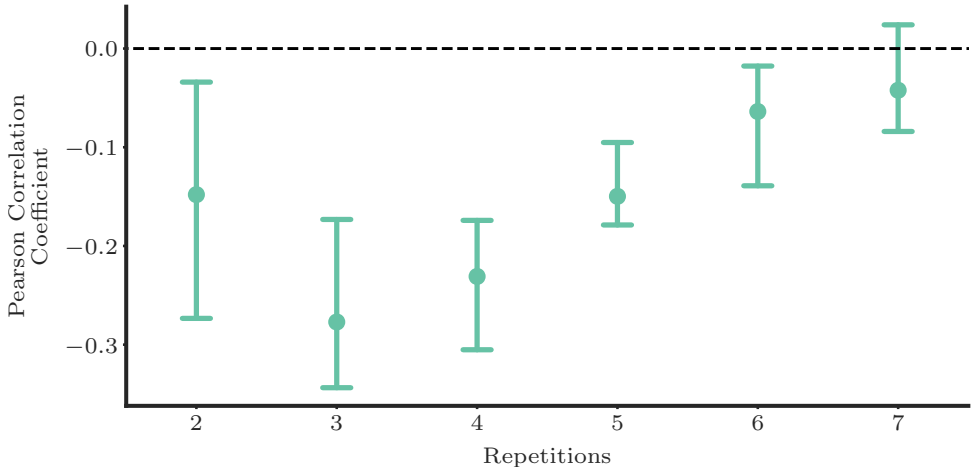


Figure 4.10: Pearson correlation coefficient between the log-likelihood of the top and bottom 50% reviewing sequences of a learner under MEMORIZE and its associated empirical forgetting rate. The dots indicate median values and the bars indicate standard error. Lower correlation values correspond to greater gains due to MEMORIZE. To ensure reliable estimation, we considered learners with at least 70 reviewing sequences with a training period $T = 8 \pm 3.2$ days. There were 322 of such learners.

bottom 50% reviewing sequences in terms of log-likelihood under MEMORIZE and compute the Pearson correlation coefficient between their empirical forgetting rate and log-likelihood values. Figure 4.10 summarizes the results, which show that users, in average, achieve lower empirical forgetting rates whenever they follow MEMORIZE more closely.

Since the Leitner system [63], there have been a wealth of spaced repetition algorithms [15, 86, 79, 69, 78]. However, there have been a paucity of work on designing adaptive data-driven spaced repetition algorithms with provable guarantees. In this work, we have introduced a principled modeling framework to design a new generation of online spaced repetition algorithms with provable guarantees, which are specially designed to adapt to the learners' performance, as monitored by modern spaced repetition software and online platforms. Our modeling framework represents spaced repetition using the framework of marked temporal point processes and SDEs with jumps and, exploiting this representation, it casts the design of spaced repetition algorithms as a stochastic optimal control problem of such jump SDEs. Since our framework is agnostic

to the particular modeling choices, *i.e.*, the memory model and the quadratic loss function, we believe it provides a novel, powerful tool to find spaced repetition algorithms that are provably optimal under a given choice of memory model and loss.

There are many interesting directions for future work. For example, it would be interesting to perform large scale interventional experiments to assess the performance of our algorithm in comparison with existing spaced repetition algorithms deployed by, *e.g.*, Duolingo. Moreover, in our work, we consider a particular quadratic loss and soft constraints on the number of reviewing events, however, it would be useful to derive optimal reviewing intensities for other losses capturing particular learning goals and hard constraints on the number of events. We assumed that, by reviewing an item, one can only influence its recall probability and forgetting rate. However, items may be dependent and, by reviewing an item, one can influence the recall probabilities and forgetting rates of several items. The dataset we used spans only two weeks and that places a limitation on the range of time intervals between reviews and retention intervals we can study. It would be very interesting to evaluate our framework in datasets spanning longer periods of time. Finally, we believe that the mathematical techniques underpinning our algorithm, *i.e.*, stochastic optimal control of SDEs with jumps, have the potential to drive a new generation of control algorithms in a wide range of applications.

Predictive performance of the memory model

In this section we evaluate the predictive performance of the exponential and power-law forgetting curve models whose forgetting rates we estimated using a variant of Half-life regression (HLR) [98]. However before doing that, we highlight the differences between the original HLR and the variant we used.

The original HLR and the variant we used differ in the way successful and unsuccessful recalls change the forgetting rate. In our work, the forgetting rate at time t depends on $n_{\checkmark}(t) = \int_0^t r(\tau) dN(\tau)$ and $n_{\times}(t) = \int_0^t (1 - r(\tau)) dN(\tau)$. In contrast, in the original HLR, the forgetting rate at time t depends on $\sqrt{n_{\checkmark}(t) + 1}$ and $\sqrt{n_{\times}(t) + 1}$. The rationale behind our modeling choice is to be able to express the dynamics of the forgetting rate using a linear stochastic differential equation with jumps. Moreover, Settles *et al.* consider each session to contain multiple review events for each item. Hence, within a session, the $n_{\times}(t)$ and $n_{\checkmark}(t)$ may increase by more than one. In contrast, we consider each session to contain a single review event for each item because the reviews in each session take place in a very short time and it is likely that after the first review, the user will recall the item correctly during that session. Hence, we only increase one of $n_{\checkmark}(t)$ or $n_{\times}(t)$ by exactly 1 after each session and consider an item has been successfully recalled during a session if all reviews were successful, *i.e.*, $p_{recall} = 1$. Noticeably, $\sim$83% of the items were successfully recalled during a session.

Table 4.1 summarizes our results on the Duolingo dataset in terms of mean absolute error (MAE), area under curve (AUC) and correlation (COR_h), which show that the performance of both the exponential and power-law forgetting curve models with forgetting rates estimated using the variant of HLR is comparable to the performance of the exponential forgetting curve model with forgetting rates estimated using the original HLR. In the above results, note that we fitted a single set of parameters α and β for all items and a different initial forgetting rate $n_i(0)$ per item i. One could think of estimating item specific (or even user specific) parameters α and

K	Interval boundaries
3	[0, 20 minutes, 2.9 days, ∞]
4	[0, 9 minutes, 21.5 hours, 5.2 days, ∞]
5	[0, 6 minutes, 1.5 hours, 1.8 days, 7.3 days, ∞]

Table 4.2: The boundaries of intervals used to divide review times into bins based on K-quantiles of inter-review times in the Duolingo dataset.

(a) Comparing $\alpha^{(\text{row})}$ and $\alpha^{(\text{col})}$

	2	3	4	5
1	0.317	0.317	0.317	0.317
2		0.312	0.312	0.312
3			0.317	0.317
4				0.318

(b) Comparing $\beta^{(\text{row})}$ and $\beta^{(\text{col})}$

	2	3	4	5
1	0.165	0.172	0.165	0.165
2		0.318	0.302	0.302
3			0.318	0.318
4				0.302

Table 4.3: p-values obtained after using Welch's t-test for populations with different variance to reject the null hypothesis that the samples (*i.e.*, 400 samples of $\{\alpha^{(i)}\}$ and $\{\beta^{(i)}\}$ for $i \in [5]$) have the same mean value. In all cases, we find no evidence to reject the null hypothesis. The results for other values of K were qualitatively similar.

β, however, we found that our dataset is not large enough to provide accurate estimates of such item specific parameters. More generally, there is always a trade-off between the complexity of the model and the size of the dataset used to fit the model parameters.

Constant vs time-varying α and β parameters

In previous studies [18, 19], it has been shown the retention rate follows an *inverted U-shape*, *i.e.*, *mass practice* does not improve the forgetting rate, and thus one could argue that our framework should consider time-varying parameters $\alpha_i(t)$ and $\beta_i(t)$. In this section, we show that, for the reviewing sequences in our Duolingo dataset, allowing for time-varying $\alpha_i(t)$ and $\beta_i(t)$ in our modeling framework does not lead to more accurate recall predictions. This was one of the reasons, in addition to tractability, for considering constant parameters α_i and β_i.

Formulation. In Eq. 4.2, we have considered α and β as constants, *i.e.*, they do not vary with the review interval $t - t_r$, where t_r is the time of last review. We have dropped the subscript i denoting the item for ease of exposition. We can make a zeroth-order approximation to time varying (α, β) by allowing them to be piecewise constant for K mutually exclusive and exhaustive review-time intervals $\left\{B^{(i)}\right\}_{[K]}$. We denote the value that α (β) takes in interval $B^{(i)}$ as $\alpha^{(i)}$ ($\beta^{(i)}$) and modify the forgetting rate update equation to

$$dn(t) = \underbrace{-\alpha^{(i)} n(t) r(t) dN(t) + \beta^{(i)} n(t)(1 - r(t)) dN(t)}_{i \text{ such that } t - t_r \in B^{(i)}}$$

If we find that $\exists \{i,j\} \subset [K]$ such that $\alpha^{(i)}$ ($\beta^{(i)}$) is *significantly* different from $\alpha^{(j)}$ ($\beta^{(j)}$), then we would conclude that α (β) vary with review-time.

We obtain repeated estimates of $\left\{\alpha^{(i)}\right\}_{[K]}$ and $\left\{\beta^{(i)}\right\}_{[K]}$ by fitting our model to datasets sampled with replacement from our Duolingo dataset, *i.e.*, via bootstrapping. The Welch's t-test is used to test if the difference in mean values of the parameters in different bins is significant.

Experimental setup. We set the bins boundaries by determining the K-quantiles of the review times in our dataset. Table 4.2 shows that the bin boundaries for different K are quite varied and adequately cover long time-scales as well as review intervals which are short enough to capture *massed practicing*. This method of binning also ensures that we have sufficient samples for accurate estimation ($\sim 5.2e6/K$) for all parameters. Then we use the variant of HLR described in Appendix 4.6 to fit the parameters in 400 different datasets using bootstrapping. The regularization parameters are determined via grid-search using a train/test dataset. We thus obtain 400 samples of $\left\{\alpha^{(i)}\right\}_{[K]}$ and $\left\{\beta^{(i)}\right\}_{[K]}$ for $K \in \{3,4,5\}$ and $i \in [K]$. Using Welch's t-test for distributions with varying variances, we observe that the mean values of the distributions of $\left\{\alpha^{(i)}\right\}_{[K]}$ ($\left\{\beta^{(i)}\right\}_{[K]}$) and $\left\{\alpha^{(j)}\right\}_{[K]}$ ($\left\{\beta^{(j)}\right\}_{[K]}$) are not significantly different for any $\{i,j\}$. As an example, the p-values obtained for $K = 5$ are shown in Table 4.3.

Bibliography

[1] O. Aalen, O. Borgan, and H. K. Gjessing. *Survival and event history analysis: a process point of view*. Springer, 2008.

[2] B. T. Adler and L. De Alfaro. A content-driven reputation system for the wikipedia. In *WWW*, 2007.

[3] A. Anderson, J. Kleinberg, and S. Mullainathan. Assessing Human Error Against a Benchmark of Perfection. In *KDD*, 2016.

[4] Richard C Atkinson. Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology*, 96(1):124, 1972.

[5] L. Averell and A. Heathcote. The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, 55(1), 2011.

[6] Oana Balmau, Rachid Guerraoui, Anne-Marie Kermarrec, Alexandre Maurer, Matej Pavlovic, and Willy Zwaenepoel. Limiting the spread of fake news on social media platforms by evaluating users' trustworthiness. *arXiv preprint arXiv:1808.09922*, 2018.

[7] W. Barnett. The modern theory of consumer behavior: Ordinal or cardinal? *Quarterly Journal of Austrian Economics*, 6(1):41–65, 2003.

[8] A. R. Benson, D. F. Gleich, and J. Leskovec. Tensor spectral clustering for partitioning higher-order network structures. In *SIAM International Conference on Data Mining*, pages 118–126. SIAM, 2015.

[9] D. P. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, 1995.

[10] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000.

[11] EL Bjork and R Bjork. Making things hard on yourself, but in a good way. *Psychology in the Real World*, 2011.

[12] Aaron Blake. A new study suggests fake news might have won Donald Trump the 2016 election. https://www.washingtonpost.com/news/the-fix/wp/2018/04/03/a-new-study-suggests-fake-news-might-have-won-donald-trump-the-2016-election/, 2018. [Online; accessed 8-July-2022].

[13] Kristine C Bloom and Thomas J Shuell. Effects of massed and distributed practice on the learning and retention of second-language vocabulary. *The Journal of Educational Research*, 74(4):245–248, 1981.

[14] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *ACM Transactions on Internet Technology*, 5(1):231–297, 2005.

[15] G. Branwen. Spaced repetition. https://www.gwern.net/Spaced%20repetition, 2016.

[16] Julia Cambre, Scott Klemmer, and Chinmay Kulkarni. Juxtapeer: Comparative peer review yields higher quality feedback and promotes deeper reflection. In *CHI*, 2018.

[17] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

[18] Nicholas J Cepeda, Harold Pashler, Edward Vul, John T Wixted, and Doug Rohrer. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological bulletin*, 132(3):354, 2006.

[19] Nicholas J Cepeda, Edward Vul, Doug Rohrer, John T Wixted, and Harold Pashler. Spacing effects in learning: A temporal ridgeline of optimal retention. *Psychological science*, 19(11):1095–1102, 2008.

[20] Laurent Charlin and Richard Zemel. The toronto paper matching system: an automated paper-reviewer assignment system. In *ICML*, 2013.

[21] Charles LA Clarke, Maheedhar Kolla, Gordon V Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM, 2008.

[22] Nicholas Confessore. Cambridge Analytica and Facebook: The Scandal and the Fallout So Far. https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html, 2018. [Online; accessed 8-July-2022].

[23] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schölkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *ICML*, 2014.

[24] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. Gomez-Rodriguez. Learning and forecasting opinion dynamics in social networks. In *NIPS*, 2016.

[25] Frank N Dempster. Spacing effects and their implications for theory and practice. *Educational Psychology Review*, 1(4):309–330, 1989.

[26] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 2016.

[27] X. L. Dong, E. Gabrilovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From data fusion to knowledge fusion. *VLDB*, 2014.

[28] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang. Knowledge-based trust: Estimating the trustworthiness of web sources. *VLDB*, 2015.

[29] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005.

[30] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent Marked Temporal Point Process: Embedding Event History to Vector. In *KDD*, 2016.

[31] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *NIPS*, 2013.

[32] H. Ebbinghaus. *Memory: a contribution to experimental psychology*. Teachers College, Columbia University, 1885.

[33] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155, 2013.

[34] Benedict Evans. A new study suggests fake news might have won Donald Trump the 2016 election. https://www.ben-evans.com/benedictevans/2020/6/14/75-years-of-us-advertising, 2020. [Online; accessed 8-July-2022].

[35] M. Farajtabar, N. Du, M. Gomez-Rodriguez, I. Valera, H. Zha, and L. Song. Shaping social activity by incentivizing users. In *NIPS*, 2014.

[36] M. Farajtabar, Y. Wang, M. Gomez-Rodriguez, S. Li, H. Zha, and L. Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *NIPS*, 2015.

[37] M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha. Multistage campaigning in social networks. In *NIPS*, 2016.

[38] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[39] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Reducing controversy by connecting opposing views. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 81–90. ACM, 2017.

[40] Kiran Garimella, Aristides Gionis, Nikos Parotsidis, and Nikolaj Tatti. Balancing information exposure in social networks. In *Advances in Neural Information Processing Systems*, pages 4663–4671, 2017.

[41] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML*, 2011.

[42] M. Gomez-Rodriguez, K. P. Gummadi, and B. Schoelkopf. Quantifying information overload in social media and its impact on social contagions. In *ICWSM*, 2014.

[43] S. Greenstein and F. Zhu. Is wikipedia biased? *The American economic review*, 102(3):343–348, 2012.

[44] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *VLDB*, 2004.

[45] F. B. Hanson. *Applied stochastic processes and control for Jump-diffusions: modeling, analysis, and computation*. SIAM, 2007.

[46] Anne-Wil Harzing, Joyce Baldueza, Wilhelm Barner-Rasmussen, Cordula Barzantny, Anne Canabal, Anabella Davila, Alvaro Espejo, Rita Ferreira, Axele Giroud, Kathrin Koester, et al. Rating versus ranking: What is the best way to reduce response and language bias in cross-national research? *International Business Review*, 18(4), 2009.

[47] Herrman, John. Inside facebook's political-media machine. *New York Times*, 2016.

[48] N. Hodas and K. Lerman. How visibility and divided attention constrain social contagion. In *SocialCom*, 2012.

[49] Lily Hu and Yiling Chen. A short-term intervention for long-term fairness in the labor market. In *Proceedings of the Web Conference*, 2018.

[50] D. Hunter, P. Smyth, D. Q. Vu, and A. U. Asuncion. Dynamic egocentric models for citation networks. In *ICML*, 2011.

[51] David R Hunter et al. Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32(1):384–406, 2004.

[52] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[53] J. Kang and K. Lerman. Vip: Incorporating human cognitive biases in a probabilistic model of retweeting. In *ICSC*, 2015.

[54] Hilbert J. Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182, 2012.

[55] M. Karimi, E. Tavakoli, M. Farajtabar, L. Song, and M. Gomez-Rodriguez. Smart Broadcasting: Do you want to be seen? In *KDD*, 2016.

[56] Jack Kiefer, Jacob Wolfowitz, et al. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3):462–466, 1952.

[57] J. Kim, B. Tabibian, A. Oh, B. Schoelkopf, and M. Gomez-Rodriguez. Leveraging the crowd to detect and reduce the spread of fake news and misinformation. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, 2018.

[58] Jooyeon Kim, Behzad Tabibian, Alice Oh, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Leveraging the crowd to detect and reduce the spread of fake news and misinformation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 324–332. ACM, 2018.

[59] Jon A Krosnick and Duane F Alwin. A test of the form-resistant correlation hypothesis: Ratings, rankings, and the measurement of values. *Public Opinion Quarterly*, 52(4):526–538, 1988.

[60] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[61] Srijan Kumar and Neil Shah. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*, 2018.

[62] N. Lawrence and C. Cortes. The NIPS Experiment. http://inverseprobability.com/2014/12/16/the-nips-experiment, 2014. [Online; accessed 3-June-2017].

[63] S. Leitner. *So lernt man lernen*. Herder, 1974.

[64] K. Lerman and T. Hogg. Leveraging position bias to improve peer recommendation. *PloS one*, 9(6):e98914, 2014.

[65] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.

[66] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[67] P. A. Lewis and G. S. Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413, 1979.

[68] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han. On the discovery of evolving truth. In *KDD*, 2015.

[69] Robert V Lindsey, Jeffery D Shroyer, Harold Pashler, and Michael C Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014.

[70] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[71] X. Liu, X. L. Dong, B. C. Ooi, and D. Srivastava. Online data fusion. *VLDB*, 2011.

[72] G. R. Loftus. Evaluating forgetting curves. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 11(2), 1985.

[73] R Duncan Luce. The choice axiom after twenty years. *Journal of mathematical psychology*, 15(3):215–233, 1977.

[74] Josh Margolin. White supremacists encouraging their members to spread coronavirus to cops, Jews, FBI says. https://abcnews.go.com/US/white-supremacists-encouraging-members-spread-coronavirus-cops-jews/story?id=69737522, 2020. [Online; accessed 8-July-2022].

[75] Arthur W Melton. The situation with respect to the spacing of repetitions and memory. *Journal of Verbal Learning and Verbal Behavior*, 9(5):596–606, 1970.

[76] M. Merrifield and D. Saari. Telescope time without tears: a distributed approach to peer review. *Astronomy & Geophysics*, 50(4):4.16–4.20, 2009.

[77] J. Mervis. Want a grant? First review someone else's proposal. *Sciencemag News*, July 2014.

[78] Everett Mettler, Christine M Massey, and Philip J Kellman. A comparison of adaptive and fixed schedules of practice. *Journal of Experimental Psychology: General*, 145(7):897, 2016.

[79] Claudia Metzler-Baddeley and Roland J Baddeley. Does adaptive training work? *Applied Cognitive Psychology*, 23(2):254–266, 2009.

[80] Hussein Mouzannar, Mesrob I Ohannessian, and Nathan Srebro. From fair decision making to social equality. In *FAT**, 2019.

[81] Timothy P Novikoff, Jon M Kleinberg, and Steven H Strogatz. Education of a model student. *Proceedings of the National Academy of Sciences*, 109(6):1868–1873, 2012.

[82] R. Olfati-Saber, J. A. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[83] A. Pal, V. Rastogi, A. Machanavajjhala, and P. Bohannon. Information integration over time in unreliable and uncertain environments. In *WWW*, 2012.

[84] Harold Pashler, Nicholas Cepeda, Robert V Lindsey, Ed Vul, and Michael C Mozer. Predicting the optimal spacing of study: A multiscale context model of memory. In *Advances in Neural Information Processing Systems*, pages 1321–1329, 2009.

[85] J. Pasternack and D. Roth. Latent credibility analysis. In *WWW*, 2013.

[86] Philip I Pavlik and John R Anderson. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, 14(2):101, 2008.

[87] Kaiping Peng, Richard E Nisbett, and Nancy YC Wong. Validity problems comparing values across cultures and possible solutions. *Psychological methods*, 2(4):329, 1997.

[88] Robin L Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

[89] E. Price. The NIPS experiment. http://blog.mrtz.org/2014/12/15/the-nips-experiment.html, 2014. [Online; accessed 3-June-2017].

[90] Azzurra Ragone, Katsiaryna Mirylenka, Fabio Casati, and Maurizio Marchese. On peer review in computer science: Analysis of its effectiveness and suggestions for improvement. *Scientometrics*, 97(2):317–356, 2013.

[91] William L Rankin and Joel W Grube. A comparison of ranking and rating procedures for value system measurement. *European Journal of Social Psychology*, 10(3):233–246, 1980.

[92] S. Reddy, I. Labutov, S. Banerjee, and T. Joachims. Unbounded human learning: Optimal scheduling for spaced repetition. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, 2016.

[93] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *LREC*, 2010.

[94] Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304, 1977.

[95] Henry L Roediger III and Jeffrey D Karpicke. Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological science*, 17(3):249–255, 2006.

[96] Kevin Roose. Facebook Thwarted Chaos on Election Day. It's Hardly Clear That Will Last. https://www.nytimes.com/2018/11/07/business/facebook-midterms-misinformation.html, 2018. [Online; accessed 8-July-2022].

[97] Philip A Russell and Colin D Gray. Ranking or rating? some data and their implications for the measurement of evaluative response. *British journal of Psychology*, 85(1):79–92, 1994.

[98] B. Settles and B. Meeder. A trainable spaced repetition model for language learning. In *ACL*, 2016.

[99] N. B Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *Journal of Machine Learning Research*, 17:1–47, 2016.

[100] N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *IEEE Transactions on Information Theory*, 2016.

[101] Ashudeep Singh and Thorsten Joachims. Equality of opportunity in rankings. In *Workshop on Prioritizing Online Content, Neural Information Processing Systems*, 2017.

[102] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2018.

[103] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. *arXiv preprint arXiv:1902.04056*, 2019.

[104] A. Somerville. A Bayesian analysis of peer reviewing. *Significance*, 13(1):32–37, 2016.

[105] N. Stewart, G. Brown, and N. Chater. Absolute identification by relative judgment. *Psychological review*, 112(4):881, 2005.

[106] Elizabeth A Stuart. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

[107] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[108] Emanuel Todorov. Efficient computation of optimal actions. *Proceedings of the national academy of sciences*, 106(28):11478–11483, 2009.

[109] A. Janet Tomiyama. Getting Involved in the Peer Review Process. Psychological Science Agenda, American Psychological Association. http://www.apa.org/science/about/psa/2007/06/student-council.aspx, 2007. [Online; accessed 27-January-2018].

[110] Rachel Toor. Reading Like a Graduate Student. The Chronicle of Higher Education. https://www.chronicle.com/article/Reading-Like-a-Graduate/47922, 2009. [Online; accessed 27-January-2018].

[111] Truyen Tran, Dinh Phung, and Svetha Venkatesh. Choice by elimination via deep neural networks. *arXiv preprint arXiv:1602.05285*, 2016.

[112] Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. Fake news detection in social networks via crowd signals. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 517–524. International World Wide Web Conferences Steering Committee, 2018.

[113] U. Upadhyay, I. Valera, and M. Gomez-Rodriguez. Uncovering the dynamics of crowdlearning and the value of knowledge. In *WSDM*, 2017.

[114] Siva Vaidhyanathan. Facebook wins, democracy loses. *New York Times*, 2017.

[115] I. Valera and M. Gomez-Rodriguez. Modeling adoption and usage of competing products. In *ICDM*, 2015.

[116] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 2018.

[117] S. Wang, D. Wang, L. Su, L. Kaplan, and T. F. Abdelzaher. Towards cyber-physical systems in social spaces: The data reliability challenge. In *RTSS*, 2014.

[118] Y. Wang, G. Williams, E. Theodorou, and L. Song. Variational policy for guiding point processes. In *ICML*, 2017.

[119] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[120] John T Wixted and Shana K Carpenter. The wickelgren power law and the ebbinghaus savings function. *Psychological Science*, 18(2):133–134, 2007.

[121] Julia Carrie Wong. Former facebook executive: social media is ripping society apart. *The Guardian*, 2017.

[122] M. Wu and A. Marian. Corroborating answers from multiple web sources. In *WebDB*, 2007.

[123] H. Xiao, J. Gao, Q. Li, F. Ma, L. Su, Y Feng., and A. Zhang. Towards confidence in the truth: A bootstrapping based truth discovery approach. In *KDD*, 2016.

[124] X. Yin and W. Tan. Semi-supervised truth discovery. In *WWW*, 2011.

[125] A. Zarezade, A. De, H. Rabiee, and M. Gomez-Rodriguez. Cheshire: An online algorithm for activity maximization in social networks. In *Proceedings of the 55th Annual Allerton Conference on Communication, Control, and Computing*, 2017.

[126] A. Zarezade, U. Upadhyay, H. Rabiee, and M. Gomez-Rodriguez. Redqueen: An online algorithm for smart broadcasting in social networks. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, 2017.

[127] Ali Zarezade, Abir De, Utkarsh Upadhyay, Hamid Rabiee, and Manuel Gomez-Rodriguez. Steering social activity: A stochastic optimal control point of view. *JMLR*, 2018.

[128] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa* ir: A fair top-k ranking algorithm. In *CIKM*, 2017.

[129] B. Zhao and J. Han. A probabilistic model for estimating real-valued truth from conflicting sources. *Proceedings of QDB*, 2012.

[130] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han. A bayesian approach to discovering truth from conflicting sources for data integration. *VLDB*, 2012.

[131] Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. In *Advances in Neural Information Processing Systems*, pages 262–270, 2011.

[132] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 167–176, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee.

[133] K. Zhou, H. Zha, and L. Song. Learning triggering kernels for multi-dimensional hawkes processes. In *ICML*, 2013.

[134] Brian D. Ziebart, J. Andrew Bagnell, and Anind K. Dey. Modeling interaction via the principle of maximum causal entropy. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.