Karlsruhe Institute of Technology

The Democratization of News

Analysis and Behavior Modeling of Users in the Context of Online News Consumption

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Jan Ludwig Reubold

Tag der mündlichen Prüfung: 15.12.2022

1. Referent: Prof. Dr. Thorsten Strufe
2. Referent: Prof. Dr. Gianluca Stringhini

ABSTRACT

The invention of the *Web* paved the way for the democratization of information. News becoming more accessible to the general public held important political promises. For example, reaching previously inactive citizens to participate in politics as one of them. While, a decade into the development, many politicians and journalists were content with the development [88], the rise of online social networks (OSNs) changed the situation drastically. Today, OSNs have a nearly ubiquitous reach, with 67% of Americans getting at least parts of their news on social media [127].

The trend to get news on OSNs further decreased the costs of publishing. While, at first, this seemed like a good development, it posed a severe issue for democracy. Instead of unlimited information making us smarter, the unbounded content became a burden. Today, over half of the OSN users do not trust the news they read (54% are concerned about what is real and fake news [127]). Recent studies underline that these users are more exposed to populism, propagated by political actors from the extreme ends of the political spectrum, than individuals without social media [60]. A balanced news selection has to give way to a choice of posts and topics reinforced by the user's chosen neighborhood in this sheer mass of information. This fosters political polarisation and ideological segregation [7, 20, 39].

To help mitigate the negative side-effect of this development, my work contributes to the understanding of the problem, counter-acting a known issue, and basic research in the field of behavior modeling. To better understand news consumption in OSNs, we analyzed the behavior of German-speaking users on Twitter [141], comparing reactions towards controversial and non-controversial content [152]. We also studied the existence and extent of echo chambers and similar phenomena. Regarding user behavior, we targeted systems that allow for more complex behavior. Mining such data demands richer models and, typically, lacks further information (such as ground-truth data). Therefore, Bayesian models are commonly preferred over neural networks. We proposed non-parametric Bayesian behavior modeling solutions for clustering [148] and segmenting [149] time-series data. Besides the contributions to the understanding of the problem (analysis and basic research), we developed solutions for detecting automated accounts [150, 151]. These bots take an important role in the early phase of spreading false news [167]. We proposed an expert model based on current deep learning solutions to identify automated accounts by their behavior [151]. This solution introduced a trade-off between the utility of an automated account and the risk of detection.

This work offers insights, solutions and tools for better use of social networks in the future and the realization of the democratization of information.

ZUSAMMENFASSUNG

Die Erfindung des Internets ebnete den Weg für die Demokratisierung von Information. Die Tatsache, dass Nachrichten für die breite Öffentlichkeit zugänglicher wurden, barg wichtige politische Versprechen, wie zum Beispiel das Erreichen von zuvor uninformierten und daher oft inaktiven Bürgern. Diese konnten sich nun dank des Internets tagesaktuell über das politische Geschehen informieren und selbst politisch engagieren. Während viele Politiker und Journalisten ein Jahrzehnt lang mit dieser Entwicklung zufrieden waren, änderte sich die Situation mit dem Aufkommen der sozialen Online-Netzwerke (OSN). Diese OSNs sind heute nahezu allgegenwärtig — so beziehen inzwischen 67% der Amerikaner zumindest einen Teil ihrer Nachrichten über die sozialen Medien [127]. Dieser Trend hat die Kosten für die Veröffentlichung von Inhalten weiter gesenkt. Dies sah zunächst nach einer positiven Entwicklung aus, stellt inzwischen jedoch ein ernsthaftes Problem für Demokratien dar. Anstatt dass eine schier unendliche Menge an leicht zugänglichen Informationen uns klüger machen, wird die Menge an Inhalten zu einer Belastung. Eine ausgewogene Nachrichtenauswahl muss einer Flut an Beiträgen und Themen weichen, die durch das digitale soziale Umfeld des Nutzers gefiltert werden. Dies fördert die politische Polarisierung und ideologische Segregation [7, 20, 39]. Mehr als die Hälfte der OSN-Nutzer trauen zudem den Nachrichten, die sie lesen, nicht mehr (54% machen sich Sorgen wegen Falschnachrichten [127]). In dieses Bild passt, dass Studien berichten, dass Nutzer von OSNs dem Populismus extrem linker und rechter politischer Akteure stärker ausgesetzt sind, als Personen ohne Zugang zu sozialen Medien [60].

Um die negativen Effekt dieser Entwicklung abzumildern, trägt meine Arbeit zum einen zum Verständnis des Problems bei und befasst sich mit Grundlagenforschung im Bereich der Verhaltensmodellierung. Abschließend beschäftigen wir uns mit der Gefahr der Beeinflussung der Internetnutzer durch soziale Bots und präsentieren eine auf Verhaltensmodellierung basierende Lösung.

Zum besseren Verständnis des Nachrichtenkonsums deutschsprachiger Nutzer in OSNs, haben wir deren Verhalten auf Twitter analysiert und die Reaktionen auf kontroverse — teils verfassungsfeindliche – und nicht kontroverse Inhalte verglichen [141, 152]. Zusätzlich untersuchten wir die Existenz von Echokammern und ähnlichen Phänomenen. Hinsichtlich des Nutzerverhaltens haben wir uns auf Netzwerke konzentriert, die ein komplexeres Nutzerverhalten zulassen. Wir entwickelten probabilistische Verhaltensmodellierungslösungen für das Clustering und die Segmentierung von Zeitserien [148, 149, 150]. Neben den Beiträgen zum Verständnis des Problems haben wir Lösungen zur Erkennung automatisierter Konten entwickelt [150, 151]. Diese Bots nehmen eine wichtige Rolle in der frühen Phase der Verbreitung von Fake News ein [167]. Unser Expertenmodell – basierend auf aktuellen Deep-Learning-Lösungen – identifiziert, z. B., automatisierte Accounts anhand ihres Verhaltens.

Meine Arbeit sensibilisiert für diese negative Entwicklung und befasst sich mit der Grundlagenforschung im Bereich der Verhaltensmodellierung. Auch wird auf die Gefahr der Beeinflussung durch soziale Bots eingegangen und eine auf Verhaltensmodellierung basierende Lösung präsentiert.

ACKNOWLEDGMENTS

My path was probably a bit bumpier than most. It was definitely marked by many setbacks. But each time I managed to get back up, improve and move on. That would hardly have been possible without the help and support of a number of people.

First and foremost, I would like to thank my supervisor Thorsten Strufe. The trust he placed in me and the freedoms he gave me in my research were a great help. He taught me a wealth of things about research, project management and putting ideas into practice. He always made time for discussions and answer my many questions. I thank him for his valuable advice and for making this work possible.
In addition, I would like to express my deep gratitude to Prof. Dr. Gianluca Stringhini for agreeing to review my dissertation. It is an honor to have his name on my dissertation.

I also thank all my collaborators for their contributions to the content of this work.
In particular, I thank Clemens Deusser, with whom I shared an apartment for years and with whom I devised a lecture, adapted it, and taught it. For the time I was able to spend with someone who thinks so differently and has had such different life experiences. There will never be anything comparable that will advance me in my development more than our discussions and conflicts.
I would also like to thank Stephan Escher for the collaborations and insightful discussions we have had over the years. It has been a pleasure to walk long stretches of this bumpy road with you again and again.
I thank the Leuphana guys. Ahcène Boubekki with whom I collaborated on papers and with whom I enjoyed the time spent in Vancouver and Sydney. And especially Ulf Brefeld with whom I sat for hours on our submissions. Thanks for your time and expertise.
Furthermore, I thank Christiane Kuhn for her helpful comments on my texts and her guidance through the bureaucratic process.
For their support, I would also like to thank Benjamin Schiller, Simon Hanisch, Dirk Habich, and Johannes Pflugmacher.
Last but not least, my special thanks go to my colleagues in my research group in Dresden and Karlsruhe, for the great time and lively discussions we had during the last years.

Finally, thank you to my family for always being there for me. Thanks to my parents and my sister, who always believed in me and encouraged and enabled me to go my own way. And finally, most of all, a big thank you to my wife. She has stood by me all these years and has always supported me, even putting up with a long-distance relationship that lasted for years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

THE PUBLIC OPINION AND NEWS PROVIDERS

Democracy derives from the Greek words *dēmos* (people) and *kratos* (rule). In contrast to forms of government where a single person or a small group holds power, democracy means *rule of the people*. Its foundation, the *public opinion*, is its greatest strength and most vulnerable part at the same time.

Aristotle once said: "If the citizens of a state are to judge and distribute offices according to merit, then they must know each other's characters; where they do not possess this knowledge, both the election to offices and the decision of law-suits will go wrong". While appropriate for a village where residents know each other and gossip and public opinion are the dominant sources of social control [137], difficulties become apparent when one thinks of large cities or even entire countries. How can *public opinion* develop/exist when we only experience and see a tiny part of the world we live in?

Inevitably, people's opinions must encompass more than what they know and what they experience/observe. The opinions are based on imagination and what others report. Therefore, information from outside our bubble is essential, influencing our actions. We treat the environment based on what we believe to be the true picture:

> There is an island in the ocean, where, in 1914, a few Englishmen, Frenchmen, and Germans lived. No cable reaches that island, and the British mail steamer comes but once in sixty days. In September, it had not yet come, and the islanders were still talking about the latest newspaper, which told about the approaching trial of Madame Caillaux for the shooting of Gaston Calmette. It was, therefore, with more than usual eagerness that the whole colony assembled at the quay on a day in mid-September to hear from the captain what the verdict had been. They learned that for over six weeks now, those of them who were English and those of them who were French had been fighting on behalf of the sanctity of treaties against those of them who were Germans. For six strange weeks, they had acted as if they were friends, when in fact, they were enemies.
>
> But their plight was not so different from that of most of the population of Europe. They had been mistaken for six weeks – on the continent, the interval may have been only six days or six hours. There was an interval. There was a moment when the picture of Europe on which men were conducting their business, as usual, did not in any way correspond to the Europe which was about to make a jumble of their lives. There was a time for each man when he was still adjusted to an environment that no longer existed. All over the world, as late as July 25th, men were making goods that they would not be able to ship, buying goods they would not be able to import, careers were being planned, enterprises contemplated, hopes and expectations entertained, all in the belief that the world as known was the world as it was. Men were writing books describing that world. They trusted the picture in their heads. And then, over four years later, on a Thursday morning, came the news of an armistice, and people gave vent to their unutterable relief that the slaughter

> was over. Yet in the five days before the real armistice came, though the end of the war had been celebrated, several thousand young men died on the battlefields. (Lippmann [108])

It is amazing how indirectly we perceive our environment. We receive information from outside via news providers in text, audio, or visual form. These sources are critical to maintaining some sort of functioning democracy. Thomas Jefferson once said: "I would rather live in a country with newspapers and without a government than in a country with a government and without newspapers".
However, our environment is complex.

> For the real environment is altogether too big, too complex, and too fleeting for direct acquaintance. We are not equipped to deal with so much subtlety, so much variety, so many permutations and combinations. And although we have to act in that environment, we have to reconstruct it on a simpler model before we can manage with it. To traverse the world men must have maps of the world. Their persistent difficulty is to secure maps on which their own need, or someone else's need, has not sketched in the coast of Bohemia. (Lippmann [108])

It is suppressed by censorship or source protection or is difficult to access due to physical and social barriers. On the other end, it is refracted "by scanty attention, [...] by wear and tear, violence, monotony" (Walter Lippmann). Therefore, people construct a pseudo-environment, their version of what they think is the real environment. Walter Lippmann aptly describes the interactions between a person, his pseudo-, and the true environment:

> It is the insertion between man and his environment of a pseudo-environment. To that pseudo-environment, his behavior is a response. But because it is behavior, the consequences, if they are acts, operate not in the pseudo-environment where the behavior is stimulated but in the real environment where action eventuates. If the behavior is not a practical act, but what we call roughly thought and emotion, it may be a long time before there is any noticeable break in the texture of the fictitious world. But when the stimulus of the pseudo-fact results in action on things or other people, contradiction soon develops. Then comes the sensation of butting one's head against a stone wall, of learning by experience, and witnessing Herbert Spencer's tragedy of the murder of a Beautiful Theory by a Gang of Brutal Facts, the discomfort in short of a maladjustment. For certainly, at the level of social life, what is called the adjustment of man to his environment takes place through the medium of fictions. (Lippmann [108])

All of this tells us that information from beyond people's reach and how we assimilate it are an essential part of the foundation of democracy. Information distribution saw three main developments, the inventions of newspapers, the Web, and online social networks. While newspapers opened a constant and reliable window to the outside world, the Web promised the *democratization of news* and online social networks (OSNs) brought us a step closer to it. Today, the costs of distributing or consuming information are close to non-existent. Each evolution lowered the threshold of social and physical barriers and significantly increased news coverage and -perspectives. In consequence, it strengthened democracies.
Today, we live in a world of a sheer unlimited amount of information, perspectives, and opinions. In theory, we have technologically reached a near-perfect setting for democratic societies. Yet, while we have seen these massive developments in information distribution, we also observe the increasingly complex task of evaluating them. With each invention, new challenges concerning the *public opinion* arose. People had to adapt to the new tools. It took a while for the newspaper form, the *free press*, an essential part of modern democracy, to become established. Similarly, the Internet changed the way politics was conducted.

> But if the successes of Internet politics are increasingly obvious, they have also tempted us to draw the wrong conclusions. If we want to understand the fate of politics in the Internet age, we also need to acknowledge new and different types of exclusivity that shape online politics. In a host of areas, from political news to blogging to issue advocacy, [] online speech follows winner-take-all patterns. Paradoxically, the extreme "openness" of the Internet has fueled the creation of new political elites. The Internet's successes at democratizing politics are real. Yet the medium's failures in this regard are less acknowledged and ultimately just as profound. (Hindman [88])

The advent of the social web era promised a path to salvation. Unfortunately, while we saw the potential of online social platforms in times of political turmoil around the world, similar problems as before surfaced in everyday life.

> We are not seeing the end of hierarchy. We may be seeing the replacement of one hierarchy with another hierarchy. We may be seeing the replacement of one set of gatekeepers with another set of gatekeepers... But, we're certainly not seeing an egalitarian world where everything has the same chance to become known or accessible. (Duncan Watts, principal researcher at Microsoft Research)

Today, not only influencers or those with large audiences but also platforms have become gatekeepers. It is almost prohibitively expensive not to participate in the platforms. Meanwhile, these platforms have perfect insight into the shared content. And they have control over the dissemination of content (e.g., soft/hard censorship through limited distribution or deletion)[1].

We are currently on the edge of what our technological achievements promise. The potential is open to us, but we cannot use it properly yet. So, what goes wrong? How can we adjust? First, let us consider what news is and what it is not. News and truth cannot be the same. To report *the truth* facts must be known and understood. News, on the other hand, reports about events. Often long before facts are understood or even known. Therefore, *the truth* and news should be understood as two different parts of our arsenal of information sources.

> The hypothesis, which seems to me, the most fertile, is that news and truth are not the same thing and must be clearly distinguished! The function of news is to signalize an event, the function of truth is to bring to light the hidden facts, to set them into relation with each other, and make a picture of reality on which men can act. Only at those points, where social conditions take recognizable and measurable shape, do the body of truth and the body of news coincide. That is a comparatively small part of the whole field of human interest. In this sector, and only in this sector, the tests of the news are sufficiently exact to make the charges of perversion or suppression more than a partisan judgment. There is no defense, no extenuation, no excuse whatever, for stating six times that Lenin is dead when the only information the paper possesses is a report that he is dead from a source repeatedly shown to be unreliable. The news, in that instance, is not "Lenin Dead" but "Helsingfors Says Lenin is Dead." And a newspaper can be asked to take the responsibility of not making Lenin more dead than the source of the news is reliable; if there is one subject on which editors are most responsible, it is in their judgment of the reliability of the source. But when it comes to dealing, for example, with stories of what the Russian people want, no such test exists. (Lippmann [108])

What was true of newspapers in 1922 is true today in the face of information overload:

[1] https://www.theverge.com/2020/...

> [...] But this development will depend on how well we learn to use knowledge of the way opinions are put together to watch over our own opinions when they are being put together.
> [...] Therefore, unless there is in the community at large a growing conviction that prejudice and intuition are not enough, the working out of realistic opinion, which takes time, [] labor, conscious effort, patience, and equanimity, will not find enough support. That conviction grows as self-criticism increases and makes us [] contemptuous of ourselves when we employ it and on guard to detect it. Without an ingrained habit of analyzing opinion when we read, talk, and decide, most of us would hardly suspect the need for better ideas, nor be interested in them when they appear, nor be able to prevent the new technic of political intelligence from being manipulated. (Lippmann [108])

A Tiny Contribution to a Staggeringly Huge Problem We need to make the transition to better use of technology. Our underlying information system must become more robust and open. To this end, many different areas need to be involved (from education to legislation to more accountability). In this paper, however, we focus on just one area of this massive effort: the consumer.
While the news landscape has grown larger and more complex with each development, it is increasingly the responsibility of the individual to deal with the information on offer. Today, spam-, malware-, phishing-, or politically motivated campaigns are part of the social web experience [62, 184]. In 2008, for example, most users (83%) received at least one unsolicited message. Now, on average, 56% of users are concerned about false news online (from 82% in Brazil to 37% in Germany) [90].

But what are the main factors of this trend? Some actors on online social network platforms spread misinformation, conspiracy theories, and propaganda with agendas ranging from commercializing click-bait to political influence and establishing opinion platforms as hidden distribution channels for marketing all kinds of products [196]. Recent publications underline that social media users are more exposed to populism promoted by political actors from the extreme ends of the political spectrum than those without social media [60]. In this sheer mass of information, a balanced news selection must give way to a selection of posts and topics that is reinforced by the user's chosen neighborhood. This fosters political polarisation and ideological segregation [7, 20, 39]. Incidental news consumption amplifies such effects [16] and leads to a reduction in political education [3]. People who place more trust in information shared by friends are likely to switch to consuming news from a narrow context [16]. This development increases the difficulty of assessing the credibility of information sources [189]. It consequently makes the emerging closed user groups more vulnerable to profit-oriented marketing, political campaigning, and general misinformation. Literature has termed such user groups *echo chambers*. A phenomenon that amplifies and reinforces common opinions within groups through repetition and mutual approval. Typically claimed to exist in social networks, they increase political polarization and ideological segregation [7]. Members of these *echo chambers* are more exposed to populism, propagated by actors from the extreme ends of the political spectrum [60]. They hence have a tremendous impact on the process of political opinion forming.

Technology advances brought further challenges. For example, approaches to automate the distribution of content via bots. Bots are, i.a., used to spread political propaganda, manipulate discussions, or influence the popularity of users/content [167, 65, 171]. Most severe, malicious bots influence political opinion-forming and change the general flow of political discussions.
While false news today is predominantly created by human authors [177], natural language processing (NLP) is catching up. OpenAI called off the release of their GPT-2, a new language processing model architecture based on transformers, in 2019. A decision based on the sur-

prisingly high quality of the generated text and, therefore, the fear of malicious application[2]. Later that year, GROVER was released [197]. GROVER is based on the architecture of GPT-2 and outputs automatically generated text that has proven to be more trustworthy than human-written false news: Zellers et al. [197] reported increased trustworthiness scores of propaganda generated by GROVER instead of humans. These results suggest that generating trustworthy propaganda automatically at scale is coming into reach.
While humans and bots, both, share more often false- than factual news (cf. Vosoughi et al. [179]), bots notably accelerate the distribution of false news.

As more and more responsibility for evaluating news falls on humans, technology needs to improve to support this process.

> Whether the information ecosystem will tip toward more gatekeepers or grassroots in terms of agenda-setting is a largely abstract question — perhaps the networked media world is now too big and diverse to expect a single answer, and it will vary from issue to issue, story to story. What is increasingly clear, however, is that a hybrid system has begun to evolve, one that can be at times just as unequal in terms of voice, power and attention, despite all of technology's promises. (John Wibbey)

Therefore, research needs to understand current shortcomings, provide mechanisms to improve known problems, and develop new approaches to support news democratization. We contribute to the understanding of the current state of news consumption and propose an approach to support news assessment.

[2] openai.com/blog/better-language-models/

CHAPTER II

BACKGROUND

Our work focuses on user behavior. It is divided into three parts: (i) the *behavior analysis* of German-speaking users on Twitter, (ii) *behavioral modeling* using complex nonparametric Bayesian methods, and (iii) the detection of automated accounts in OSNs leveraging *user behavior*.
In the following, we, therefore look at the basics of behavioral modeling before going deeper into the fundamentals of the methods used in this thesis. Therefore, the complexity of the presented topics will increase successively.

Humans follow patterns to achieve their goals. When in the past, a sequence of actions achieved what we intended, we will likely fall back to the same *learned* patterns when pursuing a similar goal in the future. The same holds for users on the Internet. People do not traverse the web at random but follow loose patterns while pursuing their goals. The patterns users follow can be complex, and users usually do not declare their intent. So, an observer sees what they do but does not know what they intend.

Human behavior is often complex and, thus, behavior modeling is a difficult task. Many of these modeling problems cannot be solved by hand. We need automated approaches to identify behavior patterns.
Typically, machine learning is used for this purpose. Here, some function f is approximated using data x, $f(x)$. In some cases, the algorithm is also provided with the desired outcome of the function f given some input data x. Then the approximation task becomes more well-defined, $f(x) = y$.
In general, the procedure is as follows:

Data. First, a representative data set of the behavior to be modeled is created. Based on this data, an automated procedure should now identify frequently occurring patterns. An important step is feature selection, i.e. the decision with which information the behavior is described. In various areas, feature engineering is also required. Here, new, more complex information is extracted from the existing information in a preprocessing step. While feature engineering is not needed for neural networks, it plays a more important role in other methods. In neural networks feature engineering is part of the optimization process.

Model Type. Depending on the problem and the nature of the data, the next step is to choose the appropriate machine learning approach. The kind of recognizable patterns depends on this decision. For example, if one wants to determine a target value based on data, $f(x) = y$, one needs labeled data, i.e., for each sample x in one's data, the respective target value y must be included. In this case, we speak of supervised learning. In the case of behavior modeling, however, we normally do not know the desired target value. Here, our goal is to learn more about our data using an automated procedure, $f(x)$. This is called unsupervised learning and is significantly more complex due to the missing information y. For this reason, it is often not possible to solve problems that can be solved with supervised learning using unsupervised learning.

Optimization. The next step is to fit the selected model to the data. This process is called optimization or learning. Here, the parameters of the model are adjusted until the best possible approximation of the data is achieved. That is until the model is adjusted in such a way that it describes the data better than all other known parameter combinations. The resulting model can then be used to study existing patterns in the data to gain new information.

In the following, we will explain the different machine learning methods and their optimization and parameter estimation before turning to probabilistic modeling and community detection.

Notations In our explanations, we use the following notations:
We use Latin letters to denote variables in our equations.

- *Lowercase letters* (e.g., a): represents scalars
- *Bold lowercase letters* (e.g., $\mathbf{x}$): represents vectors
- *Bold uppercase letters* (e.g., $\mathbf{X}$): represents matrices

When referring to model parameters, we use greek letters.
Large parts of our work deal with the approximation of a function f. Either we have labeled data, $f(x) = y$, or we have to work with inputs only, $f(x)$.
We denote approximations by a hat symbol. For example, when estimating a function f we denote it's approximation by $\hat{f}$, similarly, the approximation of the model parameter θ is $\hat{\theta}$.
If the value of an expression e_1 is *proprotional to* some other expression e_2, we write $e_1 \propto e_2$.

In the paper, we are concerned with user behavior. While in practical applications we adapt the representation of behavior to the data, in researching new behavior modeling approaches we work with a unified representation of behavior. In these cases, we represent user behavior using sessions of click traces.
A click-trace s is a sequence of measurable events x_i of a user over time i in a predefined environment Ω (e.g., the Internet or a particular domain, etc.). An event x could be defined by its execution date t and information describing the event. For example, when tracking a person traversing the web, an event can be described by the domain and subdomain of the visited websites. Following the person, we obtain a sequence of events (click-trace representing a user session) $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ summarizing the actions of a user. The training set, the data used for *training* the model, consists of the tracked user sessions, $\mathcal{D} = \{\mathbf{X}_j\}_{j=1\ldots m}$. Note that we will drop subscripts from our notations if clear from context for brevity.

II.1 Recap: Probability Theory

Probability theory is the study of uncertainty. Here, probability describes the likelihood that an event will occur. With a value between 0 (impossibility) and 1 (certainty), the higher the probability, the more likely an event is to occur.

II.1.1 Random Variables

In probability theory, we represent uncertainty by random variables (RVs). Suppose that X is an RV over a set $\mathcal{X}$, it is either continuous or discrete (see Fig. II.1). This set is also called *support* and denoted by S. Then $P(X = x)$ denotes the probability that X takes on the value $x \in S$.
Probabilities are always positive,

$$P(X = x) \geq 0,$$

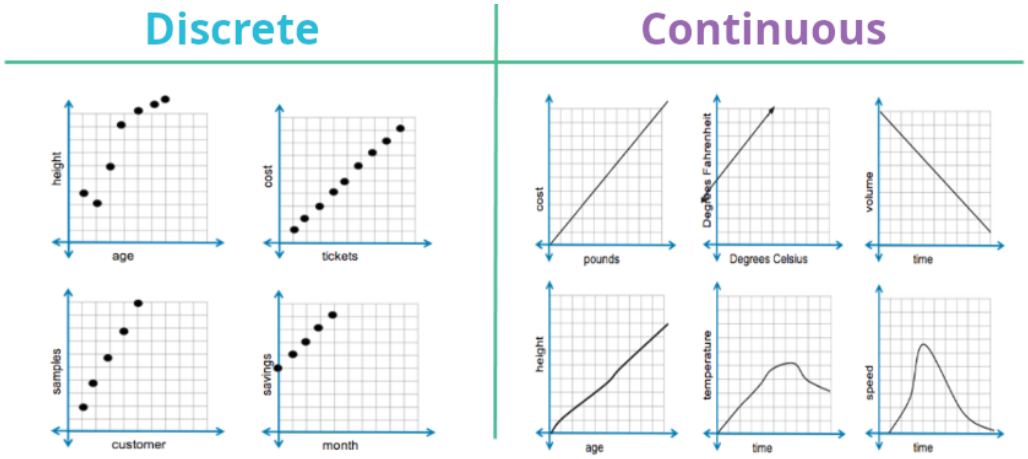


Figure II.1: A probability distribution can be defined over a discrete (left) or a continuous set (right). Image source: [Ashish]

and their values w.r.t. an RV sum to one,

$$\int_{x\in\mathfrak{X}} p(x)\,dx = 1, \text{ if } \mathfrak{X} \text{ is continuous} \qquad \sum_{x\in\mathfrak{X}} p(x) = 1, \text{ if } \mathfrak{X} \text{ is discreet.}$$

An RV is sufficiently described by a *probability distribution*, i.e., a function that provides the probability of any possible value of $x \in X$.

II.1.2 Rules of Probability

Probability theory consists of three different types of probabilities. The *marginal probability* represents the unconditional probability of an event, $P(X = x)$. The *joint probability* models the probability of two or more events occurring together, e.g., $P(X = x, Y = y)$. And the *conditional probability* describes the probability of an event x occurring, given that another event y occurred, $P(X = x|Y = y)$.
If $P(Y = y) > 0$ and $P(X = x|Y = y) = P(X = x)$, we say that X and Y are *independent*,

$$P(X = x \cap Y = y) = P(X = x|Y = y)\,P(Y = y) = P(X = x)\,P(Y = y)\,.$$

Computation-wise, probability theory is based on two rules, the product rule,

$$P(X = x, Y = y) = P(Y = y|X = x)\,P(X = x)\,,$$

and the sum rule,

$$P(X = x) = \sum_{y\in Y} P(X = x, Y = y)\,.$$

In the following, we omit the explicit declaration of RVs, e.g., instead of $P(X = x)$, we use its abbreviation $p(x)$:

Product Rule $p(x, y) = p(y|x)\,p(x)$

Sum Rule $p(x) = \sum_y p(x, y)$

Also note that $p(x, y) = p(y|x)\,p(x) = p(x|y)\,p(y)$.
To deepen the understanding, let us look at an example.

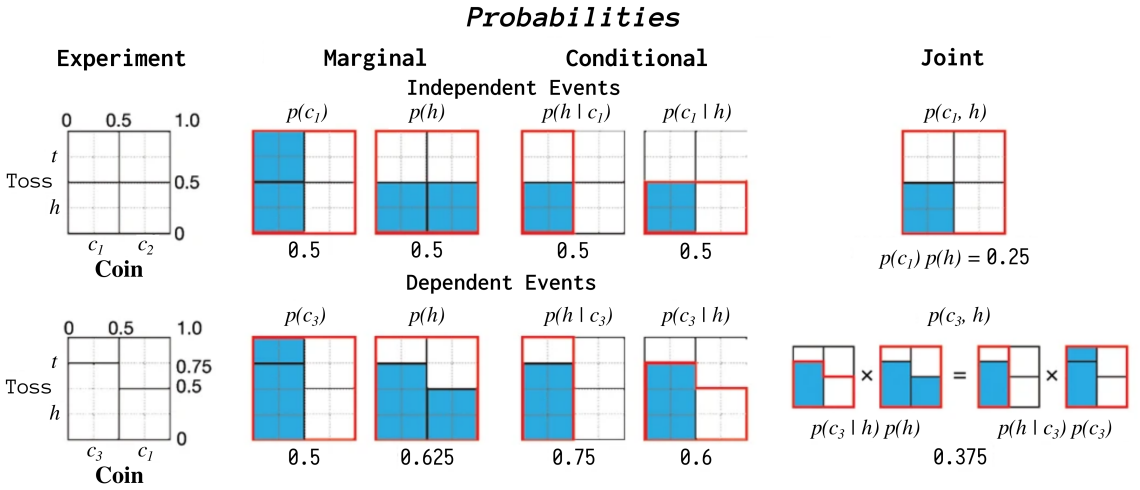


Figure II.2: The different types of probabilities and the difference between dependent and independent events. Image source: Puga et al. [144]

Example: Coin Toss (Probability)

Suppose we have three coins, two fair ones (i.e., *heads* and *tails* are equally likely) and a biased coin.

In our first experiment, we look at independent events using the two fair coins c_1 and c_2 (see Fig. II.2). The experiment consists of choosing a coin uniformly at random $P(C)$ (with $C = \{c_1, c_2\}$) that is flipped, $P(Y)$ with $Y = \{h, t\}$. We observe that the probability of heads $P(Y = h) = p(h) = 0.5$ is independent of the choice of the coin, $P(C, Y) = P(C)\,P(Y)$. We say, C and Y are independent events.

In our second experiment, we choose the fair coin c_1 and the biased coin c_3 (see Fig. II.2). We have $p(h|c_1) = 0.5$ and $p(h|c_3) = 0.75$. Clearly, the probability of turning heads depends on the chosen coin. Under these conditions, we say C and Y are dependent events. According to the product rule, the joint probability $p(c_3, h) = p(h|c_3)\,p(c_3)$ is 0.375. Also note that, while $p(h|c_3)\,p(c_3) = p(c_3|h)\,p(h)$, generally $p(h|c_3)$ and $p(c_3|h)$ are not the same (*prosecutor's fallacy*).

II.2 Machine Learning Methods

Since much of this work deals with machine learning approaches, we also briefly discuss machine learning methods. Machine learning is the discipline of predicting future events or making other decisions under uncertainty based on patterns automatically extracted from data. Methods of this field are divided into three main types of models: *predictive* or *supervised learning*, *descriptive* or *unsupervised learning*, and *reinforcement learning*.

II.2.1 Supervised Learning

Suppose that we are given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with input and output pairs $(\mathbf{x}, y)$. Here, $\mathcal{D}$ refers to the training set with N training examples.

In supervised learning (also called predictive learning), the task is to learn a mapping from input to output. Formalized as a function approximation problem, we try to estimate the function $f(x) = y$. We want the model to make predictions on novel observations (called *generalization*), $\hat{f}(\mathbf{x}) = \hat{y}$. Here, the input $\mathbf{x}$ typically consists of multiple features represented by a vector (e.g., the height and weight of a person), while the output is a scalar (e.g., the sex, the foot length, etc.). When the output is a categorical or nominal variable from some finite set

(e.g., $y \in \{1, \ldots, C\}$), the problem is called *classification* or *pattern recognition*. If the output is a real-valued scalar, it is called *regression*.

II.2.2 Unsupervised Learning

An unsupervised learning problem (also called descriptive learning) has to find descriptive patterns in the data given only inputs, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$. Compared to supervised learning, we miss the guidance on how to optimize the model that we usually get by looking up examples from the training set. So, unsupervised learning is a harder problem to solve. However, not only is collecting data cheaper – labeling data is often expensive and time-consuming – but it also conveys more information:

> When we're learning to see, nobody's telling us what the right answers are – we just look. Every so often, your mother says 'that's a dog', but that's very little information. You'd be lucky if you got a few bits of information – even one bit per second – that way. The brain's visual system has 10^{14} neural connections. And you only live for 10^9 seconds. So it's no use learning one bit per second. You need more like 10^5 bits per second. And there's only one place you can get that much information: from the input itself. (Geoffrey Hinton, 1996)

In summary, the differences between unsupervised learning compared to supervised learning are as follows:

	Supervised Learning	Unsupervised Learning
Objective	$f(\mathbf{x}) = y$	$f(\mathbf{x})$
Target Variable	y	$\mathbf{x}$
Model Types	E.g. classification, regression	E.g. clustering, dim. reduction

The missing guidance and the more complex models (multi-variate instead of univariate probability models) make unsupervised learning more challenging.

II.2.3 Reinforcement Learning

For the sake of completeness, another method, reinforcement learning, should be mentioned, although it is not relevant to this thesis. *Reinforcement learning* (RL) represents a third type of model. In RL tasks we are provided with more guidance for learning (information on the desired outcome) than in unsupervised learning, but with less information than in supervised learning. Suppose we want to train a model to play the game of Go. In this case, the model has to make decisions that provide no or only limited feedback. The quality of decisions depends on the reactions of the opponent and further decisions in the future. In contrast to chess, in *Go* it is computationally impossible to compute all possible moves and game situations. Therefore, we cannot obtain full information on the quality of each decision. We cannot make calculated decisions, we need our model to learn intuitive thinking. RL explores possible actions and reinforces the ones yielding desirable outcomes while discouraging unwanted behavior. The ultimate task (e.g., to win the game) is divided into smaller short-term goals (e.g., cutting the opponent, gaining, or capturing territory). A popular example of RL is training a dog. Before getting the dog, one defines the boundaries of what the dog is allowed to do (e.g., not go onto the sofa). From the day, the dog arrives it is trained, e.g., by discouraging unwanted behavior. When teaching the dog tricks, the environment is chosen so that the dog can reach a predefined goal. Typically, these goals are divided into intermediate stages, because otherwise, the required sequence of actions would be so complex that the dog would in reality never show the desired behavior. As soon as the dog reaches a (partial) goal through trial and error, this is positively reinforced with a reward. With correctly chosen (partial) goals, the dog will regularly show the

desired behavior after several attempts. Now the next behavior, usually based on the previously achieved behavior, and therefore more complex, can be tackled.

Compared to dogs, RL algorithms are usually inefficient learners and require a sheer unlimited amount of training data. Therefore, they can often only be applied to tasks that can be simulated (hence, provide an unlimited amount of data).
Further information on the types of machine learning and related topics can be found in Murphy [124][1].

II.3 Probabilistic Modeling

To model behavior, we use probabilistic modeling. At the beginning of this chapter II, we gave a high-level summary of the modeling process. We choose a model $\mathcal{H}$ that we adjust to data. Choosing a model and, thereby, picking the 'right' complexity is called *model selection*. Choosing how to represent events in the data is called *feature selection and -engineering*. The process of adjusting or *training* the model is called *learning* or *optimization*. Note that the feature and the model selection define what kind of patterns the algorithm can identify. In the following, we discuss these concepts in more detail.

To model behavior, we make assumptions that we encode in an abstract model $\mathcal{H}$. Typically, the model $\mathcal{H}$ depends on some parameters θ. During *training*, these parameters are adjusted to the data $\mathcal{D}$. This process is similar to buying eyeglasses in a store. The customer selects a model from a range of candidates. Then, the seller adjusts the chosen model to measurements of the eye properties of the customer. The final product is fit to the customer as well as possible based on the chosen model, the measurement, and the assessment of the customer.

In probabilistic modeling, we account for the fact that in each step, there is uncertainty. We are uncertain if we chose the correct model $\mathcal{H}$. The customer could have chosen the wrong model, e.g., a model that the seller cannot adjust to the visual impairments of the customer.
We are uncertain if we adjusted the model parameters θ correctly. The seller may have adjusted the glasses incorrectly.
We are uncertain about the input x. The seller may have adjusted the glasses correctly, but based on incorrect measurements.
We are uncertain about the output y. The customer could misjudge the quality of the vision with the help of the glasses.

So, while we try to compute $\hat{f}(\mathbf{x}) = y$ (the vision aid f, the eye property measurements x, the assesment of the customer y) with $\mathcal{H}(\theta)$ to approximate f, we are faced with uncertainties. We treat these uncertainties with the theory of probability.

Graphical Models. Sometimes it is useful to describe a model by a graphical representation. Such a representation describes the probabilistic model by a graph with the RVs and parameters as nodes and the dependencies as edges. We will encounter our first examples shortly (e.g., Fig. II.6).

In what follows, we consider Bayesian modeling and the process of estimating the true parameter values of a model. We introduce the concepts using examples of binary events and then extend them to model multivariate events as well. These basics are followed by a transition from parametric to nonparametric models. In this part, we will delve deeper and describe concepts that are used directly or in modified form in the paper.

[1] https://github.com/probml/pml-book

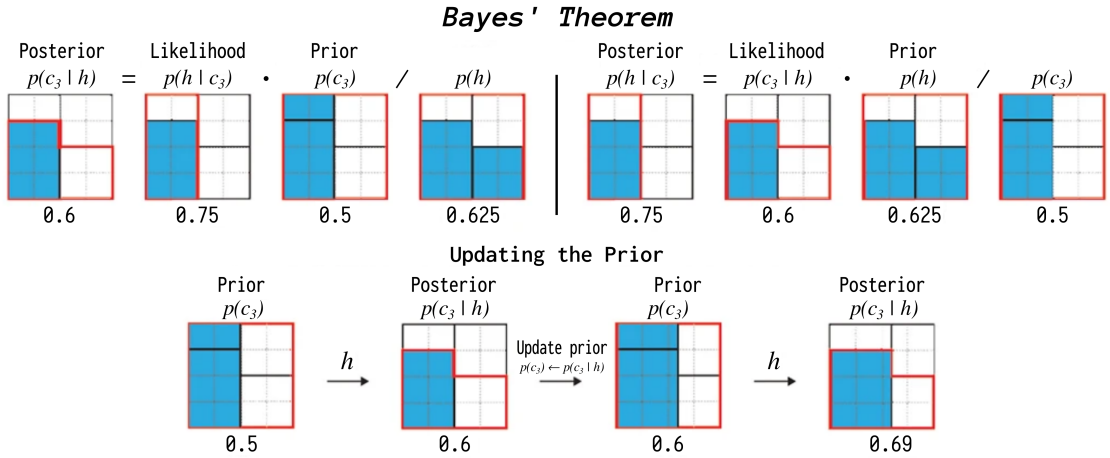


Figure II.3: Illustration of the Bayes' Theorem applied to our coin toss example. Image source: Puga et al. [144]

II.3.1 Bayes Theorem

We begin with the foundation for inference and learning in probabilistic modeling, the *Bayes Theorem*. For the description of the fundamentals of Bayesian modeling, we assume binary events in the following (e.g., a coin toss). We can derive the Bayes Theorem as follows:

$$\begin{aligned} p(\theta|x)\, p(x) &= p(x|\theta)\, p(\theta) \\ \Rightarrow p(\theta|x) &= \frac{p(x|\theta)\, p(\theta)}{p(x)}, \end{aligned} \tag{II.1}$$

with

$$p(x) = \sum_{\theta} p(x, \theta)\,. \tag{II.2}$$

The data are represented by x, the parameters of the model by θ. Then, $p(x|\theta)$ represents the *likelihood*, i.e., probability that the data was generated by the model parameterized by θ. $p(x)$ represents the model of the data, also called *normalization constant*. $p(\theta)$ denotes the *prior knowledge* about the parameter. Finally, our updated belief about the true parameter values after observing data is represented by the *posterior distribution* $p(\theta|x)$.

Example: Coin Toss (Inference)

We illustrate the application of Bayes' theorem using our running example, the coin toss experiment, in Fig. II.3. In our second experiment, we used a fair and a biased coin, c_1 and c_3. Now, we observed *heads* and want to update our belief of which coin was used $p(C)$. Using Bayes theorem, we know $P(C|X) = {}^{P(X|C)P(C)}/_{P(X)}$. Hence, we obtain the posterior probability of c_3 by:

$$p(c_3|h) = \frac{p(h|c_3)\, p(c_3)}{p(h)} \tag{II.3}$$

We know $p(h|c_3) = 0.75$ and $p(c_3) = 0.5$. Using the *sum rule*, we obtain $p(h)$ by summing over all joint probabilities $P(C, Y = h)$,

$$p(h) = [p(h|c_1)\, p(c_1)] + [p(h|c_3)\, p(c_3)] = 0.625. \tag{II.4}$$

Putting *prior*, *likelihood*, and the normalization constant together, we obtain the posterior probability $p(c_3|h) = 0.6$. Thus, because we know that $c_1 + c_3 = 1$, we can derive $p(c_1|h) = 0.4$. When observing new outcomes, our prior belief that both coins are equally likely is replaced with our belief that we chose c_3 60% of the time and c_3 40% of the time.

Equipped with these concepts, we can now introduce the Bayesian approach to probabilistic modeling.

II.3.2 Bayesian Modeling

Let us build our first model to introduce the concepts of probabilistic modeling. Suppose that we are given a coin and flip it 100 times. We observe $N_H = 60$ times heads and $N_T = 40$ times tails. We can simulate the flip of a coin as a draw from the Bernoulli distribution,

$$y \sim \text{Ber}(\theta) . \tag{II.5}$$

θ denotes the relevant properties of the coin, i.e., the probability of turning up heads. To predict future outcomes, we have to understand these properties. Then, we can make predictions on future events $\mathcal{D}^+$ conditioned on the events already observed $\mathcal{D}$.

So, let us start by formulating a probabilistic model using the Bayes theorem. Probabilistic modeling requires us to fit our model $\mathcal{H}$ to the training data $\mathcal{D}$. Therefore, we have to compute the probability $p(\theta|\mathcal{D})$, i.e., the likelihood of different versions of our model conditioned on the data. Using Bayes theorem, we get

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)\, p(\theta) . \tag{II.6}$$

Here, we compute the joint probability of θ and $\mathcal{D}$ using the *product rule*. $\mathcal{D}$ is fixed, so it is a function over θ.

We can interpret Eq. II.6 as follows: Before seeing any data, we have some prior belief about the parameters, $p(\theta)$. Based on this *prior belief*, we compute how *likely* it is that the corresponding model generated the observed data, $p(\mathcal{D}|\theta)$. Our updated belief about θ after observing data is described by the *posterior distribution* $p(\theta|\mathcal{D})$. Therefore, the Bayes theorem gives us a recipe for updating our beliefs when encountering new evidence. The appeal of this approach is that the *posterior* represents our new belief about the parameters. When again adjusting for new data our *prior belief* $p(\theta)$ is replaced with our *posterior distribution*.

According to Eq. II.6, for a fully Bayesian treatment, we have to express our prior knowledge and define a likelihood function. We know that a coin flip can result in a binary output (head or tail). We also know that flips are independent and identically distributed (i.i.d.). A flip does not influence any other coin flips (independent), and each coin flip follows the same distribution (identically distributed).

The *Bernoulli distribution* is a suitable choice for this setup. It is a discrete probability distribution that takes on (in our case) heads with probability θ and tails with probability $1-\theta$. An experiment described by a Bernoulli distribution is called a *Bernoulli trial*. The distribution describing multiple independent Bernoulli trials is called *Binomial distribution*.

II.3.2.1 Likelihood Function

Our likelihood function $L(\theta) = p(\mathcal{D}|\theta)$ is the probability of the observed outcomes as a function of θ. We know that the flips are i.i.d. Bernoulli RVs (see Eq. II.5). As we flip the coin several times, we use the Binomial distribution to describe the likelihood function,

$$L(\theta) \propto \theta^{N_H} (1-\theta)^{N_T} . \tag{II.7}$$

The likelihood is a product of small numbers and tends to underflow on computers. Therefore, one usually works with the log of the likelihood, in this case,

$$\ell(\theta) = \log L(\theta) = N_H \log\theta + N_T \log(1-\theta) . \tag{II.8}$$

Exemplary, an experiment with 100 flips, all of them turning heads $N_H = 100$ and a θ of 0.5 results in $\ell(0.5) = 100 \log 0.5 = -69.31$ instead of $L(0.5) = 0.5^{100} = 7.9 \times 10^{31}$.

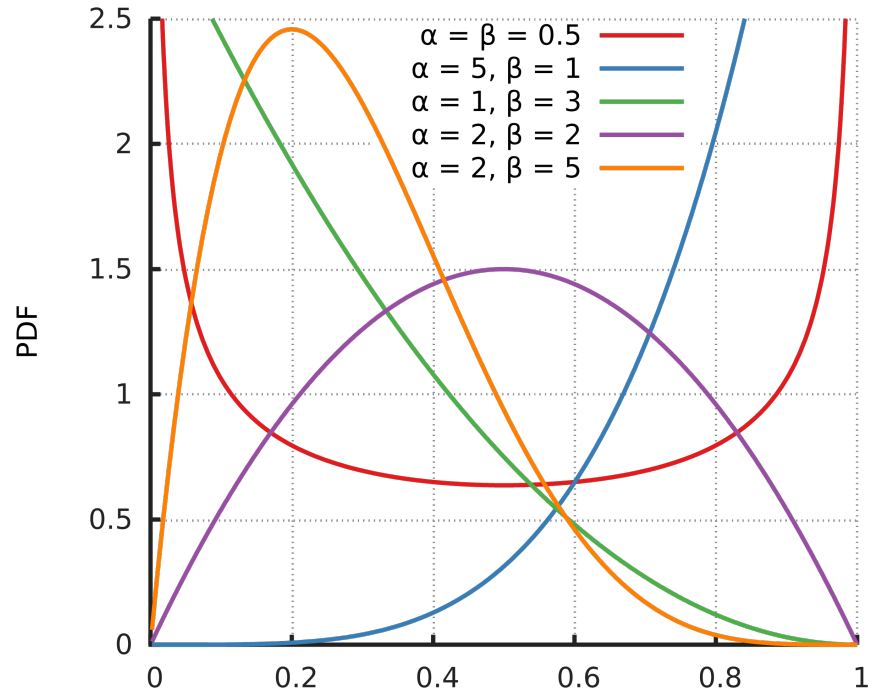


Figure II.4: Probability density function of the beta distribution with different configurations (α and β). Image source: [Wikipedia]

II.3.2.2 Prior Distribution

Before seeing any coin flip, we express our prior knowledge, i.e., our belief in the properties of the coin (regarding turning heads or tails). We can summarize this property by a value between 0 and 1, the probability of turning heads. Therefore, we want to choose a probability distribution over the unit interval. We could use an uninformative prior in form of the uniform distribution, i.e. assigning equal weights to all possible coin properties, $p(\theta) = 1$. We would assign the same probability to a coin that always turns heads (heavily bent) and one where heads or tails are equally likely (fair coin). However, from experience, we know that the latter is far more likely. Therefore, we are pretty confident that θ is around 0.5.

A convenient candidate to express this knowledge is the beta distribution. Therefore, we define our prior belief using the beta distribution,

$$p(\theta; \alpha, \beta) \propto \theta^{\alpha-1} (1-\theta)^{\beta-1} . \tag{II.9}$$

Beta Distribution. This probability distribution is defined by the parameters α and β over the unit interval centered around $\alpha/\alpha+\beta$ (see Fig. II.4). The distribution becomes more peaked with larger values of α and β. A special case of the beta distribution is when $\alpha = \beta = 1$ representing the uniform distribution. Parameters that control distributions and are fixed a priori like α and β are called *hyperparameters*.

We observe that the functional form of this prior distribution has the same form as L (cmp. Eq. II.7 and II.9). Therefore, the posterior distribution will also have the same functional form. This property is especially useful in a Bayesian setting, where we update our prior belief with the posterior distribution. Prior distributions chosen accordingly are called *conjugate priors*.

II.3.2.3 Posterior Distribution

With prior and likelihood functions as

$$\begin{aligned} p(\theta) = \text{Beta}(\theta; \alpha, \beta) &\propto \theta^{\alpha-1}(1-\theta)^{\beta-1} \\ p(\mathcal{D}|\theta) &\propto \theta^{N_H}(1-\theta)^{N_T}, \end{aligned} \tag{II.10}$$

we can write Eq. II.6 as follows:

$$\begin{aligned} p(\theta|\mathcal{D}) &\propto \left(\theta^{\alpha-1}(1-\theta)^{\beta-1}\right)\left(\theta^{N_H}(1-\theta)^{N_T}\right) \\ &= \theta^{\alpha-1+N_H}(1-\theta)^{\beta-1+N_T}. \end{aligned} \tag{II.11}$$

We see that the parameters of our prior distribution (α and β) are added to the count of heads and tails. Hence, they are also called *pseudo-counts*. With more and more observed data, the pseudo-counts get smaller in relation, i.e., the data *overwhelms the prior*. When provided with an infinite amount of data, the Bayesian approach converges to the ML estimate.

II.3.3 Model Estimation

We optimize the model to find the best possible parameterization. However, after training the model we obtain a probability for each of the possible parameter settings. Based on this outcome, we have to decide what information to use to select the best parameterization. There are several methods to do this, 3 of which we will discuss here. The greedy approach is called *maximum likelihood* (ML) estimate and only considers the likelihood function. The other two approaches, the *posterior predictive distribution* and the *maximum a-posterior approximation* assume some prior knowledge incorporated into the estimation process.

II.3.3.1 Maximum Likelihood Estimation

The most basic approach we introduce trusts the data and adjusts θ according to the ratio between heads and tails. Optimizing θ such that the likelihood function is maximized results in the *maximum likelihood* (ML) estimate of the model, $\hat{\theta}_{\text{ML}} = {}^{N_H}/_{N_H+N_T}$. While powerful, the ML estimate can heavily overfit when provided with too little data [28]. Therefore, the other approaches make use of prior knowledge.

II.3.3.2 Posterior Predictive Distribution

Our model consists of only a small amount of parameters. It is simple enough to apply the full Bayesian treatment that incorporates all information of the Bayesian model.
Our goal is to predict future outcomes. Hence, we make predictions on future events $\mathcal{D}^+$ conditioned on the events already observed $\mathcal{D}$. The past observations allow us to assign probabilities to the various settings of θ. While the ML estimate greedily chose the value for θ that maximized the likelihood function, a fully Bayesian treatment takes all information into account. Integrating over θ (marginalizing out θ) utilizes all our knowledge about θ for the prediction of future outcomes,

$$p\left(\mathcal{D}^+|\mathcal{D}\right) = \int_{\theta\in\Theta} p(\theta|\mathcal{D})\, p\left(\mathcal{D}^+|\theta\right) d\theta. \tag{II.12}$$

Finally, we obtain the *posterior predictive distribution* (PPD) over the next flip y as follows:

$$\begin{aligned}\theta_{\text{PPD}} &= \Pr(y = H|\mathcal{D}) \\ &= \int_{\theta\in\Theta} p(\theta|\mathcal{D}) \Pr(y = H|\theta)\, d\theta \\ &= \int_{\theta\in\Theta} \text{Beta}(\theta; N_H + \alpha, N_T + \beta) \cdot \theta d\theta \\ &= \mathbb{E}_{\text{Beta}(\theta;N_H+\alpha,N_T+\beta)}[\theta] \\ &= \frac{H + \alpha}{N_H + N_T + \alpha + \beta}.\end{aligned} \tag{II.13}$$

We observe that our posterior predictive distribution θ_{PPD} is similar to the ML estimate θ_{ML}. It differs the most in settings with only a few observations. In these situations, we do not have enough evidence to fully trust the data. Therefore, our prior belief regulates the posterior distribution from making too extreme predictions based on observed events. With more and more data, our prior belief gets overwhelmed by the data, and we start fully trusting the data and thereby converge to the ML solution, N_H/N_H+N_T.

II.3.3.3 Maximum a-Posterior Approximation

So far, we introduced two closed-form solutions for optimization. However, with many models, closed-form solutions do not exist. For ML estimates, we fall back to optimizing the likelihood using gradient ascent. The Bayesian approach, however, relies on integral computations. For large models, this is computationally impractical. We have to turn to approximation techniques for the fully Bayesian approach (e.g., Markov chain Monte Carlo sampling) or relax our problem and turn it into an optimization problem.

We observed that with more data, our Bayesian estimate became more peaked around a single value. Hence, we could approximate the posterior similar to the ML estimation. We can optimize θ to maximize the posterior. Instead of a distribution over all possible settings of θ, we obtain a point-estimate θ_{MAP}. The difference to the ML estimate is that it incorporates our prior beliefs. This approximation of the posterior is called the *maximum a-posterior* (MAP) approximation,

$$\hat{\theta}_{\text{MAP}} = \arg\max_\theta p(\theta|\mathcal{D}) \propto \arg\max_\theta p(\mathcal{D}|\theta)\, p(\theta) = \arg\max_\theta \log p(\mathcal{D}|\theta) + \log p(\theta). \tag{II.14}$$

According to Eq. II.11, this is

$$\log p(\theta|\mathcal{D}) \propto (N_H + \alpha - 1)\log\alpha + (N_T + \beta - 1)\log(1 - \theta). \tag{II.15}$$

Maximizing it, we end up with

$$\hat{\theta}_{\text{MAP}} = \frac{N_H + \alpha - 1}{N_H + N_T + \alpha + \beta - 2}. \tag{II.16}$$

Example: Model Estimation

To close out the section, we showcase the behavior of the different approaches using two different settings of our coin toss example. We use the beta distribution to express our prior belief of a fair coin and use the binomial distribution to compute the likelihood of an event conditioned on our knowledge. Setting the parameters of our prior distribution to $\alpha = \beta = 5$ expresses our strong belief in a fair coin. We already tossed the coin 100 times and it turned heads $N_H = 60$ times and tails $N_T = 40$ times. Suppose that in a second experiment we observed $N_H = 5$ heads and $N_T = 0$ tails. Applying the different approaches we obtain the

following parameter values for θ:

Method	5-Toss	100-Toss
$\hat{\theta}_{\text{ML}}$	1.00	0.60
θ_{PPD}	0.67	0.59
$\hat{\theta}_{\text{MAP}}$	0.69	0.59

Note that θ_{PPD} is the only exact computation of θ while the others are approximations.

II.3.4 From Binary to Multivariate Events

We started with a model to describe binary events. For this thesis, we have to extend the example to multivariate outputs. Therefore, to introduce further distributions to cope with such output, assume that we exchange the coin for a 6-sided die. Instead of repeatedly flipping a coin, we will roll a die, simulated with the *categorical distribution*, $y \sim \text{Cat}(\theta)$. We only have to change the distributions representing the likelihood and the prior distribution of the Bayesian model. Otherwise, even the i.i.d. assumptions still hold.

Likelihood function. The *multinomial distribution* is a generalization of the binomial distribution to multi-dimensional events and thus the natural pick as our likelihood function,

$$L(\theta) \propto \prod_{k=1}^{K} p_k^{x_k} \tag{II.17}$$

It describes the probabilities of the occurrence of each side of the die, in our case represented by a $6D$ vector with each probability between 0 and 1, and its sum as 1, so with support over $S_6 = \left\{x : 0 \leq x_k \leq 1, \sum_{k=1}^{6} x_k = 1\right\}$.

Prior Distribution. Our prior belief has to represent S_6, all possible 6-sided dice. A single die can be represented by a categorical distribution over its sides, a fair die, e.g., as $d_f = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6]$. Its outcome is then a number between 1 and 6. The prior distribution has to support the probability distribution of all possible configurations of a die. So, the outcome has to be a categorical distribution. Therefore, such a distribution is called a *distribution over distributions*. Note that we call the resulting distributions (e.g., the representation of a die) probability measures.

Similar to the likelihood function, we need the multivariate analog to the beta distribution, the *Dirichlet distribution*.

The Dirichlet distribution is called the multivariate beta distribution and is controlled by a parameter-vector $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$. It has support over the K-dimensional probability simplex (over a K-dimensional probability distribution), $S_K = \left\{x : 0 \leq x_k \leq 1, \sum_{k=1}^{K} x_k = 1\right\}$. Its pdf is proportional to

$$\text{Dir}(\mathbf{x}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} x_k^{\alpha_k - 1}, \tag{II.18}$$

with $\alpha_k > 0$. We further define the magnitude of the vector $\alpha_0 = \sum_{k=1}^{K} \alpha_k$ and the normalized α_k by $\tilde{\alpha}_k = {}^{\alpha_k}/_{\alpha_0}$.

The expected value and variance (e.g., of a side of the die) are given by

$$\begin{aligned} \mathbb{E}[X_i] &= \tilde{\alpha}_i \\ \text{Var}[X_i] &= \frac{\tilde{\alpha}_i (1 - \tilde{\alpha}_i)}{\alpha_0 - 1}. \end{aligned} \tag{II.19}$$

We observe that the normalized parameter-vector $\tilde{\alpha} = [\tilde{\alpha}_1, \ldots, \tilde{\alpha}_K]$ is the average measure and that the magnitude of α_0 controls how strongly draws are concentrated around this distribution.

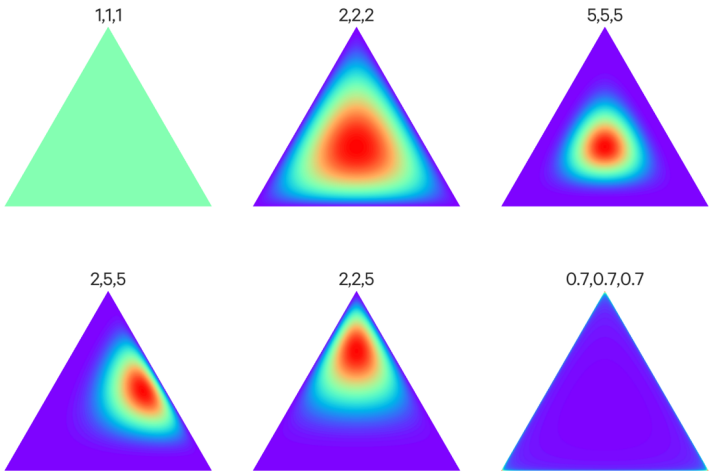


Figure II.5: Effect of the parameter-vector α on the Dirichlet distribution. Image source: [Tseng]

Exemplary, suppose we define a symmetric Dirichlet distribution by assigning all α_k the same value. Analog to the beta distribution, setting all of it to 1 resembles the uniform distribution over the probability simplex. The higher we set the value, the more peaked the distribution becomes, i.e., the draws concentrate more and more around the average distribution. The more we lower the value below 1, the more the distribution (over measures) prefers sparse measures, with most values close to 0 and heavily concentrated in only a few of the values. Figure II.5 depicts the effect of $\boldsymbol{\alpha}$ and its magnitude on the Dirichlet distribution modeling a $3D$ probability simplex in the symmetric and non-symmetric case.

Adjusting the distribution of our prior belief and the likelihood function allows us to apply a Bayesian treatment to the dice role example. We obtain the PPD by integrating out the parameters θ (Dirichlet multinoulli model),

$$\boldsymbol{\theta}_{PPD} = \tilde{\boldsymbol{\alpha}} = \sum_{k=1}^{K} \frac{\alpha_k}{\alpha_0} \delta_k \tag{II.20}$$

The approaches presented in the thesis heavily rely on the multinomial and the Dirichlet distribution.

To summarize, a categorical distribution of, e.g., a fair six-sided die,

$$\text{Cat}\left([1/6,\ 1/6,\ 1/6,\ 1/6,\ 1/6,\ 1/6]\right),$$

yields a scalar, while a Dirichlet distribution in the same configuration,

$$\text{DP}\left([1/6,\ 1/6,\ 1/6,\ 1/6,\ 1/6,\ 1/6]\right),$$

represents a *distribution of distributions* and yields a multivariate outcome. Here, the normalized values $\tilde{\boldsymbol{\alpha}}$ describe the mean distribution and the magnitude of $\boldsymbol{\alpha}$, α_0 describe how strongly the draws are concentrated around $\tilde{\boldsymbol{\alpha}}$. For example, a fair six-sided die is far more likely to be drawn from a DP([100, 100, 100, 100, 100, 100]) than from a DP([1/6, 1/6, 1/6, 1/6, 1/6, 1/6]).

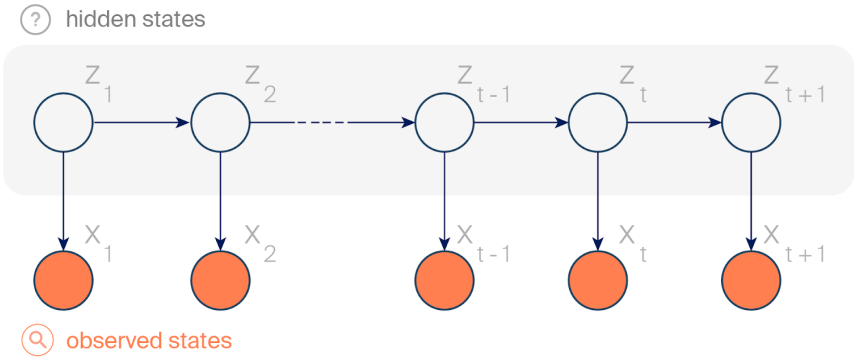


Figure II.6: The graphical model of an HMM is comprised of a layer of hidden variables z and a layer of measurable events x; here $\tau = t + 1$, the end of the sequence. Image source: [Haensch]

II.3.5 Hidden Markov Models

Based on the topics discussed so far, we can describe the probability of events of a completely observable process using probabilistic models. For our purposes, however, concerning user behavior, we assume that the behavior is controlled by unobservable states (in our case, the user's intention). Therefore, we have to introduce an additional concept to cope with such processes.

The *hidden Markov model* is a simple, yet nonetheless powerful approach we can use. It represents a mixture model, a process that consists of two parts, the *mixing proportion* and the *mixture components*.

Suppose we play a gain where, depending on the situation, different dice have to be thrown. Then, the *mixing proportion* describes the probability of switching between the different dice and the *mixture component* describes the properties of each die.

The *hidden Markov model* is a mixture model consisting of two intertwined processes, namely the *transition model* and the *observation model* (see Fig. II.6). The transition model $p\left(z_t|z_{t-1}\right)$ is hidden and represents the user's intent, while the observation model $p\left(x_t|z_t\right)$ describes the corresponding behavior pattern that we can observe. To show in the model which hidden state is active at time t, we introduce the hidden states $z_t \in \{1, \ldots, K\}$. Then, the observable state $x \in \Omega$ depends on the current active hidden state, $p\left(x_t|z_t\right)$. We assume that both state spaces are fixed and known a priori. The model makes strong assumptions about the process describing the transition model. In a fully connected model, the state of a process depends on all of its predecessors, $p\left(z_t|z_1, \ldots, z_{t-1}\right)$. In an HMM, one assumes that the previous state z_{t-1} at time $t-1$ holds all information about the entire history we need to know.

$$p(z_t|z_0, z_1, \ldots, z_{t-1}) \triangleq p(z_t|z_{t-1}). \tag{II.21}$$

Transition probabilities within the hidden process are fixed and do not change over time (*stationary process assumption*). They are denoted by a transition matrix with the transition probability of z_t conditioned on the previous state as $p(z_t|z_{t-1}) = A\left[z_{t-1}, z_t\right]$. The transitions matrix represents the adjustable parameters θ of the model. We compute the joint probability distribution of a click-trace s as

$$p\left(z_{1:\tau}, x_{1:\tau}\right) = p\left(z_1\right) \prod_{t=2}^{\tau} p\left(z_t|z_{t-1}\right) \prod_{t=1}^{\tau} p\left(x_t|z_t\right). \tag{II.22}$$

The *limited horizon* and the *stationary process* assumptions allow us to infer time-series in a stochastic manner.

With an HMM we modeled user behavior as a generative process controlled by the user's intent.

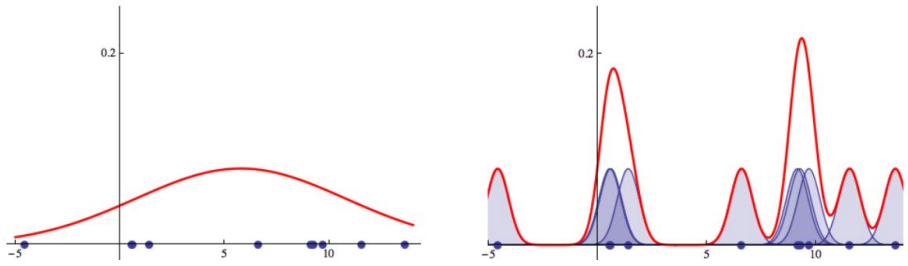


Figure II.7: ML estimate of the density using one Gaussian distribution (left) and kernel density estimation (right). Image source: Orbanz [133]

The sequence of intentions follows static patterns and each page visit is caused by one. In this simplified model, the change of intention z_t follows a static dependence A on the previous intent z_{t-1}.

Graphical Model: HMM. The graphical model of the HMM consists of two layers (see Fig. II.6). The upper layer represents the hidden states z with arrows/dependencies between z_{t-1} and z_t. The bottom layer depicts the observable events y_t conditional dependent on z_t (represented by an arrow).

The HMM oversimplifies user behavior. Two simplifications are especially hurtful. The state-space of the hidden process is fixed a priori. Hence, the number of hidden states (intentions) has to be sufficiently accurately guessed. This assumption is inappropriate for many applications and especially so in the case of user behavior. Furthermore, valuable information is lost due to the unstructured modeling of the emissions. We would assume a conditional dependency between successive page visits.

Structuring the observation models is trivial. We can reuse the first-order Markov model to enforce the same dependencies on the observation model as on the transition model. To obtain a flexible interpretation of the HMM that can adapt its state space to the data, we have to introduce nonparametric Bayesian models.

II.3.6 Nonparametric Models

The Bayesian models we encountered so far were parametric, i.e., they had a fixed number of parameters. In the dice throw example, the number of parameters corresponded to the number of sides of the dice. For the coin toss model, we needed one parameter. In both scenarios, we knew the number of parameters a priori.
However, in real-world scenarios, we often lack this information. We need algorithms that can deal with this lack of knowledge and infer the missing information from the data. For this purpose, nonparametric models have been proposed.
First of all, the term nonparametric does not mean that a model has no parameters. It states the opposite, that the number of parameters is unbounded, virtually infinite.

Example: Parametric vs. Nonparametric

The difference between parametric and nonparametric models can be illustrated by an example (for more details see Orbanz [133]). Figure II.7 shows two approaches to density estimation with Gaussian distributions. The goal is to determine the probability distribution of the random variable based on a representative data sample. In this example, both methods use the Gaussian distribution as a basis. The parametric approach, the maximum likelihood estimation, consists of a fixed number of parameters, i.e., the model consists of a single $1D$ Gaussian distribution with two parameters describing its mean and its variance. The kernel density estimator is nonparametric. In our context, it is sufficient to know that in this model

each data point is described by its own $1D$ Gaussian distribution. For a detailed description see Chen [34].
The kernel density estimator is comprised of as many Gaussian distributions as data points in the data. With each new data point, a new Gaussian distribution is added to the model. Thus, the number of parameters (mean and variance of each Gaussian distribution) grows with the data and is virtually infinite, requiring an infinite-dimensional space to express the parameter vector.

So, nonparametric models are not parameterless; rather, their parameter space adjusts as more data is observed. These models are especially useful in scenarios, where the number of parameters cannot be sufficiently accurately guessed a priori.

Our previous discussions dealt with the general principles of our work. What follows now deals with the issues that directly affect our contributions (it has been used or improved by us). For this reason, what follows now will cover the concepts in more detail.

To deal with nonparametric Bayesian models we first look at sampling methods, since they are important for the optimization of these models. Then we look at how probability distributions are defined and computed in an infinite-dimensional space. Finally, we explain approaches that we have used for behavioral modeling.

II.3.6.1 Preliminaries: Sampling Methods

We start with a simple example of how sampling can help us determine probability distributions. Assuming we want to know the average BMI of a group of 10 people. We could measure their height and weight, calculate their BMI, and get the average. Now suppose, we want the average BMI f of all N_c citizens z in our country,

$$\mathbb{E}_{f(z)} \triangleq \frac{1}{N_c} \sum_{c=1}^{N_c} f\left(z^{(c)}\right).$$

It is unrealistic to obtain the BMI of each citizen for our calculations. Thus, these calculations are not realistically feasible.

Monte Carlo In this case, we can fall back to sampling. We randomly select N_S citizens, measure their BMI and approximate the average BMI of the citizens in the country,

$$\mathbb{E}_{f(z)} \approx \frac{1}{N_S} \sum_{s=1}^{N_S} f\left(z^{(s)}\right).$$

The same holds, when applying probabilistic models. It requires the estimation of expected values or densities. However, the applicability of deterministic algorithms for posterior inference is often limited and complicated to derive. In these situations, sampling methods can be applied to approximate desired values and properties,

$$\int f(z)\, p(z)\, dz \approx \frac{1}{N_S} \sum_{s=1}^{N_S} f\left(z^{(s)}\right), \quad z^{(s)} \sim p(z).$$

The purple-colored part of the equation highlights the expressions that are approximated by sampling.
As an example, our posterior predictive distribution (see Eq. II.12 and II.13) can be approximated as

$$p(y|\mathcal{D}) \approx \frac{1}{N_S} \sum_{s=1}^{N_S} p\left(y|\theta^{(s)}, \mathcal{D}\right), \quad \theta^{(s)} \sim p(\theta|\mathcal{D}). \tag{II.23}$$

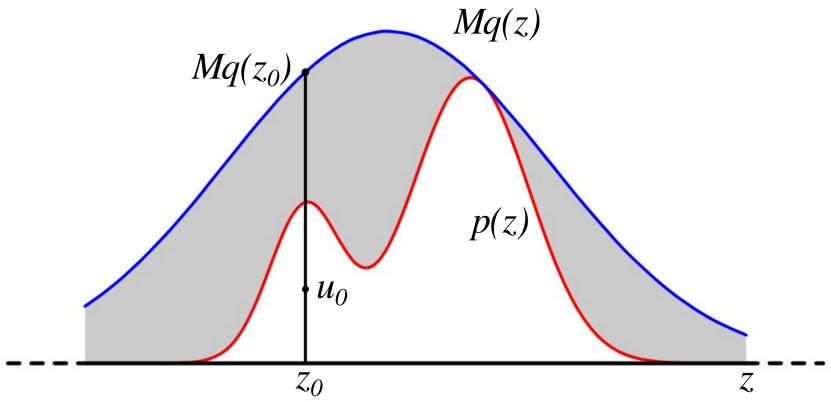


Figure II.8: Rejection sampling: Samples are drawn from a simple distribution $q(z)$; they are rejected if they fall in between the unnormalized distribution $p(z)$ and the scaled simple distribution $kq(z)$. Image source: Bishop and Nasrabadi [13]

One popular MC method is called *rejection sampling*. It approximates the expected value of some target distribution $p(z)$ with draws from a tractable distribution $z^{(s)} \sim q(z)$ with $Mq(z) \geq p(z), \forall z$ (see Fig. II.8). Here, M is a scaling parameter to ensure that our proposal distribution is at no point below our target distribution. We accept a sample with probability $p(z)/Mq(z)$. The expected value is then given by

$$\mathbb{E}_{[p(z)]} \approx \frac{1}{N_S} \sum_{s=1}^{N_S} p\left(z^{(s)}\right). \tag{II.24}$$

We can also approximate various properties of the distribution. Given unweighted samples of the target distribution and a suitable function f (e.g., a marginal) we can approximate

$$\mathbb{E}_{[f(z)]} = \int_{z \in Z} f(z)\, p(z)\, dz, \tag{II.25}$$

by

$$\mathbb{E}_{[f|\mathcal{D}]} \approx \frac{1}{N_S} \sum_{s=1}^{N_S} f\left(z^{(s)}\right). \tag{II.26}$$

Example: Monte Carlo Sampling

We explain Monte Carlo sampling using another example. Suppose we want to determine π, the ratio of the circumference of a circle to its diameter. We cannot define a function to describe π, however, we can use rejection sampling to determine the value of π.
We know functions containing π, e.g., the area under a circle, $A_c = \pi r^2$, with r as its radius. We observe that, without knowing π, we can determine if a point in the $2D$ space, $\mathbf{e} = (e_x, e_y)$, falls inside the circle or not,

$$f \triangleq \sqrt{e_x^2 + e_y^2} \leq r.$$

Given the space, the circle is drawn upon, we can calculate the fraction of the area covered by the circle, e.g.,

$$p_I = \frac{\pi r^2}{4r^2}. \tag{II.27}$$

When drawing from this space (uniformly at random) this fraction is the probability p_I of a sample falling inside the circle. Hence, we can use MC sampling to approximate p_I and then solve for π.
We define a $2D$ sample space on the $[-1, 1]$ interval and draw a circle in the center with

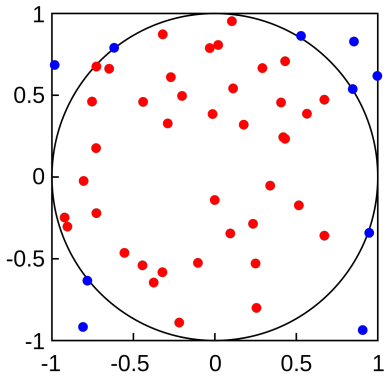


Figure II.9: Using Monte Carlo sampling to estimate π. Image source: [Wikipedia]

a radius r of 1 (see Fig. II.9). Then, we sample $2D$ points $\mathbf{e}$ uniformly from the interval $[-1, 1]$.

We obtain N_S samples and count valid samples (falling inside the circle) denoted by N_I,

$$p_I \approx \frac{N_I}{N_S}. \tag{II.28}$$

Given the sampled data points $\mathcal{D}$, we approximate the probability of obtaining a valid sample by the expected value of

$$\mathbb{E}_{[f|\mathcal{D}]} \approx \frac{1}{N_S} \sum_{s=1}^{N_S} f\left(\mathbf{e}^{(s)}\right) = \frac{N_I}{N_S}, \tag{II.29}$$

Given that this probability is calculated by dividing the area under the curve by the area of the sample space, we obtain

$$\begin{aligned} \pi r^2 &= \pi \quad \textit{(Area under the curve)} \\ 4r^2 &= 4 \quad \textit{(Area of the sample space)} \\ p_I &= \frac{\pi}{4} \end{aligned}$$

$$\frac{\pi}{4} \approx \frac{N_I}{N_S}$$

Solving for π we get

$$\pi \approx 4 \cdot \frac{N_I}{N_S}. \tag{II.30}$$

While sufficient for low-dimensional settings, these methods do not work well in higher dimensions. The reasons are the dependence of MC on independent draws from the target distribution (not possible in many Bayesian models) and the *curse of dimensionality* (sample space grows exponentially).

For example, suppose we want to use rejection sampling to approximate $p(x) = \mathcal{N}(0, \mathbb{I})$ with draws from our proposal distribution $q(x) = \mathcal{N}\left(0, \sigma^2 \mathbb{I}\right)$. We require $\sigma \geq 1$, therefore, we accept a sample with a probability of σ^{-D}, a function of the dimensionality D of the target space.

Markov chain Monte Carlo In higher-dimensional spaces, the probability of drawing valid samples decreases drastically. Moreover, when a valid sample is drawn, the algorithm does not adjust the procedure (i.i.d. sampling).

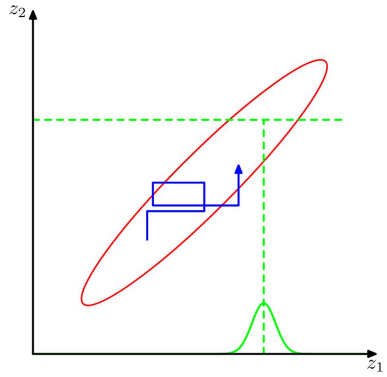


Figure II.10: The sampling process of a Gibbs sampler switching between two variables. Image source: Bishop and Nasrabadi [13]

Algorithm 1 Gibbs sampler

- **Initialize** $\mathbf{z}$
- **For** $t = 1, \ldots, \tau$ **do:**
  - **Resample** z_k **conditioned** on $\mathbf{z}_{\setminus k}$

$$z_1^{(t+1)} \sim p\left(z_2^{(t)}, z_3^{(t)}, \ldots, z_K^{(t)}\right)$$
$$z_2^{(t+1)} \sim p\left(z_1^{(t+1)}, z_3^{(t)}, \ldots, z_K^{(t)}\right)$$
$$\ldots$$
$$z_K^{(t+1)} \sim p\left(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, \ldots, z_{K-1}^{(t)}\right)$$

A sensible approach here would be to adjust the sampling to draw values closer to the valid sample $q(z'|z)$. This approach would give more weight to good samples (exploitation) but may ignore other good samples (exploration).

To reconcile exploitation and exploration we could construct a sequence of samples where each new sample depends only on the previous sample, i.e., we construct a Markov chain (see Fig. II.10). In this way, we consider the currently valid sample while continuing to explore the state space. Successive small steps can then lead to large jumps in the state space. But how can these dependent samples help to approximate the target distribution?

Suppose we have a target distribution P^* we do not know,

$$P^* = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.7 \end{bmatrix}, \tag{II.31}$$

and transitions probabilities

$$T = \begin{bmatrix} 0.34 & 0.33 & 0 \\ 0.25 & 0 & 0.25 \\ 0.41 & 0.67 & 0.75 \end{bmatrix}. \tag{II.32}$$

Then, given any initial random probability distribution P, e.g., $P = [0.9, 0.1, 0.1]^\top$, repeatedly updating $P = T * P$ converges to P^*. This property of a Markov chain is called *ergodicity* and the invariant distribution P^* the *equilibrium* distribution.

So, the initial state can be random and the target distribution is the equilibrium distribution. What we are missing is the transition function,

$$p^*(z') = \sum_{z \in Z} T(z, z')\, p^*(z)\,. \tag{II.33}$$

We assume that this proposal distribution T is sufficiently simple to sample from. Then, a sufficient condition for ergodicity is *detailed balance*,

$$p^*(z)\, T(z, z') = p^*(z')\, T(z', z)\,. \tag{II.34}$$

Therefore, in MCMC symmetric distributions are oftentimes used (e.g., a Gaussian distribution centered on the current state). Note that the whole process does not require knowing how to draw from the target distribution.

In this work, we make use of an MCMC method called *Gibbs sampling*. Suppose we are interested in the assignments $\mathbf{z}$ of observations to intentions. To learn these, we have to sample from $p(\mathbf{z}) = p(\{z_1, \ldots, z_K\})$. We choose some initial states. To now draw a new sample, we forget and resample each of the variables z_k with $k \in \{1, \ldots, K\}$ separately, conditioned on the remaining variables,

$$p\left(z_k | \mathbf{z}_{\setminus k}\right), \quad z_k \notin \mathbf{z}_{\setminus k} \quad \wedge \quad \mathbf{z} = \mathbf{z}_{\setminus k} \cup z_k. \tag{II.35}$$

The procedure is summarized in Alg. 1.

To show that the Gibbs sampler works as intended, we have to verify that it is ergodic and that the target distribution is its invariant distribution. As we are interested in $p(\boldsymbol{z})$, we have to show that it is invariant to each sampling step. Clearly, the marginal distribution $p\left(\boldsymbol{z}_{\setminus k}\right)$ is invariant (the values are unchanged). Combined with the correct conditional distribution $p\left(z_k | \boldsymbol{z}_{\setminus k}\right)$ the joint probability distribution is also invariant,

$$p(\boldsymbol{z}) = p\left(z_k | \boldsymbol{z}_{\setminus k}\right) p\left(\boldsymbol{z}_{\setminus k}\right).$$

Regarding ergodicity, we optimally want that any point in the sample space can be reached in a finite number of steps from any other point in the space. This holds if the conditional distributions are not zero anywhere. Otherwise, ergodicity has to be proven explicitly. For further discussions on MCMC and Gibbs sampling see Bishop and Nasrabadi [13]. A running example is discussed in Sec. II.3.6.3.

II.3.6.2 Nonparametric Distributions: Dirichlet Process

We start by making the following observations: Suppose we define a K-dimensional state space for the transition model of the HMM, so that the model is capable of capturing K different intentions. We observe that the parameters of the hidden process of the HMM reside in the transition matrix A. Technically, A can be interpreted as a set of K K-dimensional multinomial distributions, each conditioned on a transition state (intention). The configuration of each multinomial distribution is controlled by the Dirichlet prior distribution over the probability measures of the K-dimensional probability simplex. We note, that to change the state space dimensionality, we only have to change the support of the Dirichlet prior distribution. Therefore, it suffices to exchange the Dirichlet distribution for its nonparametric equivalent to allow for a flexible state space (i.e., a model where K adjusts to the data).

We provide a proper theoretical description of the *Dirichlet process* (DP) before presenting an extensive example. The DP is a generalized Dirichlet distribution [64]. While a Dir defines a distribution over a K-dimensional support, the DP is a generalization to arbitrary dimensions. The DP is a distribution over probability measures of an arbitrary probability simplex $G : \Theta \rightarrow \mathbb{R}^+$. It is controlled by α, a scalar value also called the *concentration parameter* and a *base measure* H, instead of the average probability measure $\boldsymbol{\alpha}$. Every possible partition of the probability simplex $(G(T_1), \ldots, G(T_K))$ is represented by a Dir,

$$\mathrm{Dir}\left(\alpha H(T_1), \ldots, \alpha H(T_K)\right). \tag{II.36}$$

For example, given $G \sim \mathrm{DP}(\alpha, H)$ with H as a $1D$ Normal Gaussian distribution, the DP has support over $[0.3, 0.1]$ as well as $[0.3, 0.1, -0.2, 0.2]$.
The marginals (i.e. each dimension) of the probability distribution $p(G(T_1), \ldots, G(T_K))$ are beta distributed:

$$\mathrm{Beta}\left(\alpha H(T_i), \alpha \sum_{j \neq i} H\left(T_j\right)\right) \tag{II.37}$$

Given some training data $\mathcal{D} = \{x_i\}_{i=1}^N$ with N data samples, and let, e.g., N_1 be the number of observed values in T_1, the posterior distribution is given by

$$p(G(T_1), \ldots, G(T_K) | \mathcal{D}, \alpha, H) = \mathrm{Dir}\left(\alpha H(T_1) + N_{T_1}, \ldots, \alpha H(T_K) + N_{T_K}\right). \tag{II.38}$$

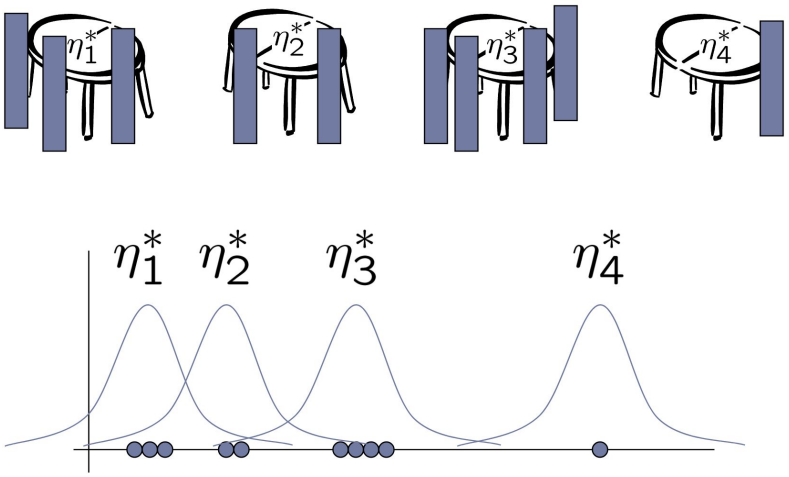


Figure II.11: Illustration of the CRP; e.g., η_1^* denotes the table with $k = 1$ and each mixture component is represented by a 1D Gaussian distribution. Image source: [El-Arini]

Hence, the updated DP is

$$G|\mathcal{D}, \alpha, H \sim \mathrm{DP}\left(\alpha + N, \frac{1}{\alpha + N}\left(\alpha H + \sum_{i=1}^{N} \delta_{x_i}\right)\right). \tag{II.39}$$

Optimizing the parameters of a DP typically requires drawing samples from the distribution. However, given the definition of a DP, the question is how to draw a valid measure from this infinite distribution. In what follows, we introduce two approaches: the *stick-breaking process*, and the *Chinese-restaurant process*.

Stick-breaking process Suppose we want to draw from a DP, a distribution that has support on infinite dimensional space. The challenge is to find an approach that yields valid samples. For example, normalizing the values so that they sum to one, as in the parametric setting, fails.

For one solution, we interpret a DP as a mixture model with $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^{\infty}$, an infinite sequence of mixture weights, and a separate representation θ_k for each mixture component. Then, an intuitive construction of the DP is given by *Sethuraman's stick-breaking process* (SBP) [166]. To obtain a sample of $\boldsymbol{\pi}$, it simulates the repeated breaking of a part of a stick. Given a stick of unit length, the repeated breaking creates a partitioning of the stick. The length of the parts represents the weights of our mixture components. It uses the Beta distribution to simulate the random partitioning of an interval of length 1 (the breaking of the stick), $\beta \sim \mathrm{Beta}(1, \alpha)$. The part β that we broke away represents the weight of a mixture component. The process is repeated with the rest of the stick. The whole process, (1) randomly dividing a stick of length 1 and (2) scaling to the length of the part of the stick that has not yet been broken away:

$$\beta_k \sim \mathrm{Beta}(1, \alpha) \qquad \pi_k = \beta_k \prod_{l=1}^{k-1}(1 - \beta_l) \qquad k = 1, 2, \ldots . \tag{II.40}$$

This process is also denoted as $\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha)$ Finally, we draw representations for each of the partitions from the base distribution of the DP,

$$\theta_k \sim H \qquad G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}. \tag{II.41}$$

It can be shown that $G \sim \mathrm{DP}(\alpha, H)$.

While this process yields an infinite number of mixture weights, the weights decrease and will be negligible at some point. Thus, this interpretation of the DP motivates truncation approximations, i.e., DP approximations that use a finite number of mixture components.

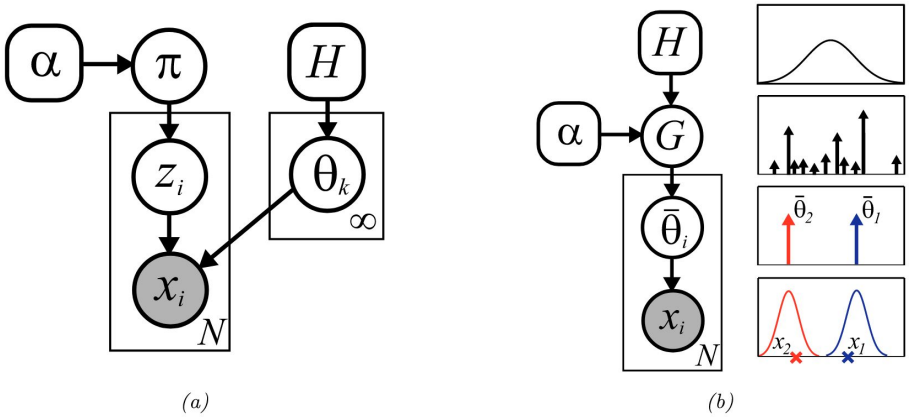


Figure II.12: Sampling from a DP mixture model: GEM-distributed cluster indicators as in the stick-breaking (left) and sampling according to the Chinese restaurant process (right). Image source: Sudderth [172]

Chinese-restaurant process Another useful interpretation of the DP is the *Chinese-restaurant process* (CRP). Imagine a restaurant with an infinite number of tables. Customers enter the restaurant and choose a table according to the following random process (see also Fig. II.11):

1. The first customer chooses the first table.
2. The N-th customer sits down either
   - at an unoccupied table with probability

$$\frac{\alpha}{N-1+\alpha}, \tag{II.42}$$

   - or at the k-th occupied table with probability

$$\frac{N_k}{N-1+\alpha}, \tag{II.43}$$

   where N_k is the number of customers already at that table.

According to the CRP, customers choose a table with a probability proportional to the number of people sitting at that table. This illustrates two properties of the DP, (*i*) the *rich-get-richer* attitude, a self-reinforcement mechanism (also observed in power-law distributions), and (*ii*) that the probability of two customers choosing the same table is non-zero, even if H is a distribution over an uncountable set. The CRP yields hard assignments where each element belongs to exactly one group. Its counterpart for soft assignments is called *Indian buffet process*.

We note that the distribution does not depend on the ordering in which the customers arrive. This property is called *exchangeability* and is used, e.g., for Gibbs sampling in the DP.

In summary, the DP is a nonparametric distribution over probability measures. Due to its base measure, it has support over all possible probability simplexes and therefore can adjust its complexity to the data. Eq. II.38 and II.39 describe how obtain the posterior when encountering data, i.e. how to adjust to data. The SBP and the CRP then tell us how to draw a probability measure from a DP (see Fig. II.12).

The question that remains is how to optimize a DP when encountering data. Directly calculating the posterior distribution is only possible in the most basic probabilistic models. For more complex models, we have to introduce approximation approaches for posterior inference.

II.3.6.3 Example: The DP mixture model

To give an example covering the different concepts of a DP (incl. the SBP and the CRP), we fit a DP mixture model to some data. Combining the DP with some observation distribution F, we get a mixture model. We assume that the mixture weights (e.g., the probability distribution over the intentions) are represented by $\boldsymbol{\pi}$,

$$\boldsymbol{\pi} \sim \mathrm{GEM}(\alpha). \tag{II.44}$$

Draws from the base measure H then represent the parameters of the mixture components,

$$\boldsymbol{\theta}_k \sim H(\lambda), \tag{II.45}$$

where λ are the parameters of the base measure.

Then, data is generated by sampling a mixture component denoted by z,

$$z_i \sim \boldsymbol{\pi}, \tag{II.46}$$

and generating data point x by drawing from the corresponding distribution $F(\boldsymbol{\theta})$ (e.g., the representation of an intent),

$$x_i \sim F\left(\boldsymbol{\theta}_{z_i}\right). \tag{II.47}$$

Now suppose we are provided with some training data $\mathcal{D}$ and we want to adjust the model accordingly. We turn to approximation techniques to come up with an updated posterior distribution of the parameters.
Using a variant of the Gibbs sampler (see Sec. II.3.6.1), the collapsed Gibbs sampler, we cope with the unbounded state space by integrating out θ. According to the sampling *one-by-one* strategy, we get

$$p\left(z_i = k|\boldsymbol{z}_{\setminus i}, \boldsymbol{x}, \alpha, \boldsymbol{\lambda}\right) \propto p\left(z_i = k|\boldsymbol{z}_{\setminus i}, \alpha\right) p\left(\boldsymbol{x}_i|\boldsymbol{x}_{\setminus i}, z_i = k, \boldsymbol{\lambda}\right). \tag{II.48}$$

The CRP tells us how to compute the mixture weights:

$$p\left(z_i = k|\boldsymbol{z}_{\setminus i}, \alpha\right) = \begin{cases} \frac{N_{k,\setminus i}}{\alpha, N-1} & \text{if } k \text{ is known} \\ \frac{\alpha}{\alpha, N-1} & \text{if } k \text{ is new} \end{cases} \tag{II.49}$$

We denote observation assigned to component k by $\boldsymbol{x}_{\cdot,k}$ where $\cdot$ denotes the set and k the *filter*, e.g., $\boldsymbol{x}_{\setminus i,k}$ denotes all $\boldsymbol{x}$ other than $\boldsymbol{x_i}$ that are assigned to component k. Then, given the assignment variable $z_i = k$, we can compute the marginal likelihood of x_i being generated by mixture component k as

$$p\left(\boldsymbol{x}_i|\boldsymbol{x}_{\setminus i,k}, \boldsymbol{\lambda}\right) = \frac{p\left(\boldsymbol{x}|\boldsymbol{\lambda}\right)}{p\left(\boldsymbol{x}_{\setminus i}|\boldsymbol{\lambda}\right)}, \tag{II.50}$$

with

$$p\left(\boldsymbol{x}|\boldsymbol{\lambda}\right) = \int \left[\prod_{i=1}^{|\boldsymbol{x}|} p\left(\boldsymbol{x}_i|\theta_k\right)\right] H\left(\theta_k|\boldsymbol{\lambda}\right) d\theta_k. \tag{II.51}$$

Finally, in case of a new k, we have

$$p\left(\boldsymbol{x}_i|\boldsymbol{\lambda}\right) = \int p\left(\boldsymbol{x}_i|\theta\right) H\left(\theta|\boldsymbol{\lambda}\right) d\theta. \tag{II.52}$$

Alg. 2 summarizes the procedure.

Algorithm 2 Collapsed Gibbs sampler for the DPMM

- **Initialize** $\mathbf{z}$
- **For** $t = 1, \ldots, \tau$ **do:**
  - **For** $k = 1, \ldots, K$ **do:**
    - Set $N_{k,\setminus i} = |\boldsymbol{x}_{\setminus i,k}|$
    - Compute marginal likelihood $p\left(\boldsymbol{x}_i | \boldsymbol{x}_{\setminus i,k}, \boldsymbol{\lambda}\right)$
    - Compute marginal posterior $p\left(z_i = k | \mathbf{z}_{\setminus i}, \alpha\right)$

  Compute the marginal L for a new component $p\left(\boldsymbol{x}_i | \boldsymbol{\lambda}\right)$

  Compute the corresponding marginal posterior $p\left(z_i = k+1 | \alpha\right)$

  Normalize posterior and sample new assignment $z_i \sim p\left(z_i | \mathbf{z}_{\setminus i}, \alpha\right)$

II.3.6.4 Hierarchical Dirichlet Process

We discussed modeling the transition matrix A of the hidden states in an HMM with a set of Dir. To allow the model to adjust the state space (e.g., number of intentions) to the data, we then exchanged each Dir with a DP. However, we are still missing the final piece. As of now, the DPs in the set are not tied together. To see why this is an issue, think of the hidden state (intentions) as nodes in a graph. The transition matrix A tells us, how to traverse between the different states. If the DPs are not tied in some way, one DP denotes the transition probabilities to a set of DPs, while the next DP describes transition probabilities to a completely different set of DPs. We would obtain a branching structure rather than a chain structure. We have to introduce a mechanism that controls the states of our DPs, thus tying them together.
Interpreting A as a graph with non-zero transition probabilities, we observe that it defines a first-order Markov model. Therefore, it has an invariant distribution, the equilibrium distribution. As we learned, this distribution is $p\left(\mathbf{z}\right)$, the distribution over the states (weighted intentions). The distribution describes how often we encounter an intent z_k. It has to be included in the model to obtain a correct representation of a graph (with a flexible state space).
We, therefore, use a hierarchical arrangement of DPs. Using the invariant distribution as the base distribution H of our DPs Teh et al. [173], the state space of the set of DPs is tied together. This model is called the hierarchical Dirichlet process (HDP) and is defined as follows:

$$G_i | G_0 \sim \mathrm{DP}\left(\alpha_i, G_0\right) \qquad G_0 \sim \mathrm{DP}\left(\alpha_0, H\right). \tag{II.53}$$

In an HDP, the realization of a DP G_0 is used as the base measure for all its subordinate DPs, $\mathrm{DP}(\alpha_i, G_0)$. The equivalent scheme of the *stick-breaking process* [173], which takes the sub-intervals β_k directly as inputs to the *stick-breaking process* of the subordinate DPs, is:

$$\beta'_{ik} \sim \mathrm{Beta}\left(\alpha\beta_k, \alpha\left(1 - \sum_{l=1}^{k}\beta_l\right)\right) \qquad \beta_{ik} = \beta'_{ik}\prod_{l=1}^{k-1}\left(1-\beta'_{ik}\right). \tag{II.54}$$

Together, the two *stick-breaking processes* represent a realization of an HDP.

II.3.6.5 Hierarchical Dirichlet Process - Hidden Markov Model

We covered all components for a flexible interpretation of the HMM, the infinite HMM [9] or the HDP-HMM [70] The HDP for the nonparametric interpretation of the mixing proportion (the process that controls the hidden states) and the categorical distributions for modeling the emission distributions.

HDPs use a DP as a Bayesian nonparametric prior over the hidden states. A DP is a distribution over probability distributions in infinite-dimensional parameter space. It is sufficiently

described by a concentration parameter α and a base measure H. The draws from the stochastic process $G \sim \mathrm{DP}(\alpha, \boldsymbol{H})$ are discrete, and multiple observations have non-trivial probabilities of assuming identical values. The generative process of the mixture proportion is as follows:

$$\begin{aligned} G_0|\gamma, H &\sim \mathrm{DP}(\gamma, H) \\ G_z|\alpha, G_0 &\sim \mathrm{DP}(\alpha, G_0) \qquad \text{for } z \in \boldsymbol{z}. \end{aligned} \tag{II.55}$$

The hyper-parameters of the model consist of α and γ, the concentration parameters, and H, the base distribution. $\boldsymbol{z}$ represents its infinite-dimensional state space. We also introduce z_0 as a predefined initial state. Then, the emission distribution is

$$\begin{aligned} z_t|z_{t-1}, G_{z_{t-1}} &\sim G_{z_{t-1}} \qquad \text{for } t = 1, \ldots, \tau \\ y_t|z_t &\sim \mathrm{Cat}_{z_t}. \end{aligned} \tag{II.56}$$

τ denotes the number of observed events. We can interpret the HDP-HMM as an HMM with virtually infinite states.

However, the model has problems with the persistence of states. It tends to explain similar patterns with redundant states that it switches back and forth between. While this is not a problem for prediction purposes, it is critical for tasks that involve inferring the hidden states. Therefore, Fox et al. [70] proposed an alternative interpretation of an HDP-HMM in which self-transitions are given special consideration. It features an alternative transition kernel G_z:

$$G_z|\alpha, \kappa, G_0, z \sim \mathrm{DP}\left(\alpha + \kappa, \frac{\alpha G_0 + \kappa \delta_z}{\alpha + \kappa}\right). \tag{II.57}$$

Here, κ represents the additional mass added to the self-transitional state of the corresponding kernel, signaled by the point mass δ_z.

We arrived at a robust interpretation of the HMM with a countably infinite dimensional (hidden) state space. To this end, we replaced the static transition matrix of the HMM with a hierarchical Dirichlet process. It is a set of Dirichlet distributions controlled by a Dirichlet process to model the state frequencies. Each subordinate DP represents a transition kernel G_z that expresses the probability of moving from state z to another state. Finally, to correctly capture the hidden states (avoiding redundant states), a self-transition bias was added.

II.4 Community Detection

In our comprehensive analysis of the German-language Twitter community (GTC), we use community detection to gain valuable information about the network structure. Here, the vertices of the network represent the users of the GTC and the edges show the interactions between the different users. The vertices and edges define a graph.

As part of our analysis, we need to identify communities in the network. However, since we do not know the memberships, we have to derive them from the properties of the network structure. Therefore, we need a cost function that we can use to decide whether we have reached a satisfactory solution.

II.4.1 Modularity

For this, modularity is often used. Modularity is the ratio of the strength of connections within a group compared to the strength of outgoing connections, i.e. connections to users outside the group. Here, we consider modularity in terms of directed graphs. A represents the corresponding transition matrix. Then, m denotes the sum of all edge weights in the graph. $k_p^I = A_{\cdot,p}$ denotes the sum of all edge weights that go to p and $k_q^O = A_{q,\cdot}$ denotes the sum of all coming from q.

$$Q = \frac{1}{m} \sum_{p,q} \left(A_{p,q} - \frac{k_p^I k_q^O}{m} \right) \delta_c(p, q), \tag{II.58}$$

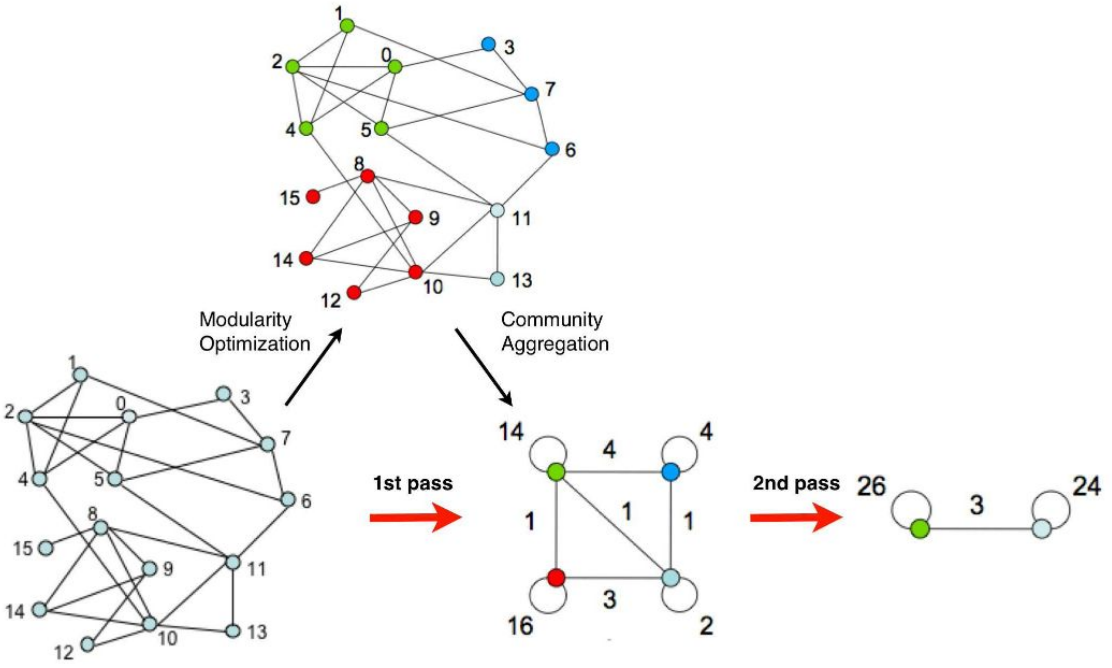


Figure II.13: Community Detection: The two phases, optimization and aggregation, of the Louvain algorithm. Image source: Blondel et al. [15]

with δ_c as the Kronecker delta function that is 1 if p and q are assigned to the same community. It is 0 otherwise.

The idea is to group users who are more closely connected through their interactions than to other users in the network. However, the graph we are looking at is so large that this is no longer possible manually. Therefore, we use so-called community detection methods.

The goal of these algorithms is to optimize the modularity (from -0.5 non-modular to 1 fully modular). Although exact optimization would lead to optimal groupings, this is often not possible due to the sheer size of many networks. Therefore heuristics are used to optimize group membership.

II.4.2 Louvain Algorithm

The Louvain algorithm is a simple, yet powerful approach to performing community detection on a large network. The algorithm is split into two phases it alternates until convergence as depicted in Fig. II.13.

Phase I. In the first phase, the modularity in the graph is optimized. For this, the change in modularity when moving a node p to a cluster of one of its neighbors is calculated,

$$\Delta Q = \frac{k_p^c}{m} - \left[\frac{k_p^O k_c^I + k_p^I k_c^O}{m^2} \right]. \tag{II.59}$$

Here, k_p^c represents the sum of weights from all edges within community c that come from or go to p. k_c^I and k_c^O then denote the sum of all edge weights coming into community c or going out, respectively.

When all possible combinations are computed, node p is assigned to the cluster with the greatest modularity increase. If none exists, its community assignment stays as it is.

Phase II. In the second phase, the communities are aggregated, i.e., all nodes of a community are combined into one node and thus a new graph is spanned. All edges within a community are

combined using a self-loop. All edges from nodes within the community to another community are aggregated (sum of edge weights) to an edge between the respective *community nodes*. Now phase 1 starts again on the new graph.

The entire process is repeated until there is no more change in cluster assignments and the maximum modularity score is achieved.

CHAPTER III

RELATED WORK

We are engaged in related research in the areas of news consumption analysis, behavior modeling, and bot detection approaches. The review of related work on news consumption focuses on effects that have implications for democratic societies. In behavior modeling, we pay particular attention to approaches that use time-series data. The section concludes with a summary of bot detection.

III.1 News Consumption

Our second contribution relates to the understanding of today's online news consumption. Online social networks (OSNs) offer the opportunity to study human interactions at scale [104]. We cover various related topics of technical and theoretical nature.

III.1.1 News Providers and Influential Accounts

Popular OSNs like Twitter have a high potential to disseminate information to large audiences. Therefore, many organizations, marketers, politicians, and generally people who want to promote products or content embrace social media. In recent years, news providers, in particular, have used social media to increase their reach, constantly adapting to the demands of a new online audience [58, 59]. In this context, the findings also indicate that journalists tend to express their opinions more freely on these platforms, which is typically observed in connection with micro-blogging but contradicts the journalistic norm of objectivity [103]. Not only do they use it as a convenient and cheap tool for spreading and supporting news, but they also use it to inform themselves [139].

The urge to use viral marketing and social recommendation networks often coincides with two related themes. Finding and understanding the most influential entities [106, 33]. A highly active area of research in this context is the identification of influential users. Here we consider a user to be influential if their actions influence the actions of many others. In other words, a user's influence is determined by how widely the information a user shares can spread within the network [119].

Researchers proposed different approaches to identify influential users within Twitter [155]. Much of the heuristics stems from user interactions (e.g., user mentions and retweets). Passive topology metrics (e.g., follower-links), on the other hand, seem to be poor indicators of actual influence [29]. An early approach by Kwak et al. [99] measured the influence of Twitter users across a vast, comprehensive network by calculating PageRank scores for all users. PageRank [135], originally developed by Google Search to rank websites, is based on eigenvectors. Eigenvectors are a measure of centrality that favors well-connected users. Results reported by Kwak et al. showed that Twitter's top users ranked by PageRank correlated with rankings based on the number of their followers. Since then, the algorithm has been repeatedly applied to Twitter

network data [154, 92, 78] and serves as basis for a variety of custom metrics, such as *Influence Rank* [83] and *InfRank* [92].

III.1.2 Political Orientation

Studies on the matter of OSNs rely on large and rich data sets. Depending on the objectives, data often has to be augmented with further information. Our studies rely on information about the political affiliation of users.

In this context, Colleoni et al. [39] worked on the complete Twitter-sphere of 2009 provided by Kwak et al. [99] to investigate the political homophily of Republicans and Democrats across the entire network. Using linguistic features extracted from annotated tweets and news texts, they utilized a supervised classification approach. While a common approach in the area of user classification on Twitter [136], research showed that the prediction of political affiliation is not reliable in multi-class scenarios, e.g., in the context of the broader political spectrum of German parties [38]. Additionally, textual features of tweets are not stable over time. Here, tweets from topical authorities, who seem to be more consistent in their messages, represent an exception [136].

Therefore, an algorithm inferring user characteristics and interest from context-specific activities is more promising for the German Twitter user base. In this context, several attempts rely on Wikipedia articles to infer the interests of users [77, 23, 50]. Wikipedia and its broad range of categorized articles, including people, events, and locations, can be utilized to build a reliable knowledge database. Faralli et al. [61] approximated user interests by finding *"friends"* they could link to Wikipedia articles. For example, if a user followed a famous basketball player, her interests included sports and basketball. The researchers proposed a hierarchical representation of user interests and conducted a large-scale homophily analysis on Twitter. Their methodology offered a compact, tunable and readable way to examine user interests.

For a more thorough understanding of user interests, Himelboim et al. [87] leveraged frequently shared hyperlinks, user mentions, and hashtags and, thereby, analyzed users based on domain-related interests and hashtags. We deploy a similar approach for inferring user attributes.

III.1.3 Community Detection and Echo Chambers

We explore the existence and spread of echo chambers. Detecting echo chambers is commonly performed by modeling the network as some graph and extracting clusters of nodes with high interconnectivity. Therefore, we rely on the information of the community structures within our data. Besides the investigations on echo chambers, we also leverage the information to understand user behavior.

Early studies investigated simple social graphs, as represented in the contact relationships on the OSN [94, 164]. These approaches assumed that online friendships inherit crucial attributes from real-world relations so that the majority of meaningful interactions in OSNs occur between friends. Wilson et al. [188] analyzed interactions, i.e., wall posts and photo comments, among Facebook users. They reported that most user interactions occur only within a tiny subset of a user's friendslist, often leaving half of the remaining friends out of all communications. They studied an interaction graph containing only edges between users interacting instead of relying on friendship links. Their evaluations on two adequate social applications demonstrated that using an interaction graph yields better results than using a friendship graph.

Himelboim et al. [87] used topic networks applied to a clustering approach to detect echo chambers on Twitter. They collected topic-related Twitter data and created multiple interaction graphs based on retweet-, mention-, and reply relationships. Using the Clauset-Newman-Moore algorithm, they identified communities of users that had frequently interacted with each other. These users created a structure of interaction silos where echo chambers might emerge. They

then assessed the occurrence of ideological similarities among users within a community by analyzing their frequently shared hyperlinks, user mentions, and hashtags for a more thorough examination of the identified communities. By assigning a political affiliation to influential users within the community, they aimed to infer its political orientation. An influential user was determined with in-degree centrality to measure his exposure to other users.

Conover et al. [40] demonstrated that detecting exposure to alternative news vs. segregation into echo chambers yields different, possibly conflicting results depending on the chosen interaction to model the graph. Their experiments illustrate the importance of the selected methods and graph modeling schemes. Utilizing two different interaction graphs, they tried to link user similarities and political orientation. Crawling 250 000 tweets during the 2010 U.S. congressional midterm elections, they modeled graphs both, depending on retweets and mentions. Detecting communities on the Retweet-graph resulted in two highly segregated communities with opposing political ideologies and only a few inter-connections. The authors concluded that the structure encoded user preferences to retweet similar political views. Community detection on the Mention-graph, however, yielded fundamentally different results. A single community emerged, containing politically heterogeneous users. Despite their opposing political ideologies, these users exhibited a high level of interaction. They concluded that users from both political sides confront each other with content, contrary to their political affiliation, which leads to a ruffled exchange of Tweets. In conclusion, they posit that meaningful analysis benefits from comprehensive, combined interaction graphs.

Besides Conover et al., several other studies in recent years dissected echo chambers within OSN. These reports describe a user's tendency to retweet content with political views similar to theirs [40, 19, 63]. They observe sharing content from such a narrow context fosters segregation by political orientation.

Other studies on this topic investigate the extent of such behavior considering the political orientation of individuals [7, 39]. Boutyline and Willer [20] observed that conservative and politically more extreme individuals showed a more pronounced tendency to form segregated user groups than liberals. While Barberá et al. [7] report similar results consistent with psychological theory and research bearing on ideological differences in epistemic, existential, and relational motivation, they conclude that previous work may have overestimated the degree of ideological segregation in online social networks.

In this context, we have to emphasize the difference between the concept of an echo chamber and an epistemic bubble. While the latter relates to information networks that exclude important information sources without their members noticing it, echo chambers actively discredit or even exclude contrary opinions [128].

III.1.4 Promotional Profiles

Besides manually controlled accounts, there also exist orchestrated and automated ones. Several guidelines recommend creating social media profiles for improved public relations and dissemination. To increase the distribution of news content in social media, Orellana-Rodriguez et al. [134] propose best practices. They suggest creating employee accounts to promote their corresponding tweets. Such accounts should contain a statement about their affiliations. News providers establish Twitter profiles to further the distribution of their articles [59].

News agencies, such as Reuters or AFP, instruct journalists using their accounts for work to include a disclaimer. The disclaimer identifies them as employees of a specific news agency [153, 72, 143]. It should also include a declaration that they speak for themselves and not their employers.

III.2 Behavioral Modeling

Time-series data plays a crucial role in capturing user behavior on the web. While snapshots of activities convey a compressed picture of behavior, time-series capture behavior over time. Therefore, before we look at behavior modeling, we take a look at time-series data, i.e., best practices and the current state of research.

III.2.1 Time-Series Data

We start with a closer look at existing approaches that project time-series data into a latent space to understand where to improve them. A simple approach to project time-series data into a latent space is to use a naïve Bag-of-Words (BoW) strategy, i.e., to express each time-series as the total number of unique observation events.

Wang et al. [182] expands this representation with additional meta-data. In the context of user modeling, this can include statistical data such as the average number of clicks a user makes within a session or the total number of clicks. These approaches ignore temporal patterns within the data, potentially leading to a significant drop in prediction performance.

The following approaches propose BoW strategies to improve temporal pattern recognition. They use constructed features representing some temporal information. Viswanath et al. [178] propose a strategy to project temporal, spatial, and spatio-temporal information onto a fixed-dimensional space. Temporal information captures characteristics over time, such as the daily number of likes a user receives on Facebook. Spatial information is encoded as a histogram over an a priori defined set of features. This transformation is similar to the BoW approach above. In the context of a Facebook user, the histogram's buckets represent *like categories* such as sports, politics, education, etc. Finally, spatio-temporal information is encoded as the temporal evolution of the spatial information of observed values. For the Facebook scenario, this means that instead of the total number of likes per category, one could capture the time series of distributions of like-categories per user and day. This approach represents a simple solution for information and computation efficiency regarding time-series data. However, the approach of Viswanath et al. [178] requires pre-processing of the time-series data combined with a feature design tailored to the task at hand. Depending on the settings (type of time-slots, buckets, etc.), the approach can result in a high dimensional latent space that requires large amounts of training data.

Solutions that explicitly model time-series data more thoroughly analyze the temporal dynamics within the data. A simple approach to capture the dynamics is n-gram models. Here an abstraction of the temporal information is obtained by running a sliding window of size n over the data and summarizing the occurrences of the various repeating patterns of length n. N-grams are commonly used for natural language processing tasks. Although it contains temporal information, it has some obvious disadvantages. When used as a projection, it potentially leads to an explosion of dimensionality in vector space, i.e., each n-gram represented as a dimension. The lack of an abstraction mechanism additionally may result in an increased information loss induced by the projection, e.g., both, n-gram A = "abacb" and B = "abaca" would represent a dimension within the latent space with a similar relation to each other as to C = "ddddd".

An adjustment to this approach is to extract patterns from n-gram representations of time-series data [25, 180]. While providing an abstraction level, it assumes that time-series follow a single pattern. To get around this limitation time-series must be split into segments that match specific underlying patterns. Figueiredo et al. [66] suggest applying a static segmentation process before mining these patterns. Modeling smaller portions of a time-series may result in more accurate representations of sequential information in data.

III.2.2 User Behavior

We established that times-series hold valuable information for behavior modeling. Therefore, we focus on related approaches.
User behavior is often modeled by clustering using probabilistic models. The most commonly studied type of approach uses Markov models [142, 112, 25, 195, 48]. Early work explores the use of probabilistic methods, and later papers use Markov chains [142, 112] to build stochastic models that capture patterns of behavior.

Cadez et al. [25] deepens the idea by proposing a mixture model of Markov chains to divide data into meaningful groups and focus analysis on those groups. Here, each manifestation of a common behavioral pattern is represented by a Markov chain. In the context of our application scenario, a user interacts with a system, switching between (possibly latent) states of a Markov model. Each state represents a possible interaction between user and system. By using first-order Markov chains, the next state is conditioned only by the previous state. The approach by Cadez et al. [25] produces interpretable results and is computationally efficient. However, model selection can lead to suboptimal results because determining the number of groups is not always obvious, and the non-convex problem may have many local optima. A similar approach using a mixture model of hidden Markov models [195] considers intertwined click traces, and Deshpande and Karypis [48] propose selective Markov models to identify user behavior patterns.

To capture user behavior in more detail, higher-order Markov models [118, 24, 10] can be used. However, approaches such as the one proposed by Mochihashi and Sumita [118] suffer from inefficient computations and poor interpretability. Other approaches require inappropriately large data sets, as the model parameters grow exponentially with the number of states N and the order o [24, 10]. Du et al. [52], Dubey et al. [53], Brown [22] use Bayesian non-parametric mechanisms to control the complexity of the respective models. For example, by combining a temporal point process with a Bayesian non-parametric prior, the relationship between the two domains is explored [52]. Compared to first-order Markov models, the resulting Dirichlet-Hawkes process allows for more detailed modeling of user behavior. However, like neural networks, point processes focus on predictive performance and are often not well interpretable.

Two models that naturally capture the dynamics caused by different types of segments are the standard and the infinite hierarchical hidden Markov models ([i]HHMM) [67, 125, 86]. Each hierarchy of an (i)HHMM is a separate hidden Markov model (HMM) in which all observations reside in the leaves, called production states. While the HHMM requires an a priori fixed number of levels, the iHHMM allows for a potentially unbounded number, growing with the data. While highly flexible, inference in these models is complicated and expensive, rendering it inapplicable to real-world problems [67, 86]. However, related work suggests that two-level analyses of dynamics are sufficient in many real-world applications [131, 130, 190]. Recent studies, therefore, investigated simplified models by restricting the depth of the hierarchy.

Stepleton et al. [168] propose to combine the infinite HMM [9] (iHMM) with a block-diagonal prior. They assume that the transition matrix of the iHMM has a nearly block-diagonal structure. Therefore, subsets of hidden states are grouped into an unbounded number of blocks. By modifying the Dirichlet process prior, the model increases the transition probability of the states within a block. We can interpret each block as a segment type. However, the model has problems with segment types with overlapping discrete-valued observation spaces.
A similar idea, a preference for self-transitions within a mixture component of the hierarchical Dirichlet process hidden Markov model [173] (HDP-HMM), is an essential component of the sticky HDP-HMM Fox et al. [71] propose. Similar to the block-diagonal iHMM, successive hidden states in this model prefer the same state. By adding an additional layer to the hidden states, the sticky HDP-HMM treats the conditional distribution of observations non-parametrically. While the model divides sequences into segments, it cannot capture dynamics within a state.

Finally, studies by Johnson [95] and Saeedi et al. [158] examine the utility of incorporating an explicit state-duration distribution rather than a preference for self-transitions [71, 168]. Both approaches are Bayesian non-parametric models that employ a two-level analysis of the dynamics within the data. While the model proposed by Johnson [95] learns a distribution expressing the total duration within a state, the segmented iHMM (siHMM) [158] models a state-duration distribution expressing the probability of change in the current state as a function of the observation and the hidden state. Both models require input sequences of equal lengths.

III.3 Bot Detection

Our third area of contribution focuses on the detection of sophisticated bots. Early approaches examined spam-related topics on the social web. To do this, Benevenuto et al. [11] collected data from users on Twitter. They manually labeled users as spammers or non-spammers and proposed an SVM classifier for detection. To help human users understand who they are communicating with, Chu et al. [35] developed a model for identifying accounts related to human, bot, or cyborg (i.e., bot-assisted human or human-assisted bot). Their approach consisted of a four-component model that combined entropy- and machine learning-based information with account characteristics into a final decision component. To make social bot detectors available to the general public, Davis et al. [44] launched the Botometer service (former BotOrNot) in 2014. The free service for evaluating accounts on Twitter uses more than 1000 features.
Then in 2017, Cresci et al. [42] reported on a new type of bots, called social bots. Empirical studies proved that humans and state-of-the-art detection approaches performed poorly in detecting these new bots, as these bots mimicked benign user behavior. The research, therefore, examined the larger context and highlighted another promising approach to the task: collective behavior detection.
In-depth analysis of the cyber-criminal ecosystem of social web platforms [191, 174, 74] provided detailed information about the activities and scale of criminal accounts. The researchers realized that coordinated campaigns often run through the same accounts. Therefore, they assumed that coordinated behaviors were largely due to malicious campaigns on the platforms. In collaboration with Facebook [27], Renren [181], or YouTube [107], researchers proposed models that leverage detailed data on social network account activity (e.g., click-stream data). For example, Chavoshi et al. [31] assumed that people are not capable of acting in a highly synchronous manner over an extended period. They proposed an activity correlation model that does not require labeled data.
However, recent work has highlighted serious weaknesses in studies throughout the discipline [56, 146, 176]. Echeverria et al. [56], for one, examined the established evaluation scheme for bot detection approaches. They emphasized the lack of generalization when approaches are trained and tested on the same pool of bot data. However, in reality, the nature and working patterns of observed bots are constantly changing [42]. Therefore, Echeverria et al. [56] proposed a Leave-One-Botnet-Out evaluation scheme (LOBO). Based on a collection of real-world data sets, the model measures the generalization ability of approaches. For this purpose, individual bot types are excluded from training and then used later for evaluation purposes. The results of the Botometer algorithm suggest that modern approaches that use meta-data do indeed fail in detecting new types of bots. Furthermore, Vargas et al. [176] challenged the assumption that humans are incapable of acting in a highly synchronized manner. They showed that coordination is indeed not uncommon in Twitter communities. When adjusted for a high detection rate of malicious behavior, 46% of legitimate activities were misclassified.
In this work, we, therefore, investigate whether generalized bot detection, based on account activities rather than coordinated campaigns, can achieve high detection rates in previously unknown bot families.

CHAPTER IV

NEWS CONSUMPTION ANALYSIS

Our first contribution focuses on the news-consuming behavior of the German-speaking Twitter population. The results of this chapter have been published in a journal article in Online Social Networks and Media (OSNEM) [152] and as work-in-progress [141].

Online social networks, such as Facebook, Instagram, YouTube, and Twitter, attract enormous attention. These networks have almost ubiquitous reach. The information circulating in these networks is manifold and comes from various sources. In particular, news providers are making great efforts to publish and disseminate their articles on multiple social platforms to reach a wider audience [59]. Politicians, too, are embracing the digital environment. They use social media for campaigning and connecting with their target audience [169]. The amount and availability of informative content have caused a rising number of social media users to consume their daily news directly on these platforms [89]. The availability of social-media mobile apps amplifies this effect and increases exposure in various everyday situations. Democratized information acquisition, dissemination, and the free flow of information are positive aspects of this development. However, at the same time, it represents potential risks to political discussion in our society.
Additional actors have emerged. Some are distributing misinformation, conspiracy theories, and propaganda with agendas ranging from the commercialization of click-bait, over political influence, to establishing opinion platforms as hidden distribution channels for marketing of all types of products [196]. Recent publications underline that social media users are more exposed to populism, propagated by political actors from the extreme ends of the political spectrum, than individuals without social media [60]. A balanced news selection has to give way to a choice of posts and topics reinforced by the user's chosen neighborhood in this sheer mass of information. This fosters political polarisation and ideological segregation [7, 20, 39]. Incidental news consumption reinforces such effects [16], leading to a reduction in political education [3]. This development arguably represents a primary risk for democratic societies.
Individuals who put more trust in information shared by friends, likely regress to consume news from narrow contexts [16]. This development increases the difficulty of evaluating the credibility of information sources [189]. It consequently makes the emerging closed user groups more vulnerable to profit-oriented marketing, political campaigning, and general misinformation. Literature has termed such user groups *"echo chambers"*. A phenomenon that amplifies and reinforces common opinions within groups through repetition and mutual approval. Typically claimed to exist in social networks, they increase political polarization and ideological segregation [7]. Members of these *echo chambers* are more exposed to populism, propagated by actors from the extreme ends of the political spectrum [60]. They hence have a tremendous impact on the process of political opinion-forming.

Our goal is to analyze the impact of anti-democratic/controversial content on a western European Twitter community. In exhaustive studies, we try to answer the following research questions:

- **RQ 1:** What is the extent and impact of controversial news content?
- **RQ 2:** To what extent are echo chambers feeding on controversial content influencing the community?

Due to data collection limitations, we have to strike a trade-off between sample size and data quality. We base our studies on a concise, well-defined, and virtually complete Twitter community, concentrating on the German-speaking Twitter community (GTC).
Selecting tweets by the language allows for a detailed observation of such a specified population. Often, culturally and geographically diverse groups speak the same language. The GTC, however, represents a large, geographically well-defined population of around 7 million active users. The majority are from Germany, Austria, and adjacent parts of neighboring countries, and they all share a relatively homogeneous political landscape and corresponding media outlets.
To study anti-democratic content, we define controversial and non-controversial content. Controversial content combines articles from providers that contribute to misinformation, conspiracy theories, political propaganda, and similar democracy decomposing elements.

We base our studies upon two building blocks:

- **Content** We propose an automated data augmentation strategy to facilitate data enrichment on large, real-world data sets. We leverage shared external content and hashtags to get a high-level understanding of discussions in an automated manner.
- **Distribution/Impact** We leverage dynamic interactions between users (i.e., mentions, retweets, quotes, and replies) to accurately measure relationships.

Thereby, we get a (*i*) high-level understanding of what is shared/discussed based on the automated categorization of content, a (*ii*) measure on the share of news-related discussions within the network, and (*iii*) can identify influential actors and multiplication networks, (*iv*) measure the presence of established news providers within the network, (*v*) analyze user engagement w.r.t. different types of news, (*vi*) study discussion patterns of users, and (*vii*) perform a community structure analysis.
Based on a comprehensive understanding of the content contributing to political opinion-forming, we study the influence of phenomenons related to controversial content.

In the remainder of this chapter, we first give an overview of the state-of-the-art in Section IV.1 and describe our approach in Section IV.2. We report on the results of our experiments and discuss them in Sections IV.3 and IV.4.

IV.1 Data Collecting in the Past

Due to the rapid development and mass distribution of social media, platforms such as Facebook, Twitter, and YouTube have gained remarkable influence in the last decade. Several studies show more and more people consume news through these channels [89, 16].

Analysis of the impact of this development requires high-quality user behavior data from these platforms [170]. Data sets on users tweeting in English are readily available. However, it is unclear whether conclusions based on these data are generalizable to other groups. A sample survey in other communities has not been convincingly conducted to date [161].

In the past, the academic community has used methods for collaborative data collection [51]. For example, in 2010, Kwak et al. [99] crawled the entire Twitter platform. Using 20 machines operating with different IPs that crawled tweets via the Twitter Search API over several weeks, they bypassed Twitter's rate-limiting. They obtained 41.7 million user profiles, 1.47 billion social relations, 4 262 trending topics, and 106 million tweets. Given Twitter's growth in recent years, this approach is very costly and time-consuming, and thus it may be considered infeasible to collect a complete data set. In addition, according to Twitter's

policies, public sharing of certain content is now prohibited [185]. Therefore, researchers have started to develop customized data crawling techniques, i.e., construction kits for individual data collection. A reliable data collection process should be transparent and reproducible to allow other researchers to replicate it.

With over 500 million tweets per day, pre-processing is not a trivial task. Current studies therefore use Twitter's streaming API. It makes it possible to obtain a limited number of real-time tweets that match a specific word filter. Thus, the amount of data collected depends on the prevalence of the specified search terms (e.g., event-related hashtags) [76, 12, 4]. However, Twitter limits the total number of retrievable tweets to 1% of all tweets per day. If the number of tweets matching a word filter exceeds this limit, the stream returns a random sample of these tweets. Studies have shown that the streaming API returns an unrepresentative sample of tweets [120, 121]. An undesired effect for research purposes. It is not sufficient to fix the discrepancy in the sample by using multiple machines to combine concurrent samples from the streaming API [96].

Despite these limitations, Scheffler [161] published a method that allowed her to collect a representative snapshot of the German-language Twitter traffic. She configured Twitter's streaming API based on a German-only word-filter list. Because the number of German tweets within the entire Twittersphere is considerably small, the number of tweets captured only slightly exceeded the 1% threshold. This allowed minimizing the impact of Twitter's downsampling. However, by capturing all tweets that matched at least one word from the word list, Scheffler also received a large number of tweets that were not German. She used a language detection algorithm to filter out these tweets. Due to insufficient labeled data, the filtering algorithm had to be evaluated manually on a small subset of the collected tweets. As for the impact of downsampling, Scheffler concluded that it was negligible, accounting for less than 3% of the missing data.

IV.2 Dissecting German Tweeting Flocks

This work provides exhaustive studies on the news consumption of German-speaking Twitter users. The basis of our approach is the data acquisition strategy (i.e., obtaining an automatically labeled, virtually complete data set) and the sophisticated, improved modeling of interaction graphs. We assume that measurements of the shared external content allow us to approximate statistics on news consumption. The classification into categories allows for an automated high-level understanding of its content. Additionally, hashtags (related to shared external content) provide further semantic understanding. The approach avoids biases due to inaccuracy during the pre-processing. An example here is utilizing NLP techniques for semantic understanding.

In the following, we introduce the various parts of the data engineering process. Therefore, we summarize Twitter functionalities before presenting our data collection strategy and explaining the automated data enrichment (e.g., promotional profile detection and domain categorization). We conclude with a detailed discussion on sophisticated interaction graph modeling.

IV.2.1 Twitter OSN and Functionalities

Twitter offers its users different types of *Tweet-Objects* to generate content on the platform. As of 2019, a user can write a message to his timeline, also known as a status update. The timeline of a user represents a roster of posts. It records activities and makes them visible to followers. The *following functionality* represents the core of the Twitter ecosystem. Based on the accounts a user follows (e.g., news providers, celebrities, and friends), Twitter compiles an overview of current events and activities. This feed displays activities of *followed* others to whom the user has subscribed. Therefore, the system provides a news-feed-like overview tailored to the user's choice.

On Twitter, the *original tweets* is the standard way of posting. *Retweets* represent another type of post, which allows a user to copy a tweet from another user to his timeline. Therefore, it is

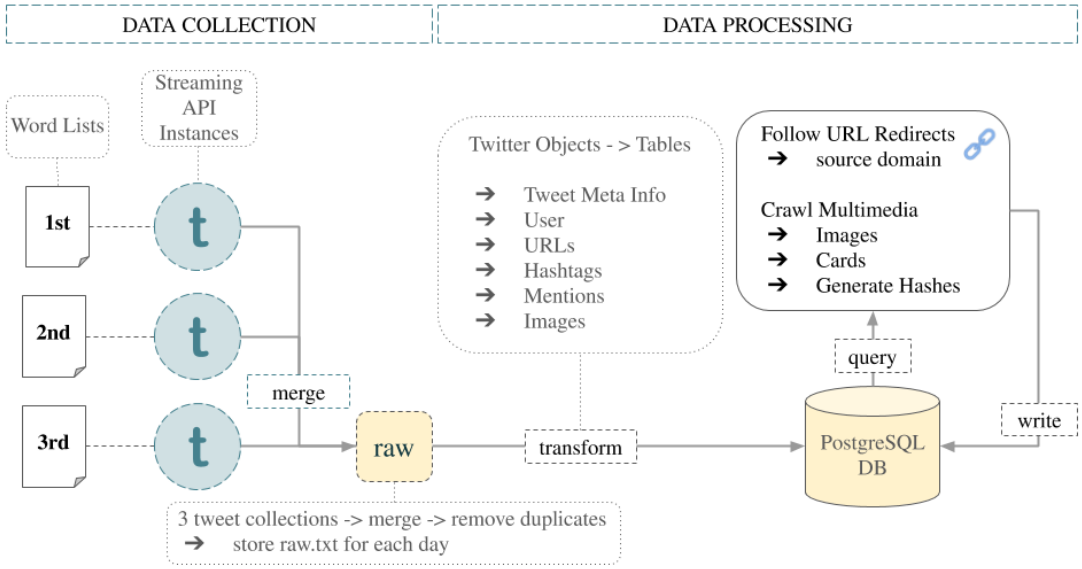


Figure IV.1: Data collection pipeline with three parallel Twitter Streaming API instances; each with a separate stop word list, including 400 frequently used terms in the German language; the output of streams is merged; duplicated entries are dropped; raw Twitter-Objects are extracted from the files and parsed into a PostgreSQL database.

visible to his respective followers and visitors. Users can also *quote* other tweets (except retweets). Thereby, they can re-post a user's message with a comment of their own. Lastly, there are *replies* to comment on any given tweet, except retweets.

Besides textual content, such a *Tweet-Object* can also contain multimedia content (*photos*, *videos*, *animated GIFs*), interactive content (*hashtags*, *user mentions*), places (*geolocation*), or links (*URLs* linking to external sources, which commonly are visualized as *Twitter Cards*). In addition to manually embedded user mentions (@username), Twitter automatically adds *mentions* in front of content that implies an interaction between users (retweets, replies, and quotes). Further, every Tweet-Object has an attribute (*source*) that describes the service used to post the tweet. We extracted the service from each tweet in our data set to estimate their usage. Besides official Twitter clients, there are also third-party services. These services allow accounts to post tweets in an automated manner.

User-objects provide a variety of meta-data. It contains multiple free-text fields (e.g., name, description, URL), statistics about the social links of a user (e.g., follower-, and friend count), and statistics about her activities (e.g., favorites-, and tweet count).

Users can interact with others via direct User Mentions within a tweet or indirectly via connected tweet types, such as retweets, quotes, and replies. Compared to static follows, interactions allow capturing relationship dynamics over time.

IV.2.2 Data Acquisition

The studies depend on a virtually complete snapshot of our target community. Therefore, we propose a comprehensive data collection scheme, i.e. an extension of Scheffler's approach [161] (cmp. Fig. IV.1).

Our evaluation of different collection methods confirmed Scheffler's findings. Geolocation-based filters only capture tiny amounts of German tweets. We hence decided to utilize word lists for our purpose. In contrast to Scheffler, we do not collect-then-filter to remove tweets in other languages, but we leverage the built-in language identification of Twitter. We thus created word filters, encompassing the 1 200 most frequent German words (see Table IV.1). We base

Table IV.1: Text Corpora Ranking

Ranking	Name	Text material	Source	Words Used
1.	Web11	random websites	LCC	400
2.	News15	news websites	LCC	180
3.	Wiki16	wikipedia dumps	LCC	233
4.	Mix11	mixture of sources	LCC	99
5.	Sub16	movie subtitles	OS	242
6.	News17	random websites	LCC	46

our choice on multiple text corpora, provided by the Leipzig Corpora Collection [81] and one corpus of frequently used words from OpenSubtitles.org[1]. The latter encompasses terms that are more prevalent in informal conversations. Twitter enforces a maximum of 400 keywords per instance, so we divided our word filter into 3 different lists and used three individual, parallel data streams. All streams obtained many tweets from 600k to 1.2M on average. Thus our approach does not exceed the rate limitations of 1% (≈ 5M tweets). We drop duplicated entries and merge the stream outputs.
Findings in Morstatter et al. [122] suggest that German tweets are sufficient to capture political debates of the German-speaking population as non-German Tweets are ignored by the community. So, relying on Twitter's language detector, we exclusively capture German tweets. Therefore, we sidestep Twitter's rate limitations and, thereby, avoid down-sampling. While the detector lacks thorough documentation, research showed that, in some cases, it outperforms established alternatives such as Google's Compact Language Detector [140].

We enrich recorded tweets with additional data. Besides the attributes, we further extracted child objects (original tweets, replies, quotes) from collected Tweet-Objects. The latter may entail collecting additional (non-German) Tweet-/ User-Objects. We argue that we need to include users who do not tweet in German but interact with German tweets.

We also developed an algorithm that resolves shortened URLs to identify their source domains. Using McAfee's domain categorization tool TrustedSource[2] we categorized the domains (e.g., News, Lifestyle, Political Opinion, Spam, etc.).

Additionally, we analyzed the distribution of the most popular OSNs on Twitter. We measured this by the amount of external content shared from these platforms. To measure the influence of specific personalities, we developed web crawlers for the most popular platforms, i.e., YouTube, Facebook, and Instagram. By using the YouTube Data API v3[3] and the HTML and JavaScript sources from Facebook and Instagram, we were able to identify YouTube Channels, Facebook Pages, and Instagram profiles shared by users in our corpus.

We also found that Twitter truncates the text of Retweets that exceed the character limit of a message. Because the user name and the prefix *RT* are appended to Retweet-Objects, sometimes the appended URLs were cropped out of the message even though they were visible in the actual tweet. We solved this problem by extracting the URL information from the original tweet sources, adding a total of 4.7 million URLs to our data set.

Additionally, we enrich recorded tweets with additional data. Besides the attributes, we further extracted child objects (original tweets, replies, quotes) from collected Tweet-Objects. The latter may entail collecting additional (non-German) Tweet-/ User-Objects. We argue that we need to include users who do not tweet in German but interact with German tweets.

[1] https://github.com/hermitdave/FrequencyWords/
[2] https://trustedsource.org/
[3] http://youtube.googleapis.com

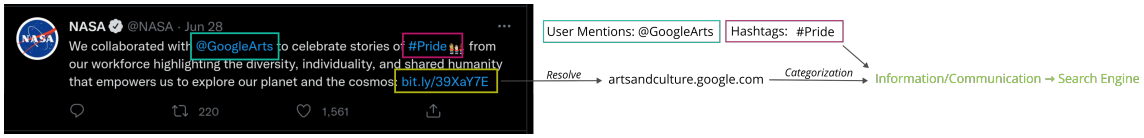


Figure IV.2: Identification of meta-information in large-scale networks; Information at the core of our research: *User Mentions*, **categorized** *Hashtags*, and **categorized** *URLs*.

IV.2.3 Data Enrichment

We obtained a virtually complete snapshot of the GTC, collected during 2 months surrounding the European Parliament Election in 2019 [?]. Still, we have to understand the content to analyze news consumption on Twitter. We base subsequent studies on this understanding. Thus, we need a robust, generalized strategy. In the following, we propose an automated, sophisticated, and comprehensive data enrichment strategy evolving around shared external content (see Fig. IV.2).

We focus on embedded news: shared external links presented as a preview within the social media platforms (for instance, Twitter cards with a headline, thumbnail, and summary on Twitter). Our analysis terminally requires extracting the category and type from the shared tweets as well as additional meta-information (e.g., promotional profile, controversial user), which we perform in the following ways:

IV.2.3.1 Content Understanding

We begin by augmenting tweets with meta information to obtain a high-level understanding of their contents.

Functional Groups: Categorizing Domains We categorize domains leveraging McAfee's TrustedSource[4] (2019) to obtain a comprehensive understanding of the shared external content. We tested different categorization services, and McAfee's TrustedSource successfully identified the highest number of domains. Further, it provides a fine-grained set of 100 hierarchical categories (e.g., News, Lifestyle, Political Opinions, or Spam). McAfee also provides semantic subsets that split the categories into 12 so-called Functional Groups (FGs). Using TrustedSource, we categorized 98.3% of the URL-tweets in our data set. In the remainder, we sort URLs based on their domains into **FGs** and its related **categories** (**FG→category**).

News Group To identify all domains that influence the forming of political opinions, we manually investigated the most-shared websites from every category in our data set. Based on this research, the following set of domain categories, distinguished by the objectivity of reports (from moderate-, over tendentious- to extreme views), comprises the ***News Group***:

- Information/Communication → **General News:** Domains that generate daily news, political opinion sections, and educational content.
  Top 5: spiegel.de, welt.de, bild.de, sueddeutsche.de, and *zeit.de*.

- Society/Education/Religion → **Education/Reference:** Web pages that relate to educational content, for example, classic literature, history, art, and other academic-related content.
  Top 5: de.wikipedia.org, spektrum.de, fridaysforfuture.de, kurierdeswissens.de, danisch.de.

[4] https://trustedsource.org

- Society/Edu./Religion → **Non-Profit/Advocacy/NGO:** Web pages run by charities and or educational groups or campaigns.
  Top 5: change.org, correctiv.org, peta.de, deutschland-kurier.org, mimikama.at.
- Society/Education/Religion → **Government/Military:** Web pages provided by governmental or military organizations, including national branches as well as supranational entities, such as the United Nations or the European Union.
  Top 5: bundestag.de, polizei.bayern.de, auswaertiges-amt.de, bundeswahlleiter.de.
- Society/Education/Religion → **Major Global Religions:** Web pages that provide information about major religions (e.g., Buddhism, Chinese Traditional, Christianity, Hinduism, Islam, Judaism, etc.) and include discussions and non-controversial commentary.
  Top 5: katholisch.de, catholicnewsagency.com, kath.net, vaticannews.va, evangelisch.de.
- Society/Edu./Religion → **Politics/Opinion:** Web pages that cover political parties and opinions on various topics such as political debates.
  Top 5: tichyseinblick.com, jungefreiheit.de, achgut.com, politikstube.com, volksverpetzer.de.
- Lifestyle → **Controversial Opinions:** Web pages that share extreme opinions, which are offensive to political or social sensibilities. Examples include xenophobic, fundamentalist viewpoints, and disinformation campaigns.
  Top 5: journalistenwatch.com, pi-news.net, philosophia-perennis.com, anonymousnews.ru, der-dritte-weg.info.
- Risk/Fraud/Crime → **Discrimination:** Web pages that provide content that explicitly encourages the oppression or discrimination of a specific group of individuals. There are only a few domains that McAfee classifies as discrimination and only a few found in our data.
  Top 5: metapedia.org, theeuroprobe.org, renegadetribune.com, vanguardnewsnetwork.com, nordfront.se.
- Risk/Fraud/Crime → **Historical Revisionism:** Web pages that spread misinformation, or offer divergent interpretations of, significant historical facts (e.g., Holocaust denial).
  Top 5: renegadetribune.com, vho.org, altright.com, dailystormer.name, johndenugent.com, codoh.com.

Accessing OSN Links In addition to external sources referring to news content, we want to gain insight into content included in links to posts on other social platforms. Thus, we developed web crawlers for *YouTube*, *Facebook*, and *Instagram*, the most shared platforms in our data set. By utilizing the *YouTube Data API v3* and the *HTML* and *JavaScript* sources from Facebook and Instagram, we identify corresponding external profiles and their influence on news distribution on Twitter.

Political Hashtags To investigate user discussions about shared news content, we also consider the hashtags they contain. We automatically categorize the corresponding tweets by leveraging the co-occurrence of hashtags with URLs, as classified above. For example, if the hashtag #CDU appears in a tweet that also shares an article from *Spiegel*, we assign the #CDU hashtag to the category General News. This approach allowed us to assign categories to 60% of the hashtags in our data set. Note, however, that a hashtag is assigned to several categories depending on its usage w.r.t. URL-tweets.

IV.2.3.2 Impact Measurement

Besides understanding content, we also want to study its distribution and impact. Therefore, we introduce the following definitions.

Table IV.2: Sample of self-promotional Twitter profiles from Spiegel.

Screen Name	@SPIEGEL_Politik	@joleffers
Name	SPIEGEL ONLINE Politik	Jochen Leffers
Description	Hier twittert das Politik-Ressort von @SPIEGELONLINE. Datenschutz: http://spon.de/afemu	ist bei SPIEGEL ONLINE im einestages-Ressort, twittert hier aber-so-was-von-privat
URL	spiegel.de	spiegel.de
Journalist	✗	✓
Feed	✓	✗

User Engagement To measure how *users engage* with news or external political content, we define *reaction-tweets* in addition to simple tweeting and retweeting. Reaction-tweets contain direct responses (replies and quotes) and retweets. We attribute them to the original tweets they are referencing.
To measure content popularity, we leverage related reactions. We also identify self-promotional profiles and influential users to complement our studies on *engagement*.

Promotion Profiles and Automated Accounts We identify promotional profiles to measure their impact on the distribution of news. We base our automated detection of self-promotional profiles on the guidelines of news agencies such as Reuters or AFP (see Section III.1.4). This process yields two types of promotional profiles: (*i*) journalists and (*ii*) feeds (see Table IV.2). We identify a journalist's profile by checking if it stated a news source in the free-text *URL-field* (e.g. *spiegel.de*) as well as the respective news domain in the *description text* (e.g. *Spiegel*) of their profile. Feeds, we identify using the above and check if their screen name contains the respective news domain (e.g., @spiegelonline).

These feeds often act as the official publisher of articles. They generate automated content and rarely interact with other users. Websites often create multiple feeds solely to disseminate their articles.

This approach has obvious limitations. We cannot automatically detect promotion profiles that do not follow the journalistic guidelines. Therefore, we conducted a manual search for additional promotion profiles for the 30 top content providers. As it did not yield any additional profiles, we are confident that our findings below are representative in this regard.

Besides official automated profiles, malicious bots exist. Concerning these bots, we pursue a different route. In general, bot detection is an unsolved problem. For this reason, scientists resort to heuristics. Often, suspended accounts are interpreted as bots. However, a recent study [111] reports that less than 1% of the suspended accounts were suspected or potential bots. In line with other research, they found that suspended accounts pursued specific polarizing political agendas. Another approach to identify bots is to use tools such as the BotOrNot service. While often used by scientists, research shows how limited this approach is [56, 146]. Twitter adds, that binary judgments have real potential to poison our public discourse.[5] Based on this evidence, we argue that using these heuristics to exclude bots from our study provides no guaranteed benefits while seemingly introducing significant amounts of noise.

Influential Users Researchers proposed different measurements to identify influential accounts on Twitter [155]. In this work, we follow the approach of Kwak et al. [99] by applying the PageRank algorithm to our network. However, we augmented the approach in two ways. Instead of the passive topology metric (i.e., follower-links) – a poor indicator of actual influence

[5] https://blog.twitter.com/en_us/topics/company/2020/bot-or-not

[29] – we utilize interaction activities of users (i.e., retweets, mentions, quotes, and replies) to form our edges. Therefore, our approach relies on similar information as Himelboim et al. [87] (see Section III.1.2). However, we do not rely on an undirected network, assigning symmetric values to interactions between users, but construct a directed graph by calculating scores that indicate how much a user interacts with another user. The resulting weighted PageRank score for each user contributes to a more precise examination of influential nodes in our network.

Furthermore, we expand our research regarding the detection of influential news providers. We determine the influence of news providers by influence measurements. These influence measurements include provider abilities to spread news articles and how many users they can reach. Our approach complements usual methods to measure the popularity profiles in online social networks (e.g., surveys) [89]. In contrast to surveys, a methodology based on sharing and commenting on news provides a more detailed depiction of user behavior. Also, unlike surveys based on self-reports, it is not vulnerable to social desirability bias [175]. PageRank measures the global influence of nodes in a network and, thereby, lends itself to this task.

IV.2.3.3 Understanding Users

So far, our data enrichment strategy allows us to understand the content and distribution of tweets. However, we also want to gain insights into the political attitude of users. Therefore, we augment user information by leveraging their interests.

User Interests Prior work [61] identified user interests based on language processing and augmented this information into the friendship graph. This approach yields a more static assignment and relies on potentially error-prone text extraction. We aim to capture the dynamics of interest more accurately. Therefore, we identify it according to the hyperlinks the users interact with and share to avoid language processing and ambiguities. Using our approach, we leverage the categories of shared domains and hashtags. Briefly, we consider a user who regularly shares or replies to a specific news domain interested in related topics.

Controversial Users The majority of studies classify users based on a political spectrum. Expressing opposing views in the political landscape of the U.S., researchers often label users as either Democrats or Republicans. Since the U.S. has a virtually two-party system, this is a justified and sensible approach. The political landscape in German-speaking countries, however, is more diverse. The political agendas of parties, e.g., tend to overlap. Also, deducing opinions based solely on hashtag information does not distinguish between support and opposition. Therefore, we do not rely on party references in tweets for estimating political affiliation.

The 'Hidden Tribes' study [85] took a more nuanced approach to analyze America's political landscape. Surveying 8 000 Americans, they identified seven groups based on shared beliefs and behaviors. Interestingly, the groups furthest to the right and left of the political spectrum were similar in surprising ways (e.g., color and wealth) and, most importantly, these two groups are the driving force behind the widening of the gulf between the two political factions. Therefore, we distinguish between moderate and extreme users, labeled as non-controversial and controversial.

Based on McAfee's TrustedSource database, our domain categorization approach identifies domains that produce extreme political content and misinformation. While the category *Politics/Opinion* already contains domains with extreme and inflammatory content, categorizing users as controversial based on a shared article of these domains would lead to imprecise labels. Hence, we only include domains with extreme political views that, e.g., deny the Holocaust or encourage the oppression or discrimination of specific groups.

In this context, we assume that retweeting indicates an interest in a topic or even agreement with the sentiment of a message [114]. Therefore, after investigating all of the domains, we posit that people, who support these contents by sharing them in the network and contributing to its distribution, are likely to hold extreme political views. The categorization in our database classify

these domains as Controversial Opinions, Discrimination, and Historical Revisionism. We define a group of **Controversial Users** comprised of users that shared at least one of these URLs. Accordingly, we specify users who share non-controversial content as **Non-Controversial Users**. While we cannot deduce their political affiliations, we assume they manifest less extreme views.

IV.2.4 Extracting Interaction Graphs

Our strategy for data acquisition- and augmentation provides an understanding of tweet content and distribution. However, further information on interactions is necessary to understand news-related dynamics. Therefore, we introduce a sophisticated modeling scheme for interaction graphs.

Two major approaches exist, where one utilizes static follower-relations, and the other leverages dynamic interactions between the users in the network. The *following*-functionality of Twitter offers users a way to keep track of each other's content. By following the activities of others, users express endorsement or even take part in sweepstakes. Since links between users can be one-sided or reciprocal, many users try to expand their influence in the network by offering reciprocal following-relationships to like-minded people. However, follower-relations are insufficient to understand the relationships [188].
Interactions between users can either be found in direct user mentions or indirectly by using connected tweet variants, such as retweets, quotes, and replies. In contrast to static follower-relations, these interactions gather more information about relationship dynamics over time. For example, users frequently retweeting each other's content during a political election seem to share the same political orientation. Retweeting indicates that a user is interested in a topic or even agrees with the sentiment of a message [114]. An extensive (reciprocal) retweeting among users could also show a certain level of trust and appreciation for each other. Quotes allow to retweet content with additional commentary. It can either express opposition or praise the quoted post and its originator. The use of mentions and replies is more prevalent between users having opposing views on a specific topic [40]. While retweets provide no platform for further interaction, quotes and replies allow for comment. Thus, reaction-tweets can start discussions.

Examining conversations within Twitter is a promising strategy to gain insight into the relationship between users. Therefore, we rely on user interactions. In the following, we describe the community detection algorithm.

IV.2.4.1 Interaction Graph

We want to model the exposure of users and their communities to news and categories. For that purpose, we model the users V as the vertices and all Twitter interactions as connecting weighted edges within a graph. We want the weights to represent similarity for later community detection.

The semantics of distance in social graphs depends on the type of interaction. Gadek et al. [73] posit that quantified interaction is a promising metric to estimate a distance between users. We thus quantify interactions between users, combining the four interaction types: retweets, replies, quotes, and user mentions. Each interaction has its semantics. Therefore, we calculate one metric for each interaction type and accumulate the scores to a final edge weight, denoting the distance.

Given a set of N users, $U \triangleq \{u_i\}_{i=1}^{N}$, and the different types of tweets Ω,

$$\Omega \triangleq \{\alpha = \text{original tweet}, \beta = \text{retweet}, \gamma = \text{reply}, \tau = \text{quote}\},$$

we break down the count of all tweets T by their type with $T_\omega\ (u_a, u_b)$ as the total number of tweets of $\omega \in \Omega$ user A posted. When A posts an original tweet, B is the empty set. Otherwise, B is the author of the original tweet. Further, the total number of tweets T user A posted, for

example, is expressed as

$$T_{\Omega}\left(u_A, \cdot\right) = \sum_{\omega}^{\Omega} \sum_{u}^{U} T_{\omega}\left(u_A, u\right).$$

Accordingly, $T_{\Omega\setminus\beta}\left(u_A, \cdot\right)$ represents the total number of tweets of user A that were not retweets.

The Retweet score is based on (i) the number of retweets from tweets of user B shared by user A ($T_{\beta}\left(u_a, u_b\right)$) and (ii) the number of all tweets of users B that are no retweets ($T_{\Omega\setminus\beta}\left(u_b, \cdot\right)$ and defined as

$$S_{\beta}\left(u_a, u_b\right) \triangleq \frac{1}{2}\left(\frac{T_{\beta}\left(u_a, u_b\right)}{T_{\Omega\setminus\beta}\left(u_b, \cdot\right)} + \frac{T_{\beta}\left(u_b, u_a\right)}{T_{\Omega\setminus\beta}\left(u_a, \cdot\right)}\right). \tag{IV.1}$$

Note that we exclude retweeted retweets from the equation because these tweets essentially are retweets of the original tweet, e.g., $A \rightarrow B \rightarrow C$ we capture as $A \rightarrow C$.

The idea behind the metric is that user A retweets a specific number of tweets from user B. The more content users retweet from each other, the closer their distance in the graph. For example, if A retweets every tweet from B, they are closer together in the graph since A shares the same content as B. Therefore, two profiles that were to retweet each other's every tweet, virtually mirroring one another, would represent the closest profile distance.

The corresponding scores for quotes and replies are defined accordingly, as

$$S_{\tau}\left(u_a, u_b\right) \triangleq \frac{1}{2}\left(\frac{T_{\tau}\left(u_a, u_b\right)}{T_{\Omega\setminus\beta}\left(u_b, \cdot\right)} + \frac{T_{\tau}\left(u_b, u_a\right)}{T_{\Omega\setminus\beta}\left(u_a, \cdot\right)}\right), \tag{IV.2}$$

and

$$S_{\gamma}\left(u_a, u_b\right) \triangleq \frac{1}{2}\left(\frac{T_{\gamma}\left(u_a, u_b\right)}{T_{\Omega\setminus\beta}\left(u_b, \cdot\right)} + \frac{T_{\gamma}\left(u_b, u_a\right)}{T_{\Omega\setminus\beta}\left(u_a, \cdot\right)}\right). \tag{IV.3}$$

In contrast to the other interactions, user mentions are not tweet-variants but interactive elements added to tweets. Every tweet potentially contains a User Mention that links a specific user profile. Profiles hence are closer to each other if they have frequent, mutual mentions. For calculating the User Mention metric, we need two statistics: The number of user mentions between respective users and the total number of user mentions per user. We then encode Mentions similar to tweets and define the User Mention Score as follows:

$$S_{M}\left(u_a, u_b\right) \triangleq \frac{1}{2}\left(\frac{T_{M}\left(u_a, u_b\right)}{T_{M}\left(u_b, \cdot\right)} + \frac{T_{M}\left(u_b, u_a\right)}{T_{M}\left(u_a, \cdot\right)}\right). \tag{IV.4}$$

Our final *Interaction Score* combines all interaction metrics mentioned above as

$$S\left(u_a, u_b\right) \triangleq \frac{1}{4} \sum_{\nu}^{\tilde{\Omega}} S_{\nu}(u_a, u_b), \tag{IV.5}$$

with $\tilde{\Omega} \triangleq \{\beta, \gamma, \tau, M\}$, representing the mean value of all scores combined.

S thus encompasses all interactions between users, and we apply it as the final weights to the edges of our graph. The edge weight ranges from 0 to 1. A higher score results in closer distances in the graph, therefore supporting the detection of user groups that frequently interact with each other. For studies on the communities, we perform additional analyses on the separate metric scores (1) - (4).

IV.2.4.2 Community Detection

We define a community as a sub-graph of the network. The literature distinguishes between soft- and hard clustering, where nodes may be associated with several different communities in the former, but only a single one in the latter case. Soft clustering commonly identifies much higher numbers of communities compared to hard clustering. For a fine-grained analysis, soft-clustering hence is intractable on such massive graphs. Further, focusing on the big picture of the

network at hand, the results of a hard clustering approach identify communities that are most densely connected. Due to the large scale of the graph, we chose to apply the Louvain method [15]. It represents a hard-clustering approach based on a greedy algorithm that optimizes modularity. It runs for several iterations on weighted graphs and detects hierarchies of clusters in this process. The hierarchical partitioning of communities allows for a more detailed analysis of the discovered communities.

An issue with modularity optimization is the so-called resolution limit, i.e., there is no guarantee to detect small communities or combinations of small, weakly interconnected communities. The discovered structure does not necessarily correspond to the most pronounced community structure. Fortunate and Barthelemy studied the effects of the resolution limit and questioned the usefulness of modularity in practical applications [69]. However, by utilizing the hierarchical approach of the Louvain method, we can circumvent the issue. Taking successive iterations into account, we can identify small communities in early iterations of the hierarchical process.

IV.2.4.3 Quality Indicators

There is no correct quality assessment strategy to measure the goodness of fit of an identified community structure. However, related work leverages various indicators for this task. Besides the modularity score, the size distribution and the ratio between intra- (communications within a community) and inter-scores (communications between communities) serve as indicators.

Modularity Modularity measures the difference between the original graph and a randomized graph. The value ranges between -1 and 1, where a positive value indicates that the edges within communities exceed the expected connectedness compared to random connectivity. According to Reichardt and Bornholdt [147], the expected maximum modularity for a random network is $Q = 0.15$. Wang [183] compared modularity values across different clustering algorithms. They reported that $Q \geq 0.4$ is a sufficient threshold for detecting meaningful, distinct communities in a graph.

Size Distribution The Interaction Graph represents a network built on the individual activities of people using Twitter. Therefore, the community structure potentially includes both small groups and large communities. A commonly observed indicator of real networks is the heterogeneity of their size distribution. It means most community detection methods find skewed distributions of community sizes [126, 36, 102].

Score-Ratios Based on the average interaction score, we compute the ratios. The desired outcome, a higher score for intra-edges, indicates communities with more densely connected users. On the other hand, a higher value for inter-edges suggests a slight imprecision in the separation of communities.

IV.3 A Study on News Consumption

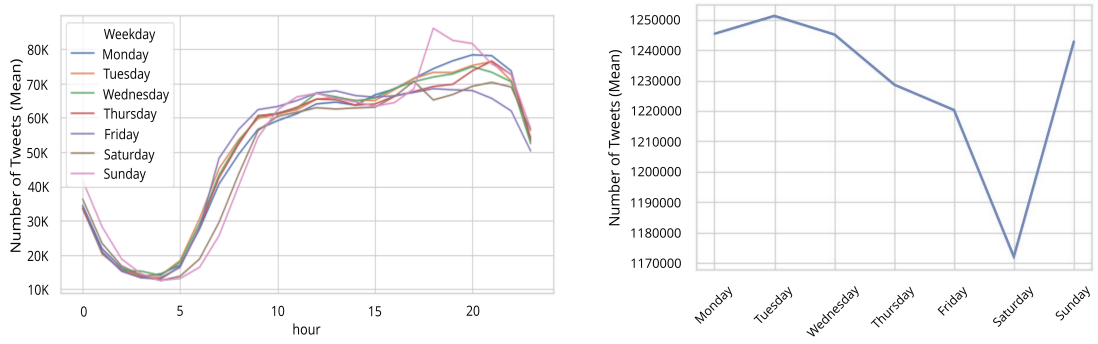
In the following, we present our exhaustive studies on the news consumption characteristics of German-speaking Twitter users. Our goal is to answer questions on controversial news content (see RQs [IV]). However, we require supplementary studies to classify findings regarding controversial users. Therefore, we examine different aspects related to news content.
In the following, we introduce the data set and report relevant statistics. We study news-related content and its overall share within the network leveraging figures on *Functional Groups* (URL categories for automated content understanding, see Section IV.2.3.1), hashtag usage, and external OSN content. We also explore distribution patterns, reach and impact. Then, we complement our studies by analyzing the behavior patterns of news-interested users. Finally,

Table IV.3: Twitter-Objects captured during the data collection process (T = Tweets, U = Users).

Object-Type	Count	T (%)	U (%)
Tweet	77 390 122	-	-
User	6 919 206	-	-
Mention	85 155 158	72	80
URL	18 358 074	23	25
Hashtag	39 197 019	22	29
Multimedia	19 702 261	19	56
Place	1 189 696	2	2

Table IV.4: Distribution of Tweet variants when performing actions.

Action	Tweet Variant (%)		
	Original	Reply	Quote
Retweeting	66.8	21.7	11.5
Replying to	24.7	71.7	3.6
Quoting	76.5	14.5	9.0



(a) Twitter activities throughout every weekday.

(b) Twitter activities over a week.

Figure IV.3: Tweets over time.

we turn to controversial users. Here, we concentrate on the behavior of controversial users in the Twitter community and investigate the existence of isolated controversial user groups.

IV.3.1 The German-Speaking Twitter Community

To obtain a representative sample of the Twitter-sphere of the German-speaking user base, we collected tweets throughout two months – between the 2nd of April and the 2nd of June 2019. The sample contains 77 million tweets and 6.9 million user profiles (ref. Tab. IV.3 for an overview).

IV.3.1.1 Tweet Types

Categorizing these Tweet-Objects by tweet type (i.e., original tweet, retweet, reply, quote) revealed that the most frequent action was retweeting. The majority of activity in our sample was reactive. Retweets account for 38% of all tweets in our corpus and are used to distribute content from other users, Replies for 31%, and original tweets, creating novel content or initiating conversations, account for only 27%. Quotes are rarely used at all (3.7% of the sample). Interestingly, we observed fewer users in our sample using replies (23%) than retweets (64%).

Besides investigating tweet types, we also analyzed their interactions. Table IV.4 shows that most often original content was retweeted (66.8%), followed by replies (21.7%) and quotes (11.5%). Looking at quoting, the distribution is very similar. Regarding replies, however, most of these tweets react to other replies (71.7%).

Table IV.5: Most shared hashtags during busiest days of data collection; here GTNM stands for *Germany's Next Top Model*.

Period	Hashtag	Count	Category	Event
May, 26th – 28th	#Europawahl2019	105 498	politics	European Parliament Election 2019
	#CDU	42 570	pol. party	European Parliament Election 2019
	#AfD	36 038	pol. party	European Parliament Election 2019
	#EUWahl19	25 562	election	European Parliament Election 2019
	#AKK	24 697	politician	European Parliament Election 2019
	#Europawahl	24 185	election	European Parliament Election 2019
	#Rezo	23 498	controversy	European Parliament Election 2019
	#SPD	21 519	pol. party	European Parliament Election 2019
	#Zensur	16 087	controversy	European Parliament Election 2019
	#Grüne, #Grünen	14 290	pol. party	European Parliament Election 2019
	#Sachsen	11 272	election	European Parliament Election 2019
June, 2nd	#NichtOhneMeinKopftuch	124 218	politics	Political campaign
	#Nahles	17 991	politician	Politician resignation
	#SPD	14 698	pol. party	Politician resignation
	#뷔	9 237	music	Campaign by BTS
	#태형	9 234	music	Campaign by BTS
	#태태	9 066	music	Campaign by BTS
May, 18th	#EurovisionSongContest2019	67 188	music	Eurovision Song Contest 2019
	#Eurovision	45 210	music	Eurovision Song Contest 2019
	#Strache	30 218	politician	Ibiza Affair
	#esc2019	23 038	music	Eurovision Song Contest 2019
	#strachevideo	12 911	controversy	Ibiza Affair
May, 22nd – 23rd	#rezo	30 742	controversy	European Parliament Election 2019
	#GNTM	22 980	entertainment	TV Show
	#CDU	20 216	pol. party	European Parliament Election 2019
	#Amthor	19 006	politician	European Parliament Election 2019
	#EuropaWahl2019	15 595	election	European Parliament Election 2019
	#RezoVideo	14 515	controversy	European Parliament Election 2019

IV.3.1.2 Tweets over Time

The volume of daily captured tweets varies from 1M to 1.6M messages with an average of 1.2M. By examining the average collection of tweets by weekdays, we observed that German-speaking Twitter users were more active from Sunday to Tuesday and had a decreasing interest in Twitter from Wednesday to Saturday, with the lowest activity on Saturdays (see Fig. IV.3b). The overall daily usage (see Fig. IV.3a) is moderate in the morning, increases during after-work hours, and drops to its lowest point at night between 1 am and 5 am. At the weekend, Twitter usage naturally starts a few hours later in the morning. The oddly shaped peak on Sunday evenings results from high volumes of tweets during the night of the 2019 European Parliament election. The daily Twitter activities match Central European Time and the working schedule of people from Germany and Austria.

IV.3.1.3 Tweet Content

The content of each tweet can consist of text and additional, interactive content. Table IV.3 shows statistics on the usage of different content types.
The most prominent type is *user mentions* (85M). Since every retweet, reply, and quote contains at least one mention to the originator, these automated user mentions make up for 35%, 29%, and 3%, respectively. Therefore, 33% of the 85M mention-objects (28M) are user mentions, which are added manually into a tweet (@*username*). *URLs* (18M) are the second most prominent objects found in 23% of all tweets. There were 6 667 962 distinct *URLs* shared that originated

Table IV.6: FG and categorical distribution of the 17 million URL-Tweets (multiple assignments per domain possible) corresponding tweets (T), users (U), and form of distribution, such as Original Tweets (OT), Retweets (RT), Replies (RP), and Quotes (QT); statistics on third-party services (Third) are included; FGs and categories with less than a 1% share of all tweets are excluded; FGs containing categories of the ***News Group*** are depicted in brown.

Category	T %	U %	URL %	OT %	RT %	RP %	QT %	Third %
Information/Communication	**47**	**35**	**39**	**46**	**52**	**2**	**1**	**29**
General News	32	21	23	40	57	2	1	23
Blogs/Wiki	10	16	10	52	44	3	1	36
Public Information	2	3	2	68	29	3	1	59
Portal Sites	2	5	2	46	52	2	1	20
Technical/Business Forums	1	2	1	66	31	2	0	52
Forum/Bulletin Boards	1	2	1	64	33	3	0	43
Entertainment/Culture	**15**	**43**	**13**	**44**	**50**	**5**	**1**	**21**
Streaming Media	10	36	8	42	52	6	1	17
Media Sharing	8	33	6	39	54	7	1	14
Entertainment	4	10	4	56	41	2	1	38
Internet Radio/TV	1	1	0	69	29	2	1	53
Art/Culture/Heritage	1	2	0	37	60	2	1	21
Lifestyle	**12**	**23**	**17**	**65**	**34**	**1**	**0**	**55**
Social Networking	7	18	12	69	30	1	0	64
Sports	3	4	3	66	32	1	1	49
Controversial Opinions	1	1	0	29	70	1	0	11
Travel	1	1	1	84	13	3	0	73
Society/Education/Religion	**9**	**13**	**7**	**35**	**59**	**6**	**2**	**17**
Politics/Opinion	3	5	1	27	68	4	2	14
Education/Reference	2	5	2	43	46	9	3	23
Non-Profit/Advocacy/NGO	2	6	2	35	60	4	3	10
Government/Military	1	3	1	30	61	8	4	16
Health	1	1	1	52	41	5	2	35
Purchasing	**8**	**10**	**10**	**75**	**22**	**3**	**1**	**58**
Marketing/Merchandising	3	5	4	73	24	3	1	59
Online Shopping	3	4	3	71	25	3	0	52
Auctions/Classifieds	1	1	1	91	9	1	0	57
Business/Services	**6**	**10**	**8**	**71**	**26**	**2**	**1**	**52**
Business	4	8	5	65	31	3	2	42
Finance/Banking	1	2	2	78	20	2	1	63
Job Search	1	1	1	92	8	0	0	86
Information Technology	**5**	**12**	**7**	**69**	**28**	**3**	**1**	**56**
Internet Services	3	7	4	73	24	2	1	60
Software/Hardware	1	3	2	83	15	2	0	72
Pornography/Nudity	**4**	**5**	**3**	**43**	**56**	**1**	**0**	**43**
Pornography	2	2	2	49	51	0	0	63
Incidental Nudity	2	3	1	35	64	1	0	19
Games/Gambling	**3**	**5**	**2**	**57**	**42**	**1**	**1**	**38**
Games	3	5	2	57	42	1	1	37
Risk/Fraud/Crime	**1**	**1**	**1**	**65**	**33**	**2**	**0**	**77**

from 275 078 different domains. Since 1/4 of all users in our corpus actively shared at least one URL, it seems typical for the German user base to consume and share content from external sources.

Beside these external sources we extracted 19.7 million (5 874 013 distinct) multimedia-objects. The majority of the multimedia contents shared are photos (82%), followed by videos (12%) and animated GIFs (4%), shared by a total of 56% of the users. Note that we can only obtain multimedia content from text tweets, as at least a single word is needed to identify a tweet to be German. Further, 29% of the users in our data set shared 39 million hashtags in 22% of all tweets. However, while we observe more tweets with hashtags than multimedia content, more

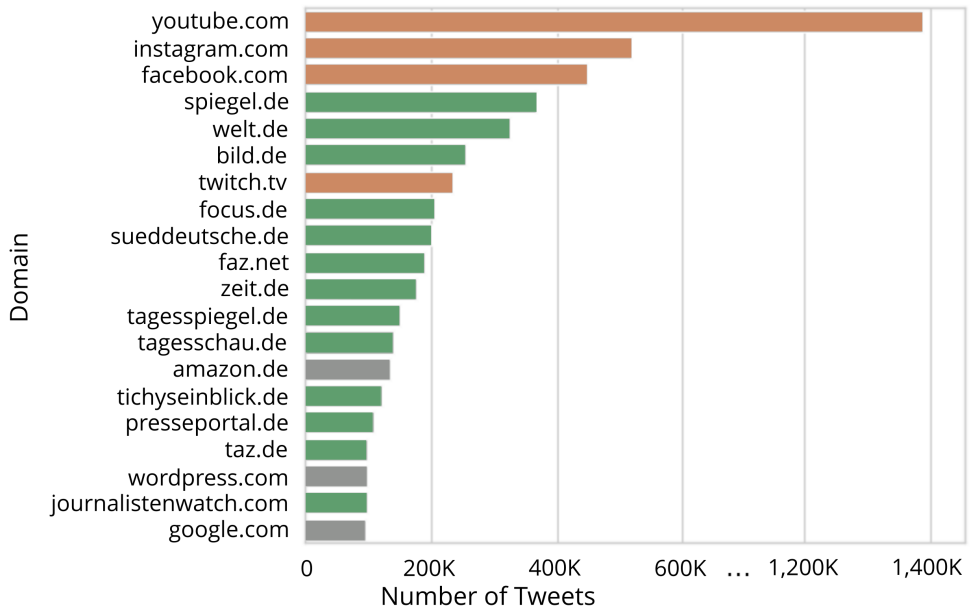


Figure IV.4: Tweet volume of top 20 external sources (orange: OSN, green: news content, gray: other)

users share multimedia content (56%) than hashtags (22%). Users using hashtags are about two times more active on Twitter than users sharing multimedia content, which, to some extent, explains this effect. A feature almost entirely neglected by users in our data set is the submission of geolocation data (*Places*). Only 2% of the users share their location when tweeting.

We want to understand the type of content circulating in the German-speaking Twitter community and measure the share of news-related contributions. Leveraging hashtags, shared external content, and the concept of FGs, we report on the media-consuming behavior of the German Twitter population.

Hashtags In addition to external sources, users produce a high amount of hashtags. By examining popular hashtags shared during unusual high peaks in daily usage, we could identify the related influential events (see Tab. IV.5). We observe a peak in activity at the end of the 2019 European Parliament election. The election and discussion on the results dominate the hashtags during that time.

We also observe that hashtag usage does not reflect election results. The far-right party *Alternative für Deutschland (AfD)*, for instance, is close to leading the hashtag ranking (#AfD), even though it only came in 4^{th} place in the election.

Besides political events, pop-cultural events also caused an increase in daily Twitter volume, e.g., a non-German hashtag referencing the Korean pop band *BTS* or *Germany's Next Topmodel* (#GNTM) and the *Eurovision Song Contest* (#ESC2019). Here, the band *BTS* achieved high music chart rankings over several weeks in Germany, released a single, and, thereby, generated several trending hashtags. Nevertheless, most top hashtags correspond to events within German-speaking countries. These events also dominated the news in Germany during the data collection period.

Functional Groups Table IV.6 details the distribution volumes of the Top 10 FGs and their categories. The most traffic is generated in the *Information/Communication* FG (47% tweets). Large amounts of its content are related to news (General News: 32%) and personal blogs (Blogs/Wiki: 10%), mainly consisting of content from online news media and personalized political websites. Based on the high number of retweets in this group (52%), news and blog content seems to be well-received by the user base. We observed the same popularity of political domains in the

Table IV.7: List of popular services used to distribute social media URLs.

Platform/App	Tweets	Users	Official
YouTube			
Android	33	48	✓
Web Client	26	17	✓
iPhone	17	27	✓
Web App	6	6	✓
IFTTT	3	0	✗
Instagram			
Instagram	62	42	✗
Android	14	25	✓
iPhone	8	17	✓
IFTTT	6	3	✗
Web Client	5	9	✓
Facebook			
Facebook	56	54	✗
Android	16	17	✓
Web Client	10	12	✓
iPhone	8	11	✓
Web App	4	4	✓

Table IV.8: Top shared external OSN content providers broken down by tweet type

Platform	Count	Original	Retweet	Reply
YouTube	1 402 441 (374 414)	38	55	7
Instagram	520 466 (370 510)	72	27	1
Facebook	454 128 (292 316)	67	31	1

Table IV.9: YouTube URLs: Most shared video categories.

Category	Share (%)	User (%)	Video Count
Music	20	34	47 018
News & Politics	19	18	18 484
Gaming	14	10	40 427
People & Blogs	13	20	29 610
Entertainment	12	21	21 982
Education	4	7	10 274
Science & Technology	4	8	8 332
Film & Animation	3	7	7 967
Nonprofits & Activism	2	4	3 868

FG *Society/Education/Religion*, comprised of even more elaborate political content. McAfee's TrustedSource grouped *Controversial Opinions* into the FG *Lifestyle*. The number of retweets in this category is 70%, further supporting the assumption that political content on Twitter is widely distributed and acknowledged.

IV.3.1.4 External Media Usage

A closer look at the 20 most shared external sources (see Fig. IV.4) revealed that 13 link to popular German news providers such as *Spiegel*, *Welt* or *Bild*, as well as to smaller news/opinion blogs, such as *Tichy's Einblick* and *Journalistenwatch*.
However, it turned out that the top domains are external OSNs, led by YouTube, followed by Instagram and Facebook (see Fig. IV.4). These platforms have a significantly higher distribution and more users sharing content from these platforms than any other domain (see Tab. IV.7 and IV.8). They are platforms for a variety of content providers. Therefore, we resolved links to *YouTube*, *Facebook*, and *Instagram* to identify popular *YouTube Channels*, *Facebook Pages*, and *Instagram profiles*.

YouTube content varies from music, gaming, and political opinions to educational content (see Tab. IV.9). We identified single videos accounting for large chunks of the YouTube links on Twitter. For example, a newly released single of a Korean pop band (BTS) or a video of a channel called *Rezo* belonging to a person who was at the center of a political controversy surrounding the 2019 European Parliament election. He published a video with the title "Die Zerstörung der CDU" (Engl.: the destruction of the CDU) that went viral, expressing concern regarding the political course of the *CDU*. In general, there is only a small number of frequently shared content providers from YouTube (see Tab. IV.10). Half of these Channels are related to political topics. Moreover, they show a specific political affiliation. Channels belonging to the right-wing political party AfD are shared more often than channels of any other party. This observation indicates a high activity during their election campaign and shows a trend towards utilizing multimedia content to reach a broader spectrum of users.
Instagram links are mostly apolitical and dominated by profiles from the entertainment industry. Looking at the most shared Facebook profiles (see Tab. IV.10), we observe a relatively small user base that only supports a handful of Facebook pages or profiles, with a low distribution factor. We notice, however, that most Facebook profiles are politically motivated and shifted

Table IV.10: Top external social media profiles (Brown: Political Emphasis)

Provider	Tweets #	Users #	URLs #	Category	Description
YouTube					
책코리아	284 980	282 903	1	music	South Korean singer
Rezo ja lol ey	50 277	30 802	29	political cont.	Rezo controversy
AfD Kompakt TV	14 323	3 413	99	political party	Political party: AfD
Rammstein Official	11 793	8 664	70	music	German band
ProDogRomania e.V.	10 265	617	637	activism	Dog rescue Romania
AfD-Fraktion Bundestag	8 201	2 351	309	political party	Political party: AfD
ibighit	8 094	7 328	36	music	Korean pop band
Joko & Klaas	7 540	6 231	35	entertainment	German entertainers
RT Deutsch	7 138	2 321	1 003	news/politics	Russian news media
Gottfried Curio	6 904	2 330	50	politician	Politician from AfD
Instagram					
@zkdlin	16 845	6 933	202	music	South Korean singer
@oohsehun	9 752	7 102	91	music	South Korean singer
@ksh7909	4 342	3 878	4	music	South Korean singer
@sooyoungchoi	3 579	1 720	9	music	South Korean singer
@daniel.k.here	3 093	3 038	9	music	South Korean singer
@taeyeon_ss	2 666	1 155	22	music	South Korean singer
@saulami1g	2 453	7	2 443	gaming	Gaming/Streaming
@svchicas	2 113	219	2	nudity	Explicit Content/Spam
@stephenathome	1 957	1 953	5	politics	Late night show host
Facebook					
@aliceweidel	16 433	3 867	327	politician	Party member of AfD
@alternativefuerde	11 762	2 930	190	political party	Facebook page of AfD
@Prof.Dr.Joerg.Meuthen	8 149	2 496	85	politician	Party member of AfD
@Bjoern.Hoecke.AfD	3 498	1 523	54	politician	Party member of AfD
@Pazderski.Georg	2 408	910	59	politician	Party member of AfD
@Academia-Para-C...	2 131	222	1	nudity	Explicit Content/Spam
@Deutschland3000	2 012	1 988	15	education	Educational (politics)
@GegenDieAfD	1 740	825	140	activism	Activism against AfD
@GottfriedCurio.AfD	1 694	1 138	10	politician	Party member (AfD)
@app: rossmann.de	1 472	1 079	1	advertisement	Facebook app (shop)

towards the right-wing party AfD. One exception to this rule is a frequently shared page that directly opposes said party (@GegenDieAfD).

Overall, the top content providers from YouTube and Facebook are mostly related to political parties and activism. We observed that the German political party *AfD* was highly active on social media. Regarding shared links from Facebook (6 out of 10) and YouTube (3 out of 10), *AfD*-related topics dominated this content.
In general, concerning the total number of tweets, only *BTS* and *Rezo* were able to generate reach comparable to other popular content providers on Twitter.

IV.3.1.5 Community Structures

We further explore user behavior related to political discussions. Therefore, we study the community structures of the German-speaking Twitter community. Statistics on activity, tweeting behaviors, and communication allow us to analyze group dynamics and -characteristics.
Our studies are based on a holistic interaction graph that stems from 29 098 133 retweets, 24 432 025 replies, 2 907 173 quotes, and 37 979 345 mentions to users within the network. The final graph encompasses 6 809 903 users connected via 32 984 267 edges.

Table IV.11: Hierarchical partitions of the Louvain Method of the unweighted (w^-) and weighted ($^+$) graph; CU depicts the number of communities with controversial users.

Hierarchy Level	Modularity Q w^-	Modularity Q w^+	# of Communities w^+	# of Communities CU
0	0.398	0.7300	872 581	1 734
1	0.496	0.8800	334 656	435
2	0.529	0.9040	274 689	162
3	0.532	0.9056	270 437	117
4	0.532	0.9057	270 184	112
5	0.532	0.9057	270 177	112

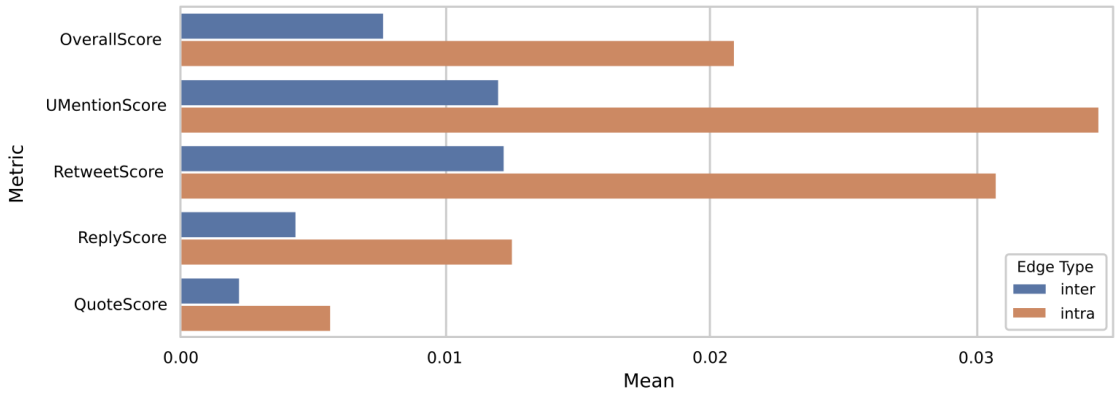


Figure IV.5: Comparison of the interaction metrics based on edges within communities (intra) and edges between communities (inter).

In the list of identified communities, we observe a consistent number of groups with 2 – 3 members (2: 206 054, 3: 38 551). Due to their inactivity, these tiny groups do not get merged into larger communities. We consider these as noise. They interact with one or two other users and do not contribute to conversations and controversies. The remaining community structure of our Twitter corpus encompasses 25 572 communities. In the following, we study the quality of detected communities according to the mentioned quality indicators (see Section IV.2.4.3).

Table IV.11 reports on the modularities of detected communities (weighted- and unweighted). The modularities are $Q = 0.53$ and $Q = 0.91$, respectively. Thereby, they exceed the expected maximum modularity for a random network ($Q = 0.15$), suggesting reliable community structures. The modularity score of 0.91 indicates that the discovered communities describe groups of users having a significantly higher exposure to each other than to users from other groups. Table IV.11 summarizes the 5 iterations of the community detection approach. It depicts 6 partitions of communities with increasing modularity scores. In the first iteration, the unweighted graph barely reaches the acceptance threshold of the modularity score of 0.4. Acceptable modularity scores in the initial aggregations are advantageous. Including the lowest hierarchical level in the analysis circumvents the *resolution limit* of modularity optimization [15]. Lancichinetti and Fortunato confirmed this in a comparative analysis of community detection methods, using the lowest hierarchical level to improve their performance [101]. Therefore, during the analysis, we take every partition into account. Figure IV.5 depicts the expected values of the different interaction metrics (final iteration). Scores of partitions from lower hierarchical levels show similar results but produce inter and intra-edges with slightly higher expected values.
These results suggest that the final partition represents a more generalized overview of the user groups. Partitions in earlier iterations, however, depict smaller communities in more detail.

Table IV.12: News Group Volume: Number of URL- and Reaction-tweets.

Data Set	# of Tweets	%	# of Users	%	# of URLs	%
URL-tweets	17 478 261	100	1 720 752	100	6 667 962	100
News Group	7 247 843	41	454 381	26	1 903 133	29
Reaction-tweets	9 582 682	100	1 222 863	100	1 193 232	100
News Group	5 660 382	59	391 139	32	515 883	43

They allow for a better understanding of these groups within the network.
Finally, we consider the size distribution of the identified groups. We observe the desired skewed distribution in our community structure. 45% of our users belong to the 10 largest communities. The lowest level of the hierarchical partitions shows a flatter distribution with only 13% of the users belonging to the 10 largest communities.

IV.3.2 News Content Analysis

We aim to investigate informational and political content on Twitter and how it influences the German user base. We defined the News Group as a collective term that comprises external domains related to news, political/controversial opinions, and educational content (see Section IV.2.3.1). In the following, we leverage our knowledge on shared content to further our understanding of news-related information.
Table IV.12 shows the volume of tweets, users, and URLs within the News Group. Approximately 41% of all URL-tweets distribute content that belongs to this group. However, only 26% of the users sharing URLs belong to this group. The ratio between URL-tweets (41%) and distinct URLs (29%) in the News Group implies that the average URL is shared 3.81 times. Compared to the average distribution of non-members with a distribution factor of 2.15, the News Group is more active in sharing the content of interest. Therefore, URLs shared on Twitter predominantly link to news-related content.

IV.3.2.1 News Exposure

We established that 25% of the user in our Twitter corpus shared at least one URL-tweet, and 18% of the users replied. With 26% news-related URLs, we have 6.5% of the users actively sharing news content. However, these numbers are only a lower bound on the percentage of users exposed to external content. The challenge is identifying users that read and consume but do not react to URL-tweets. These users only use Twitter as a newsfeed and show no measurable activity towards URLs at all. Although we cannot accurately estimate the number of these users, it is possible to identify their position in the network. The attempt we follow is to find the communities that share URL-tweets. Since communities are densely connected, their users are also more exposed to content from within the community. Therefore, we assume that URL-sharing communities expose their members to external news sources.
Hence, by counting the members of (news-related) URL-sharing communities, we obtain an upper bound on users exposed to external content.
Overall, 23% of the communities share URLs. They encompass 91% of all users. 28% of these, 1 678 communities, share news-related content. They still combine 90% of all users and produce 99% of all tweets. A mean percentage of 62% news-related links indicates that most URLs within these communities relate to news, political, or educational topics. With an average of 72%, the ratio within Reaction-tweets is even higher. In total, 35% of their users produced 34% of the tweets while immersed in news-related discussions. Projected onto the entire data set, 31% of all tweets in our data sample discuss external news content. These results suggest that URLs from

Table IV.13: Categorical usage and distribution of 7M URL-tweets (450k users), 6M reaction-tweets (390k users) within the News Group (URL/reaction); categories are distinguished by political views from: moderate- to extreme.

Category	Tweets	Users	Distribution (%)				
	%	%	OT	RT	RP	QT	Third
General News	77 / 82	79 / 85	41	57 / 41	2 / 49	1 / 22	23 / 3
Politics/Opinion	8 / 8	19 / 20	27	68 / 47	4 / 44	2 / 25	14 / 3
Education/Reference	5 / 4	20 / 15	43	46 / 36	9 / 54	3 / 22	23 / 4
Non-Profit/Adv./NGO	5 / 3	21 / 14	35	60 / 48	4 / 40	3 / 32	10 / 4
Controversial Opinions	3 / 3	2 / 3	29	70 / 68	1 / 25	0 / 14	11 / 1
Government/Military	3 / 3	13 / 14	30	61 / 43	8 / 46	4 / 34	16 / 3
Major Global Religions	1 / 1	3 / 3	42	54 / 35	4 / 54	2 / 20	20 / 3
Discrimination	< 1 / < 1	< 1 / < 1	42	32 / 51	24 / 41	2 / 12	5 / 1
Historical Revisionism	< 1 / < 1	< 1 / < 1	59	19 / 60	22 / 32	< 1 / 34	2 / < 1

external news sources attract more attention than any other content. Informational content has a massive influence on the German-speaking Twitter community.

IV.3.2.2 Engagement

We established that 13 of the 20 most shared external sources (see Fig. IV.4) link to popular German news providers such as *Spiegel*, *Welt* or *Bild*, as well as to smaller news/opinion blogs, such as *Tichy's Einblick* and *Journalistenwatch*. To further our understanding of news distribution within the German-speaking Twitter community, we analyze the subset of shared external sources that link to news-related content. Subsequent analyses are based on the 30 dominant news providers in our data set.

We turn our attention to user engagement. We compare the popularity and reach of content providers within the News Group by analyzing the volume of tweets they generated, the number of users they mobilized, and the number of reactions they prompted.
With the bouquet of actors and news providers, we shed light on the most influential distributors and how users support and react to these diverse options. Regarding controversial content, we further analyze the influence on the general public (on Twitter).

We study user engagement by measuring two factors: reach and impact. We approximate reach by the spread of URLs from a content provider and calculate the impact by the number of reactions to these tweets. In the following, we report on reach and impact w.r.t. two different aspects: (*i*) category, and (*ii*) news provider. We also cover engagement towards links of external OSNs.

Reach The *News Group* comprises 9 categories. These categories allow us to examine the reach w.r.t. different types of news. Table IV.13 gives an overview of the sharing behavior viewed by category. Most URL-tweets originate from moderate domains (General News: 77%). Besides religious- (20%) and educational content (23%), general news is with 23% on the top of the list w.r.t. the distribution via third-party services. Regarding support via retweets, we observe that news sources that offer tendentious to extreme views on politics (i.e., Politics/Opinion and Controversial Opinion) are the most supported domains (retweet factor: 68 – 70%). However, the average distribution of URLs via retweets is consistently high in almost all categories. An exception is URLs propagating extreme political views, i.e., discrimination and historical revisionism. With a retweet factor of 19 – 32%, such content experiences significantly less support via retweets. Interestingly, however, these links seem to be often used within discussions, resulting in a 22 – 24% URL-tweet share via replies (others: 1 – 9%).
The data suggests three different support patterns: (*i*) highly shared and discussed articles, (*ii*) highly distributed articles via retweets (68 – 70%), and (*iii*) articles supported via replies

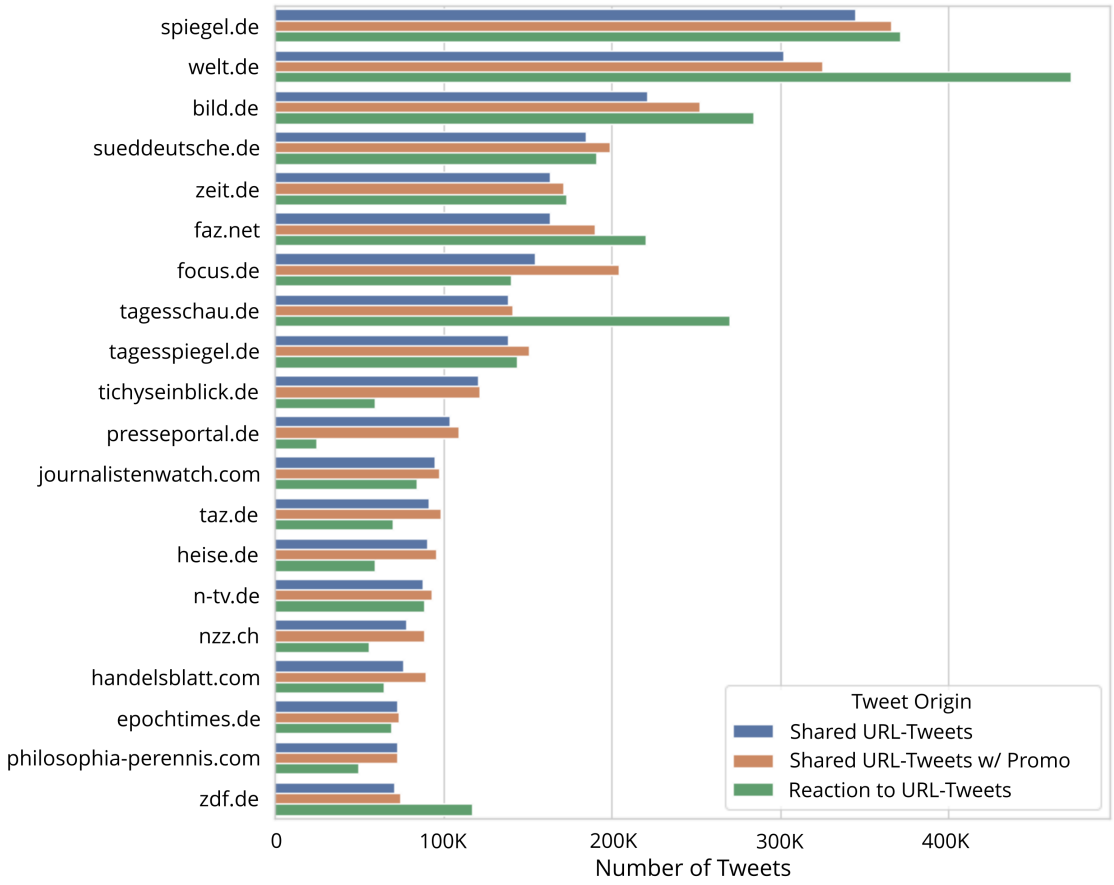


Figure IV.6: Tweet volume of the top 20 news domains, comparing URL-tweets (with and without promotion profiles) and Reaction-tweets.

(22 – 24%) but mainly ignored by the general public (≤ 3% of all users). These support patterns correlate strongly with the subjectivity level of shared content, i.e., *moderate* domains are supported by (*i*), *tendentious* outlets by (*ii*), and *extreme* domains by (*iii*). Overall, we rarely observe extreme external content. A share of < 4% of URL-tweets, actively shared by < 4% of the users, and rarely replied to, extreme content seems to be widely ignored by most Twitter users.

Concentrating on news providers, Figure IV.6 depicts the tweet volume broken down by provider. Further, table IV.14 (right column) shows additional data regarding tweet distributions. The user/tweet ratio reveals two distinct user types. Followers of domains such as *tichyseinblick.de*, *journalistenwatch.de*, or *philosophia-perennis.com* (tendentious to extreme views) have the most active users with a user to tweet ratio of 10.83 (tendentious) and 12.20 (extreme). In comparison, readers of traditional news outlets such as *Spiegel* or *Zeit* only have a ratio of 4.54 and 3.21, respectively (traditional German news providers: 3.91). Also noticeable, Twitter users sharing less moderate outlets retweet more often, with *tichyseinblick.de*, *jungefreiheit.de*, *taz.de*, and *philosophia-perennis.com* as top domains in this category and retweet counts ranging from 81% to 91%. Note that similar to *Bild*, while *taz* is part of moderate news media, in the past several articles with tendentious, disputable content were rebuked by the German Press Council[6]. Traditional media sources, in contrast, reach a broader spectrum of users, but their popularity partially depends on the number of articles they publish.
Users neither share the links from moderate nor tendentious media via replies, indicating that users less often reference such content within discussions. In this category, the governmental

[6] https://de.wikipedia.org/wiki/Die_Tageszeitung#Presseratsr%C3%BCgen

Table IV.14: Statistics on identified promotional profiles (left) and reach (right) of the most shared content providers within the News Group (U = Users, J = Journalists); well-respected traditional German news providers (acc. to [186]) are highlighted in blue; for corresponding URLs see Table IV.19.

Content Prov.	U #	J #	F #	Tweets #	URL %	OT %	RT %	RP %	QT %	3rd %	UTweets #	Users #	U/T #	URLs #	OT %	RT %	RP %	QT %	3^{rd} %
Spiegel	98	68	30	20 850	24	86	13	0.42	0.18	72	344 946	76 028	4.54	47 805	31	66	2	1	12
Welt	72	53	19	23 017	39	88	12	0.23	0.08	90	302 259	47 139	6.41	43 842	26	71	3	0.40	7
Bild	138	59	79	31 059	52	97	3	0.05	0.01	94	221 394	30 547	7.25	24 526	21	78	1	0.21	5
Sueddeutsche	76	38	38	14 231	27	91	8	0.43	0.16	81	184 966	55 572	3.33	20 990	24	74	2	0.43	9
Zeit	80	59	21	7 724	24	88	11	1	0.27	82	163 648	51 052	3.21	21 314	23	73	4	1	9
FAZ	59	34	25	26 322	35	94	6	0.16	0.08	89	163 531	40 784	4.01	32 909	28	70	2	1	8
Focus	20	5	15	49 952	53	81	19	0	0	81	154 124	23 751	6.49	25 050	27	71	2	0.25	11
Tagesschau	10	8	2	2 463	21	98	2	0.04	0.04	95	138 522	39 178	3.54	10 770	27	71	2	0.46	14
Tagesspiegel	77	57	20	12 559	38	45	53	1	2	0	138 274	38 231	3.62	13 590	18	79	2	1	6
Tichys Einblick	2	1	1	1 483	29	88	9	1	1	15	120 412	9 344	12.89	2 315	8	91	1	0.34	3
Presseportal	3	1	2	5 550	10	100	0	0	0	100	103 418	10 401	9.94	52 888	55	44	1	0.48	43
Journalisten. . .	1	0	1	2 151	56	100	0	0	0	100	95 225	6 470	14.72	3 789	22	78	0	0.04	3
taz	43	26	17	6 754	44	91	8	1	0.07	83	91 674	29 327	3.13	8 432	16	82	2	1	7
Heise	26	14	12	4 946	23	76	24	0.22	0.14	93	90 829	25 667	3.54	15 330	38	58	3	2	22
n-tv	12	5	7	5 177	24	95	4	0.19	1	85	88 036	21 254	4.14	19 304	32	66	2	0.39	13
NZZ	82	55	27	10 395	35	58	38	4	0.14	20	78 256	22 146	3.53	14 883	33	64	2	0.39	11
Handelsblatt	66	60	6	13 090	36	74	25	0.34	0.11	60	76 174	22 765	3.35	23 212	38	60	2	0.44	19
Epochtimes	2	1	1	192	2	100	0	0	0	100	73 086	6 898	10.60	8 313	25	74	1	0.13	6
Philosophia p. . .	0	0	0	0	0	0	0	0	0	0	72 926	6 408	11.38	1 457	19	81	1	0.23	2
ZDF	64	35	29	3 877	29	80	14	5	0.48	19	70 492	27 887	2.53	9 914	18	79	2	1	6
Der Standard	33	28	6	8 131	39	96	4	0.33	0.09	89	66 846	15 745	4.25	14 495	37	60	3	1	13
change.org	11	8	4	169	0	51	31	14	4	3	61 586	27 915	2.21	34 674	59	39	3	1	3
Junge Freiheit. . .	2	1	1	672	26	83	17	0	0	1	56 823	7 026	8.09	1 864	14	85	1	0.07	5
Deutschlandf. . .	4	3	2	3 824	31	86	9	5	0.18	4	49 799	19 815	2.51	9 178	25	71	3	1	9
WDR	38	24	16	3 966	36	72	15	9	5	15	44 134	18 345	2.41	5 218	18	80	2	1	7
bundestag.de	4	1	3	71	1	100	0	0	0	8	43 280	19 733	2.19	4 233	17	76	6	6	9
BR	59	40	20	8 937	63	87	10	2	1	47	42 108	16 190	2.60	7 805	23	74	3	1	9
Stern	17	10	7	5 588	29	98	2	0.18	0.07	93	41 611	14 892	2.79	16 231	43	55	2	0.46	27
NDR	37	22	15	3 860	32	64	34	2	1	9	40 851	16 025	2.55	6 851	28	70	2	1	13
RT	10	0	10	2 334	40	91	1	7	0	0	39 262	6 683	5.87	5 189	31	67	2	0.31	7

outlet *bundestag.de* shows the highest reach in this context with distributions via replies and quotes of 6%, each.

We complement information on the reach of news providers by studying promotional profiles, i.e., we consider self-promotional tweets produced by feed-profiles and corresponding journalists. Table IV.14 (left column) details the results of our promotional profile detection process for each of the 30 content providers. For instance, we identified 98 promotional profiles from *Spiegel*, comprised of 68 journalist-profiles and 30 feed-profiles. Over two months, these profiles produced 20 850 tweets, which constitutes a daily average tweet volume of ~ 342 tweets (per account: ~ 3.49). These accounts mainly distributed content via original tweets (86%) and shared them via third-party services (72%). In the process, they actively distributed 27% of the distinct URLs from *spiegel.de* shared during the two months.

In general, we observed that predominantly traditional news media sources, such as *Spiegel*, *Welt*, *Bild* and *FAZ* disseminate their articles via third-party services to extend their reach on Twitter. In particular, *Focus* utilizes a sophisticated feed-profile network that produces a massive volume of tweets. Besides *Bild* with 52%, *Focus* also covers (53%) most of their articles circulating on Twitter, only topped by *Journalistenwatch* (56%) and *BR* (64%). In contrast, non-commercial public news media such as *Tagesschau* and governmental news providers such as *bundestag.de* only generate small amounts of such tweets utilizing significantly more diminutive Feed-networks.

We observed that tendentious to extreme outlets, such as *Tichys Einblick*, *Philosophia perennis*, *Journalistenwatch* and *Epochtimes*, generate much less self-promotional tweets than traditional media. Note, however, that these findings could also be an artifact due to our detection approach, i.e., the corresponding promotional profiles could not adhere to media best practices (see Section III.1.4).

Table IV.15: Reaction-tweets towards the 30 most distributed content providers from the News Group

	Tweets		Users		URLs		Distribution (%)			
Provider	#	%	#	%	#	%	RT	RP	QT	Third
Spiegel	371 725	10	72 881	14	17 884	35	36	56	20	2
Welt	472 260	11	62 868	15	24 592	46	37	56	19	2
Bild	283 968	10	43 158	13	14 790	48	40	52	15	1
Sueddeutsche	190 727	9	53 591	12	9 088	40	37	53	25	3
Zeit	173 291	9	45 878	12	9 024	41	31	60	21	2
FAZ	220 151	10	46 895	13	14 820	40	34	57	23	2
Focus	140 549	7	23 089	15	8 960	19	46	46	15	1
Tagesschau	270 323	7	50 758	10	4 623	43	41	53	18	2
Tagesspiegel	144 017	8	37 048	12	7 176	50	37	53	27	2
Tichys Einblick	59 531	3	10 331	9	983	42	42	48	20	1
Presseportal	24 369	5	9 297	9	5 233	10	41	49	24	5
Journalistenwatch	84 145	8	7 616	10	2 450	64	68	25	14	1
taz	69 589	8	24 225	9	4 547	50	38	51	32	3
Heise	59 171	10	18 783	12	5 024	31	63	30	15	11
n-tv	88 275	11	23 658	13	8 027	39	44	48	16	2
NZZ	56 114	10	18 572	13	5 306	31	39	51	23	2
Handelsblatt	64 816	10	22 968	13	7 456	26	38	50	27	2
Epochtimes	68 923	7	7 542	14	3 144	37	63	30	18	0
Philosophia perennis	49 726	6	7 558	11	721	49	68	26	12	1
ZDF	117 548	8	31 993	8	4 559	44	33	58	23	2
Der Standard	60 219	13	14 356	14	6 770	37	47	44	23	3
change.org	17 808	6	9 697	8	3 277	9	55	38	20	3
Junge Freiheit	45 106	5	8 928	10	752	40	41	50	21	1
Deutschlandfunk	49 611	11	18 625	11	4 290	45	31	57	24	3
WDR	40 341	9	16 336	9	2 525	45	38	50	30	3
bundestag.de	26 863	6	12 334	8	1 135	27	45	44	40	3
BR	43 720	11	16 508	10	4 325	38	44	46	28	3
Stern	46 682	13	16 232	9	5 411	29	32	60	15	2
NDR	36 454	10	15 287	11	2 948	41	40	50	22	3
RT	36 089	12	7 524	11	3 193	58	58	34	18	1

Impact Besides reach, we measure the impact of news categories, -providers, and external OSNs. Table IV.12 shows the number of tweets commenting on or referencing URL-tweets. We found that most reaction-tweets (59%) occurred in the News Group. Furthermore, 43% of the URLs that prompted reactions on Twitter originated from the News Group. The proportion of users (32%) and tweets (59%) indicates a highly active News Group.

We analyzed the distribution of reaction-tweets considering each category of the News Group (see Table IV.13). In contrast to the distribution of URL-tweets, we registered almost no Reaction-tweets from third-party services. Regarding discussions, users commented on moderate content actively (replies + quotes: 71 – 80%), followed by tendentious articles (69%). The more extreme the content, the more "discussions" via retweets (extreme content: > 50%) with an active discussion ratio (replies + quotes) of 53 – 66%. These results suggest that some users continue to disseminate and support controversial opinions, while others are less likely to respond to such content (reply rate: General News 49% vs. Controversial Opinions 25%).
In terms of activity levels, it can again be seen that users discussing extreme content are the most active, with a ratio of tweets per user of 8.93. Users discussing tendentious content are this time more similar to users discussing moderate content, with 5.43 and 4.84 respectively. We find that users are more active in discussing than sharing moderate content 4.84 versus 3.91. The reverse is true for tendentious (5.43 vs. 10.87) and extreme (8.93 vs. 12.20) content. Throughout, we observe many replies and quotes. Therefore, we assume that users heavily

engage in political discussions. Note that the cumulative percentages of retweets, quotes, and replies exceed 100% because retweets may contain nested quotes and replies, which we counted as a retweet of each instance in this case.

Figure IV.6 depicts the URL-tweet volumes and the number of Reaction-tweets concerning each content provider. Traditional news media, such as *Spiegel*, *Welt*, and *FAZ*, trigger a high amount of Reaction-tweets, exceeding the number of tweets that share their articles. It suggests that users actively discuss their content. Of note are the statistics of *Tagesschau*. While showing only moderate amounts of URL-tweet shares, it prompted the 4^{th}-highest number of Reaction-tweets. Not reaching the number of their respective URL-tweets, tendentious and extreme media providers, in contrast, receive fewer reactions.
Table IV.15 gives a more detailed overview of the reactions prompted by the 30 most shared domains in the News Group. For instance, *Spiegel* received 371 725 Reaction-tweets to 10% of tweets that shared a *Spiegel* article. In total, 72 881 users reacted to 17 884 distinct URLs from *Spiegel*, encompassing 35% of all unique *Spiegel* URLs shared on Twitter. Thereby, only 14% of the users that shared a *Spiegel* article received any reaction.
The outlets of *Welt*, *Bild*, *ZDF*, and *Tagesschau* trigger a significantly larger user base discussing their content than the user base that shares it.

Finally, we explore content from external OSNs. We start by further measuring the reach of content by the ratio of shares per unique URL (*S/U*). URLs of the 10 most shared content providers reach an average *S/U* ratio of 12.58. YouTube- (*S/U*: 3.75), Instagram- (*S/U*: 1.40) and Facebook links (*S/U*: 1.55) were shared less often, and, hence, failed to generate reach and impact. A portion of 55% retweets and 7% replies when sharing YouTube-URLs suggests that users distribute YouTube videos to support the content and communicate with other users. On the other hand, the Twitter user base widely ignores Facebook and Instagram URLs. These links get mostly shared via original tweets (Instagram: 72%, Facebook: 67%). While users distribute YouTube links primarily via the official mobile and web clients from Twitter, they share most Facebook and Instagram content via third-party services. We assume that most Facebook and Instagram users share their content passively while actively using Facebook and Instagram clients. They share content on these platforms and forward them to their Twitter profiles to extend their reach. However, the low number of retweets (Instagram: 27%, Facebook: 31%) indicates that this strategy is not very effective. Consequently, we conclude that Facebook and Instagram content is perceived less distinctly than YouTube or other shared media content. Overall, compared to news media sources that distribute their articles directly on Twitter, content providers that operate from other social media networks attract considerably less attention.

So far, our analyses on shared content (Sec. IV.3.2) and the engagement within the News Group (Sec. IV.3.2.2) showed that political content produces the most activity within the German user base. Further, the high number of reaction-tweets to URL-tweets from the News Group suggests keen interest in such content and that users use Twitter as a platform for political discourse. Regarding political content, non-controversial content attracts more users than controversial topics. Traditional news providers distribute most of the news articles (see Fig. IV.4). However, articles from tendentious to extreme outlets generated significantly more retweets per user. Still, the majority of users support moderate views (via retweets). Only a small group (< 4% of the users) supports extreme political views. These users tend to use the reply functionality instead of retweets, implying that they share their opinions within discussions.

IV.3.2.3 Political Hashtags

Next, we analyze hashtag-usage w.r.t. categories to further our understanding. Table IV.16 gives an overview of popular hashtags within news categories and compares URL- and Reaction-tweets. A variety of content providers report on the same events. Therefore, many popular hashtags, such as #AfD, #Europawahl2019, and #Rezo, appear in multiple categories. We also

Table IV.16: Popular Hashtags within the News Group.

Category	Hashtags (URL-Tweets)	Hashtags (Reaction-Tweets)
General News	AfD, SPD, Berlin, CDU, ots, news, Europawahl2020, Merkel, FridaysForFuture, EU, Europawahl, Deutschland, NotreDame, Klimaschutz, Polizei	AfD, SPD, CDU, Europawahl2019, Merkel, FridaysForFuture, EU, Berlin, NieMehrCDU, Deutschland, Europawahl, Rezo, Strache, FPÖ, Klimaschutz
Politics, Opinion	AfD, Europawahl2019, EU, Europawahl, PIRATEN, SPD, Europa, Bundestag, EP2019, CDU, Prüffall, Deutschland, FridaysForFuture, Liebe, ReconquistaInternet	AfD, Europawahl2019, CDU, SPD, NieMehrCDU, PIRATEN, Europawahl, TERREG, EU, Piraten, NieMehrSPD, Uploadfilter, FridaysForFuture, FDP, CSU
Education, Reference	FridaysForFuture, Rezo, Europawahl, Europawahl2019, Klimaschutz, FFFfordert, actnow, Digitalisierung, OSTSTEINBBEKKER音, GrimmsWort, OTD, Berlin, DOYOUNG, KI, Stellenangebot	FFFfordert, actnow, wespoke, OER, GoBlue, kangdaniel, 김종현, WelcomeBackDaniel, 임영민, ABSOLUTE6IX, Marburg, noplanetB, twitterlehrerzimmer, wählengehen, Twitterlehrerzimmer
Non-Profit, Advocacy, NGO	Europawahl2019, Rezo, Zensur, Homöopathie, Meinungsfreiheit, Klimaschutz, Europawahl, Europa, FridaysForFuture, Klimakrise, EU, Uploadfilter, AfD, Berlin, Transsexuellengesetz	Lifeline, Scientists4Future, Florida, unteilbar, Atheisten, AlleGegenRWE, Weimar, OperationSophia, SafePassage, GrandTheftEurope, Economists4Future, Garzweiler, Thema, Upskirting, GamerGate
Controversial Opinions	FFD365, AfD, anonymous, anonymousnews, NotreDame, Merkel, EU, SPD, Antifa, EU19, Berlin, Grüne, CDU, Papst, Migration	anonymous, OliverFlesch, RRG, anonymousnews, MiloYiannopoulos, ramadan, Sperre, Obdachloser, MeinungsfreiheitAuchFürDumme, Schönleinstraße, FFD365, Grosz, einschönesOsterfest, pädophil, homophob
Government, Military	Bundestag, AfD, keinluxus, Klimaschutzgesetz, Feuerwehr, Polizei, Klimaschutz, Europawahl2019, FridaysForFuture, ParentsForFuture, Petition, Fahndung, EU, Braunkohle, Urheberechtsreform	Urheberechtsreform, Feuerwehr, Urheberrechtsreform, BVerfG, Protokollerklärung, Fahndung, KeinAber, copyright, 1919LIVE, SPC_Watch, Vermisstenfahndung, txwx, NRWE, Barcelona, Rossell
Major Global Religions	Kirche, AfD, NotreDame, Frauen, Europawahl, Missbrauch, ZdK, Sternberg, Karwoche, Ostern, PapstFranziskus, Woelki, Europa, GehtWählen, Papst	Karwoche, BenediktXVI, Glaube, Ratzinger, Benedikt, Maria20, kirche, Tagesevangelium, klerikal, Kirchenkrise, Kirchenaustritt, Sexualität, berührende_Erzählung, Gründonnerstag, Freitagsworte
Discrimination	ISIS, falseflag, Churchill, H8Front, H84U, Weltkrieg, PeterPadfield, KJM, RudolfHess, niemehrCDU, niemehrSPD, sydney, kalergiplan, Gunskirchen, IMMIVASION	falseflag, ISIS, Afd, leftwing, Gruene, Gewalt, H8Front, H84U, Ibizagate, Linke, Podcast
Historical Revisionism	Churchill, Weltkrieg, Grundgesetz, PeterPadfield, RudolfHess, GG70, Freimaurerei, Verfassungsschutz, Kommunismus, 1Mai, Kühnert, Verfassung, Nationalsozialismus, Sozialismus	Grundgesetz, GG70, Euro, Verfassung, Verfassungsschutz, Verfassungsrichter

observe that most of the hashtags in General News exhibit a political background. Based on the reaction-tweets prompted by these categories, we observe that users often reply to political news concerning the CDU with hashtags that dissent the party and its coalition partner (e.g., #NieMehrCDU, #NieMehrSPD). Reaction-tweets indicate that users discuss the shared news and use hashtags to express their opinion. While many news articles express less extreme opinions, reaction-tweets express their views more directly. Tweets that distribute controversial news content also receive attention from users with opposing opinions, observable by the usage of hashtags like #MeinungsfreiheitAuchFürDumme (Engl: free speech even for idiots) and #homophob within the reaction-tweets. Users sharing discrimination sources use hashtags opposing the *CDU* and *SPD*. In contrast to other categories, Reaction-tweets contain fewer opposing hashtags. Only a minor fraction of the user base discusses content from discrimination sources without attracting much attention from users opposing their views.

Table IV.17: The 20 largest communities with their distribution of Tweets and additional information; μ and $\tilde{\mu}$ represent mean- and median values, respectively; UwE describes users with edges (to other communities).

																Dominance of **Inter-Edges**				Dominance of **Intra-Edges**			
	Users	Tweets	Per User		OT	RT	RP	QT	3rd	Inter-Edges		UwE	User Deg.			S_β	S_γ	S_τ	S_M	S_β	S_γ	S_τ	S_M
ID	#	#	μ	$\tilde{\mu}$	%	%	%	%	%	#	μ_U	%	μ	$\tilde{\mu}$		%	%	%	%	%	%	%	%
Core	816 677	47 737 955	67	4	24	37	36	8	10	9 436	1.15	34.75	45.59	3		34.07	22.06	3.52	40.35	39.07	21.32	2.18	37.43
109805	498 305	1 403 790	3	1	10	73	12	7	1	1 751	1.69	9.30	4.69	1		59.53	12.74	2.85	24.88	68.36	5.98	1.44	24.21
262453	381 148	1 236 799	3	1	13	67	18	3	3	1 560	1.86	14.76	4.37	1		64.51	16.02	2.58	16.88	87.28	6.36	0.96	5.40
34263	261 370	936 647	4	2	11	72	11	7	1	457	1.31	21.01	4.92	2		72.86	9.18	2.32	15.63	80.22	5.72	2.61	11.44
249774	261 057	440 439	2	1	7	86	4	7	1	1 786	1.88	17.56	2.87	1		91.59	2.39	0.93	5.10	92.28	3.07	1.79	2.87
150111	181 828	350 654	2	1	14	67	8	17	2	587	1.36	18.60	2.88	1		69.47	8.65	4.96	16.92	75.80	5.31	5.97	12.93
142499	179 695	1 376 486	9	1	35	53	10	3	21	1 049	2.86	24.49	6.55	1		56.89	10.13	2.63	30.34	62.58	3.85	1.05	32.52
219563	172 529	5 924 941	38	3	26	18	55	3	3	568	2.44	41.63	17.53	2		36.88	29.08	2.36	31.67	24.12	35.19	1.45	39.24
224357	165 974	342 187	2	1	10	73	10	13	3	596	1.53	15.52	3.36	1		77.25	9.38	2.28	11.09	72.73	10.72	2.95	13.60
257645	160 591	449 073	3	1	24	41	25	15	11	607	2.61	20.17	6.17	2		25.26	14.60	7.88	52.25	22.35	13.41	2.35	61.89
242038	149 786	276 990	2	1	11	76	6	13	1	381	1.42	14.16	2.91	1		85.18	3.54	1.40	9.88	79.55	6.13	3.53	10.79
143859	147 317	345 582	3	1	18	61	14	11	5	429	2.06	23.34	4.61	2		49.48	9.22	3.78	37.51	48.22	9.62	1.92	40.24
225111	135 624	319 732	3	1	13	64	13	18	1	506	1.58	20.73	3.42	1		72.53	7.58	3.28	16.60	63.11	11.74	4.98	20.17
96059	132 954	1 313 323	11	1	29	48	20	3	21	370	1.97	13.48	7.95	2		40.61	26.52	2.54	30.33	40.00	13.57	1.76	44.67
182077	132 654	1 663 479	13	2	25	32	40	6	7	784	3.83	32.75	11.69	2		31.77	22.47	2.19	43.57	29.62	22.00	1.74	46.64
34582	99 098	1 256 112	15	2	33	30	35	3	20	305	2.95	35.73	11.00	2		28.73	28.20	1.61	41.45	28.07	20.26	1.07	50.60
129034	97 746	343 890	4	2	13	72	8	8	2	246	1.94	24.79	5.60	2		69.24	8.66	2.60	19.49	75.62	4.29	2.96	17.13
195201	96 761	197 615	2	1	22	43	26	12	13	239	2.87	8.71	4.41	1		43.52	13.43	4.64	38.41	29.03	19.10	2.18	49.69
231865	81 920	298 713	4	1	14	66	16	6	3	210	2.21	7.28	4.90	1		49.07	22.29	3.69	24.95	64.92	7.86	2.95	24.28
32152	78 812	288 402	4	1	25	31	37	10	9	274	3.59	16.97	4.44	1		36.04	19.02	3.50	41.44	36.97	13.03	3.93	46.06

IV.3.2.4 Communities

We complement our studies, including information on community structures. Table IV.17 provides detailed information on the 20 largest communities. With 62% of all tweets in our Twitter corpus, the largest community substantially determines the content we observe in the German-speaking Twitter community. We reference this community, comprised of 816 677 users, as the German Twitter Core Community (Core).
Further, since we identified many communities, it is sensible to obtain an overview of popular users and hashtags. Although not detailed enough to understand the content discussed within a cluster, it provides a high-level approximation. We report on the 10 largest communities, including the most popular users and hashtags (see Table IV.18). User popularity is measured by PageRank, indicating how well-connected someone is.

Popular Hashtags Table IV.18 shows that most of the Core's top hashtags are related to politics (e.g., #AfD, #Europawahl, etc.) and activism (e.g, #FridaysForFuture). Moreover, users like Rezo (@rezomusik) and A. Kramp-Karrenbauer (@akk) also relate to political events. On the other hand, we found multiple clusters engaged in Asian pop-culture personalities and events. We examined the community 109805 and found numerous users distributing music and entertainment content. By looking at the top users and hashtags, we can assess the general orientation of many communities. For example, the community 262453 shows several gaming-related profiles and hashtags, whereas 150111 and 142499 are related to a mix of lifestyle topics and hobbies. With community 257645, we discovered a political group exclusively discussing non-German content. The central users within this community are related to US politics, including Donald Trump (@realDonaldTrump) and his daughter (@IvankaTrump, @FLOTUS). We also found that mainly international media sources and YouTube videos are distributed in this community. International communities do not necessarily mean that they contain no German users but that their interests include global content.

Popular FGs By measuring the most popular FGs, we identify the dominant domain category of a community (see Fig. IV.7). We report on 5 860 communities containing users sharing URL-tweets. We can observe that most communities evolve around external sources classified as Lifestyle, Information/Communication, and Entertainment/Culture. Spam-like external sources, such as Pornography/Nudity, Risk/Fraud/Crime, and Drugs, only dominate a fraction

Table IV.18: Popular users and Hashtags for the largest communities in our network. We also assigned different subjects to the communities based on our investigation of the respective data.

ID	Top User (PageRank)	Top Hashtags	Subjects
Core	*rezomusik, CDU, akk, janboehm, ChangeGER, SPIEGELONLINE, welt, tagesschau, faznet, DB_Bahn*	AfD, Europawahl2019, CDU, EU, SPD, FridaysForFuture, Berlin, Europawahl, Rezo, NotreDame	German, Politics, News, Election
109805	*BTS_twt, kbs_exclusive, nochuucometru, cookiesketches, jasonaron, H0NEYMYG, mygwithIuv, uchiha_jungkook, AshToTheBashh, Taeholic_V*	방탄소년단, BTS, 뷔, 태형, BBMAsTopSocial, V, BTSV, taehyung, 방탄소년단뷔, 태태	Pop-Culture, Entertainment, Korea, Music
262453	*KPrime86, nusr_ett, Doodlelot, ikuchan_kaoru, pegushi_, _bazztek, Kitsune__Zakuro, modernmodeIs, steelix666_, Crystal_herb*	魔道祖, FFXV, FFBE, 天官福, nsfw, FFXIV, MoDaoZuShi, FGO, LeagueOfLegends, xenoblade2	Entertainment, Gaming
34263	*VXyeontan_, WayV_official, ShunJou, donlaima, petiteyoona, babykihyunie, itfeelquotes, __CONY13, The_LordOfSalem, OfficialMonstaX*	WayV, 威神V, WeiShenV, WINWIN, 董思成, 윈윈, TEN, 李永, ウィンウィン, NCT127	Pop-Culture Asia
249774	*yalibragirl, kellieeastwood, rihanna, __hazelr, ltsesthetics, femmeduart, itsbaddies, Stripx777, onIybaddies, RomeTrumain*	MetGala, WayV, 威神V, WeiShenV, TEN, 李永, The1975, WINWIN, 董思成, 뷔	International, Art, Entertainment, Movies, Culture
150111	*JayeCooley, Baddie___bey, ylreajd, LeanandCuisine, OModutle, RocFoster4, JeiMonroe, kodonism, BakingSodaYola, Thundercat*	METGala, Endgame, themasters, US, MetGala, TBT, 90s, dogs, pets, iPhoneVsHuaweiP30	International, Entertainment, Technology, Lifestyle
142499	*TravelVida, humorandanimals, F1, TraveIPage, archpng, radnature, MercedesAMGF1, BestMovieLine, MercedesBenz, LetsbeAdventure*	art, photography, F1, debk, ebook, travel, porsche, Amazon, porsches, NowPlaying	Traveling, Technology, Hobbies
219563	*MontanaBlack, unge, NVIDIAGeForceDE, NetflixDE, Taddl, GermanLetsPlay, MaxAdlersson, Zombey, Paluten, FF_XIV_DE*	NintendoSwitch, ESC2019, SURO, Europawahl2019, Fortnite, Eurovision, GNTM, Rezo, Splatoon2, Artikel13	German, Entertainment, Politics, Gaming
224357	*KPWKM, rrrrrafla, asmolsushi, dauspozi, rlthingy, youngkessi, AfiqBushido, JKMHQ, latifborgiva, Nsyuhailarahmat*	SedangDiMainkan, WayV, 威神V, WeiShenV, UCL, 李永, TEN, SudirmanCup2019, WINWIN, DrakeCurse	Asian Pop-Culture
257645	*realDonaldTrump, bobcesca_go, marklevinshow, IvankaTrump, CNN, SirajAHashmi, nytimes, Twitter, _rshapiro, FLOTUS*	WWG1WGA, Weathercloud, Germany, Deutschland, MAGA, Election2020, Iran, Bayern, YesWeCan, Allemagne	International, Politics

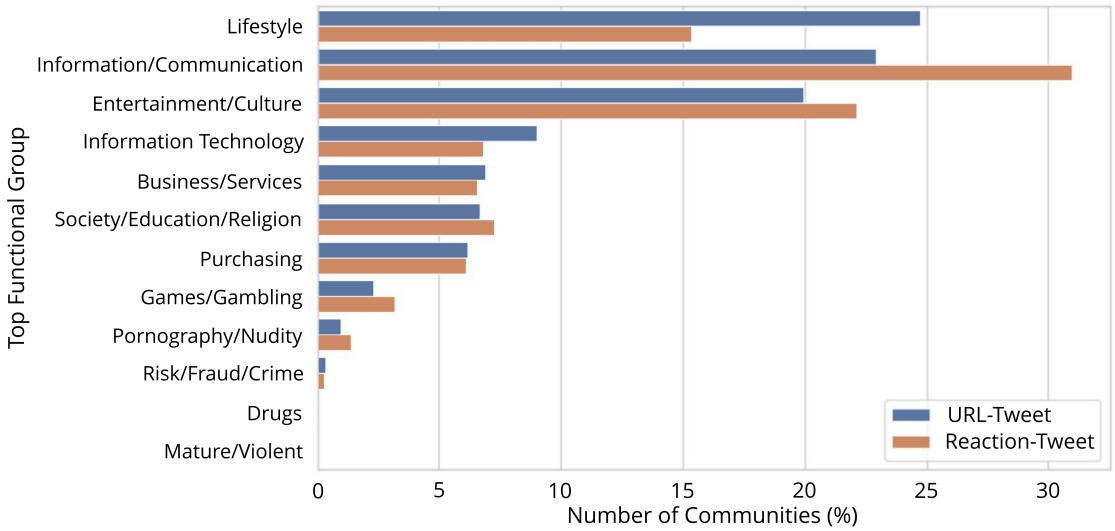


Figure IV.7: The most shared Functional Groups.

of communities. The relatively low percentage of communities that react to Lifestyle sources confirms our assumption that users mainly ignore these tweets, including links from other social

Table IV.19: Media influence in the community structure based on global PageRank percentiles and interconnectedness.

Content Provider Domains	Shared URL-Tweets Com. #	PRank ∅	Deg. ∅	Reaction-Tweets Com. #	PRank ∅	Deg. ∅
spiegel.de	413	0.74	261	318	0.78	289
welt.de	320	0.75	330	300	0.77	300
bild.de	251	0.74	367	244	0.78	341
sueddeutsche.de	350	0.76	299	251	0.81	334
zeit.de	301	0.77	314	226	0.82	359
faz.net	247	0.78	352	194	0.82	359
focus.de	213	0.78	436	149	0.84	468
tagesschau.de	281	0.77	359	271	0.79	338
tagesspiegel.de	223	0.80	386	176	0.84	411
tichyseinblick.de	69	0.79	511	66	0.85	583
presseportal.de	134	0.82	546	82	0.86	590
journalistenwatch.com	53	0.82	585	58	0.85	615
taz.de	181	0.80	398	133	0.86	484
heise.de	226	0.75	314	159	0.81	381
n-tv.de	178	0.81	483	144	0.84	469
nzz.ch	203	0.77	368	140	0.83	453
handelsblatt.com	182	0.82	433	124	0.86	492
epochtimes.de	79	0.81	590	60	0.87	649
philosophia-perennis.com	63	0.81	586	62	0.85	631
zdf.de	223	0.80	415	177	0.84	421
derstandard.at	163	0.79	433	128	0.83	482
change.org	194	0.70	199	126	0.84	455
jungefreiheit.de	58	0.81	580	51	0.85	599
deutschlandfunk.de	139	0.83	492	118	0.88	544
wdr.de	148	0.84	482	114	0.87	534
bundestag.de	156	0.80	422	103	0.87	566
br.de	144	0.83	508	116	0.88	558
stern.de	160	0.81	515	137	0.83	519
ndr.de	151	0.84	518	124	0.88	560
rt.com	105	0.78	543	108	0.83	594

media networks. In contrast, many communities predominantly react to Information/Communication sources, including news and blogs. Our approach of classifying communities based on their URL-sharing behavior is also sensible for filtering communities. For example, we detected numerous communities that only share spam URLs and inappropriate or malicious content. Overall, communities significantly differ in tweeting behavior, interest, and connectedness.

Popular News Categories To extend our approximations on the popularity of news-related topics, we calculated their spread within the community structure. By observing the most shared domain category, we observed that most News Group communities (67%) engage in content from General News sources such as Spiegel, Welt, or FAZ. Additionally, we identified many communities sharing Education/Reference content (17%). These are relatively small in terms of user size. They frequently used sources regarding environmental activism and university-related information sites. Communities sharing Non-Profit/Advocacy/NGO sources (7%) show more ties to political activism and local charitable projects. Communities preferring Political/Opinions content (4%) support opposing views, e.g., sharing left-wing- (e.g., avaaz.org and campact.de) or right-wing sites (e.g., infowars.com). Users discussing governmental/military

Table IV.20: Most influential users of the German Twitter Core Community

Screen Name	Name	PageRank	Degree	Tweets #	3rd %	Label
@rezomusik	Rezo	0.999996	66 244	378	0	contro.
@CDU	CDU Deutschlands	0.999995	75 704	1 710	0	party
@akk	A. Kramp-Karrenbauer	0.999993	63 310	517	9	politician
@ChangeGER	Change.org DE	0.999991	20 939	312	0	activism
@SPIEGELONLINE	SPIEGEL ONLINE	0.999988	50 051	2 994	53	media
@welt	WELT	0.999986	45 410	11 424	99	media
@tagesschau	tagesschau	0.999982	52 230	2 816	87	media
@faznet	FAZ.NET	0.999978	36 622	4 810	78	media
@DB_Bahn	Deutsche Bahn	0.999977	17 508	12 462	100	info
@janboehm	Jan Böhmermann	0.999977	33 045	861	1	contro.
@BILD	BILD	0.999976	35 475	8 161	97	media
@DiePARTEI	Die PARTEI	0.999976	28 915	402	0	party
@KuehniKev	Kevin Kühnert	0.999975	39 372	289	0	politician
@Gronkh	GRONKH	0.999973	19 953	713	39	influencer
@zeitonline	ZEIT ONLINE	0.999972	36 751	3 585	95	media
@SZ	Süddeutsche Zeitung	0.999971	39 390	3 608	92	media
@sebastiankurz	Sebastian Kurz	0.999970	22 778	357	0	contro.
@nicosemsrott	Nico Semsrott	0.999970	33 749	270	0	politician
@spdde	SPD Parteivorstand	0.999969	36 288	4 988	57	party
@iBlali	Vik	0.999965	20 141	520	0	influencer

content and groups discussing religious topics make up 4% and 1%, respectively.

News Providers Next, we refine our understanding of news providers by measuring their influence within the communities. Table IV.19 provides an overview of the tweet distribution within the network for each content provider. We distinguish between URL- and Reaction-tweet statistics. For example, *spiegel.de* is shared within 413 communities and reacted upon within 318. The table also provides the mean PageRank of the user base that shared the tweets. In the case of *spiegel.de*, the mean PageRank of its supporting users is in the 74^{th} percentile. This percentile of the PageRank indicates that the average Spiegel reader is better connected than 74% of the users in our Twitter corpus. Furthermore, the mean degree shows the number of connections Spiegel readers have with other users in the graph. An average user who shares *spiegel.de* articles interacted with 261 other Twitter profiles during the two months of our data collection. We also observe that the users who reacted to *spiegel.de* articles are better connected in the graph (Mean PageRank: 78^{th} percentile; Mean degree: 289) than people who share the articles. We observed the same pattern for most news media and political blogs. This finding indicates that people, who comment on news articles, are overly active on Twitter in general and better connected than users who only share URLs. Interestingly, readers of controversial media, such as journalistenwatch.com, epochtimes.com, and philosophia-perennis.com, are noticeably well connected on average. These readers are, to a great extent, members of large communities. Traditional news providers, such as Spiegel, SZ, and Welt, spread in considerably more communities than newer providers. Therefore, they generated a greater reach with a broader audience. The massive audience also reflects itself in the lower PageRank of traditional media since many casual users are not well-connected in the network.

IV.3.2.5 German Twitter Core Community

Finally, we explore the German Twitter Core Community (Core) (see Section IV.3.2.4) as it substantially determines the content we observe in the German-speaking Twitter community.

Table IV.20 lists the most influential users within the Core by their PageRank percentile. At the top of the list, we find *Rezo*, the YouTube influencer, who was at the center of a political controversy surrounding the 2019 European Parliament election[7]. Besides one of the most discussed news topics revolving around *Rezo*, his profile also is an active, influential part of the Twitter community. By comparing his user degree with the degrees from news provider accounts, such as *Spiegel*, *Welt*, *FAZ*, etc., it is also apparent that more users interacted with him than with already established media profiles. Other political actors of the controversy, such as *Annegret Kramp-Karrenbauer* (@akk), and the official *CDU* profile (@CDU), are also present in the top user list. Users also utilized the Twitter profile of the *change.org* petition website (@ChangeGER) to attract attention to various topics throughout the Rezo controversy. We observe that most popular profiles are related to content contributing to political opinions. Furthermore, we found that news providers and politicians could establish widely popular Twitter profiles that stand at the top of the communities we discovered in our network. Therefore, we conclude that politicians adapted to the digital environment of Twitter and that German Twitter users show massive reactions towards them.

We analyzed the URL distribution in the Core and found that the user base is highly interested in external content from the News Group. 57% of all shared URLs contribute to political discussions. Additionally, 68% of the reaction-tweets relate to content from the News Group. Overall, we observed that 42% of the users in the Core discussed or shared news-related URLs. The observations suggest that the Core mainly discusses political content and consumes news media. This large-scale community indicates that active German Twitter users form a well-connected cluster rather than several smaller groups.

IV.3.3 News Discussion Analysis

Besides information on news content, we are interested in the user behavior related to discussions (distinguished by the type of supported content). We augment our findings with information on the community structures of the German-speaking Twitter community. Statistics on their activity, tweeting behaviors, and communication with other groups allow us to analyze group dynamics and -characteristics.

Tweeting Behavior Based on the tweeting behavior (see Table IV.17), we see that different communities exhibit diverse and partly contrasting tweeting practices. The willingness to communicate varies significantly between them. We identified two generic types we reference as *active-* and *passive groups*. For example, a high percentage of replies and quotes within the Core suggests that its users frequently engage in discussions (see Table IV.17 [left column]). Similar behavior is measurable within all communities of the active group. Groups related to politics show further emphasis on replies. On the other hand, passive communities mainly retweet ($\geq$ 60%). These figures indicate that their user bases mainly distribute content from other users.

Further statistics confirm these characteristic differences in tweeting behavior. Dissecting the interaction metric by its separate scores, we observe that, in most cases, one indicator dominates the others. For example, if user A frequently shares the contents of user B via retweets but only occasionally replies to them, the result will show a high retweet score and a low reply score. We consider the metric with the highest score as the dominant metric of an edge. These dominant metrics give us a more detailed view of the structure of communities. Table IV.17 (right column) gives an overview of the dominant metrics broken down by inter and intra-edges. We observe that active communities show a high percentage of user-to-user links dominated by replies (S_γ) and user mentions (S_M).
An essential trait of a sound community structure is not just isolated groups but users that

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

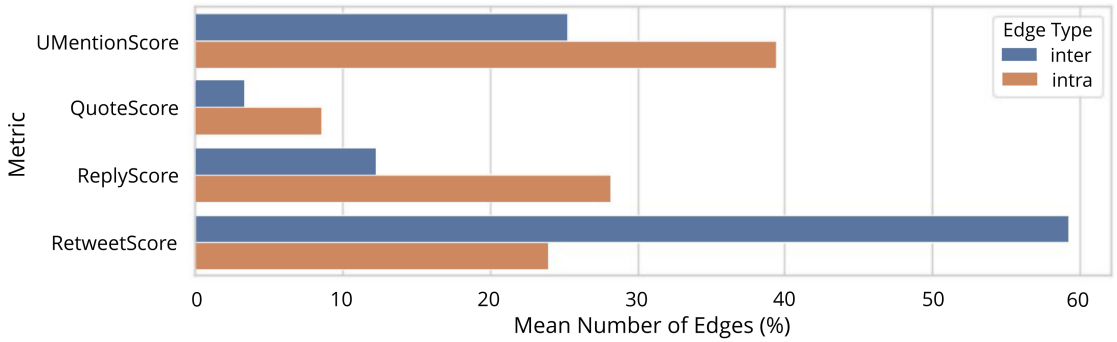


Figure IV.8: Comparison of dominant interaction metrics within the communities.

have ties with users outside their community. Table IV.17 (middle column) details statistics on interconnections. We identify a perceptible difference between users of passive and active communities. Active communities tend to have a high share of inter-connected users, suggesting a more active engagement in conversations than groups with a low amount of connected users. Utilizing the Pearson correlation coefficient, we observe that the share of inter-connected users positively correlates with the ratio of replies ($p = 0.63$) and original tweets ($p = 0.49$) in communities. In contrast, retweet-heavy communities have fewer connections to other groups ($p = -0.6$).
These findings suggest that communities with more inter-edges actively discuss the same content as their adjacent communities. Further, active discussion culture seems to bring users from different communities together.
Finally, we observe that while passive communities related to Asian pop or entertainment and gaming show coherent activeness (mean-median ratio $2-3$), active groups related to German politics exhibit significantly different ratios (≥ 13). This discrepancy between a high mean value of tweets per user to a significantly smaller median indicates a small group of very active users within a community.

Communication Patterns We further observe peculiarities in internal- versus external communications. Figure IV.8 depicts the average shares of dominant metric scores per community, separated by edge type. The most striking difference indicates that retweets are more frequent between communities (Inter: 59%) than within communities (Intra: 24%). Furthermore, user mentions are the predominant type of interaction in communities (S_M (Intra): 39%), whereas retweets come in third after replies (S_γ (Intra): 28%). It suggests that discussions are the main factors for the forming of communities. In contrast, retweets dominate the connections between communities. We believe that users share content discovered outside of their community, supporting it via retweets. On the other hand, it is rare for these users to comment on content from users outside their community via replies or quotes. Nonetheless, they reference users from other communities via user mentions (S_M (inter): 25%), showing a certain level of direct interaction.

IV.3.4 Controversial Users

Up to that point, we concentrated on content-related characteristics and behavior patterns. We complement our studies, by exploring behavior patterns of users sharing controversial, anti-democratic content.
According to Section IV.2.3.3, we label users as either Controversial- or Non-Controversial Users. We detect 11 129 Controversial Users, identify their most influential members, and subsequently survey them to validate the group of Controversial Users. Thereby, we found several profiles from political personalities within the far-right ideological spectrum. These

included well-known activists from the alt-right movement and German politicians from the right-wing party AfD. We also discovered authors from political blogs such as *philosophia-perennis.com*.
Related work [40, 19, 63] reported on political echo chambers from the extreme ends of the political spectrum. A common assumption regarding users within these chambers is that they only inform themselves based on a small and narrow set of information sources. McPherson et al. [113] reported this biased information consumption in social networks, called selective exposure. We focus on potential differences between controversial and non-controversial users and possible echo chambers revolving around anti-democratic content.

IV.3.4.1 User Base

Overall, we have a group of 11 129 users who support anti-democratic content. In the top 30 news providers on Twitter, there are also 3 which spread anti-democratic content. *Epoch times*, supported by 6 900 users, *Journalistenwatch* supported by 6 471 users and *Philosophia perennis* supported by 6 408 users. The group of users supporting at least one of these 3 domains includes 10 694 accounts. 3 555 of which share articles from each of these 3 sources. Furthermore, it can be observed that a large part of these users also share articles from politically right-winged platforms that we do not consider to be extreme (*Tichy's Einblick*, *Junge Freiheit*), e.g. there are still 2 922 users who share articles from each of the 5 platforms (*Epoch Times*, *Philosophia perennis*, *Journalistenwatch*, *Tichy's Einblick* and *Junge Freiheit*.

In terms of responses to URL tweets from these providers, 12 809 users participated in the discussions (*Epoch Times* 7 542, *Philosophia perennis* 7 558, *Journalistenwatch* 7 616). Combined, this results in a group of 15 811 users who share or discuss these articles. Including *Tichy's Einblick* and *Junge Freiheit*, this figure grows to 22 334 with 19 043 users that responded to these URL tweets.

IV.3.4.2 Tweeting Behavior

Based on their PageRank (Mean PageRank: All 0.52 / Non-Controversial 0.64 / Controversial 0.76), Controversial Users are considerably well-connected in the network. The high reach of their tweets suggests that their overall influence is above average within the German Twitter user base. To understand how this influence manifests itself in the network, we study which hashtags they distribute.

Controversial Users produce a large share of political hashtags (see Def. IV.2.3.1). For example, the #AfD hashtag appears in 469 987 tweets shared by 49 883 users. While Controversial Users only make up for 15% (7 239) of these users, they generated 55% of these tweets. We made similar observations for most of the other tweets regarding political hashtags, such as #Merkel, #Islam, and #Flüchtlinge (eng.: refugees), and #Migranten (Engl.: immigrants). Despite their small numbers, Controversial Users, on average, distribute 42% of the tweets that contain political hashtags.

We further analyze the distribution and commenting behavior of Controversial Users. While these users prefer controversial information sources, they also share many articles from traditional news providers. In particular, articles from the large daily newspapers *Welt* and *Bild* (both conservative) attract a considerable attention from Controversial Users. Overall, 92% of the Controversial Users shared a traditional news provider at least once. They also use a wider variety of content providers (⌀ 6) than Non-Controversial Users (⌀ 3) to inform themselves.

So far, we only considered domains of external sources shared by Controversial Users. We extend our studies by exploring their reactions towards domains. We discovered that Controversial Users mainly react to traditional news sources. Articles from *Welt* caught the attention of many users in this group. Primarily, these users commented on political news articles that voice critical opinions about the *AfD* or reports about topics like immigration or

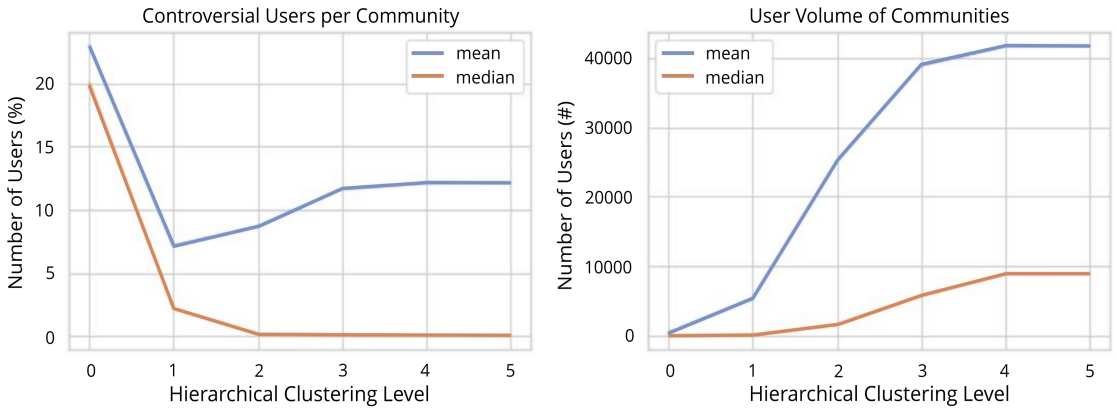


Figure IV.9: Detailed statistics on the distribution of controversial users per iteration of the Louvain Method.

political and climate activism. A closer look at related articles revealed several instances of comments on misinformation content (e.g., faked statistics) in favor of critical views about immigration. Moreover, Controversial Users fiercely commented against news articles that were generally positive about Islam and immigration. In contrast, extreme information sources received virtually no attention from Non-Controversial Users.

Our findings contradict the assumption that users with extreme views tend to form closed systems only reaffirming each other's beliefs. Based on the PageRank scores, the average Controversial User is more active than the average Non-Controversial User. Its average member achieves a higher reach in the German Twitter network than people with moderate political views. Most of their interactions with external political content are responses to Tweets from Non-Controversial Users. They actively engage in many discussions and confront people with opposing views. Non-Controversial Users, on the other hand, tend to remain in their moderate area of political discussions, ignoring external content that supports extreme political ideologies.

IV.3.4.3 Controversial Communities

These findings contradict the notion of echo chambers. With high confidence, we can rule out epistemic bubbles. In general, however, they are no proof of the non-existence of echo chambers. We have to analyze controversial groups further. Do they form a type of echo chamber, which persistently discredits contrary political opinions (see Nguyen [128])?

In 2016, Zick et al. [199] reported a rising social acceptance of right-wing world views in Germany. This trend resulted in people expressing political opinions in public that would have been socially unacceptable before. So, while our results confirm that the average Controversial User does not withdraw into segregated groups reaffirming their political views, the question remains if this observation correlates with the development reported by Zick et al. [199]. We perform further studies to understand the diffusion of Controversial Users within the network. We leverage the hierarchy of our community structure. By examining each iteration of the Louvain method, we trace small user groups before they get merged into larger communities. Focusing on Controversial Users, we study the evolution of their memberships in communities through the hierarchy.

The first iteration places the Controversial Users into 1 734 communities (see Tab. IV.11). With further iterations, the number of communities drastically shrinks. It shows that the algorithm merges these user groups into larger, more general groups. After the last iteration, we observe 112 communities that include at least one Controversial User.

Figure IV.9 gives an overview of the number of Controversial Users per community at different levels of Louvain clustering. We see the share of Controversial Users at the first level is relatively

high. On average, 23% of the users in these communities are Controversial Users, which indicates denser controversial groups. However, we notice that most of these dense controversial groups are considerably small. User clusters at this resolution only reflect the communication between a few people and do not necessarily indicate political echo chambers but rather identify small-scale relationships between a few users. Nevertheless, we also identified several user groups ranging from 20 to 80 members that solely shared extreme domains and hashtags. For example, several small communities mainly supported content from *anonymous.ru*, *pi-news.com*, or *philosophia-perennis.com*. They also showed little to no interest in traditional news media sources.

At successive levels, we register that most communities get merged with others. As a result, the median share of Controversial Users per community decreases. Interestingly, the mean share, after dropping on the second iteration, increases again on successive iterations. This effect continues with each hierarchical level. After the first iteration, most of the dense controversial groups we found at the bottom level are already part of the Core. The increasing mean share reflects the handful of controversial clusters that remain, for example, a network of 63 members that heavily support content from, e.g., *anonymousnews.ru*.

IV.4 Discussion on the State of News Consumption

The experiments provide extensive insights into the news consumption patterns of German Twitter users. In the following, we discuss the state of news consumption within the GTC.

General State Interested in the share of news consumption within the GTC, we measured the exposure to news-related content. Leveraging shared external content, we observed that 25% of all users actively shared external sources. Regarding URL-tweets – posts that contained at least one URL to an external source – 41% were related to news content. Accounting only for unique URLs, we observe that news-related content makes up 29% of these. The discrepancy, a share of 29% unique URLs making up for 41% of all links, further emphasized the popularity of news-related content.

To refine our measurements of news consumption, we incorporated information about the community structures within the network. Studies on shared content within communities revealed that ~23% of the 25 572 identified communities, including 90.6% of all users, shared URL-tweets. ~28% of these communities shared news-related content. These 1 678 news-related communities (6.56% of all communities) produced 99% of all tweets within the network. Overall, 33% of all tweets in our data set supported or discussed news-related content. A similar picture emerged when analyzing the Core, the largest community within the GTC. With 57% of the URLs and 68% of Reaction-tweets related to news, 42% of its users shared or actively discussed such content.

Regarding the impact of news content, especially, the high number of replies w.r.t. tweets sharing and discussing such content suggests that users are willing to discuss or comment on others' content. The ratio of retweets of shared content (news related: 3.81; others: 2.15) reaffirmed this observation. Statistics on Reaction-tweets depicted a similar picture. News-related content and, especially, news providers triggered many Reaction-tweets. These figures suggest a high interest and participation in news consumption and political discussions. Only self-promotional profiles seemed to fall short of their intended goals, yielding minor to no effects concerning user engagement.

Regarding traditional news providers, the most popular outlets successfully established influential accounts within the GTC, with *Spiegel*, *Welt*, *Bild*, *Sueddeutsche*, *Zeit*, and *FAZ* at the top of all content providers within the GTC. Further, related German TV stations (e.g. *ZDF*, *WDR*, *BR*), related content (e.g. *Tagesschau*), and traditional news outlets from Switzerland (*NZZ*) and Austria (*derstandard.at*) were also part of the top 30 content providers. We also identified political blogs (e.g. *Tichyseinblick*, *Journalistenwatch*, *Epochtimes*, *Philosophia-perennis*) with a tendency

towards tendentious to extreme content within the top 30. In this context, we also observed that the German political party *AfD* was highly active on social media. Regarding shared links from Facebook (6 out of 10) and YouTube (3 out of 10), *AfD*-related topics dominated this content. Besides news providers directly operating on Twitter, only *BTS* and *Rezo* were able to generate reach with links from *YouTube*. Here, one event showed the impact the *BTS*-community can have on networks. A single link was shared 284 980 times by 282 903 users. In comparison, the most shared news provider, *Spiegel*, reached a total tweet-count of 344 946 with 47 805 different links. However, *BTS* and *Rezo* are the exception. Regarding other content, links from YouTube, Instagram, and Facebook failed to generate reach and impact. Here, users shared most of the content from Instagram and Facebook via third-party apps, indicating a more passive Twitter use.
Taking community structures into account, influential nodes, besides traditional news providers, were politicians, the political parties *CDU*, *Die PARTEI* and *SPD*, streamers/influencers, and accounts in conjunction with controversial topics. The only two other accounts in the 20 most influential accounts were the activism platform *change.org* and the Twitter account of *Deutsche Bahn*, the national railway company of Germany. The data revealed that young politicians (@KuehniKev, @nicosemsrott) established widely popular Twitter profiles. Further, political controversies seemed to have an immediate and significant impact on the network structure. For example, the controversy surrounding *Rezo* significantly influenced the reach and visibility of not only himself but also of profiles affected by the event (see, e.g., *@akk*, *@CDU*). It demonstrates the impact a political actor can achieve on short notice.
Finally, we observed that activeness is a characteristic of news-related communities. Others, such as entertainment-, gaming-, or lifestyle-related ones, showed a high share of retweets. News-related communities, however, exhibited a more dynamic behavior via replies and quotes. We also observed that large communities include users from the whole political spectrum.

Controversial News-Content To classify observations on controversial news content, we need to look at related work. Bor and Petersen [17] examined the question of why online discussions seem more hostile than their offline counterparts. They examined eight studies using cross-national surveys and behavioral studies and concluded that it is not that people are more hostile online, but that hostile people gain greater visibility online. Additionally, other studies report that emotion-triggering posts [21], especially posts about political opponents are substantially more likely to be shared [145]. Combined, these effect seems to be amplified by the fact that moderate users turn away from discussions because of this hostile behavior [85]. This inevitably leads to the behavior of the few receiving a disproportionate amount of attention. In the U.S., this seems to be compounded by the fact that the most extreme left- and right winged political groups not only attack users with opposing views but are particularly hostile to moderates who espouse their beliefs [85, 75, 123]. "Those who express sympathy for the views of opposing groups may experience backlash from their cohort." (Hawkins et al. [85]) This behavior undermines discussion between people with different opinions and even causes social media to have a detrimental effect on democratic societies [110].

Our work now sheds light on the situation in German-speaking countries. Established news providers dominate news-related content within the GTC. Nonetheless, actors spreading and supporting controversial opinions are also part of the landscape. We observed striking differences in the supporting patterns of different news types. While moderate news was widely shared and discussed, users supported tendentious news sources mainly via retweets. Supporters of extreme political content use the Reply-function (22 – 24%) to inject their content into discussions. Moderate users, however, mostly ignore it.

We extended our research on controversial news content, focusing on providers and users supporting tendentious to extreme sources. Here, two strongly varying pictures emerged. On the one hand, content providers that distribute tendentious to extreme political content play

only minor roles in the network (see Table IV.13). On the other hand, their supporters are highly active and noticeably well connected.
At first glance, this high frequency of interactions with various users contradicts the assumption that people with more extreme political ideologies tend to form echo chambers [20]. However, similar to Zick et al. [199], our findings suggest the existence of a more self-confident form of echo chambers. By dissecting the different layers of the network partition, small coherent groups with selective exposure to extreme political content emerged. Interestingly, these groups became part of larger clusters that predominantly engaged in discussions of moderate political content. Taking their high activity, hashtag usage, content, and shared URLs into account, a picture similar to the findings in Hawkins et al. [85], Bor and Petersen [17] emerged. A minor group of extreme users – formed according to standard echo chambers – spread out to aggressively support their opinions in public. From a group – repellent to opposing views and reassuring in their political positions – these users evolved to highly active members of larger communities.
These users drastically increased their reach and visibility. While popular domains in the Top 30 that share anti-democratic content only have roughly ≈6 500 supporters (combined: 10 694) and an active audience of ≈7 500 users, *Journalistenwatch* (Position in Top30: 12^{th} with 95 225 Tweets supporting and 84 145 Tweets discussing the content), *Epochtimes* (18^{th}: 73 086 / 68 923), and *Philosophia perennis* (19^{th}: 72 926 / 49 726) are among the 30 most shared news providers in the GTC.

For example, 9 088 articles of the renowned newspaper *Süddeutsche Zeitung* were discussed by 53 591 users in 190 727 tweets (*Tweets/audience*: 3.56, *Tweets/article*: 20.99), while only 721 articles of the anti-democratic domain *Philosophia perennis* were discussed by 7 558 users in 49 726 tweets (*T/audience*: 6.58, *T/article*: 68.97). So, 7-times more people discussed articles of the moderate outlet in comparison to articles of the anti-democratic domain. The moderate discussions, however, only generated 3.8x more tweets with only 2x the number of retweets involved in the anti-democratic discussions. Here, 68% of the 'discussions' were in form of retweets.

In summary, we conclude that a similar behavior from users of the extreme ends of the political spectrum as reported in Hawkins et al. [85] can be observed in the GTC. The average controversial user has a high PageRank, i.e., a user's profile connects to other well-connected users within the GTC. Interestingly, however, it seems that these users are largely ignored in discussions by the moderate majority of users in the GTC.

Due to missing data from previous years, we could not study potential developments, e.g., if it correlates to the rise of social acceptance of their opinions [199].

IV.5 Limitations

We reported exhaustive studies on the influence and impact of anti-democratic news content. To cope with a large data set, we formulated several assumptions. Thereby, we accepted certain limitations of our approach.

Content Understanding We based our study on a large data sample. Thereby, we decided to rely on automated methods for content understanding. Studying the content discussed on Twitter via shared external content seems a rough estimate in the first place. However, curated third-party services significantly reduce the complexity of content understanding. Looking at a handful of domains to understand an FG and, thereby, thousands of articles/domains helped us cope with the sheer amount of data. Also, due to the restrictions on tweet length, URLs offer themselves an easy way to share opinions.
Statistics on our data set support and confirm our abstraction approach. Alone 1/3 of all tweets discussed news-content. Including other discussed content, the method allows understanding

large parts of discussed topics.

Data Collection We collected users active during the collection phase. Therefore, we missed all inactive users, even if these users passively consumed content on Twitter. Follower-information would have provided data on passive users (having other drawbacks). The information would also have allowed for more detailed approximations of reach and impact of content. However, concentrating on a virtually complete snapshot of the targeted community made it impossible to collect this information (request limitations).

Promotional Profiles Finally, our crudest approximation regards promotional profiles. To rely on voluntarily provided information from the relevant account carries some risks. Especially the striking difference between traditional news providers (where we identified plenty of promotional profiles) and tendentious to extreme news-content providers (almost none) needs further investigation. To mitigate these uncertainties, one could use shared external content information.

Further research on the detection of automated accounts is also needed. We decided to ignore the noise introduced by bots because recent reports question current detection solutions. According to Majó-Vázquez et al. [111], e.g., accounts we focused on in our research are especially prone to get suspended due to their behavior rather than bot activities.

Controversial Users Controversial users are almost surely correctly labeled. To ensure this, we only labeled users as *controversial* that actively shared an article from extreme, anti-democratic domains. This probably leads to the fact that we have uncertainty in the group of non-controversial users. However, we argue that the imprecision introduced in this way has a smaller impact (arguably none) because it affects by far the larger group of users to a much smaller extent.

CHAPTER V

BEHAVIOR MODELING

Our analysis of news consumption patterns is based on behavioral data. We used the tweets and the URLs they contain to understand user behavior. The data and the domain abstractions gave us a fairly rough reconstruction of behavior. The question is, can we do better?

In our behavioral analysis, we looked at user behavior based on a sequence of events (tweets) of a given user. In addition, we summarized the behavior based on further information (meta-information). Especially the sequence of events, the actions of a user, holds a lot of information about the intention of a user. However, Twitter offers only a limited set of possible actions for a user. In networks such as Facebook, this is quite different. Here, the user has a large number of possible actions and his behavior can be identified much more precisely based on the observable events. From the ML perspective, however, it makes the task more complicated. The algorithm must be able to identify patterns in data of arbitrarily long time series. In the following, we will develop such an algorithm step-by-step. The results of this chapter have been published at the International Conference of Machine Learning (ICML) Workshop on Time-Series [150] and other venues [148, 149].

Understanding user behavior is of great importance in a variety of research areas. Especially in topics related to user experience, deep insights into user patterns are required. The ability to translate a user's behavior into an educated guess of their intent is often the key to a satisfying user experience.

Users exhibit different behaviors in different contexts to satisfy their needs, accomplish a task, etc. [117]. Characteristic behavioral traits can therefore serve as indicators of future behavior, and capturing these traits is important in many application domains:

Content providers on the Internet often rely on repeat visits from users. Their success depends largely on how well they can anticipate user needs by providing the right content at the right time and place. Accurate modeling of user behavior is used to predict user actions and inform design and content decisions. This includes predicting which links a user will click, deciding where to place webpage components, and what content to deliver.

A similar problem arises in emerging areas such as educational research, which aims to provide tailored learning environments and tutoring systems to children and students. Often, it is either undesirable or not possible to create personalized models. And even when such models are available, they suffer from the cold start problem or cannot account for contextual variations in user behavior. Accurately modeling user behavior leads to a precise assessment of a user's competence and enables the selection of next tasks, appropriate feedback, etc.

Recently, user behavior has played an increasing role in security-related areas. Behavioral models are being explored as a replacement for passwords, and smart parts of operating systems are being developed to actively block security-related components, e.g., access to a corporate database when the user checks messages on Facebook. Similarly, security-relevant functions can be blocked by such a system if user behavior deviates from expected behavior, e.g., to prevent a stolen device from being hacked.

V.1 The Clustering Approach

Facilitating a satisfying user experience requires a detailed understanding of user behavior and intentions. The key is to leverage observations of activities, usually the clicks performed on Web pages. A common approach is to transform user sessions into Markov chains and analyze them using mixture models [25, 112, 142, 82]. The idea is to exploit the sequential nature of user behavior and translate user sessions into Markov processes.
However, model selection and interpretability of the results are often limiting factors. Using Expectation-Maximization-based approaches (EM) [47], sessions are grouped to draw inferences about different types of users and their behaviors. Although there is nothing wrong with the general design of these analyses, they often suffer from being parametric approaches and using greedy optimization strategies that can lead to poor local optima. The problem is that the optimal number of clusters is unknown *a priori* and must be determined using heuristics (e.g., Schwarz [165], Akaike [2]) or trial and error. This often leads to repeated parameter estimates on subsets of the data. Moreover, EM-based algorithms potentially converge to local optima, requiring multiple repetitions of the same experiment with random initializations. Given today's data set sizes, the multiplicative consequences of using heuristics with EM-based algorithms quickly become prohibitive.

As a remedy, we propose a nonparametric Bayesian interpretation of this problem. Empirical results on a social network and an electronic textbook show that our approach reliably identifies underlying behavioral patterns and proves more robust than baseline competitors. We conclude by discussing the resulting models and how these findings impact future developments and design decisions.

V.1.1 User Behavior: A Non-parametric Bayesian Interpretation

To model user behavior, we use click-trace data. A click-trace $\boldsymbol{s} = \{\boldsymbol{y}_i\}_{i=1}^{\tau}$ is a sequence of observable events $\boldsymbol{y}$ of a user over time τ in a predefined environment Ω (e.g., the Internet or a particular domain, etc.). An event $\boldsymbol{y}$ is defined by its execution date t and information describing the event. Depending on the task, the description can be as short as the domain the user visited, or it can include more details about sub-domains, categories, locations, and other information. We assume a minimal setting, i.e., events solely described by domain and sub-domain information.
The scenario is as follows: We observe the websites a user visits during a browsing session. In the morning, he visits news-related websites, while later in the day, he searches Google for work-related topics before watching Netflix or Amazon Prime content in the evening. Clearly, the order of websites visited depends on the user's intentions. Thus, these intentions manifest as patterns in the data. We make the following assumption: User behavior is not random but is controlled by hidden processes (the intentions).
Our task is to reverse the process. We assume that a set of processes exists that sufficiently explains the observed events. Further, we assume that the number of processes L is significantly smaller than the number of users N_u. Therefore, given a set of traces, we group them by similarity. The resulting clusters we use for representing the underlying processes.

The mixture model of Markov chains (MMC) Cadez et al. [25] represents a model that aligns with our assumptions. However, the standard approach has serious drawbacks when applied to complex real-world data. Therefore, we propose a natural evolution of the MMC.
We start by reviewing the MMC. After discussing its drawbacks, we introduce its nonparametric Bayesian interpretation.

V.1.1.1 Mixtures of Markov Chains

Consider a user session $s = (y_1, \ldots, y_\tau)$ of length τ over the alphabet $y \in \Omega$. We extend the sequence by an additional start and end symbol. To avoid complicating the notation unnecessarily, we omit subscripts when the context allows. The scenario is as follows: A user traverses the web with a goal in mind. The users intent can be diffuse (e.g., to get information) or concrete (e.g., what year am I living in?).
In terms of our model, this process can be interpreted as follows: The user's intent is not directly observable (hidden state), but can only be guessed from the sequence of web pages visited (observable events).

This interpretation allows us to use an extension of the HMM (see Sec. II.3.5), i.e. the MMC [25], to model the process. The MMC is a mixture model of first-order Markov models. Think of these Markov models as graphs. Thus, the model has an a priori fixed number of graphs whose parameters must be estimated so that the likelihood function is optimized. The MMC models an intent as a graph with parameters specific to that intent, i.e. the next website is selected based on the hidden state and the currently visited website. If the user's intent changes, the user switches to a different graph. By introducing start and end symbols, we can capture preferred entry- and exit points. Note that for this contribution, we make the rather hard assumption that a sequence belongs to exactly one component. The corresponding mixture model with L components has the form:

$$p(s|\Theta) = \sum_{l=1}^{L} p(z_l|\Theta)\, p(s|z_l, \Theta)\,. \tag{V.1}$$

It has two components, the prior distribution (mixture components weights) $p(z_l|\Theta)$ and the likelihood (mixture component description) $p(s|z_l, \Theta)$. $\Theta = \{\theta_1, \ldots, \theta_L\}$ denotes the parameters of the model.

The prior distribution, the marginal distribution of the l^{th} component, tells us how likely it is to observe events that follow the pattern of the l^{th} component. Mixture components that represent the patterns as a graph are then described using the parameter set $\theta_l = \left\{\theta_l^I, \theta_l^T\right\}$:

$$p(s|z_l, \theta_l) = p\left(y_1|\theta_l^I\right) \prod_{i=2}^{\tau} p\left(y_i|y_{i-1}, \theta_l^T\right). \tag{V.2}$$

The parameters Θ are estimated by a maximum likelihood approach [25].
Using Bayes' rule and a trained model, we can assign new sequences to mixture components as follows:

$$p(z_l|s, \Theta) \propto p(z_l|\Theta)\, p(s|z_l, \Theta)\,. \tag{V.3}$$

While EM-based approaches provide interpretable results that can be computed efficiently, they also have major drawbacks. First, the actual number of components is generally unknown, so L is a parameter that must be adjusted during model selection. Second, the greedy inference by EM-based approaches can converge to local optima. This not only renders a single solution unquantifiable but also necessitates repetitions of the same experiment (e.g., using different initializations). Combining the two arguments leads to complex experiments and quickly becomes tedious.

V.1.1.2 Infinite Mixtures of Markov Chains

In the following, we describe our contribution that addresses both limitations of MMC. Based on the same building blocks as the MMC (graphs representing intentions), we use a nonparametric Bayesian model approach to address the issues around model selection and training. Being a nonparametric Bayesian interpretation of the mixture of Markov chains, the number of

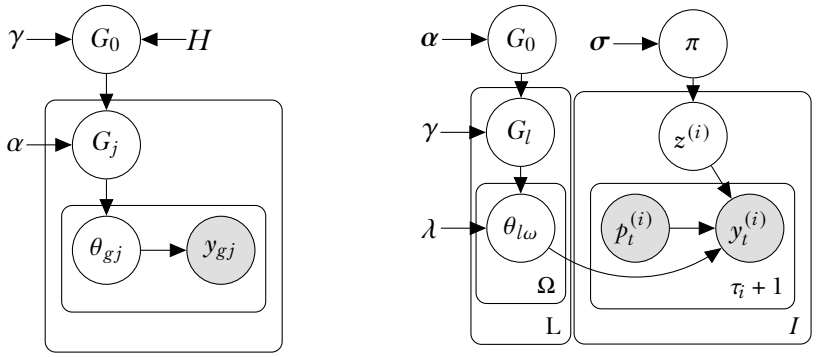


Figure V.1: (left) Graphical model of an HDP mixture model; (right) graphical model of the proposed iMMC; $\boldsymbol{\pi}$ denotes the mixture weights; G_0, G_l, and θ model the mixture components; G_0 denotes the base distribution of the subordinate Dirs G_l, and $\theta_{l\omega}$ represents the transition distribution in mixture component l conditioned on $p_t = \omega$; α, γ, λ, and σ represent the hyperparameters of the model; Ω is the set of events; I is the cardinality of the set of input sequences with τ_i as the length of the corresponding sequence; $i \in I$ and $t \in \{0, \ldots, \tau_i + 1\}$; white- and gray nodes represent hidden states (z) and observed states (p and y), respectively.

components is adjusted in a data-driven way during the optimization. Optimization itself is then performed by a Gibbs sampler that does not share the greedy nature of EM-based methods.

The model is built on two previously covered concepts (see Section II.3.6.4), the Dirichlet distribution and the finite-dimensional hierarchical Dirichlet process [173, 91]. The Dirichlet distribution is used to substitute the prior distribution of the MMC (weights of its components) to allow for an adaptive interpretation. A mixture component is modeled by an HDP.
We make use of a computationally efficient approximation to the hierarchical Dirichlet processes (HDP) [173], known as the degree L weak limit approximation [91]. The limiter L denotes the maximum cardinality of the approximated distribution. The approach encourages the learning of models with a state space of less than L components while allowing for the creation of new ones (up to L). It can be shown that this approximation converges to the original HDP as $L \to \infty$ and provides a common solution to efficient Bayesian nonparametrics [98].

Graphical Model Our model consists of a flexible set of components (intents), which may however grow to at most L components. It is represented by a graph modeled by an HDP with its base distribution G_0 and a child distribution G_l. Recursively, each subordinate distribution G_l serves as a base distribution for the transition graph of each component, i.e., the set of $\theta_{l\omega}$ for each element in the observation space, $\omega \in \Omega$. Thus, G_l models the state distribution within a component. As before, we distinguish between observations x and latent variables z that assign sequences to components.

Note that the distributions related to the initial state (θ^I) and exit state (θ^E) are combined with the standard transition distribution (θ^T) in θ. Figure V.1 (right) shows the graphical model, and the generative process is given by

$$z|\pi \sim \pi \qquad y_t|z, p_t \sim \theta_{zp_t} \qquad t \in \{1, \ldots, \tau_i + 1\}\,, \tag{V.4}$$

where p_i denotes the observation of the previous timestep.

Inference To estimate the parameters, we use a two-step sampling algorithm consisting of alternating sequence assignments and parameter updates. In the assignment step, we assign observations to components In the maximization step, we use the updated assignments to adjust the prior distributions accordingly. These two steps are then repeated until convergence.

At the start, realizations of the uninformative prior distributions π, G_0, G_l and $\theta_{l\omega}$ are obtained

by

$$\begin{aligned} \pi|\boldsymbol{\sigma} &\sim \mathrm{Dir}(\boldsymbol{\sigma}) \\ G_0|\boldsymbol{\alpha} &\sim \mathrm{Dir}(\boldsymbol{\alpha}) \\ G_l|\gamma, G_0 &\sim \mathrm{Dir}(\gamma G_0) \\ \theta_{l\omega}|\lambda, G_l &\sim \mathrm{Dir}(\alpha G_l), \end{aligned} \tag{V.5}$$

with $|\Omega|$ as the size of the observation space.

V.1.1.3 Assignment Step

Considering the prior distributions, we calculate the probability of a sequence s as

$$p(\mathbf{s}|\Theta) = \sum_{l=1}^{L} p(z=l|\Theta) \prod_{t=1}^{\tau+1} p(y_t|p_t, z=l, \Theta) = \sum_{l=1}^{L} \pi(l) \prod_{t=1}^{\tau+1} \theta_{lp_t}(y_t), \tag{V.6}$$

where y_0 and $y_{\tau+1}$ represent the artificial boundary nodes. The distribution of a single subordinate process is

$$p(y|z=l,\Theta) \propto \pi(l) \prod_{t=1}^{\tau+1} \theta_{lp_t}(y_t). \tag{V.7}$$

Therefore, the assignments are straightforward,

$$z \sim \mathrm{Mu}\left(\sum_{l\in L} p(y|z=l,\Theta)\delta_l\right), \tag{V.8}$$

where δ_l represents the point mass at position l.

V.1.1.4 Update Step

Sufficient statistics are collected during the assignment step. We have tracked the frequency of the different components and the transitions within the graphs. Therefore, b_l represents the number of observations assigned to component l. $d_{l,\omega}$ captures the number of observations $y=\omega$ associated with component l. Finally, o_{l,ω_n,ω_m} records the number of transitions from ω_n to ω_m in l.
Using the statistics and the assignments, we can update the prior distributions:

$$\begin{aligned} \pi|\boldsymbol{\sigma} &\sim \mathrm{Dir}(\boldsymbol{\sigma} + \boldsymbol{b}) \\ G_0|\boldsymbol{\alpha} &\sim \mathrm{Dir}\left(\boldsymbol{\alpha} + \sum_{l=1}^{L} \boldsymbol{d}_{l\cdot}\right) \\ G_l|\gamma &\sim \mathrm{Dir}(\gamma G_0 + \boldsymbol{d}_{l\cdot}) \\ \theta_{l\omega}|\lambda, G_l &\sim \mathrm{Dir}\left(\lambda G_l + \boldsymbol{o}_{l,\omega,\cdot}\right), \end{aligned} \tag{V.9}$$

A summary of the inference process is given in Listing 3. It should be noted that the Gibbs sampler, although similar to the classic EM approaches, is a stochastic process and not a greedy optimization. Therefore, it can be shown that the sampler converges to the global optimum under certain conditions [156].

V.1.2 Experiments: Clustering

We start by evaluating the effectiveness of our model. Therefore, we build a controlled artificial environment. This allows us to accurately evaluate the clustering performance of our approach

Algorithm 3 Blocked Gibbs sampler for iMMC

Given the hyperparameters $\boldsymbol{\sigma}$, $\boldsymbol{\alpha}$, γ, λ

(i) Initialize prior distributions according to Eq. V.5

Until convergence **do**:

(ii) Assignment Step

→ Obtain a realization of z according to Eq. V.8

→ Update auxiliary variables as follows:

- $b_{z=l} \equiv$ # observations assigned to component l
- $d_{z=l,y_l=\omega} \equiv$ # observations ω assigned to l
- $o_{z=l,p_l=\omega_n,y_l=\omega_m} \equiv$ # transitions from ω_n to ω_m in l

(iii) Re-sample prior distributions (Eq. V.9)

(iv) Build a final model from multiple sample sets of the parameters

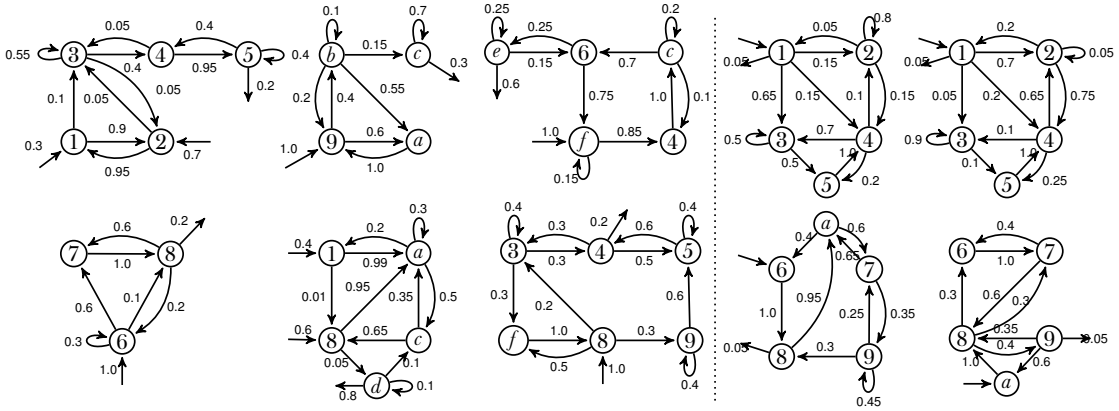


Figure V.2: Generative processes of scenario II (left) and scenario III (right); states are indexed by hexadecimal numbers (1-f).

(unsupervised setting with perfect gound-truth knowledge). Thereafter, we focus on the interpretability of the resulting groups (graphs representing intents). We extract patterns of users surfing a social network website. We conclude with an analysis of the behavior in an electronic textbook. Using the iMMC, we show that the obtained patterns correlate with the success of the students. The findings suggest that behavior patterns in learning environments hold sensitive personal information.

V.1.2.1 Synthetic data

In this section, we compare the clustering performance of the iMMC to the traditional MMC. Additionally, we also pick the latent Dirichlet allocation (LDA) [14] as a baseline to assess the importance of the sequential information contained in the observations. LDA only makes use of the frequency count of events within a sequence.

We build three artificial web surfing scenarios. A scenario constitutes a set of behavior patterns modeled as graphs. It represents the causal reason for an observed sequence of events: graphs thus serve as proxies for user intent. The shared observation space is comprised of all events that are associated with one or more patterns of the scenario.
The scenarios are designed to pose different levels of complexity for a clustering approach:

- **Scenario I (easy):** Disjoint observation spaces.
  E.g., *cooking* and *driving a car*, sequences of actions without any overlapping actions

Table V.1: Error rates for the synthetic clustering tasks; each data set consists of 10*k*, 100*k*, and 250*k* data points (small, medium, large).

	Scenario I			**Scenario II**			**Scenario III**		
	Small	Medium	Large	Small	Medium	Large	Small	Medium	Large
LDA	20.92%	28.14%	28.62%	14.69%	12.09%	20.20%	27.95%	29.54%	29.06%
MMC	19.60%	9.90%	5.13%	5.94%	6.78%	4.77%	14.26%	20.36%	8.47%
iMMC	0.14%	2.23%	0.26%	0.00%	0.54%	2.78%	8.61%	5.82%	5.15%

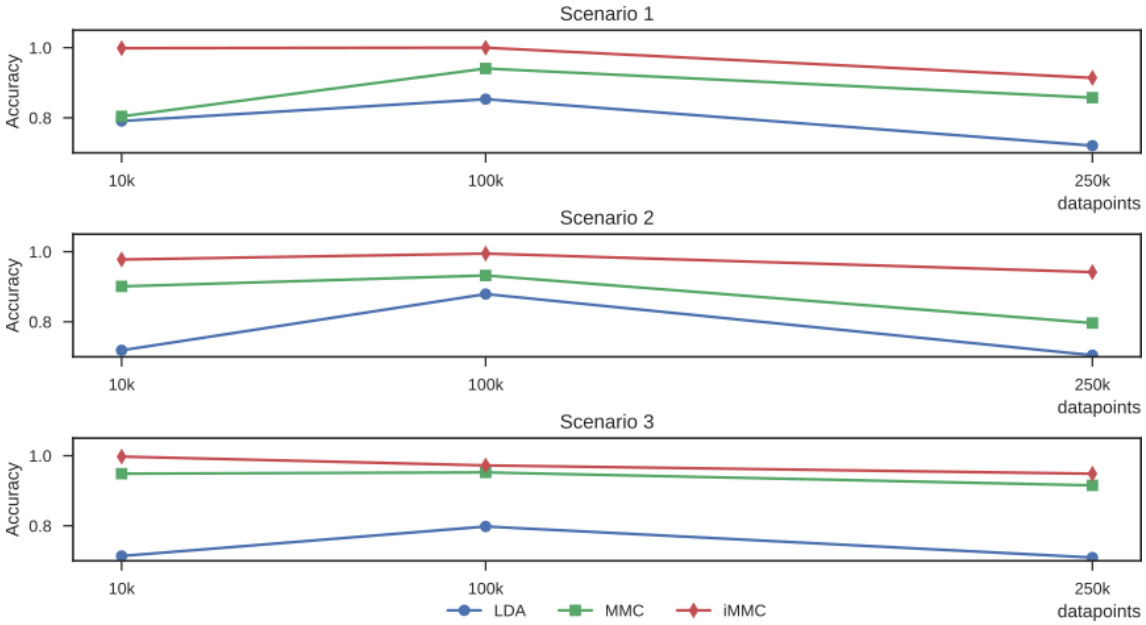


Figure V.3: Accuracy of each method on different scenarios and for dataset sizes.

- **Scenario II (medium):** Partly overlapping observation spaces.
  E.g., *cooking* and *cleaning the kitchen*, sequences with partly overlapping actions (e.g., *open the oven*)

- **Scenario III (challenging):** Fully overlapping observation spaces.
  E.g., *cooking* and *baking*, or *driving a car* and *driving a motorcycle*

A learning task is simpler when the observation spaces are disjoint (Scenario I). Examples are clusters like 'cooking' and 'driving a car' that have no state spaces of events in common. Learning tasks with fully overlapping state spaces is more difficult (Scenario III, Fig. V.2 (right)). Examples are clusters that share many events such as 'cooking' and 'baking' or 'driving a car' and 'driving a motorcycle'. The learning task in Scenario II (Fig. V.2 (left)) addresses both characteristics.

The data for our experiments we generate as follows: We assume that each *intent* is equally likely. Therefore, we select a behavior pattern uniformly at random. The sequence of visited websites is realized by a random walk in the corresponding graph. The procedure is repeated until we reach an a priori defined number of observations. Each scenario is evaluate on data sets of three different sizes: 10 000, 100 000 and 250 000 data points. We use 10 artificially generated data sets per setting (scenarios + data set size) and report on the averaged performances. While we use an uninformed value for all hyper-parameter of our algorithm (each is set to 1), we supply the MMC with the correct number of clusters and apply soft clustering. For LDA we transform each sequence into a frequency vector of the events.

Table V.1 and Figure V.3 shows the clustering performance of the algorithms w.r.t. the different data sets sizes and scenarios. Performance is measured as the accuracy of sequence-

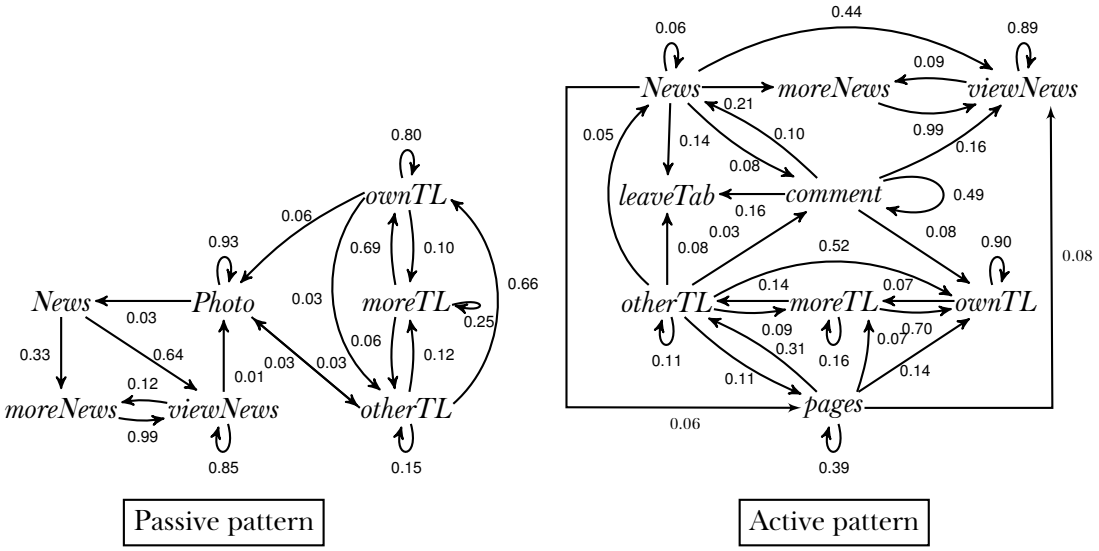


Figure V.4: An exemplary solution of the identified clusters; exit states are omitted, their probability equals 1 minus the sum of emission probabilities of a state.

component associations. In all cases, our algorithm outperforms MMC. In theory, provided with the correct number of clusters, the MMC should converge to the same result as the iMMC. The discrepancy is caused by the average score. Due to the unsupervised nature of the task, selecting the best-performing model is not suitable for evaluations. Therefore, caused by the inefficiency of MMC's optimization algorithm, poor results caused by local optima significantly influence the overall performance of the MMC.

V.1.2.2 Facebook data

We demonstrate how the model can be applied for information extraction tasks. This is especially useful for tasks that come with no or only little prior knowledge. The data set for the next evaluation contains user navigation data from Facebook [138]. For each user, the visited pages are recorded. Examples of pages visited on Facebook are *'Login', 'Newsfeed', 'Load more news', 'Like', etc*. The data set contains 152 unique pages, contained in 49 479 sequences of 2 749 users with 8 197 308 page requests. In our experiments, every session is interpreted as a sequence of observations.

The most frequently identified behavioral pattern shows a user checking for updates on the newsfeed by *waking up* the device and, without performing any additional activity, *put to sleep* shortly after. Figure V.4 depicts two more complex patterns observed on Facebook. The first pattern, on the left, describes passive user behavior without any direct communication. Users tend to look at their newsfeed (*News*) or at their timeline (*ownTL*). While updating (represented by the loop on *ownTL*) or scrolling (*moreTL*) their own timeline, they get sometimes interested in someone else's timeline (*otherTL*). They then scroll through it before going back to their timeline. They tend to look at more entries (*viewNews*) from their newsfeed and interact (self-loop) with them. If they open a gallery (*Photo*), they look at several pictures (self-loop on *Photo*) before returning to their previous activity.

The pattern, on the right, describes a more active behavior. While surfing their Facebook universe, users frequently *comment* on newsfeeds' and timelines' entries. They also visit fan and company pages (*pages*) more often.

The iMMC algorithm successfully distinguished different session behaviors without any prior knowledge of the data, or dependencies between events.

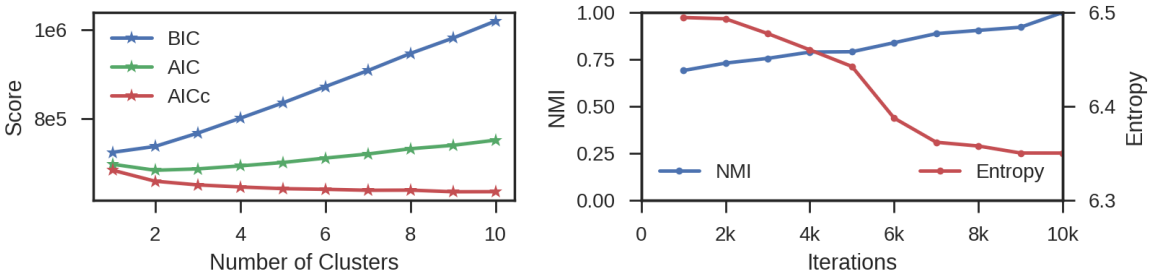


Figure V.5: Left: BIC, AIC and AICc for MMC. Right: NMI and entropy for iMMC

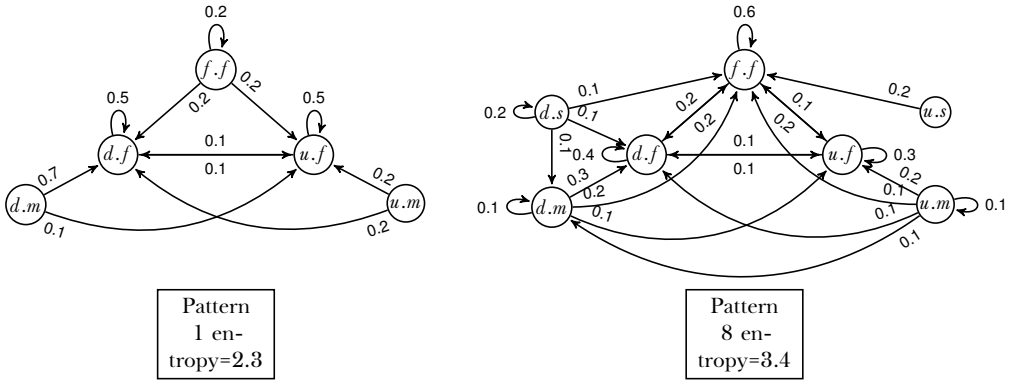


Figure V.6: Two exemplary scrolling patterns.

V.1.2.3 Electronic text books

We present insights on the usage behavior of students interacting with an electronic textbook for history called the mBook [162] next. Our experiments show that the identified usage patterns correlate with psychometric scores.

Among others, the mBook has been successfully deployed in the German-speaking community of Belgium. Together with psychologists and didacticians the aim is to evaluate the pros and cons of daily use in classrooms on children and teachers. In addition to an event log that tracks all user actions in the book, demographic variables and ones measuring competencies and interest are regularly assessed. Since 2013, about 3 000 users have created 370 000 sessions. In this experiment, we focus on 803 sessions of a subset of 286 users between February and March 2017. We aim to identify characteristic usage patterns to later search for correlation with psychometric variables.

Related studies reveal that time-on-page and cursor trajectories often serve as indicators for student engagement [37, 160]. However, in our case, the textbook is mainly used on tablets in classrooms and, hence, cursors or eye tracking are not available. We thus aim to identify alternative indicators precise enough to capture characteristic traits of different behavior. We define and differentiate 75 atomic events that a user can trigger, ranging from pressing a button to various scrolling performances. The latter are further divided into 9 events: *scroll.direction.duration*. The direction can be *up*, *down* or *fix* if the movement is of less than 10 pixels. The duration can be *fast*, *medium* or *slow* for event duration of respectively less than 1 second, between 1 and 3 seconds and more than 3 seconds. In the following, node names will be abbreviated using only the first letter. For example a *scroll.down.fast* is reduced to *d.f*.

In contrast to the analysis of the Facebook data set, where the huge amount of data allowed for the deployment of MMC, in this case, an MMC would fail due to the lack of a sufficient amount of data; information criteria are known to perform poorly when the sample size is smaller than the number of parameters [80] as shown in Figure V.5 (left). The evolution of three information

Table V.2: The most strongly correlated event transitions for each score.

Score	Max Corr.	Event	Min Corr.	Event
Competence	0.697	$f.f \rightarrow u.f$	-0.719	$u.m \rightarrow u.s$
Knowledge	0.962	$d.m \rightarrow u.f$	-0.947	$d.s \rightarrow d.m$
Motivation	0.748	$f.f \rightarrow f.f$	-0.714	$f.f \rightarrow u.f$
IT Access	0.751	$d.s \rightarrow u.f$	-0.735	$f.f \rightarrow d.f$
IT Skill	0.837	$d.s \rightarrow u.f$	-0.743	$d.m \rightarrow d.s$

criteria AIC [2], AICc [132], and BIC [165] is depicted for different numbers of clusters where every point in the figure denotes the best result out of 30 repetitions. Theoretically, the minima of these curves are supposed to give the optimal solutions given the involved parameters. Due to the ill-posed optimization problem, however, the criteria grow almost linearly. The AIC curves reach a minimum for two clusters, which is not interesting. Thus information criteria do not allow to conclude.

By contrast, our Bayesian approach successfully groups the data using $\gamma = 2$, $\sigma = 1.5$, $\lambda = 2.4$, $L = 100$ in 10 000 iterations. After every 1 000 iterations, an intermediate clustering is computed as the average of the last 1 000 iterations. The first intermediate clustering is based on 34 clusters, the final solution settles on 32 clusters. The evolution of the solution is shown in Figure V.5 (right). The blue line (left scale) represents the evolution of the normalized mutual information (NMI) relative to the final solution. The red line (right scale) refers to the entropy of the clustering for the actual iteration. After 7 000 iterations the NMI indicates that the clustering is already 90% similar to the final one. The decrease in entropy shows that the algorithm merges the data into fewer clusters. The plateau after 7 000 iterations indicates fine granular changes in group memberships.

There are eight resulting groups with at least 20 sessions. We focus on the scrolling events and show two patterns in Figure V.6 realizing the smallest and highest entropy, respectively. Note that the weights do not sum up to one, as we ignore outgoing edges to non-scroll events in this analysis.

The first thing to notice is that in Pattern 1, *scroll.fix.** cannot be reached from another type of scroll. Either it starts a scrolling sequence or it indicates misusage or hesitation of the user. Although Pattern 8 is more complex, it shares the fact that users tend to not transit to slower scrolls. This can be interpreted by the observed behavior that 'longer' scrolls are corrected by faster ones. This is typical behavior for users who are scrolling while reading the text on the page. It also reflects in high self-transition probabilities of *scroll.down.slow* and *scroll.fix.fast*. Multiple ways to reach this last event are likely caused by stopping a scroll with a small scroll and keeping the finger on the tablet.

Psychometric Correlations During the 4 years of the experiment, the children are assessed at the end of each school year. Five factors are measured. Competency and knowledge in the field are assessed using item response theory [6, 46]. Additionally, their motivation, access to digital devices and their skills in the usage of these are assessed by multiple choice questionnaires (advanced skills weigh more than simple ones).

To correlate the assessed variables with our clustering, we represent groups by the average score of all children who share a specific behavior pattern. We compute Pearson correlation coefficients [132] that are adjusted for small sample sizes for the 81 possible transition probabilities between scroll events and the 8 resulting clusters with at least 20 elements.

The maximum and minimum correlations for the assessed variables are reported in Table V.2. Except for motivation, high correlated transitions for every variable end with a *scroll.up.fast*

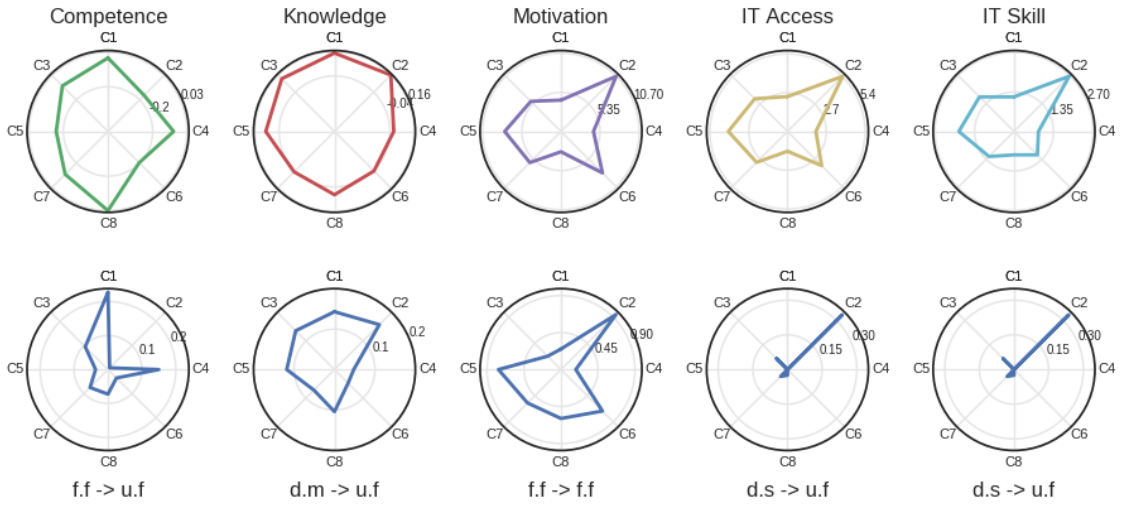


Figure V.7: Scores and probabilities of their most correlated transition for the 8 biggest clusters.

and a change in direction. Knowledge has a correlation of almost 1.0 with *scroll.down.medium* → *scroll.up.fast*, and of nearly −1.0 with *scroll.down.slow* → *scroll.down.medium*. Pattern 8 is the only pattern containing these two edges. However, the correlations cancel out in the final result. Figure V.7 confirms that cluster 8 loads only weakly on knowledge compared to the others.

The first row in Figure V.7 shows the loadings for the 8 biggest clusters. The clusters are organized from top to bottom according to their entropy.

Patterns 1 and 8 (see Fig. V.6) are extracted from clusters 1 and 8, respectively. Both patterns are often observed by pupils with high competencies in history. Therefore, these patterns may serve as behavioral indicators for a user's competency. This finding is supported by the high correlation of cluster 1 with the prior knowledge of the user. Seemingly, knowledgeable children prefer simpler scrolling patterns. By contrast, cluster 2 contains highly motivated children that possess high computer skills. The pupils in cluster 6 are also motivated but do not possess such a high ICT literacy and thus do not know to handle electronic devices that well.

The second row in Figure V.7 displays the values among clusters of the most strongly correlated transitions to the corresponding score. Negative correlations are not shown for interpretability. These plots give an impression of the correlations. For knowledge and motivation scores, the probability of *scroll.down.medium* → *scroll.up.fast* and *scroll.fix.fast* → *scroll.fix.fast* could be used to predict their respective scores in the assessment. Concerning competence, a high transition probability seemingly also implies a high score in the assessment. However, the opposite does not hold. Cluster 8, as also seen in Figure V.6, has a smaller probability of transitioning from *scroll.fix.fast* to *scroll.up.fast*, although the average competency score of the cluster is the largest.

Our results show for the first time that behavioral indicators in electronic textbooks can be identified to discriminate between children. Results like this will have a high impact on the next generations of electronic textbooks so that they become adaptive and provide individual learning environments for every child.

V.2 The Segmentation of Click-Traces

So far, we have considered an approach to clustering time-series data. It showed the advantages of nonparametric Bayesian models. However, we also introduced a rather restrictive assumption. That is, we assumed that each session is based on exactly one intention (mixture component). While this is true for some scenarios, it does not hold for most real-world situations.

In our next contribution, we, therefore, relax this assumption. Again, we propose a nonparametric Bayesian mixture model for prediction and information extraction tasks. From now on, however, we assume an arbitrary number of intentions underlying each session. When this assumption is relaxed, the clustering task becomes a segmentation problem.

Different data, such as traces of user behavior, exhibit dynamics that follow different patterns. In this context, a pattern is considered as a region of a sequence, called a segment, in which successive observations exhibit low-level dynamics (consider: transitions in a graph). Thus, while observations within a segment exhibit low-level dynamics (represented by a graph), transitions between different patterns exhibit high-level dynamics (switching to a different graph). We represent the abstraction of different segments representing the same underlying pattern by a mixture component in the form of a first-order Markov model (a graph). We assume that patterns can be approximated by grouping similar segments in time-series data.

Modeling such data is challenging because the models must be flexible and become very complex very quickly: Bayesian nonparametric models successfully capture data that have complex low-level dynamics [71, 8]. However, approaches that aim to capture dynamics at different levels struggle with either their efficiency [67] or their flexibility. Nonetheless, such models are critical for addressing natural processes that exhibit both low-level and high-level dynamics, such as the navigation strategies of users searching for information on the Internet [187] or Facebook [138], human activities of daily living [54], natural language [105], or motion recognition [86].

The goal of this paper is to develop an approach for segmenting discrete-valued sequential data that can be used for prediction and information extraction. Considering user behavior on the Web, the models should help to predict future behavior and understand sequences of observed actions. Therefore, we assume that users can be described by a set of intentions observable from their paths through the Web. Starting from a data set of observed user behavior, the idea is to approximate intentions by identifying similar segments with relatively stable low-level dynamics. Therefore, each set of segments can be interpreted as a manifestation of a particular intention. It is assumed that user behavior in this constellation is fully observable given the intent. The requirements are as follows: (i) the algorithm should perform a multi-level analysis covering at least two levels of the dynamics (e.g. number of patterns and their manifestations), (ii) the number of these patterns should be unbounded, (iii) have generative/predictive capabilities; and (iv) provide human-readable results. While the first two requirements relate to the segmentation task, the last two are equally important for user understanding.

A nonparametric Bayesian treatment satisfies requirement (ii). First-order Markov models represent a generative approach and therefore guarantee a certain degree of predictive power (iii) as well as easily interpretable results (iv) and a simple inference scheme. By combining both concepts in a hierarchical order, we can also perform a two-level analysis of data dynamics (i).

Our goal is to use segmentation to improve both the prediction of future activities and the understanding of the dynamics of behavioral data. Therefore, we evaluate the segmentation performance of our model on synthetic data to understand its effectiveness and test it for extreme cases. We also apply our model to a novel user understanding task, where we segment behavior traces of users on Facebook to understand their behavior and predict their next steps. Our empirical findings show that our model successfully identifies underlying patterns and can be effectively transformed into a predictor for future observations.

V.2.1 An Infinite Mixture Model of Markov Chains

In this section, we present our approach to a two-level analysis of data dynamics. The main novelty of the model is its ability to capture the fully observable graphs describing the patterns without violating the exchangeability assumption.

The model is comprised of four key parts:

- An HDP-HMM [II.3.6.5], modeling the high-level dynamics of the transitions between patterns
- A self-transition bias [71], representing our prior knowledge that successive observations are more likely to belong to the same pattern
- An annealing process, that changes the source of information behind the self-transition bias during the optimization phase
- First-order Markov models the low-level dynamics representing the patterns as graphs

V.2.1.1 Mixing proportion

The hierarchical Dirichlet process (HDP), which consists of a two-level hierarchy of DPs, the realization of one DP $G_0 \sim \text{GEM}(\alpha)$,

$$\begin{aligned} \beta_k &\sim \text{Beta}(1, \alpha) & \pi_k &= \beta_k \prod_{l=1}^{k-1}(1-\beta_l) & k &= 1, 2, \ldots \\ \theta_k &\sim H & G_0 &= \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}. \end{aligned} \tag{V.10}$$

is used as the base measure for all its subordinate DPs, $G_i \sim \text{GEM}_2(\alpha, G_0)$,

$$\beta'_{ik} \sim \text{Beta}\left(\alpha\beta_k, \alpha\left(1 - \sum_{l=1}^{k}\beta_l\right)\right) \qquad \beta_{ik} = \beta'_{ik}\prod_{l=1}^{k-1}\left(1-\beta'_{ik}\right) \qquad G_i = \sum_{k=1}^{K}\beta_{ik}G_{0,k}\delta_k. \tag{V.11}$$

Therefore, these DPs represent distributions over measures on the same discrete, finite space. They resemble a transition matrix that can change its size. Thus, the resulting HDP-HMM represents an HMM with an unbounded state-space and is well suited to model the hidden states of our model.

Self-transition bias To address the problem of fast switching between redundant states in the HDP-HMM and avoid slowing mixing rates and a possible decrease in predictive performance [71], we make use of the self-transition mechanism proposed in Fox et al. [71]. Therefore, GEM_2 is slightly modified to incorporate a bias towards self-transitions

$$G_i \sim \text{GEM}_2\left(\alpha + \kappa, \frac{\alpha\boldsymbol{\beta} + \kappa\delta_i}{\alpha + \kappa}\right), \tag{V.12}$$

where $\kappa > 0$ is the amount added to the ith mixture component (self-transition bias).

This implementation of a self-transition bias is rather uninformative. Over time, we gain more information on the process which allows us to formulate a more informed bias. Remember, we represent mixture components by graphs. A graph consists of starting and exit nodes. Therefore, when learning about these nodes, we can augment our uninformative self-transition bias with this information. Our final approach represents an annealing strategy, that, over time, incorporates more and more information on the starting and exit nodes into the bias. Here, reaching an exit node influences the current hidden state through a 'flagged' previous activity state.

V.2.1.2 Mixture components

The model so far consists of the HDP-HMM. It is suitable for representing the mixing proportion, i.e., the underlying hidden process of changing intentions. Remember though that the emission distributions of this model (for our purpose: representation of intentions) are discrete probability distributions, which are separately specified. Hence, we cannot represent our mixture component (i.e., the intentions) by graphs. To do so, we need more information on emissions.

To represent the mixture components by graphs, we can use transition information rather than single activities as observations. Note however that in an HMM, successive observations only depend on the previous hidden state (the intent) and not previous observations (the clicks). Adding dependencies between successive observations would make the model optimization significantly more complex and time-consuming. We can avoid this problem with a simple trick. Defining an observation not as a single activity (a click) but as a 2-gram that combines information of the current with the previous activity. Hence, an observation represents a transition without introducing dependencies between successive observations. We define an observation as a 2-gram (p_t, y_t), where $\boldsymbol{p}$ is an additional layer of states representing observations of the previous time-step.

V.2.1.3 Formal description

We now give a more formal description of the model. Let Ω denote a finite activity space and Ω^* any sequence of possible combinations over Ω. Then, $\boldsymbol{y}^{(s)} \in \Omega^*$ denotes a sequence of activities with s as its index. We assume a data set $\mathbf{Y} = \left\{\boldsymbol{y}^{(s)}\right\}_{s=1}^{S}$ of S sequences with arbitrary length τ_s.

The mixing proportion is represented by an HDP-HMM with a self-transition bias,

$$G_0 \sim \mathrm{GEM}\left(\gamma\right) \qquad \pi_i \sim \mathrm{GEM}_2\left(\alpha, \boldsymbol{\beta}\right). \tag{V.13}$$

With the corresponding biased weights, $\hat{\pi}$ as

$$\hat{\pi}_i \sim \mathrm{GEM}_2\left(\alpha + \kappa, \frac{\alpha\,\boldsymbol{\beta} + \kappa\delta_i}{\alpha + \kappa}\right). \tag{V.14}$$

Here, the mixing weights of G_0 are denoted by $\boldsymbol{\beta}$, the ones of G_{i0} by $\boldsymbol{\beta}_i$. It controls the hidden states (intentions),

$$z_t \sim \pi_{z_{t-1}}. \tag{V.15}$$

We represent each mixture component by a transition matrix describing the pattern as a graph. We assume, contrary to the hidden states, that the activity space is known in advance. Thus, we use Dirichlet distributions (Dir) instead of DPs. Each mixture component consists of a base distribution

$$\boldsymbol{\omega}_i \sim \mathrm{Dir}\left(\sigma\right) \tag{V.16}$$

and corresponding subordinate Dirichlet distributions

$$\theta_{ik} \sim \mathrm{Dir}\left(\lambda\,\boldsymbol{\omega}_i\right). \tag{V.17}$$

It denotes the probability measure of traversing from activity k in the i^{th} mixture component to any of the K activities, θ_{ik}.

Besides these parameters, we have the states of the model. When making observations, we have to account for the currently active mixture component (intention), the previous activity (previously visited page), and the current activity (current page). Therefore, our approach consists of three layers, the hidden states denoted by $\boldsymbol{z} = \{z_t\}_{t=1}^{\tau+1}$, the previous activity denoted by $\boldsymbol{p} = \{p_t\}_{t=1}^{\tau+1}$, and the current activity denoted by $\boldsymbol{y} = \{y_t\}_{t=1}^{\tau+1}$. Here, a hidden state z_t depends on the previous hidden state z_{t-1} and the previous activity p_t,

$$p\left(z_t | z_{t-1}, p_t\right) = \left(1 - \psi\right)\hat{\pi}_{z_{t-1}} + \psi\left(\mathbb{1}\left(p_t \neq B\right)\delta_{z_{t-1}} + \mathbb{1}\left(p_t = B\right)\pi_{z_{t-1}}\right), \tag{V.18}$$

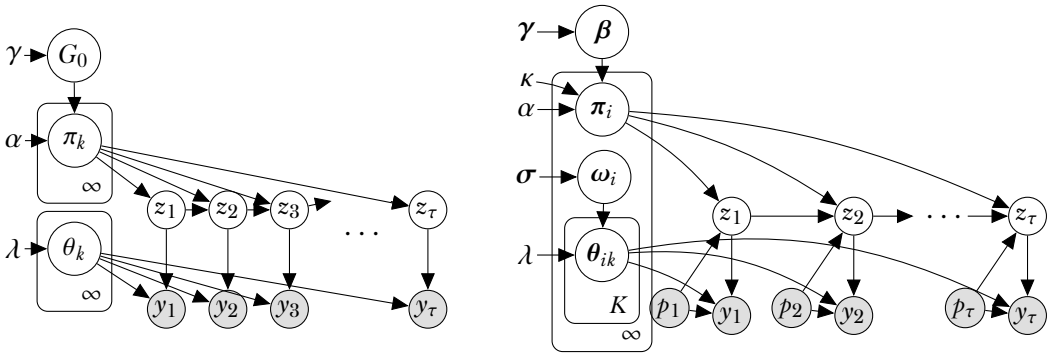


Figure V.8: (left) Graphical model of an HDP-HMM [173]; (right) graphical model of IMMC; K is defined as $|\Sigma| + 1$, where $+1$ represents the auxiliary boundary node of the lossless concatenation; $\boldsymbol{\beta}$, $\boldsymbol{\pi}$, $\boldsymbol{\omega}$, and $\boldsymbol{\theta}$ represent Dirs; the model's hyperparameters are denoted by γ, κ, α, σ, and λ; white nodes represent hidden states (z), gray nodes observed states (p and y).

where ψ denotes the annealing parameter that changes during optimization, shifting from 0 to 0.9. Note that the previous activity p_t belongs to the current observation at time-step t. If $p_t = B$, we say $p\,(z_t|z_{t-1})$ as the second part of Eq. V.18 is zero. Finally, the current activity x_t depends on the previous activity p_t and the current hidden state z_t,

$$p\,(y_t|z_t, p_t) = \theta_{z_t p_t y_t}. \tag{V.19}$$

During optimization, a time series is augmented by virtual activities marking the boundaries of segments denoted by B. For example, the first observation consists of the virtual boundary symbol B in p_1 ('start of a segment') and the first observed activity as y_1, while the last observation consists of the last observed activity in $p_{\tau+1}$ and the virtual boundary symbol B in $y_{\tau+1}$ ('end of a segment'). At each time step at which the model switches its hidden state, an additional virtual boundary symbol is inserted.
The resulting graphical model is depicted in Figure V.8 (right).

V.2.1.4 A Blocked Gibbs Sampler

We present a truncated blocked Markov chain Monte Carlo (MCMC) HDP sampling algorithm, similar to the one Fox et al. [71] propose, to estimate the parameters of our approach.

Fox et al. [71] show that a truncated blocked Gibbs sampler allows to jointly sample hidden states and exploit the Markovian structure. The joint mechanism obtains faster mixing rates than for instance a direct assignment sampler. To sample distributions of theoretically infinite cardinality, we make use of the degree L weak limit approximation [91], where L denotes the maximum cardinality of the approximated distribution. It follows, that in practice L needs to exceed the number of true mixture components.

Thus, a DP is approximated by a Dirichlet distribution (Dir), with $\mathrm{Dir}(\alpha/L, \ldots, \alpha/L)$. The prior distributions $\boldsymbol{\beta}$, $\boldsymbol{\pi}$, $\boldsymbol{\omega}$, and $\boldsymbol{\theta}$ are initialized by

$$\begin{aligned} \boldsymbol{\beta} &\sim \mathrm{Dir}\,(\boldsymbol{\gamma}) & \boldsymbol{\pi}_i &\sim \mathrm{Dir}\,(\alpha\,\boldsymbol{\beta} + \kappa\delta_i) \\ \boldsymbol{\omega}_i &\sim \mathrm{Dir}\,(\boldsymbol{\sigma}) & \boldsymbol{\theta}_{ik} &\sim \mathrm{Dir}\,(\lambda\,\boldsymbol{\omega}_i)\,, \end{aligned} \tag{V.20}$$

where $\boldsymbol{\beta}$ are the mixture weights of G_0, $1 \le i \le L$, $K = |\Omega| + 1$, and $1 \le k \le K$.

To update the prior distributions after each iteration, we have to keep track of observation assignments and hidden state transitions. Therefore, d stores the number of observations assigned to each mixture component and $f_{ik_1k_2}$ records the number of transitions within component i, where k_1 and k_2 represent the row and column of the transition matrix. Finally, $n_{i_1 i_2}$ keeps track of the transitions between mixture components i_1 and i_2. For each iteration, the auxiliary variables document the assignment step.

Algorithm 4 Blocked Gibbs sampler for IMMC

Given the hyperparameters $\boldsymbol{\beta}, \boldsymbol{\pi}, \boldsymbol{\omega}, \boldsymbol{\theta}, \kappa$

1. Initialize prior distributions according to Eq. V.20

Until convergence **do**:

2. Perform Baum-Welch algorithm Eqs. V.21 - V.24
3. During the forward steps, update auxiliary variables as follows:
   - Increment

$$
\begin{aligned}
&d_{z_t}, && \text{if } y_t \neq B \\
&n_{z_{t-1}z_t}, && \text{if } z_t \neq z_{t-1} \text{ or } \epsilon \\
&f_{z_t,B,y_t} && \text{if } z_t \neq z_{t-1} \text{ or } \epsilon \text{ or } p_t = B \\
&f_{z_t,p_t,B} && \text{if } z_t \neq z_{t-1} \text{ or } \epsilon \text{ or } y_t = B \\
&f_{z_t,p_t,y_t}, && \text{if } z_t = z_{t-1} \text{ and } \neg\epsilon
\end{aligned}
$$

4. Compute posterior distributions according to Eq. V.26

Sampling z_t We obtain a realization of the latent states z_t by making use of the Baum-Welch algorithm. Applying the algorithm backward in time, from the last to the first observation of a sequence, the backward step describes the transition from p_t to y_t given that we know the probabilities *from there on* (coming from the future, learning about the past). Hence, we have three different cases. An observation can describe the beginning of a sequence denoted by the virtual boundary symbol B in p_t. Therefore, the backward probability is multiplied by the probability of a mixture component to start with y_t. Further, an observation describes the end of a sequence when y_t is B. In this case, we know that the model exited the segment on p_t. Therefore, the backward probability is multiplied by $\theta_{ip_t}(B)$. Finally, if none of the observable states contains a boundary symbol, the situation is more complex. We have to account for the continuation of a segment ($\theta_{ip_t}(y_t)$), while also incorporating the case where the model switches to another segment ($\theta_{ip_t}(B)\,\pi_{ij}\theta_{jB}(y_t)$). We obtain the backward probabilities $m_{t,t-1}$:

$$
\begin{aligned}
&m_{\tau+1,\tau}(i) = 1 \\
&m_{t,t-1}(i) = \begin{cases} m_{t+1,t}(i) \cdot p_I & \text{if } B \in (p_t, y_t)\,; \\ m_{t+1,t}^{\top}\left(p_I \delta(i) + p_E\right) & \text{otherwise.} \end{cases}
\end{aligned}
\tag{V.21}
$$

With

$$
\begin{aligned}
p_I &= p\left(y_t | z_t = i, p_t\right) \\
p_E &= p\left(B | z_t = i, p_t\right) \sum_{j=1}^{L} p\left(z_t = j | z_{t-1} = i, p_t\right) \cdot p\left(y_t | z_t = j, p_t = B\right) \cdot \delta_j .
\end{aligned}
\tag{V.22}
$$

In the forward pass of the Baum-Welch algorithm, we draw a sample of the hidden states based on the updated backward probabilities, the current hidden state, and the observation. Again, we have to account for three cases. The start of a sequence ($p_t = B$) represents the case with the least information. We sample z_t based on the weight of a mixture component and the probability that y_t is its starting node. Observing the end of a sequence ($y_t = B$) represents the most trivial case, the hidden state does not change, $z_t = z_{t-1}$. In the final case, we again have to account for two possible events, the continuation and the change of a segment. In summary, we have

$$
p(z_t | z_{t-1}, p_t, y_t, \boldsymbol{m}) = \begin{cases} \delta_{z_{t-1}} & \text{if } y_t = B \\ p(z_t | z_{t-1}) \cdot p_I \cdot m_{t+1,t}(z_t) & \text{if } p_t = B \\ \left(p_I \delta_{z_{t-1}} + p_E\right) \cdot m_{t+1,t}(z_t) & \text{otherwise.} \end{cases}
\tag{V.23}
$$

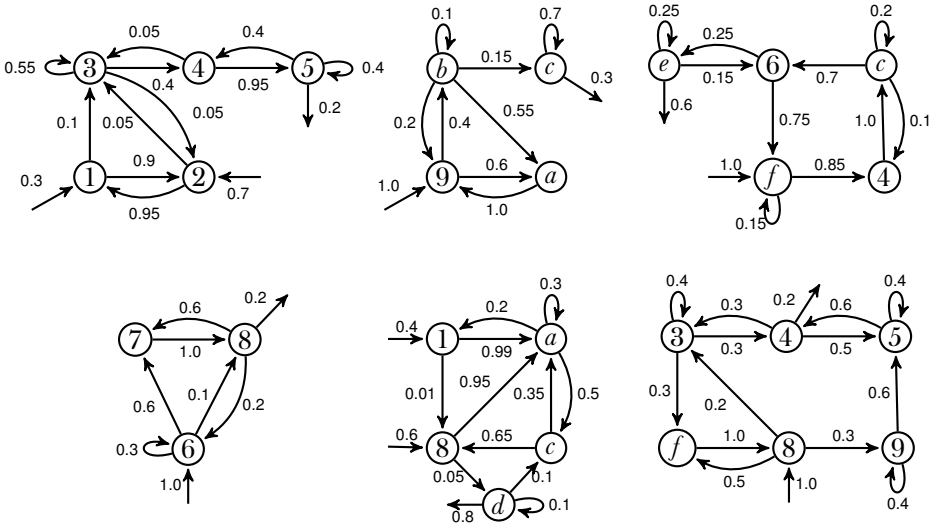


Figure V.9: Generative processes of Scenario II; observations are indexed by hexadecimal numbers (1-f).

The assignments are sampled from the probability distribution for z_t,

$$z_t \sim \text{Cat}\left(p(z_t|z_{t-1}, p_t, y_t, \boldsymbol{m})\right). \tag{V.24}$$

During the sampling process, the auxiliary variables keep track of the sufficient statistics (depicted in Algorithm 4) to update the prior distributions afterward. To this end, we introduce an additional variable ϵ to allow the model to distinguish between the continuation of a segment and successive segments of the same type, i.e. a transition from a segment of a specific mixture component to a new segment of the same mixture component. So, if $z_t = z_{t-1}$ and $B \notin (p_t, y_t)$, we compute epsilon as follows:

$$\epsilon \sim \text{Ber}\left(\frac{p\left(B|z_t, p_t\right) p\left(z_t|z_{t-1}\right) p\left(y_t|z_t, B\right)}{p(y_t|z_t, p_t) + p\left(B|z_t, p_t\right) p\left(z_t|z_{t-1}\right) p\left(y_t|z_t, B\right)}\right), \tag{V.25}$$

where $\text{Ber}(\cdot)$ denotes the Bernoulli distribution. Given a realization of $\boldsymbol{z}$, the prior distributions of the parameters are updated accordingly,

$$\begin{aligned} \boldsymbol{\beta} &\sim \text{Dir}\left(\boldsymbol{\gamma} + \boldsymbol{d}\right) & \boldsymbol{\pi_i} &\sim \text{Dir}\left(\alpha\,\boldsymbol{\beta} + \boldsymbol{n}_{i\cdot} + \kappa\delta_i\right) \\ \boldsymbol{\omega}_i &\sim \text{Dir}\left(\boldsymbol{\sigma} + \boldsymbol{f}_{i\cdot}\right) & \boldsymbol{\theta}_{ik} &\sim \text{Dir}\left(\lambda\,\boldsymbol{\omega}_i + \boldsymbol{f}_{ik}\right), \end{aligned} \tag{V.26}$$

where $\boldsymbol{f}_{ik}$ denotes the frequency count of different events in state k of mixture component i. Then, $\boldsymbol{f}_{i\cdot}$ denotes the frequency count over all states in mixture component i, $\boldsymbol{f}_{i\cdot} = \sum_{k=1}^{K} \boldsymbol{f}_{i,k}$.

V.2.2 Experiments: Segmentation

We first evaluate the segmentation performance of our model in a controlled environment using synthetic data. It allows us to understand its effectiveness and test it for extreme cases. We also apply our model to a novel user understanding task, where we segment behavior traces of users on Facebook to understand their behavior and predict their next steps.

V.2.2.1 Artificial Data

We evaluate the segmentation performance of our model to understand its effectiveness and test it for extreme cases. Therefore, we apply our model to three artificial datasets. Given the assignments as ground truth, we report on the accuracy of our approach.

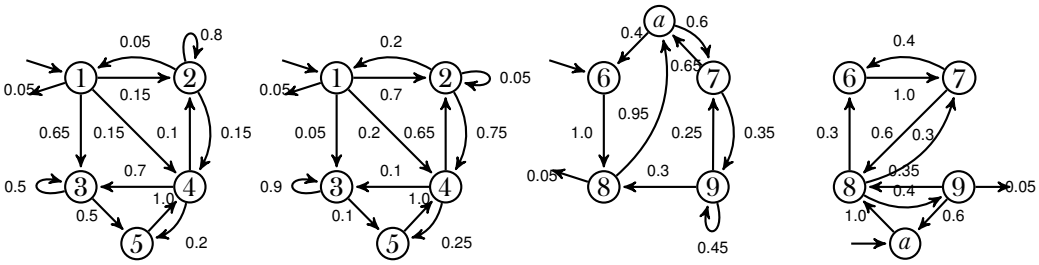


Figure V.10: Generative processes of the artificial scenario III.

Table V.3: Error rates for the artificial segmentation tasks.

	MMC	IMMC
Scenario I	3.64%	0.07%
Scenario II	2.38%	4.73%
Scenario III	15.66%	0.63%

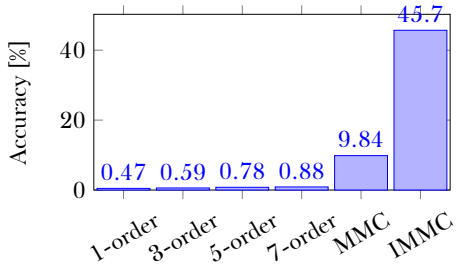


Figure V.11: Prediction performances.

The scenarios consist of behavior patterns emulating different intentions of a user. The level of difficulty is controlled by the similarity between different patterns.
Scenario I: Patterns with no overlapping events. Technically, the simplest scenario. Any segmentation approach should be able to distinguish between these patterns.
Scenario II: Patterns with partially overlapping events but completely different transition probabilities (see Fig. V.9).
Scenario III: Two pairs of patterns with entirely overlapping event spaces (see Fig. V.10). One pair shows the same event transitions only differing in transition probabilities, the other with already differing transitions.

Each scenario is represented by a training set of ≈30 000 *observed* events combined into 1 000 sequences to assess the performance of the algorithms.

Segmentation performance We compare our approach to the parametric counterpart proposed by Cadez et al. [25]. This approach represents a parametric interpretation of mixture models of Markov chains (MMC). Due to its lack of flexibility, it is unable to segment sequences, but it rather clusters them. Thus, we provide the MMC with information on segment boundaries. All results are reported as the average of 25 recorded runs. However, in the case of the MMC, every time we run it 10 times with varying cluster initializations and record the best result for our comparison.

Table V.3 depicts error rates for the segmentation task. Even though MMC has additional information, our approach outperforms it in both Scenarios I and III. In Scenario II the additionally provided information about the segment boundaries is even more vital than in the other scenarios. Our algorithm, for example, split pattern 2 into two parts, $\{9, a, b\}$ and $\{b, c\}$ (see Fig. V.9).

V.2.2.2 User Navigation on Facebook

The Facebook dataset contains user navigation data from Facebook [138]. For each user, the invoked pages are recorded and grouped into sessions. Exemplary invoked pages are *'Login', 'Newsfeed', 'Load more news', 'Like', etc.* The dataset contains 152 unique invoked pages (Ω), 49 479

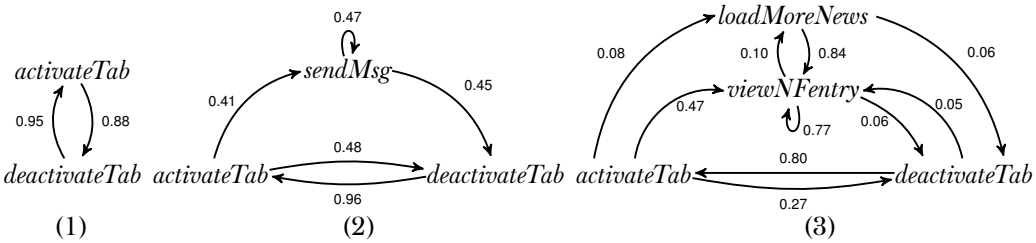


Figure V.12: Examples of the identified processes; exit nodes are omitted, their probability equals 1 minus the sum of emission probabilities of a node.

sessions of 2 749 users, and 8 197 308 activities (visited pages). Every session is interpreted as a sequence of activities.

We use our algorithm to identify the user intentions to improve our understanding of their behavior. Further, we use the identified patterns to predict the next step of a user. The prediction performance provides a measure of the quality of the identified patterns.

Prediction performance To measure the prediction performance, we split the Facebook data into a training- and an evaluation set. Given a sequence, the goal is to predict the final activity. The rest of the sequence is used as evidence for the algorithm. This situation simulates the prediction of future observations in a sequence given only past and present observations.

For the prediction process, we optimize the model using a Gibbs sampler. As the final step of optimization, an activity is assigned to the component where it was most often assigned during the iterations of the sampling process. For prediction, we compute the MAP estimate based on the likelihood of all mixture components for a given sequence and the transition probabilities in each component from the most recent observation to all possible future observations.

We compare the performance of our model to the prediction performance of Markov models of different orders (1-7) as well as to the MMC. Markov models have natural prediction capabilities and, similar to our model, MMC represents its mixture components as MC. Thus, this comparison provides some insights into the benefits of higher-order Markov models as well as additional clustering or segmenting of the input data. Due to the parametric nature of MMC, we perform a grid search for the optimal parameterization that results in the highest data likelihood (40 components).

Figure V.11 shows the accuracies of predicting the next observation following the most recent one within the Facebook dataset. Our model outperforms MMC as well as the Markov models. The results suggest that on this level of abstraction a more detailed model significantly increases the quality of prediction results.

Interpretability. Finally, we demonstrate how the model can be applied to information extraction tasks. This is especially useful for tasks that come with no or only little prior knowledge. Being a nonparametric model that adjusts its complexity to the data, our approach is a promising candidate for such tasks. Representing clusters by Markov models makes it easy to interpret the resulting patterns.

Figure V.12 depicts three frequently observed behavioral patterns of users on Facebook. *(1)* shows a user checking for updates on the *newsfeed* or waiting for *new messages*. The user *activates* the Facebook tab and without doing any additional activity *deactivates* it shortly after. *(2)* represents users communicating with each other. *(3)* shows users who are interested in updates of their friends. After *activating* the Facebook tab, scrolling the *newsfeed* and visiting specific *newsfeed entries*, users *deactivate* the tab again. These types of segments represent user behavior focused on specific tasks. Our results give a detailed insight into how users interact on Facebook.

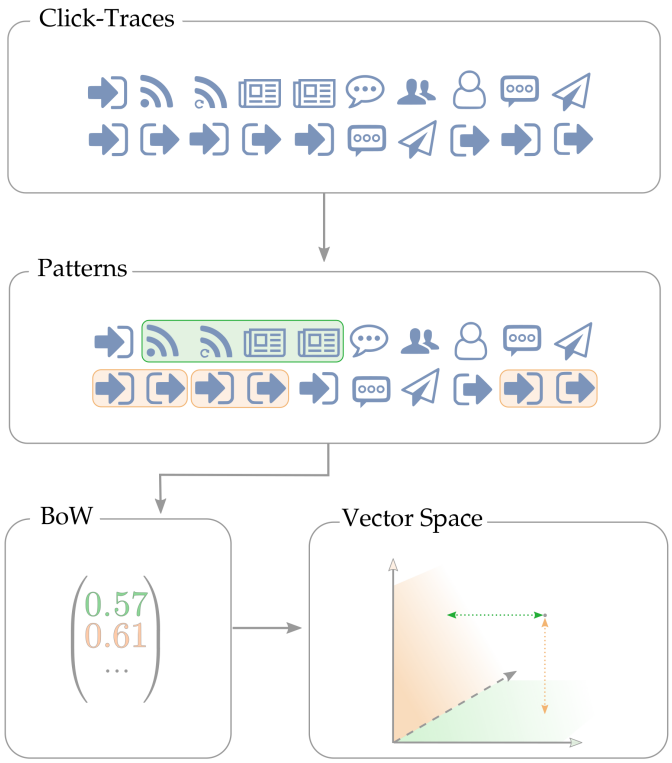


Figure V.13: Overview of behavior patterns embedded in a vector space: 1) time-series (click-traces) are mined for patterns and split into corresponding segments 2) pattern frequencies over the segments of a user are transformed into vectors and 3) represented in a corresponding vector space.

V.3 The Latent Behavior Space

Time-series are ubiquitous, yet current approaches to mining information from such data come with significant limitations. Algorithms specifically designed for time-series data are computationally expensive and restricted in their application. Alternative solutions that project time-series data into a fixed-dimensional space to apply proven machine learning algorithms make inefficient use of temporal information inherent in such data.

In our next contribution, we propose an unsupervised approach to identify a fixed-dimensional latent space that can be interpreted as behavior embeddings. The embedding is spanned by the patterns that are extracted from data using a non-parametric Bayesian segmentation algorithm. The algorithm adjusts the number of patterns to the structure of the data. It is robust to noise, easily deployable in industrial settings, and generally applicable to time-series inference tasks.

We perform exhaustive experiments on three different tasks. The two major findings are as follows: (*i*) the group of general approaches including our work slightly outperforms state-of-the-art solutions for social bot detection that were specifically designed for that task (Twitter Experiment), (*ii*) our algorithm outperforms the other state-of-the-art general approaches (Artificial Experiment).

Additionally, we show the benefits of our approach to data science (Electronic Textbook Experiment). On event logs from an electronic textbook, we show that behavior patterns identified by our approach can be easily understood and interpreted by humans.

V.3.1 Inference on Time-Series

Monitoring systems over time generate a series of observed *events*. These events, including their order and timing, can be represented as time series data. As systems evolve and choices are

not independent of the past, processing time series holds promise for many different domains: Initial results demonstrate noteworthy improvements for applications ranging from mining health records [93], forecasting of financial data [26], to recommending locations, music, and products of potential interest [66]. Other promising applications include Web usage analysis to personalize Web pages adaptively, behavior modeling to detect anomalous accounts, and building adaptive and intelligent educational systems [157].

We propose an approach that excels at two highly diverse application scenarios, namely *social-bot detection* and *educational science*.

Making sense of observed behavior in an automated fashion is key to these applications. However, observed behavior traces are typically characterized by complex properties and are usually not independent and identically distributed. The literature so far addresses these concerns in two different ways. Approaches of one family explicitly aim at modeling the underlying processes that cause the characteristics in observed time series [71, 95, 66, 159]. They leverage temporal information that is inherent in the observed behavioral patterns and are considered state-of-the-art for sequential prediction tasks [198]. The second family projects time series data into fixed dimensional latent spaces, thereby, assuming that observations within time series are i.i.d. While losing temporal information, the representation in these latent spaces allows for standard learning principles and algorithms for inference tasks. While the former group of approaches assumes labeled data, the latter can be applied to unsupervised settings. The application of machine learning algorithms that leverage temporal information in an unsupervised setting hence is not straightforward. It is yet unclear how to leverage the rich temporal information and perform inference in face of the increased complexity and unlabeled data sets.

To motivate our contribution, we take a look at a related field of research. In NLP, a similar problem has hampered research for years. Computers do not understand the semantic meaning of words. Therefore, there are methods to transfer words into formats that can be understood by computers. One approach of earlier years was one-hot encodings, also known as count vectorization. Here, each word is represented by its own dimension in a vector. Thus, a word can be represented as a vector where the dimension representing that word is one, and all others zero. One gets huge sparse vectors that do not contain any relations (e.g. meaning, morphology, context). This changed with the invention of word embedding. Mikolov et al. [115] provided an efficient method for identifying high-quality vector representations that capture semantic word representations. The underlying idea can be summarized by the distribution hypothesis [84]: "A word is characterized by the company it keeps" (Firth [68]). It made the unsupervised vectorization task tractable.

We can draw many parallels between this development in NLP and today's challenges in processing time series. Suppose, we observe an entity in a system. Interpreting events as characters in a text, a sequence of events then constitutes a word, i.e., a series of observations represents an action. A sequence of actions, in turn, is not random but holds specific information (like a sentence). Similar to texts for NLP tasks, semantic information is crucial for subsequent tasks.

In this paper, we present a vectorization method for time series data that encodes semantic behavior information. Due to the more complex dynamics of user behavior compared to text corpora, we apply a nonparametric Bayesian model.

In the context of user behavior, we assume that the events follow patterns caused by user intents. Examples could be browsing a news feed, posting, or responding to messages. Our approach inspects the behavior of entities in the system (i.e. their click traces). It identifies segments of arbitrary length that represent often occurring patterns in the data. Based on the learned behavior patterns, we transform time series into vectors according to the identified patterns. Users are then represented by the amount of time and frequencies they typically spend

throughout these patterns.

To demonstrate the benefits of our approach, we compare our behavior representations to standard behavior-specific one-hot encodings. Therefore, we designed an artificial data set to measure the capabilities of the approaches w.r.t. the frequency count of observed states, temporal patterns inherent in the data, and durations that pass between successive observations.

We further demonstrate its applicability to real-world scenarios on two highly diverse application scenarios, namely *social-bot detection* and *educational science*. On a Twitter corpus containing the Tweet behavior of bots and humans, we utilize behavior embeddings for the tasks of *social bot detection*. Compared to algorithms specifically tailored to the task, the *behavior embeddings* performs similarly to the best-performing state-of-the-art algorithm reported in [43]. In a second scenario we predict the competency scores of pupils based on their behavior when interacting with an electronic textbook: On usage data of pupils interacting with an electronic textbook, we mine patterns, show the prediction performance, and ease of interpretation of obtained results.

The remainder of this section introduces the related work in §1, describes our behavior embeddings in §2, reports the results of our experiments, utilizing the embeddings for social-bot detection and user understanding in §3, and concludes our contribution in §4.

V.3.1.1 Existing Projections and time series Representations

We commence by taking a closer look at existing approaches that project time series data into a latent space, to understand where they can be improved. A simple approach to project time series data into a latent space is to apply a naïve Bag-of-Words (BoW) strategy, i.e. to express each time series as the count of occurrences of the unique observations it contains.

Wang et al. [182] extend this representation by additional metadata. In the context of user modeling, this may comprise statistical data like the average number of clicks a user performs within a session, or the total count of her clicks. These approaches disregard temporal patterns within the data entirely, which may cause a significant drop in prediction performance.

To improve the recognition of temporal patterns the following approaches propose BoW strategies that are based on constructed features that represent some temporal information. Viswanath et al. [178] suggest a strategy to project temporal-, spatial-, and spatio-temporal information onto a fixed-dimensional space. Here, temporal information captures some characteristics of the system over time, such as the number of likes of users on Facebook per day. Spatial information is encoded as a histogram over a set of features that are defined a priori. This step of the transformation is similar to the simpler BoW approach above. In the context of a user on Facebook, the buckets of the histogram could represent some categories of likes, such as sports, politics, education, etc. Finally, spatio-temporal information is encoded as the evolution of the spatial information of observed values over time. For the Facebook example this means that instead of the total number of likes per category, one could capture the time series of distributions of liked categories per user and day. This approach represents a simple solution to the question of information- and computation efficiency regarding time series data. However, the approach of Viswanath et al. [178] requires pre-processing of the time series data combined with a feature design tailored to the task at hand. Depending on the type of time slots and buckets the approach can furthermore lead to a high dimensional latent space requiring a huge amount of training data.

Approaches that explicitly model time series data, analyze the temporal dynamics within the data more thoroughly. A simple approach to capture the dynamics is n-gram models. Here, an abstraction of the temporal information is obtained by running a sliding window of size n over the data and summarizing the occurrences of the different recurring patterns of length n. This is commonly used for natural language processing tasks. However, incorporating temporal information has some obvious drawbacks. For one, when used as a projection scheme such a representation could potentially result in an explosion of dimensionality in the vector space,

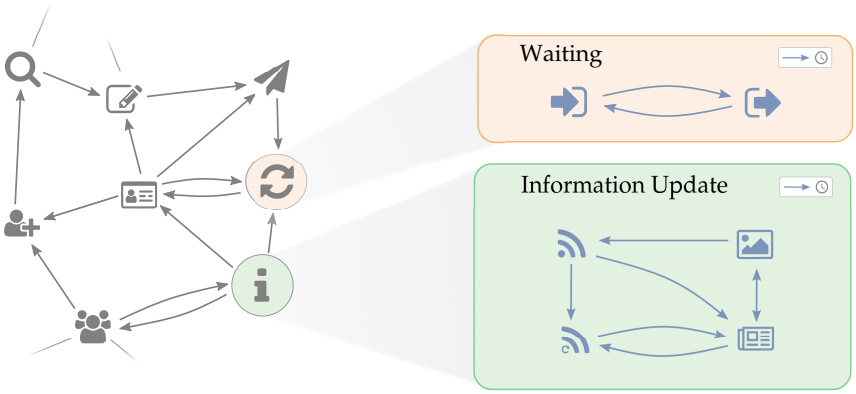


Figure V.14: Transition graph of user behavior patterns [mixture model] (left) shows identified dependencies between different behavior patterns [mixture components]; examples of transition graphs corresponding to specific behavior patterns (mixture components: *Waiting* and *Information Update*) are shown on the right. The transitions within a behavior pattern encode duration information.

i.e. each n-gram would be represented as a dimension. The lack of a reasonable abstraction mechanism additionally may result in an increased information loss that is induced by the projection, e.g. both, n-gram A = "abacb" and B = "abaca", would represent a dimension within the latent space with a similar relation to each other as to C = 'ddddd'.

An adjustment to this approach is to mine patterns from n-gram representations of time series data [25, 180]. This mechanism provides a level of abstraction. However, it assumes that a time series follows a single pattern. To circumvent this restriction it is necessary to split a time series into segments that correspond to the specific patterns of individual intentions. Figueiredo et al. [66] propose an approach that applies a static segmentation process before mining underlying patterns. Modeling smaller chunks of a time series potentially result in more accurate representations of sequential information within data.

Considering both, the time series modeling approaches and the projection schemes, and their respective strengths, we suggest that the latter will benefit from projections that better model the temporal properties. By building the projection space upon identified temporal patterns, our projections capture valuable temporal information that, therefore, is accessible to standard machine learning approaches such as SVMs, kNNs, etc.

V.3.2 Behavior Embeddings

We assume that we can summarize observed user behavior, e.g., on an online social network (OSN) such as Facebook by a set of user intentions. User behavior is captured as a series of events in a certain time frame representing user sessions. Within such a session a user can pursue an arbitrary number of intentions. We assume that we observe a set $\{\mathbf{x_s}\}_{s=1}^{S}$ of S time series, each representing a user session. A time series is an ordered list of events that originate from an arbitrary number of patterns (intentions). Accordingly, each event of a time series, x_{st}, represents an action of a user within a session. Here, t denotes the position of the event within the time series.

The patterns that govern the observed behavior are the alphabet of the presented behavior representation. These patterns manifest themselves as sub-sequences within sessions. In the following, we refer to such sub-sequences as segments. To identify the underlying patterns, our algorithm splits the data into meaningful segments of observations. This process is based on an internal representation of the set of identified patterns that generalize the behavior of segments. These representations are modeled by transition graphs. Considering the Web user example, the observation space contains all possible activities (clicks) a user can perform. The graph then

describes a pattern of frequently observed user behavior in a compact and interpretable fashion.

The extracted patterns themselves yield valuable information. In the second step, we use the identified patterns to vectorize the time series data. We project the time series data onto a vector space spanned by the identified patterns (Section V.3.2.2). Hereby, the projection scheme can vary depending on the task, e.g. to represent users of an OSN as data points in the vector space, one could combine all vectorized sessions into a single vector. Figure V.14 gives an overview of the design of our approach.

V.3.2.1 Identification of Behavior Patterns

Similar to our last contribution, we apply a mixture model consisting of an arbitrary number of patterns modeled as transition graphs. The architecture of our model has a simple and efficient design. In large parts, it follows the IMMC (cf. V.2.1). It consists of three parts:

- **Mixing proportion:** Number of and dependencies between the mixture components (*Graph of flexible size*)
- **Mixture component:** Pattern (activity and time) as a transition graph (*Graph of fixed size*)
- **Duration model:** The averaged time that elapses between events in a pattern (*Distribution over a fixed space*)

Engaging our OSN example, the mixing proportion describes the pool of possible intentions. A user behaves according to one of these mixture components at a time. At first, however, it is unknown to the observer which one the user follows. Therefore, the connection between the activity (x and p) and its corresponding mixture component is modeled by a hidden state (z). By observing a crowd of OSN users, we identify the mixture components and express our belief of which intent a user will pursue next by a simple multinomial distribution. So, using a divide-and-conquer approach, we end up with simple statistical models that describe a complex behavior (see Fig. V.8).

In comparison to the IMMC, we extend the model's capabilities with a duration model, i.e., a probability distribution to capture the time typically spend on an activity before moving on to the next one.

Duration Model In the literature, there are commonly two concepts for the notion of duration in the context of segmentation approaches, namely the average number of events per segment and the time between successive events. In the context of our running example, one would like to answer the questions: (i) how many actions will a user take in pursuit of an intention, and (ii) how much time will elapse between the user's successive actions.

The most common approach to capturing (i) is to fit a probability distribution over the number of elements that make up the corresponding segments (intentions) [95, 159]. While this approach provides a rough understanding of segment durations, it ignores information about the actions that lead to the termination of a segment. We propose a simple, elegant, and natural approach that incorporates this information. The transition graphs that model the patterns underlying the data already capture this information. By extending these graphs with additional boundary markers B (see V.3.2.1) representing the beginning and end of each segment, our model captures a natural interpretation of the duration of the segments.

The second notion of duration (ii) is also of interest to our approach. The data d_{ij} may contain important information about temporal patterns. Although mixture models would theoretically be suitable for our purpose, we discard their use due to potential problems caused by sparsely populated mixture models. We rather turn to approximations. In a preprocessing step, we fit a Gaussian mixture model Λ to all durations d_{ij} and obtain a simplified grid model

of the durations. Now the duration model of each transition in the intention graphs can be described using a distribution over the simplified grid.

$$d_{ij} \mapsto \hat{d}_{ij} \in \mathbb{R}^M,$$

where M denotes the cardinality of the identified duration space.

That is, to capture the duration model for each transition in each graph, we fit a Dirichlet distribution $\hat{\theta}$ to the corresponding duration grid space. In this way, we circumvent the problem of sparsely populated mixture models. Moreover, the approximation leads to a significant reduction in model complexity while preserving most of the information.

Combining both duration types gives the model a natural and detailed duration model. Note that for time series data without duration information, the model collapses to the mixture model described above and only the representation of the duration space is lost.

Formal Description The hierarchical structure of distributions is repeatedly applied, to model the mixture components ($\boldsymbol{\omega}_i$ and $\boldsymbol{\theta}_{ik}$), the mixing proportion ($\boldsymbol{\beta}$ and $\boldsymbol{\pi}_i$), and the duration model of each component and transition ($\hat{\boldsymbol{\omega}}_i$ and $\hat{\boldsymbol{\theta}}_{im}$). For example, a mixture component consists of a set of transition distributions. Each transition distribution $\boldsymbol{\theta}$ in the graph is modeled by a Dirichlet distribution (*Dir*). Its shape is controlled by the Dir $\boldsymbol{\omega}_i$ that describes the observation frequency of nodes in the corresponding i^{th} graph (*intention*).

The hierarchical ordering encodes our current belief about what the true transition probabilities are and how confident we are in our statements. The input of a Dirichlet distribution Dir (x) affects the distribution as follows: its mean distribution is proportional to its input vector x and its variance is inversely proportional to the mean of the values in x. During optimization, we adjust the proportions of these input vectors, and the more confidence we gain in our beliefs, the higher the mean over their values becomes.

The model of the mixing proportions and large parts of the mixture components model follow the IMMC (cf. V.2.1). In the following, we discuss extensions to the existing model, i.e., we introduce our notion of duration. It consists of a set of prior distributions, $\hat{\omega}_i$, that represent the overall duration behavior in each graph,

$$\hat{\omega}_i | \hat{\sigma}, \hat{d}_{i\cdot k} \sim \text{Dir}\left(\hat{\sigma} + \hat{d}_{i\cdot k}\right). \tag{V.27}$$

The priors $\hat{\omega}$ govern $\hat{\theta}$, the duration distributions of the transitions in the graphs,

$$\hat{\theta}_{ik_1k_2} | \hat{\lambda}, \hat{\omega}_i, \hat{d}_{ik_1k_2} \sim \text{Dir}\left(\hat{\lambda} \cdot \hat{\omega}_i + \hat{d}_{ik_1k_2}\right), \tag{V.28}$$

where, analogous to the transition model, $\hat{\sigma}$ and $\hat{\lambda}$ denote the hyper-parameters and $\hat{d}_{ik_1k_2}$ the statistics on recorded rasterized durations.
Note that $\hat{\omega}$ and $\hat{\theta}$ model the rasterized durations $\hat{d}_{ik_1k_2}$, while ω and θ model the events of the system in form of observed activities.

In combination, θ and $\hat{\theta}$ express our assumption that the next element in a time series depends on the current graph (intention), its current state (last observed activity) and the time elapsed since entering that state.

To integrate the duration information into the inference process, we update the probabilities defined by Eq. V.19.

$$p(x_j | z, p_j, \hat{d}) \propto \theta_{zp_tx_j} \cdot \hat{\theta}_{zp_tx_j\hat{d}}. \tag{V.29}$$

The modular construction of the individual parts of the model allows for an efficient optimization, where, given the hyper-parameters and an upper bound on the number of graphs, the algorithm adjusts the true number of graphs to the data.

To infer the model parameters, we make use of a truncated Gibbs sampler. Further information can be found in Section V.2.1.

V.3.2.2 Vector Representation

We now define a suitable projection for our time series data. The latent vector space, the *behavior embeddings*, is spanned using the identified patterns as its axes. It accounts for the sequential information inherent in the data while allowing the application of vector space methods, e.g., NNs, SVMs, kNNs, or other well-known approaches.

In this latent vector space, each of our example web users can be represented in different ways. For example, suppose we want to represent each user by a single point in the *behavior embeddings*. Then the projection set of a single point is all the time series of a user. It is defined as the set of time series projected onto a single point in the identified *behavior embeddings*. In this way, intentions are encoded instead of exact click sequences, i.e., a user is represented by his intentions and the related time series. Note that by choosing each time series as its own projection set, each sequence can be projected individually.

The *behavior embeddings* are defined as follows: Assume that K is the number of patterns. Then the vector space is spanned by the identified patterns, i.e., each axis corresponds to one pattern. Thus, a set of S time-series $\mathbf{X} = \{\mathbf{x}_s\}_{s \in S}$, is expressed as a point $p \in \mathbb{R}^D$,

$$p = \phi_{\mathbf{z}}(\mathbf{X}), \tag{V.30}$$

where $\phi_{\mathbf{z}}$ denotes the projection as a function of the latent variable assignments $\mathbf{z}$. The projection function accumulates time spent by a user in following a particular behavior pattern (ϕ^D), as well as the frequency of patterns shown (ϕ^F),

$$\begin{aligned} \phi^F_{\mathbf{z}}(\mathbf{X}) &\triangleq \frac{\sum_{i,j \in \widetilde{\mathbf{z}}} \mathbb{1}_{z_{ij}}}{|\widetilde{\mathbf{z}}|}, \\ \phi^D_{\mathbf{z}}(\mathbf{X}) &\triangleq \frac{\sum_{1 \le i \le S} \sum_{1 \le j \le l_i} \mathbb{1}_{z_{ij}} \hat{d}_{ij}}{\sum_{1 \le i \le S} \sum_{1 \le j \le l_i} \hat{d}_{ij}}. \end{aligned} \tag{V.31}$$

Compared to approaches that represent the data using simple BoW projections, our representation preserves much of the temporal information contained in time series data. In the context of the web user scenario, users can be compared based on their behavior rather than just the frequency of their observed actions.

V.3.2.3 Evaluating Behavior Embeddings

In the following, we evaluate the proposed behavior representations. Our analysis consists of three experiments:

- Evaluation of the modeling capabilities of the proposed approach in comparison with related vectorization approaches
- Analyzation of the validity/usefulness of the proposed behavioral representations to real-world tasks
- Demonstration of further benefits (e.g. interpretability) of our approach

A basic controlled setting allows us to illustrate and discuss the advantages and disadvantages of the different algorithms. The experiment is followed by two experiments on real-world applications. Using Twitter data [43], the performance of the algorithm in detecting anomalous accounts is examined. Finally, using an electronic textbook as an example, we mine behavior patterns for user understanding.

As baselines, we focus on the work of Viswanath et al. [178] and Wang et al. [182]. Both propose vectorizations with an abstraction mechanism to capture temporal behavior patterns.

When comparing with Viswanath et al. [178], we apply the approach following their paper as follows:

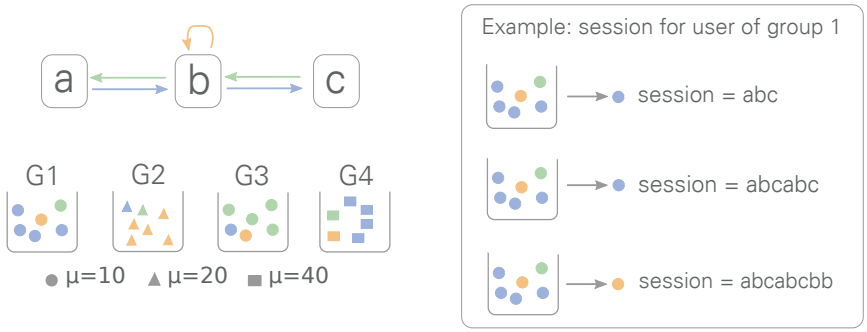


Figure V.15: Setting and sampling example used for the experiment on artificial data.

- temporal feature: histogram of clicks
- spatial feature: histogram of categories of clicks
- temporal-spatial feature: entropy of spatial features per session

Regarding Wang et al. [182], we vectorize time series data using the following 12 features:

- ∅ clicks per session
- ∅ length of a session (duration)
- ∅ duration between clicks
- total number of sessions
- histogram of 8 categories of clicks

Synthetic Experiment We create synthetic data to demonstrate the strengths and limitations of competing algorithms. To evaluate the ability to capture the temporal patterns inherent in the data, we design a simple scenario with three different behavior patterns. We further define four groups that are characterized by the frequency of the temporal patterns exhibited by their users and the speed at which the actions are performed.

The scenarios are designed to test three different types of distinguishable features: (*i*) frequency of actions, (*ii*) temporal patterns, and (*iii*) speed of execution in the pursuit of an intention. The groups are devised so that one group ($G2$) is distinguishable only by considering the frequency of activities. Two groups ($G1, G4$) exhibit similar frequency counts and can be distinguished only by considering temporal patterns. Finally, two groups ($G1, G3$) exhibit similar behavior in terms of frequency and temporal patterns. Therefore, users in these groups can only be distinguished based on the amount of time they spend on their activities.

The artificial data is generated as follows: The dataset is based on only three different behavior patterns (latent processes),

$$C = \{\text{'abc'}, \text{'bb'}, \text{'cba'}\}.$$

In addition, we define four user groups that describe the behavior of the users,

$$U = \left\{ \left[\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\right], \left[\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right], \left[\frac{1}{6}, \frac{1}{6}, \frac{2}{3}\right], \left[\frac{2}{3}, \frac{1}{6}, \frac{1}{6}\right] \right\}.$$

When generating users, group membership is assigned uniformly at random. Based on a user and the corresponding group, sessions are generated based on the frequency and the patterns themselves. A generated session might look like 'abcabcbbabcabc' and follow the patterns $[1, 1, 2, 1, 1]$ in the order shown. Each user is represented by at least 30 observed activities in

Table V.4: Performances on the classification tasks based on four artificial behavior groups

	Behavior Embeddings			Viswanath			Wang		
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
G1	1.00	1.00	1.00	0.33	0.36	0.34	0.53	0.40	0.46
G2	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
G3	0.99	0.99	0.99	0.33	0.25	0.29	0.50	0.62	0.55
G4	1.00	1.00	1.00	0.35	0.48	0.37	1.00	1.00	1.00
Avg	0.99	0.99	0.99	0.50	0.50	0.50	0.76	0.76	0.76

the dataset. For example, users in the first and fourth groups follow pattern 1 ("abc") 66% of the time, while they exhibit the other behaviors only 17% of the time each.

Finally, we also add information about the duration a user stays in a state. The dwell time of the group members in the respective states is modeled by a Gaussian distribution. Groups 1 and 3 show similar duration behavior, i.e., both Gaussians have a mean of 10 and a variance of 1. Group 2 is slower, as expressed by a shifted Gaussian mean of 20, with group 4 being the slowest with a mean of 40. Figure V.15 shows the setting and sampling process.

To evaluate the performance of the algorithms, we generate 10 000 users and split the data into training, development and test set, the latter containing 10% of the original users. Using stratified random partitioning ensures that the percentage of samples for each group is preserved. For classification, we train standard SVMs with a randomized search in the hyper-parameter space for model selection.

The obtained results show that our algorithm has no difficulty in distinguishing between the four different groups (see Tab. V.4). An F_1 value of 0.99 shows that the approach efficiently uses not only the general patterns but also the duration information. Viswanath's approach seems to fail completely in capturing the differences between groups 1, 3, and 4 (F_1-value of 0.34, 0.29, and 0.37, respectively). In contrast, Wang's method allows discrimination between groups 1 and 4, but has problems with discrimination between groups 1 and 3. The results show the shortcomings of the corresponding vectorization methods. Group 2 is easily detected by both approaches. Here, the temporal and temporal-spatial features of [178] capture the more frequent occurrence of 'b'. Given the similar event space of groups 1, 3, and 4, the same features are not sufficient to capture the differences in the order of execution. Similarly, Wang et al. [182] capture the distinct features of group 2 through both the histogram of categories and the average duration between activities. This also allows them to distinguish between groups 1 and 4. However, their approach does not capture the temporal sequence of activities. Therefore, the classification algorithm is not able to distinguish between groups 1 and 3.

Our model represents users by their observed behaviors, i.e., how often they follow certain patterns and how much time they spend doing so. The patterns themselves also take into account the time elapsed between the current and the next activity. Therefore, our algorithm captures both temporal order and duration. The results suggest, that, assuming that time series data are generated by latent processes, our algorithm is capable of recognizing these patterns and translating the results into a projection that allows us to represent arbitrary time series as points in a fixed-dimensional vector space.

Twitter Experiment While the previous experiment allowed us to assess the strengths and limitations of the different approaches, real-world applications are necessary to justify the proposed behavior representations. Therefore, we turn to real-world applications. The following measurements focus on detecting social spambots on Twitter. The task is to identify the type of account (bot or human) based on tweets from the respective accounts. The data was provided by [43]. It contains 5 912 social spambot accounts with a total of 3 602 227 tweets

Table V.5: Performance comparison to results reported in Cresci et al. [43].

Approach	Method	Precision	Recall	F_1
Human anno.	manual	0.267	0.080	0.123
Davis et al. [45]	supervised	0.471	0.208	0.289
Yang et al. [192]	supervised	0.563	0.170	0.261
Miller et al. [116]	unsupervised	0.555	0.358	0.435
Ahmed et al.[1]	unsupervised	0.945	0.944	0.944
Cresci et al. [43]	unsupervised	0.982	0.972	0.977
Our*	unsupervised	0.981	0.982	0.981

and 1 083 accounts of real users comprising 2 839 361 tweets.

Consistent with Cresci et al's approach [43], we define a set of states that distinguish between different types of tweets, e.g., simple tweet, re-tweet, or reply. By including information such as whether a tweet contains links or mentions, we obtain 24 states. For our experiment, we use a balanced dataset that includes all 1 083 accounts of benign users and a randomly selected subset of 1 083 accounts of social spambots. We use an AdaBoost classifier trained on 90% of the available data. For this, we again use a stratified split. Due to the limited informativeness of user behavior on Twitter, we extend the latent space with an additional dimension that encodes the average duration between a user's tweets (elapsed time between one tweet and the next).

We compare our approach with the results from Cresci et al.[43] (see Tab. V.5). These solutions are tailored to the task at hand and are not general approaches for arbitrary data.

The results show that our approach can keep up with the state-of-the-art algorithm even without adapting to the task. Using behavior embeddings to project time series data into a vector space, the algorithm captures sequence and duration information from arbitrary time series data.

The result shows that behavior embeddings allow for the accurate identification of known social spambots.

User Behavior in Electronic Textbooks We now describe an experiment based on an electronic history textbook. The mBook [163], which has been used in the German-speaking community of Belgium since 2013, is a project resulting from a partnership between didacticians, psychologists and computer scientists. It aims to study the impact of electronic teaching materials on students and teachers.

Before proceeding, it should be noted that count-based analyses are usually the means of approaching discrete sequential data in educational science (see [5]). Thus, patterns are defined by transition frequencies [194, 30]. The count-based approach of Bakeman et al. [5] gives us 101 frequent transitions that have a z-score greater than 1.96 for our data sample. Therefore, computing scores for longer patterns quickly become impractical and we forgo the inclusion of these baselines.

Alternative approaches group entire sequences into clusters using mixture models and Bayesian approaches [18, 149]. While these approaches lead to interesting insights, the resulting models represent averages of the included sequences and thus average out small but potentially meaningful variations. Nonetheless, we include sequence clustering as a baseline whenever possible.

We now show that the patterns extracted using our approach can serve as indicators of the psychometric variables that are collected offline. We train our approach on 5, 000 iterations with an upper bound of 20 patterns. We label the identified patterns, $C = \{C1, \ldots, C20\}$. To keep them as diverse as possible, the hyperparameters λ and ψ are set to $1/27$, while γ follows an annealing process [100], decreasing gradually from 1 to $1/20$ every 50 iterations to avoid local

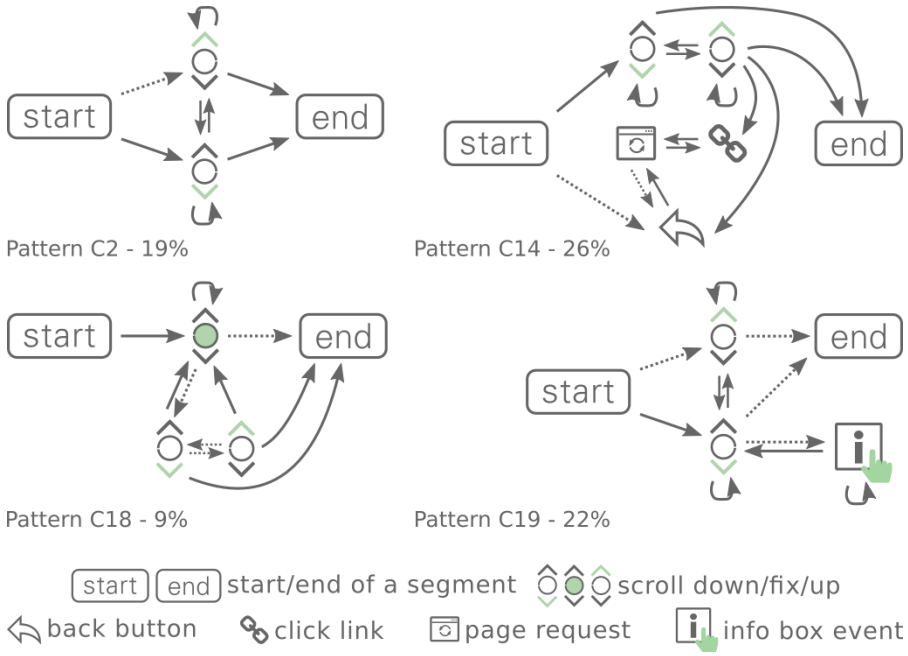


Figure V.16: The four most frequent behavior patterns of pupils interacting with a history textbook. Only the most probable transitions are displayed.

Table V.6: MAEs for predicting psychometric scores based on segmentations (our approach) and clusterings (MMC, iMMC). The scores range from 0 to 100.

	Our	MMC	iMMC
Competency	3.9	4.0	3.7
Knowledge	2.3	3.2	3.2
Motivation	3.6	9.2	9.8
IT Access	2.3	1.8	2.5
IT Skill	2.7	2.4	2.5
Avg	3.0	4.1	4.3

extrema. The hyperparameter for stickiness ρ is set to 5.

The model identifies 14 patterns with a frequency greater than 1%. The four most frequent patterns, C2, C14, C18, and C19, occurring in at least 9% of the sessions are shown in Figure V.16. Patterns C2 and C19 are long scrolling events. Pattern C19 is sometimes interrupted by clicks on information boxes where students are given additional information. The segments generated by C14 revolve around page turns. They begin either with a scroll followed by the opening of a link or directly with a press of the tablet's textitback button. Once a new page is loaded, the user either returns to the previous page or navigates through the page. The sequences generated by pattern C18 essentially consist of scrolls of less than 10 pixels. These occur when the user hesitates or reads and scrolls simultaneously.

We now test the hypothesis that user behavior can provide inferences about indicators of psychometric variables. We compare our approach to two sequence clustering baselines using a mixture of Markov chains (MMC) learned with an EM-based algorithm [18] and a Bayesian version (iMMC) of it [18].

We try to predict the score of the 5% best students on each offline test. For this, we use the individual distributions of the patterns (our approach) or the assignment probabilities to clusters (MMC and iMMC). We used linear ridge regressions to calculate the prediction. The results of a 10-fold cross-validation are shown in Table V.6.

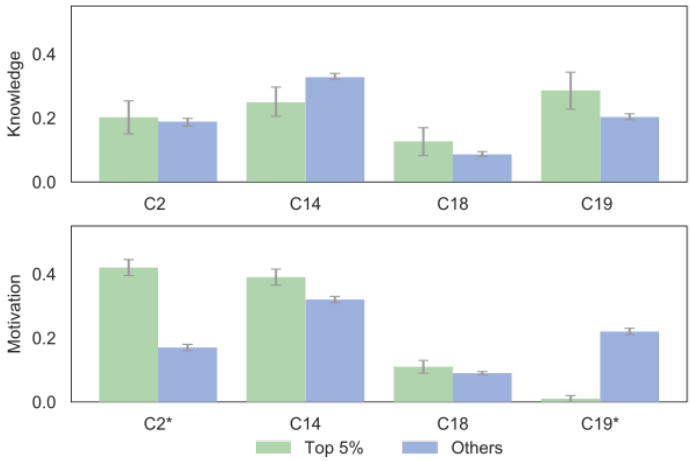


Figure V.17: Contrasting the top 5% in Knowledge and Motivations tests to their peers. Stars indicate significant differences.

All approaches obtained similar results in predicting psychometric traits, except for the motivation score, where the proposed model performed significantly better. This oddity is due to the unique behavioral features that characterize highly motivated students.

Figure V.17 shows the average pattern distributions of the top of the class (top 5%) students in terms of knowledge and motivation scores compared to the rest of the students. The error bars indicate the standard errors. Students who perform well on the knowledge test tend to follow certain patterns. However, the differences are not statistically significant. On the other hand, the behaviors of the best-motivated students in patterns C2 and C19 differ significantly from those of their peers. They interact intensively with the book and, for example, open more pages than their peers (C14). Scrolling patterns such as C2 are twice as common among these highly motivated students, and the C19 pattern is rarely used. Interestingly, this is in complete contrast to the best performance in the knowledge test.

While the baseline MMC and iMMC solutions averaged over the patterns observed in the execution order of a session, our approach divides them into segments of frequently observed behaviors. Since there are no significant differences in student behavior, all represented algorithms perform similarly. However, if a group exhibits characteristic patterns, the behavior embeddings successfully exploit these differences, while the baseline algorithms do not. These differences in predictive performance and pattern distributions support the claim of a relationship between psychometric variables and user behavior at the segment level. Further studies need to investigate whether behavior affects performance or vice versa.

V.3.2.4 Conclusion

In our final contribution to behavior modeling, we proposed a vectorization method to represent behavior in the vector space. We devised a Bayesian non-parametric approach, to segment categorical-valued time series data. The temporal order of events and durations between successive events of time series were mapped into a vector space of recurring patterns. Operating in these vector spaces facilitated inference and reduced problem complexity. Furthermore, the vectorized behavior representation allowed for augmenting the data with additional information while retaining the characteristic traits of the time series. The Bayesian setting allowed us to cope with noisy data, e.g. the irrational, biased decisions of humans observed as activity traces. It further provided a gateway to insert prior domain knowledge. By the substitution of the proposed segmentation approach with sticky HDP-HMMs [71] the framework can, additionally, be utilized for continuous-valued time series.

In experiments, we have shown the superiority of our algorithm vectorization method

compared to related algorithms. The results of real experiments, i.e. *Social Spambot Detection* and *User Understanding*, as well as synthetic experiments suggest that the proposed *Behavior Embeddings* for categorical-valued time series provide crucial modeling capabilities for inference tasks with inherent temporal patterns.

CHAPTER VI

SOCIAL BOT DETECTION

Information distribution saw three main developments, the inventions of newspapers, the Web, and online social networks. While newspapers opened a constant and reliable window to the outside world, the Web promised the *democratization of news* and online social networks (OSNs) brought us a step closer to it. Today, the costs of distributing or consuming information are close to negligible. Each evolution lowered the threshold of social and physical barriers and significantly increased news coverage and -perspectives. In consequence, it theoretically strengthened democracies.

Today, we live in a world of a sheer unlimited amount of information, perspectives, and opinions. In theory, we have technologically reached a near-perfect setting for democratic societies. Yet, while we have seen these massive developments in information distribution, we also observe increasingly complex difficulties in evaluating them. With each invention, new challenges concerning the *public opinion* arose.

A decade ago, the rise of Facebook marked the beginning of the era of the social web. Today, online social networks (OSNs), such as Facebook, Instagram, YouTube, and Twitter, are attracting enormous attention and have a nearly ubiquitous reach. Unfortunately, while we recognized the potential of online social platforms during times of political turmoil around the world, similar problems surfaced in everyday life as before. After initial praise, research began to highlight the potential negative impact on democracies [109]. Some of the studies focused on automated accounts. According to these, bots are used to spread political propaganda, manipulate discussions, or influence the popularity of users/content, among other things [167, 65, 171]. In particular, the influence of malicious bots on political opinion-forming and political discussions poses a threat to democratic societies. Today, on average, 56% of OSN users are concerned about online false news [90]. While false news is predominantly created by human authors [177], natural language processing (NLP) is catching up.

GROVER, a language processing model based on the architecture of GPT-2[1] outputs automatically generated text that is more trustworthy than human-written false news [197]. These results suggest that the automatic generation of trustworthy propaganda on a large scale is within reach.

While both humans and bots are more likely to spread false news than factual news (cf. Vosoughi et al. [179]), bots significantly accelerate the spread of false news. Shao et al. [167] reported how bots target influential users via replies and mentions, reinforcing the early stage of extended spread.

Thus, in recent years, research and society have recognized that bots play a key role in the context of malicious behavior in OSNs. In the face of reported election meddling[2], reliable detection of automated accounts is an essential building block for healthy public opinion.

So, what is the state-of-the-art in social bot detection? In many analyses, scientists resort to heuristics. Often, suspended accounts are interpreted as bots. However, a recent study by

[1] openai.com/blog/better-language-models/
[2] https://reut.rs/3AovisG

Majó-Vázquez et al. [111] reports that less than 1% of the suspended accounts were suspected or potential bots. In line with other research, they found that suspended accounts pursued specific polarizing political agendas.

Taking a look at other fields that rely on reliable bot detection (e.g., social science), we see that the 'Botometer' is the go-to choice [146]. While often used by scientists, research shows how limited this approach is [56, 146]. Regarding Botometer, Twitter remarked that binary judgments have real potential to poison our public discourse.[3]

General bot detection seems to be an unsolved problem. The detection of known bot types can be solved with a labeled data set and a state-of-the-art classification approach. However, while the authors of Botometer report about near-perfect detection performances [44], Echeverria et al. [56] argue that the established evaluation methods are rigged and, thus, reported performance results are misleading. When the goal is to distinguish between automated and manual accounts, the detection performance regarding known bot types may be interesting but is not a reliable statement of the detection rate of automated accounts in general. Additionally, this approach leads to an arms race between development and detection [42]. While Echeverria et al. [56] proposed a new evaluation scheme to measure the detection performance of unknown bot types, we lack an approach for reliable detection of these types of bots.

In the following, we contribute to the discussion on social bot detection with a novel approach for generic bot detection. The results of the following study are to be published in a conference article at the international conference on BigData (IEEE BigData).
Recent bot detection algorithms [44, 193] are optimized based on collected and labeled data sets of bot- and benign accounts. These models are thus trained and tested on the same pool of bots.

Echeverria et al. [56] emphasized that these approaches do not generalize. They overfit due to a feature selection process that focuses on the best combination of features for a given data set. As a result, they often include information that exploits artifacts in the data, which reduces generalization to other types of bots.

Our next contribution focuses on the detection of unknown bot types. To detect unknown bots and break the arms race, we have to shift to general bot detection approaches. We pose the following question:

RQ: Is it possible to define/identify generic bot behavior that enables generalized bot detection on Twitter?

Therefore, we assume that bot behavior manifests itself in the form of patterns in aggregated activity data, and consider only behavioral characteristics. We ignore information that only exploits artifacts of specific bot types in the data (e.g., username length or profile description), although this can improve performance in detecting specific bot families.

To achieve the best possible generalization, we use an ensemble of neural networks that filter different aspects of the available information. To measure and compare the performance and generalization capabilities of our approach, we use the evaluation strategy and data sets proposed and published in Echeverria et al. [56]. The results of extensive evaluations of Twitter data sets show that our model significantly outperforms the Botometer in terms of accuracy and stability, and generalizes therefore significantly more to new, previously unknown bot families.

VI.1 The Current State of Social-Bot Detection

We start by looking at the current approaches to bot detection. Early approaches examined spam-related topics on the social web. Benevenuto et al. [11] collected a data set of Twitter usage. They manually labeled users as spammers or non-spammers and proposed an SVM classifier for detection.

[3] https://blog.twitter.com/en_us/topics/company/2020/bot-or-not

Table VI.1: List of feature-sets used in our studies; *User centric*: a collection of statistics on the tweeting behavior of a user; *Content centric*: statistics on the content of a user's Tweets combined with a machine-readable summary of the content; *Response-centric*: statistics of how others reacted towards the content of an account.

Categories	Features
User-centric	# Tweets, Lifetime, Ø Duration (Tweets), # Statuses, # Friends, # Followers, # Favorites, # Listings
↪ Tweet-behavior	# Original Tweets, # Retweets, # Replies, # Quotes Ratio of Tweets, -Retweets, -Replies, -Quotes
Content-centric	# Mentions, # Hashtags, # URLs, # Domains
↪ Mean-Tweet	Ø Tweet-length, Ø Mentions, Ø Hashtags, Ø URLs, Domaindiversity, Ø Pictures, Ø Geo-locations
↪ Text	BERTweet (vectorized Tweets)
Response-centric	# retweeted Tweets, # received Replies, # favorited Tweets, Ø retweeted Tweets, Ø received Replies, Ø favorited Tweets

To help human users understand who they are communicating with, Chu et al. [35] developed a model for identifying accounts as human, bot, or cyborg (i.e., bot-assisted human or human-assisted bot). Their approach consisted of a four-component model that combined entropy and machine learning-based information with account characteristics into a final decision-maker component.

To make social bot detectors available for the general public, Davis et al. [44] launched the Botometer (former BotOrNot) service in 2014. The free social bot assessment service uses more than 1000 features.

Then, in 2017, Cresci et al. [42] reported a new type of bot, called social bots. Empirical studies suggested that humans and state-of-the-art detection approaches performed poorly in detecting these new bots because they closely mimicked benign user behavior. Research, therefore, examined the larger context and highlighted another promising approach to the task: collective behavior detection.

In-depth analyses of the cybercriminal ecosystem on social web platforms [191, 174, 74] provided detailed information about the activities and scale of criminal accounts on Twitter and Facebook. The researchers recognized that coordinated campaigns often operate through the same set of accounts. Therefore, in cooperation with Facebook [27], Renren [181], or YouTube [107], researchers proposed models that leverage detailed data on social network account activities. These models detect coordinated behavior patterns caused by malicious campaigning on the platforms.

For example, Chavoshi et al. [31] assumed that humans are not able to be highly synchronized over a long period. Therefore, they proposed an activity correlation model that does not require labeled data.

However, recent work has identified serious limitations of studies across the discipline [56, 176]. Echeverria et al. [56], for one, discussed the established evaluation scheme for bot detection approaches. They emphasized the lack of generalization when approaches are trained and tested on the same pool of bot data. Therefore, they proposed a Leave-One-Botnet-Out evaluation strategy (LOBO). Based on a collection of real-world data sets, the model measures the generalization capability of approaches by running tests on held-out bot types, i.e., bot types that were ignored during optimization. The results of the Botometer algorithm, e.g., suggest

that modern approaches that use metadata do indeed fail in detecting new types of bots.

Vargas et al. [176] on the other, challenged the assumption that humans do not act in a highly synchronized manner. They showed that coordination is indeed not uncommon in Twitter communities. With a high detection rate for malicious coordination, 46% of legitimate coordinated activity was misclassified.

In this work, we, therefore, investigate whether generalized bot detection based on account activities rather than coordinated campaigns can achieve high detection rates in previously unknown bot families.

VI.2 Designing a General Bot Detector

We can divide most bot detection models into two general groups. One uses the content and metadata of individual accounts on social networks [97, 44, 56]. The other uses coordinated activities in the network [181, 27, 107]. Recent studies have shown that the basic assumptions underlying coordinated behavior approaches may be flawed [176]. Therefore, we focus on the behavior of each account and ignore the coordinated behavior. However, Echeverria et al. [56] recently reported serious generalization problems with account-based metadata approaches. The introduction of variations in bot signatures – similar to encountering instances of new types – led to poor performance.

In our work, focused on unknown bots, we reconsider bot detection. We assume that the intentions of malicious activities leave detectable traces in the data. Therefore, we focus on metadata and ignore features that do not contain behavioral information. In particular, we ignore information that exploits artifacts of specific bot types. To measure the detection performance of unknown bots and potentially uncover generalization problems, we use the *Leave-One-Botnet-Out* (*LOBO*) evaluation strategy proposed by Echeverria et al. [56].

VI.2.1 Behavioral Features

Our model distills the data to identify characteristic patterns of behavior. We rely on similar metadata to related approaches, but disregard features that do not contain information about the activity. Examples include account name length or profile descriptions.

We represent behavior by a set of 33 aggregated *user-*, *content-*, and *response-centric* features (see Table VI.1). The *user-centric* data provide a general overview of an account, such as its overall lifetime, the number of Tweets published, or the number of friends and followers. It also contains information summarizing user activity by breaking down the total number of published Tweets into the number of Tweets, retweets, replies, and quotes. *Content-centric* information provides more details about an account's tweet activity. The data includes statistics about the content of Tweets, such as # of mentions shared, hashtags, or URLs. It also consists of a representation of an account's average Tweet (e.g., its Ø length or the Ø mentions, hashtags, and URLs). In this context, we define *domain diversity* as the number of unique hosts normalized to the total number of URLs. Finally, *response-centric* features contain information about the response to an account's activity. We measure response by the number of retweets, replies, and favorites an account or its average Tweet receives. In what follows, we refer to these features as metadata.

In addition, we consider the published Tweets. Therefore, we convert the raw Tweets into numerical vectors through tokenization and a BERT model. BERT, Bidirectional Encoder Representations from Transformers [49], is a sequence transduction model that replaces the recurrent layers with multi-headed self-attention and represents the state-of-the-art for various NLP tasks. Transformers can be trained much faster than recurrent or convolutional neural networks. The variant we use for our experiments, BERTweet [129], is pre-trained based on English Tweets.

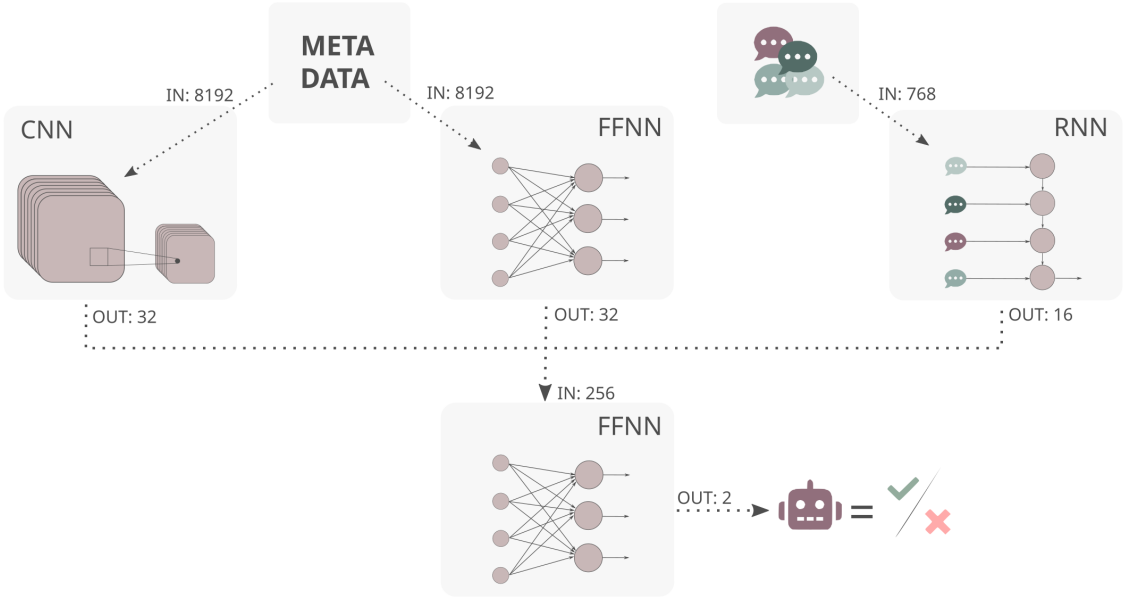


Figure VI.1: Architecture of the proposed model; Meta-data is processed by a CNN and FFN separately; vectorized Tweets are run through an RNN; the outputs are combined by a final feed-forward unit.

VI.2.2 Model Architecture

In addition to feature selection, model architecture also plays a crucial role in abstraction. We chose a standard feed-forward neural network (FFNN) and a convolutional neural network (CNN) as candidates for processing the metadata. The latter is because it is capable of highlighting certain feature combinations. The candidates for text processing were a standard recurrent neural network (RNN) and a long short-term memory RNN (LSTM). We found that the metadata performance of different architectures varied depending on the bot types tested. Therefore, we assume that they provide different generalizations of the data. For texts, the simple RNN consistently performed better than the more sophisticated LSTM. Overall, we obtained the highest average and peak performance by combining the different metadata approaches with the RNN to form an ensemble model of neural networks. Figure VI.1 shows the final architecture.

The resulting model consists of (1) an FFNN, (2) a CNN, and (3) an RNN component combined by a final FFNN. Here, each of the 3 components processes the provided data and outputs a *summary*, i.e., a numerical vector. Depending on the architecture (FFNN vs. CNN), the model appears to take into account different aspects of the data.

Metadata is processed by an FFNN and a CNN in a normalized and standardized form. The FFNN component consists of 9 fully-connected layers arranged in a funnel shape. The 33-dimensional input vector (metadata) is connected with the first, 8192-dimensional (2^{13}) network layer. Accordingly, the layer sizes are $\{2^{13}, 2^{12}, \ldots, 2^5\}$, resulting in an 32-dimensional output vector.

In the CNN framework, the 33-dimensional input data is extended to 1024 dimensions using a linear layer. The convolutional network consists of a single 1D-convolutional layer with 30 output channels and a kernel size of 3. This is followed by a *1D-MaxPool* layer, also with a size of 3. The output of the 30 channels is flattened and passed through a linear component-layer normalization combination. The length of the resulting output vector is 32.

Tweet texts are processed by an RNN. To obtain numerical vectors, Tweet texts are pre-

processed with a transformer model. BERTweet transforms each Tweet into a 768-dimensional numerical vector. Then, a single RNN unit processes all transformed Tweets (768 × #Tweets) from a user and returns a 16-dimensional vector (final state of the RNN).

Combined are the three components by a final FFNN. Thus, we concatenate the outputs of the experts. The 80-dimensional (32 FFNN + 32 CNN + 16 RNN) input is fed into a 256-dimensional (2^8) network layer. We arranged the hidden layers in a funnel shape, with layer sizes $\{2^8, 2^7, \ldots, 2^4\}$. A final linear layer (16 to 2) and a softmax unit give the final result.

VI.3 Evaluation of the Detection Performance

In this section, we report evaluation results focusing on whether it is possible to define/identify generic bot behavior to enable generalized bot detection. To this end, we compare our model to state-of-the-art algorithms to evaluate its generalization capabilities. We use the model proposed by Echeverria et al. [56] and the popular Botometer as baselines. The evaluations are performed according to the LOBO scheme with balanced data sets. In addition to comparing with state-of-the-art approaches, we investigate the performance of our model in more detail, i.e., considering additional metrics and using performance progression results.

VI.3.1 Evaluation Methodology: Leave-One-Botnet-Out

We are interested in measuring the detection performance in the context of variations in the signatures of different bot types. These variations can lead to poor performance when encountering instances of new types of bots. By using a *metadata set* consisting of different real bots, such a scenario can be simulated.

VI.3.1.1 Methodology

While previous approaches used the same data basis for optimization and evaluation, *Leave-One-Botnet-Out* (*LOBO*) [56] relies on a collection of different real-world bot data sets. Bot types range from traditional- to social spam bots, to honeypot bots, to bots that attack individuals. The evaluation process, which is similar to cross-validation in its approach, then proceeds as follows: We optimize a model based on a training set with data samples from all but one bot type, augmented by an equal number of samples of benign users. Performance is measured on a data set consisting of samples of the withheld bot type balanced with benign user samples. We repeat the process for all bot types.

In addition, Echeverria et al. [56] proposed sub-sampling of bot data to ensure that each bot type is represented with the same number of data samples during training. Realizing that one does not always have the advantage of a large bot data corpus, they set the sample size to 500 ($C500$) and excluded all bot data with less than 500 samples from the evaluations.

We use their evaluation method, with minor adjustments: While we follow their strategy of excluding data of bot types with less than 500 samples for training, we still include them in the measurement of detection performance.

VI.3.1.2 Data

The subsequent evaluations use data from 20 real-world data sets. The different data sets contain bot types, ranging from political bots, phishing bots, content polluters, or fake followers to silent accounts. An overview can be found in Table VI.2.

The *metadata set* was published in Echeverria et al. [56] and contains content from various bot data sets. Some of these are from research [55, 42, 79], while others were reported by

Table VI.2: Overview of the data sets, information on their size, whether they were used for the Botometer optimization, and how they are used in our experiments (Train/Test).

Name	Size	Botometer	Train	Test
Social Spambots 1 (SSB1)	551	✓	✓	✓
Social Spambots 2 (SSB2)	3 320	✓	✓	✓
Social Spambots 3 (SSB3)	458	✓	×	✓
Traditional Spambots 1 (TSB1)	872	✓	✓	✓
Traditional Spambots 2 (TSB2)	1	✓	×	✓
Traditional Spambots 3 (TSB3)	283	✓	×	✓
Traditional Spambots 4 (TSB4)	977	✓	✓	✓
Fake-followers FSF	33	✓	×	✓
Fake-followers INT	64	✓	×	✓
Fake-followers TWT	624	✓	✓	✓
Human Annotated 1k (B1k)	387	✓	×	✓
Human Annotated 100k (B100k)	534	✓	✓	✓
Human Annotated 1M (B1M)	229	✓	×	✓
Human Annotated 10M (B10M)	26	✓	×	✓
Darpa	2 521	×	✓	✓
Attack on Brian Krebs (Krebs)	728	×	✓	✓
Attack on Ben Nimmo (Nimmo)	1 558	×	✓	✓
StarWars Bots	357 000	×	✓	✓
Bursty Bots	500 000	×	✓	✓
DeBot	700 000	×	✓	✓

journalists who fell victim to a botnet-attack. The data is supplemented by an equal number of benign user samples. Each sample includes information on the user profile and published Tweets.

Spam bots range from the simplest bot type (TSB) to sophisticated social spambots (SSB) that mimic real user behavior. The TSB data sets consist mainly of bots used for traditional spam campaigns (TSB1, TSB2), with two of them (TSB3, TSB4) specifically spreading job offers.

SSB records contain accounts that mimic the behavior of real users, which makes bots more difficult to detect. Here, SSB1 consists of spammers of paid apps for mobile devices, while others (SSB2) retweet content from an Italian politician and (SSB3) promote products on Amazon.

All data sets (TSB, SSB) were previously used in Cresci et al. [42] and Echeverria et al. [56].

Fake-follower bot types consist of accounts that can be purchased by customers to follow their accounts to push them in visibility. The corresponding data sets contain fake followers from different services (fastfollowerz (FSF), intertwitter (INT), and twittertechnology (TWT). These types of accounts can be identified by synchronized behavior, but are very difficult to detect by behavioral analysis. For more information see Cresci et al. [41].

Attack-bots are Twitter accounts that participated in an attack on two journalists, Brian Krebs and Ben Nimmo (Krebs, Nimmo), in 2017. The journalists logged and published a list of the

Twitter accounts involved[4].

Campaign-bots are bots detected by a bot detection service (DeBot) [31, 32]. The service provides daily reports on bot activity, focusing on warped correlation in Tweet timings of different accounts. Echeverria et al. [56] used the API to query over 700 000 accounts that were identified as bots. Therefore, the data set represents a potentially noisy sample as it is based on real detection results.

Mixed-bots contain data sets that were labeled by humans or were captured by honeypots (Darpa [171]). Thus, they may contain different types of bots. The different manually labeled bot accounts are grouped by the size of their followings:
B1k → follower counts between 900 and 1 100.
B100k → follower counts between 90 000 and 110 000.
B1M → follower counts between 900 000 and 1 000 000.
B10M → follower counts over 9 000 000.

Other bots finally belong to none of the above categories. These two data sets (StarWars, Bursty) contain samples of discovered botnets, one quoting from Star Wars novels and the other luring users to dubious websites through mentions. The StarWars bots were all created during a small window of time and have only a small number of friends and followers. The Bursty bots, on the other hand, all have similar characteristics in terms of a lifetime (only a few Tweets shortly after account creation), with no friends or followers. Both were reported by Echeverria *et al.* in Echeverria et al. [57], Echeverria and Zhou [55].

VI.3.1.3 Optimization

Our model consists of one CNN, one RNN, and two FFNN units.The hyper-parameters given are the result of an exhaustive model-selection process. All layers have a dropout ratio of 0.3 and use a *LeakyReLU* activation function, with only the recurrent layer using a *ReLU* function. The learning rate is fixed at 10^{-5}. Our models are trained to convergence, with a simulated annealing strategy to adjust the learning rate during training.

VI.3.2 Performance Comparison

In this section, we report on the performance of the algorithms. Our goal is to measure their abstraction capabilities, i.e., their detection performances on new, previously unknown bot types. Unlike related work, the feature selection of our algorithm is limited to behavioral information to obtain more abstract representations. In addition to the baseline comparisons, we are also interested in the impact of the different information sources. Therefore, we report the performance of an FFNN model (referred to as *META*) fed only with metadata information. In initial experiments, we have already excluded the Tweet-text-only approaches as they showed worse performance.

Besides the META model, all other algorithms are provided with the same information sources. Note, however, that Echeverria's model uses information extracted from Tweet texts but does not use NLP approaches for text understanding. The test data is only used for the final evaluation, not for model selection.

In our comparison, we include Echeverria's approach[56] as a representative of the algorithms using all features with current classification approaches for detecting unknown bots. We also compare against the Botometer to highlight the shortcomings of the current go-to approach.

[4] https://krebsonsecurity.com/tag/twitter-bots/

Table VI.3: General Performance: Results of evaluations of Botometer (Bmeter), Echeverria's model, our META model (only using meta information), and our ensemble of experts; evaluations are split into 6 groups of bot sets; average accuracy and standard deviation of the approaches are at the bottom.

Data Set	Botometer	Eche	**META**	**Expert**
SSB1	0.924	0.492	0.763	**0.949**
SSB2	**0.994**	0.007	0.871	0.938
SSB3	**0.941**	–	0.745	0.919
TSB1	**0.983**	0.022	0.750	0.846
TSB2	**1.0**	–	0.893	**1.0**
TSB3	0.661	–	0.566	**0.827**
TSB4	**0.978**	0.020	0.789	0.948
FSF	**1.0**	–	0.474	0.909
INT	**1.0**	–	0.563	0.891
TWT	**0.953**	0.888	0.698	0.757
B1k	0.209	–	0.821	**0.875**
B100k	0.109	0.660	0.717	**0.798**
B1M	0.013	–	0.688	**0.883**
B10M	0.000	–	0.476	**0.980**
Darpa	0.277	0.779	0.680	**0.835**
Krebs	0.831	–	**0.861**	0.817
Nimmo	0.591	**0.898**	0.807	0.754
StarWars	–	0.620	0.601	**0.949**
Bursty	0.028	**0.981**	0.898	0.975
DeBot	0.077	0.848	0.720	**0.862**
Avg. Acc.	0.609	0.565	0.719	**0.886**
↪ std	0.406	0.361	0.126	**0.071**

Note however that the approach can only be evaluated through the provided API. Thus, training data cannot be controlled and indeed violates the LOBO strategy. Nevertheless, due to its popularity as a bot detection service, we include the Botometer as a baseline. We report on Botometer results published by Echeverria et al.[56] using the model accessible through the public API. Note that experiments with bot types included in the training set can be interpreted as loose upper bounds. For an overview of the data used to optimize the Botometer model[5] see Tab. VI.2.

Echeverria Tab. VI.3 reports the evaluation performances. Overall, Echeverrias' model shows severe performance issues when detecting spam-bots (SSB, TSB). While the average accuracy on other bot types is 0.811, the performance on spam-bots is 0.135, only.

Botometer The Botometer model performs best with an average accuracy of 0.697 on bot types known from training (remember: in violation with the LOBO evaluation strategy) (including acc. of SSB, TSB, FSF, INT, TWT, B1k, B100k, B1M, B10M). It performs poorly on human-annotated bots (B1k, B100k, B1M, B10M) with an average accuracy of 0.083, while showing peak performance on the other known bot types (Ø 0.943). When detecting

[5] https://botometer.osome.iu.edu/bot-repository/datasets.html

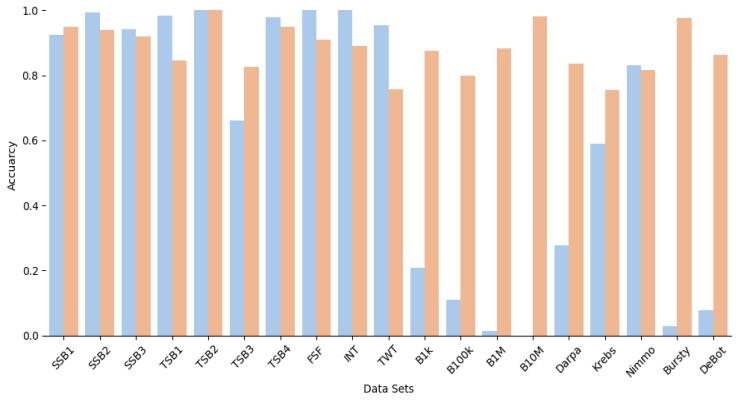


Figure VI.2: Accuracy comparison between the ensemble of experts model (red) and the Botometer (blue).

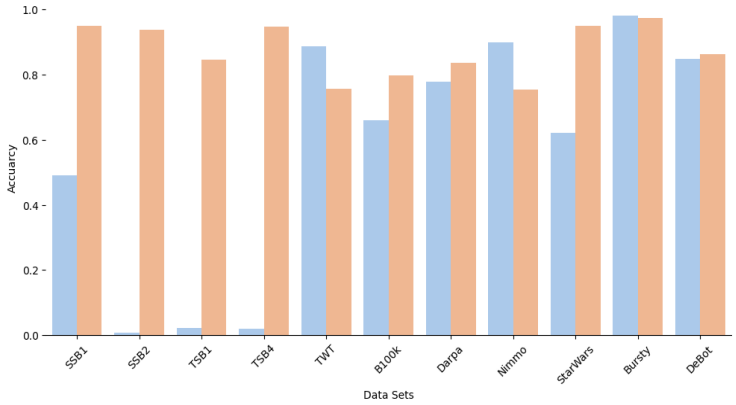


Figure VI.3: Comparison between the ensemble of experts model (red) and Echeverria's approach (blue).

unknown bot types, the model accuracy drops drastically to 0.361.

Interestingly, according to these results, the Botometer actually outperforms the model proposed by Echeverria et al. [56] in terms of average accuracy (0.609 to 0.565). We also note that the performance of both approaches varies significantly depending on the bot type (standard deviation (std): 0.406 and 0.361).

META model Our metadata-based META model achieved the best detection rates compared to the other sub-models (CNN + metadata, RNN + text), with the Tweet texts model performing the worst. With the feed-forward component, we achieve accuracy between 0.474 and 0.898. While peak performance is significantly below baselines (≤ 0.526) in some cases, the model is more stable and outperforms baselines in terms of average performance (across all bot types), i.e., 0.719 compared to 0.609 and 0.565. Moreover, a standard deviation of 0.126 indicates a better generalization than baselines. This is especially true for fake and spam bots.

Ensemble of Experts Our ensemble model additionally incorporates information from Tweet texts extracted using BERTweet. The model achieves an average accuracy of 0.886 with a standard deviation of 0.071. Compared to the Botometer model (cf Fig. VI.2) or Echeverria's model (cf Fig. VI.3), it increases the overall performance by 45.48% and 56.81%, respectively. In 11 of 20 experiments, it outperforms the baselines and never shows serious performance degradation. As for the peak performance, the worst performance degradation is 0.196 on

Table VI.4: Average detection accuracy w.r.t. the bot categories.

Categories	Bmeter	Eche	**META**	**Expert**
Spam	**0.926**	0.135	0.768	0.918
Fake	**0.984**	0.888	0.578	0.852
Attackers	0.711	**0.898**	0.834	0.786
Campaigns	0.077	0.848	0.720	**0.862**
Mixed	0.122	0.720	0.676	**0.874**
Other	0.028	0.801	0.750	**0.962**
Avg. acc.	0.475	0.715	0.721	**0.876**
↪ std	0.408	0.266	0.080	**0.055**

TWT (compared to Botometer), while the largest gain is 0.980 on B10M.

VI.3.3 Bot Categories

Next, we report on the performance w.r.t. the different bot categories (see Tab. VI.4). Our approach yields the most stable results with competitive performances in all categories. Interestingly, in contrast to the average performance results in the previous section, in this section, Echeverria's model significantly outperforms the Botometer and indeed achieves similar results to our approach, except in the spam category.

The results show the instability of the Botometer, which delivers top performance on spam and fake bots, but fails on campaign-, mixed-, and other bots. Our META model already delivers decent performance across the board (0.578 – 0.834), but fails to deliver consistent peak performance. The results of this model highlight the importance of feature selection and confirm our assumption that bots can be detected based on behavioral data.

Overall, the performance of our approach is significantly more stable than related work, suggesting better generalization capabilities.

VI.3.4 Performance Details

In the following, we focus exclusively on the ensemble of experts model to obtain a differentiated understanding of its performance. Thus, more information on the bot detection results can be found in Tab. VI.5.

While *accuracy* is a measure of the overall detection accuracy, we are interested in more detailed measurements. We consider the F_1 *score*, which provides information about the *precision* and *recall* of the model. Here, *recall* denotes the percentage of bots that were missed by the algorithm, while *precision* denotes the percentage of detected accounts that are actually bots. In general, misclassifying a user is a more serious error than overlooking a bot. Thus, *precision* is more crucial than *recall*.

We note that F_1 scores are similar to the accuracy measures. This is to be expected since we worked with balanced data sets. It confirms that the algorithm is generally balanced between detecting bots and detecting benign users. Nevertheless, we consider precision and recall separately.

Regarding this ratio (between precision and recall), we find that results vary slightly, with some showing similar results while others tend to have higher precision or recall. We note that the bots belonging to the same category show the same tendencies. In total, we achieve an average precision of 0.905 and an average recall of 0.856.

In general, *traditional spam bots* and *campaign bots* show balanced results. Experiments with social spam bots (SSB) and the *other bots* (StarWars, Bursty) show lower precision compared

Table VI.5: Further performance measures: F_1 score, recall, and precision of the ensemble of experts; evaluations are split into 6 groups of bot sets.

Data set	Accuracy	F_1	Precision	Recall	P/R ratio
SSB1	0.949	0.944	0.904	0.987	0.916
SSB2	0.938	0.932	0.878	0.994	0.883
SSB3	0.919	0.917	0.869	0.971	0.895
TSB1	0.846	0.848	0.850	0.846	1.005
TSB2	1.000	1.000	1.000	1.000	1.000
TSB3	0.827	0.828	0.831	0.825	1.007
TSB4	0.948	0.951	0.907	1.000	0.907
FSF	0.909	0.905	0.962	0.854	1.126
INT	0.891	0.859	0.951	0.784	1.213
TWT	0.757	0.724	0.873	0.619	1.410
B1k	0.875	0.867	0.934	0.809	1.155
B100k	0.798	0.780	0.895	0.691	1.295
B1M	0.883	0.872	0.947	0.807	1.173
B10M	0.980	0.964	0.990	0.941	1.052
Darpa	0.835	0.842	0.875	0.811	1.079
Krebs	0.817	0.783	0.893	0.698	1.279
Nimmo	0.754	0.714	0.856	0.612	1.399
StarWars	0.949	0.951	0.919	0.986	0.932
Bursty	0.975	0.944	0.905	0.986	0.918
DeBot	0.862	0.871	0.858	0.886	0.968

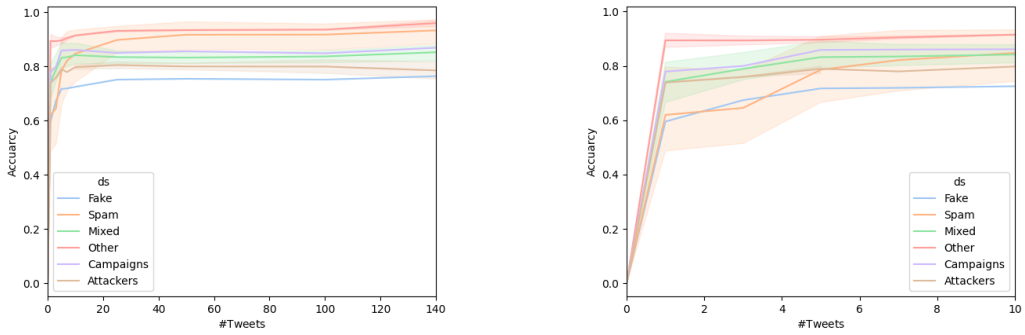
to recall. This is to be expected since social spam bots mimic human behavior and the other bots are non-commercial *concept bots* (StarWars) or are only active for a short period (Bursty). Finally, for the remaining bot types (fake followers, mixed bots), precision is the higher score. For human-labeled data sets, we have the most significant imbalance between precision (0.920) and recall (0.760).

VI.3.5 Performance Progression

Last, we investigate the performance progression of the model as a function of the number of observed Tweets. To this end, we simulate data aggregation based on a theoretical number of observed Tweets. While we can adjust the number of Tweets, we need to interpolate the corresponding metadata. Note that by doing so, we introduce a bias towards better (averaged) metadata and avoid artifacts that might be caused by small amounts of data. We perform the evaluations on all experiments and limit the number of observed Tweets to $\{1, 3, 5, 7, 10, 25, 50, 100, 140\}$.

Fig. VI.5 shows the performance w.r.t. bot groups, while Fig. VI.4a shows the progression w.r.t. categories. In both cases, the experimental results indicate that the algorithm requires only a small number of Tweets to achieve a high level of accuracy. On average, this level is reached after observing 20 Tweets. Fake and social spam bots are an exception. Here, the algorithm takes longer to collect its 'sufficient statistics'.

If we take into account the performance for the first 10 observed Tweets (cf Fig. VI.4b), we see that the algorithm only needs a single Tweet from some bot types. Due to the biased metadata, these results imply that for some bot types, statistics on behavior are sufficient for detection, while for others, Tweet texts provide valuable information. For example, the results



(a) Accuracy progression per bot category.

(b) Accuracy progression with the first 10 Tweets.

Figure VI.4: Performance progression of the classification task.

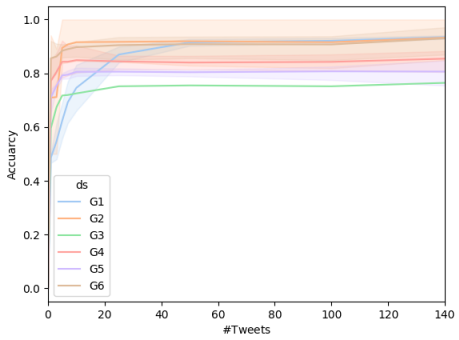


Figure VI.5: Avg. accuracy per bot group; (1) SSB, (2), TSB (3), Fake (4), Mixed (5), Darpa + Attack, (6) Campaign + Other.

suggest that the content of Tweets is particularly important for social spam bot detection, which explains the poor performance of Echeverria's model in this context. On the other hand, it seems that the detection of the StarWars- and Bursty bots does not benefit from Tweet content information.

CHAPTER VII

CONCLUSION

This thesis addresses topics surrounding social media and our democratic societies. The contributions of this paper can be broadly divided into three subsections.

- We addressed the negative impact of social networks on our democratic societies. In particular, the thesis examines the situation in German-speaking countries.
  We focused on the extent to which the trend toward the division of society and radicalization by extreme-minded parts within a group can be found in the German-speaking Twitter space.

- To better understand user behavior, we designed and investigated possible approaches to behavior modeling. Especially the assumption that unsupervised learning methods have to be used, was a major challenge here. Starting with a nonparametric clustering algorithm, we laid the groundwork for developing methods that can handle the complex, noisy, large datasets of collective user behavior.

- Vosoughi et al. [179] found that automated accounts in social networks strongly support the spread of fake news by targeting influential accounts. Since the detection of such social bots was insufficiently solved, we developed a deep learning approach that combines different neural networks into an ensemble of experts.

In the following, we will summarize our contributions and conclude our findings.

VII.1 News Consumption of the German-Speaking Twitter Community

In the United States, it can be seen that the most extreme parts of the left-wing and right-wing groupings strongly influence the dialog in social networks through aggressive behavior and a strong presence. In our work, we investigated the extent to which this development has progressed in Germany. Therefore, we focused on Twitter's German-speaking user base and their behavior. We emphasized external information sources contributing to the forming of political opinions. The goal was to estimate the influence of anti-democratic political information on the German community.

We captured the Twitter traffic of German-speaking users over two months during the 2018 European Parliament election. By utilizing the Twitter Streaming API for our systematic collection approach, we obtained a representative snapshot of the German-speaking Twitter community, comprised of 76M tweets from 6.9M users. To further study the news consumption of users, we categorized external content. The automated categorization successfully assigned categories to 97% of the URL-sharing tweets. We believe that such an approach provides a powerful tool for identifying meta-information in large-scale networks.

Information that contributes to shaping political opinion received the most responses from the German user base. The most prominent representatives include traditional news media, official government sites and political blogs from the far-right political spectrum. Due to the election period, official government information providers also received a lot of attention and were mentioned by users on the platform. Traditional content providers also went to great lengths to create sophisticated promotional networks within Twitter. News feed accounts and journalists contributed to the dissemination of articles and participated in political discourse.

Overall, external news sources attract more attention than any other content. We concluded that news content has a massive impact on the German-speaking Twitter community.

With the acquired background information, we focused on the research questions. We studied the scale and influence of anti-democratic news content. Thus, we defined the groups of controversial- and non-controversial users. Comparing their tweet behavior, we found striking differences. Moderate news providers received significant attention from the user base and contributed to a lively discussion culture. In contrast, people who consumed tendentious to extreme politically opinionated blogs were overly supportive, but users discussed their content far less. These small blogs, supporting extreme political ideologies, had a small but loyal user base that distributed their content extensively via retweets in the network. However, these were consistently ignored by others.

Further, we observed that most communities include users from all over the political spectrum. While our results provided evidence that small-sized user clusters, supporting extreme views, exist in the Twitter network, most of these communities became part of large-scale clusters. Thus, we could not confirm the existence of massive networks of ideologically segregated user groups (cf. Boutyline and Willer [20]). However, similar to the study by Zick et al. [199], the data revealed the development of a new self-assured form of echo chambers. Members of existing echo chambers (identified in early iterations of the Louvain algorithm) support their opinions in discussions with dissenters (thereby, get merged into diverse clusters).

Similar to findings reported in the 'Hidden Tribes' study [85], these politically motivated controversial users are overly active on Twitter. Despite their small size (11 128 users), they generate large amounts of tweets. Overall, people with extreme political views are well-connected and frequently engage in discussions with users that share moderate information sources. However, information on used hashtags suggests that these users propagate their opinions rather than discuss topics. Due to their high activity, this small group of users is overly influential and visible in the GTC.

Regarding our research question, the behavioral analysis of the German-speaking users showed that similar behavioral patterns could be observed especially among far-right groups. However, extreme content seems to be ignored by the middle of society. Thus, the effects on society currently appear to be less fatal than in the United States.

Our findings add to a growing body of literature on political polarization and the forming of echo chambers. We devised an innovative strategy to evaluate how small-sized political clusters become part of large-scale communities and used Twitter data to provide meaning to these structures. We highly suggest that researchers apply similar methods to conduct their studies on large-scale snapshots rather than small network samples.

VII.2 Behavior Modeling: The Bayesian Approach

We started by presenting a nonparametric Bayesian approach to modeling user behavior via clustering. The nonparametric nature of our approach allowed for the efficient adjustment to, and identification of the underlying clusters within user event data. Our model showed significant improvements over related approaches when analyzing such data. We further obtained a natural state-duration model by capturing the start- and exit distributions of the clusters. Therefore,

we capture state durations based on the dynamics of the cluster. Furthermore, representing each cluster as a Markov chain (graph) led to easily interpretable results that may impact design decisions and future developments of the respective service.

Building upon the clustering approach, we proposed a more sophisticated algorithm that allowed us to segment time series data instead of only clustering them. The nonparametric Bayesian approach performed a two-level analysis of the dynamics in discrete-valued time series. By interpreting the two levels as the hidden states of an unbounded mixture model with first-order Markov models as its mixture components, our model shows significant improvements over related approaches when analyzing time series. We obtain a natural state-duration model by representing each pattern with a Markov model. The model excels in prediction and information extraction tasks.

Finally, we proposed behavior embeddings, a Bayesian non-parametric approach, to segment and project categorical-valued time series into latent spaces. The temporal order of events and durations between successive events of time series were mapped into a vector space of recurring patterns. Operating in these vector spaces facilitated inference and reduced problem complexity. Furthermore, the vector representation of the time series allowed for augmenting the data with additional information while retaining the characteristic traits of the time series. The Bayesian setting allowed us to cope with noisy data, e.g. the irrational, biased decisions of humans observed as activity traces. It further provided a gateway to insert prior domain knowledge. By the substitution of the proposed segmentation approach with sticky HDP-HMMs [71] the framework can, additionally, be utilized for continuous-valued time series.

Experiments showed that our algorithm outperforms state-of-the-art algorithms, both compared to general approaches and compared to approaches specialized to the tasks of the experiments. The results of real experiments, i.e. *Social Spambot Detection* and *User Understanding*, as well as synthetic experiments suggest that the proposed *Behavior Embeddings* for categorical-valued time series provide crucial modeling capabilities for inference tasks with inherent temporal patterns.

VII.3 The Current State of Bot Detection

The combination of the prevailing bot detection evaluations and performance-based feature selection has led to poor generalization performance in the past. While the Botometer achieves peak performance on known bots, the learned representation of bots seems too narrow to identify new bot types.

In this work, we investigated whether it is possible to identify generic bot behavior for generalized bot detection. We devised a model based on the assumption that bot behavior manifests as patterns in aggregated behavior in the form of statistics and the content of the Tweets. In particular, we ignored information that only exploited artifacts of specific bot types in the data. Experiments on a standard feed-forward model showed that selecting features that are limited to general behavior data increases the overall generalization performance of bot detection approaches. To achieve the best possible generalization, we developed an ensemble of neural networks to combine different aspects of the information.

In general, it is complicated to classify the peak performances correctly, since most of the datasets are noisy. However, the difference regarding the generalizability of the learned bot representations is clear. The performance of our behavior-based approach significantly outperforms the others.

A look at the performance of the categories reveals the weakness of the currently preferred solution, the Botometer. With an average accuracy of 0.475 and a variance of 0.408, this method is unsuitable for detecting bots in general.

Echeverria's approach, on the other hand, shows much more consistent performance across categories. With 0.715 accuracy, it is much closer to the performance of our approach than

the Botometer. However, the use of all available information seems to lead to a too narrow representation of general bot behavior. This is also indicated by a relatively high variance of 0.266.

Our approach shows very consistent performance across all bot types. As expected, it performed worst in detecting fake accounts, since the objective of fake accounts depends only very weakly on their behavior. Dedicated methods are preferable here.

In terms of error type, we have an average precision of 0.905 and an average recall of 0.855. Thus, a bot is overlooked more often (Type 2 error) than a user is detected as a bot (Type 1 error). Only in the case of SSB, Debot, StarWars and Bursty does the more expensive type 1 error occur more frequently. Here, another striking finding is the imbalance between precision and recall for the bot types labeled by humans. While we have an average precision of 0.920, the recall is significantly worse at 0.760.

The performance difference between our method and Echeverria's method concerning social spambots suggests the importance of tweet content in detecting these bot types. Here, a semantic understanding of textual information appears to be critical for consistent competitive performance.

The results of our extensive experiments suggest that generic bot behavior can be extracted and used for reliable bot detection. Using more general features combined with a BERT model to incorporate textual information yields competitive performance with better consistency across bot types. Especially in networks like Facebook, which offer the user a larger action space, this seems to be possible with sufficient accuracy. Twitter presents a more difficult task because users here have only very limited options for action.

The approaches investigated here require high-quality labeled data. Obtaining this data is an expensive and lengthy process. However, as long as clustering approaches are far behind these methods in terms of performance, this is the only realistic way.

Final Thoughts...

The relationship between social media and our Western democracies is complex. While Germany, for example, currently seems more resistant to the divisive forces of social networks than, say, the U.S., this problem is not national but global. We don't seem to have much time to rethink how we use social media. By now, everyone is aware of how much power there is in social media, and a corresponding amount of effort and money is being invested to harness that power.

BIBLIOGRAPHY

[1] Faraz Ahmed and Muhammad Abulaish. A generic statistical approach for spam detection in online social networks. *Computer Communications*, 36(10-11):1120–1129, 2013.

[2] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 1974.

[3] Alberto Ardèvol-Abreu, Trevor Diehl, and Homero Gil de Zúñiga. Antecedents of Internal Political Efficacy Incidental News Exposure Online and the Mediating Role of Political Discussion. *Politics*, 2019.

[4] Farzindar Atefeh and Wael Khreich. A Survey of Techniques for Event Detection in Twitter. *Computational Intelligence*, 2015.

[5] Roger Bakeman, Byron F Robinson, and Vicenç Quera. Testing sequential association: Estimating exact p values using sampled permutations. *Psychological methods*, 1(1):4, 1996.

[6] Frank B Baker. *The basics of item response theory*. 2001.

[7] Pablo Barberá, John T Jost, Jonathan Nagler, Joshua A Tucker, and Richard Bonneau. Tweeting From Left to Right: Is Online Political Communication More Than an Echo Chamber? *Psychol. Sci.*, 2015.

[8] Matthew Beal and Praveen Krishnamurthy. Gene expression time course clustering with countably infinite hidden markov models. *arXiv preprint arXiv:1206.6824*, 2012.

[9] Matthew J Beal, Zoubin Ghahramani, and Carl E Rasmussen. The infinite hidden markov model. In *Advances in neural information processing systems*, pages 577–584, 2001.

[10] Ron Begleiter, Ran El-Yaniv, and Golan Yona. On prediction using variable order markov models. *Journal of Artificial Intelligence Research*, 2004.

[11] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on twitter. In *Collaboration, electronic messaging, anti-abuse and spam conference (CEAS)*, page 12, 2010.

[12] James Benhardus and Jugal Kalita. Streaming Trend Detection in Twitter. *International Journal of Web Based Communities*, 2013.

[13] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

[14] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 2003.

[15] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast Unfolding of Communities in Large Networks. *Journal of statistical mechanics: theory and experiment*, 2008.

[16] Pablo Boczkowski, Eugenia Mitchelstein, and Mora Matassi. News Comes Across When I'm in a Moment of Leisure: Understanding the Practices of Incidental News Consumption on Social Media. *New Media Soc.*, 2018.

[17] Alexander Bor and Michael Bang Petersen. The psychology of online political hostility: A comprehensive, cross-national test of the mismatch hypothesis. *American political science review*, 2022.

[18] Ahcène Boubekki, Ulf Kröhne, Frank Goldhammer, Waltraud Schreiber, and Ulf Brefeld. Data-driven analyses of electronic text books. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pages 362–376. Springer, 2016.

[19] Antoine Boutet, Hyoungshick Kim, and Eiko Yoneki. What's in Twitter, I Know What Parties are Popular and Who You are Supporting Now! *Social Network Analysis and Mining*, 2013.

[20] Andrei Boutyline and Robb Willer. The Social Structure of Political Echo Chambers: Variation in Ideological Homophily in Online Networks. *Political Psychology*, 2017.

[21] William J. Brady, Julian A. Wills, John T. Jost, Joshua A. Tucker, and Jay J. Van Bavel. Emotion shapes the diffusion of moralized content in social networks. *Proceedings of the National Academy of Sciences*, 2017.

[22] Duncan P Brown. Efficient functional clustering of protein sequences using the dirichlet process. *Bioinformatics*, 2008.

[23] Ceren Budak, Anitha Kannan, Rakesh Agrawal, and Jan Pedersen. Inferring User Interests From Microblogs. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2014.

[24] Peter Bühlmann and Abraham J Wyner. Variable length markov chains. *The Annals of Statistics*, 1999.

[25] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of navigation patterns on a web site using model-based clustering. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–284. ACM, 2000.

[26] Li-Juan Cao and Francis Eng Hock Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks*, 14(6): 1506–1518, 2003.

[27] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, pages 477–488, 2014.

[28] Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. 2010.

[29] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2010.

[30] Kuo-En Chang, Chia-Tzu Chang, Huei-Tse Hou, Yao-Ting Sung, Huei-Lin Chao, and Cheng-Ming Lee. Development and behavioral pattern analysis of a mobile guide system with augmented reality for painting appreciation instruction in an art museum. *Computers & Education*, 71:185–197, 2014.

[31] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. Debot: Twitter bot detection via warped correlation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 817–822. IEEE Computer Society, 2016.

[32] Nikan Chavoshi, Hossein Hamooni, and Abdullah Mueen. Identifying correlated bots in twitter. In *International Conference on Social Informatics*, pages 14–21. Springer, 2016.

[33] Wei Chen, Chi Wang, and Yajun Wang. Scalable Influence Maximization for Prevalent Viral Marketing in Large-Scale Social Networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010.

[34] Yen-Chi Chen. A tutorial on kernel density estimation and recent advances. *Biostatistics & Epidemiology*, 2017.

[35] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Who is tweeting on twitter: Human, bot, or cyborg? In *Proceedings of the 26th Annual Computer Security Applications Conference on - ACSAC '10*, page 21, 2010.

[36] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding Community Structure in Very Large Networks. *Physical review E*, 2004.

[37] Mihaela Cocea and Stephan Weibelzahl. Cross-system validation of engagement prediction from log files. In *European conference on technology enhanced learning*, 2007.

[38] Raviv Cohen and Derek Ruths. Classifying Political Orientation on Twitter: It's Not Easy! In *Proceedings of the International AAAI Conference on Web and Social Media*, 2013.

[39] Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. Echo Chamber or Public Sphere? Predicting Political Orientation and Measuring Political Homophily in Twitter Using Big Data. *Journal of communication*, 2014.

[40] Michael D Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political Polarization on Twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2011.

[41] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. Fame for sale: Efficient detection of fake twitter followers. *Decision Support Systems*, pages 56–71, 2015.

[42] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th international conference on world wide web companion*, pages 963–972, 2017.

[43] Stefano Cresci, Roberto Di Pietro, Marinella Petrocchi, Angelo Spognardi, and Maurizio Tesconi. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 963–972. International World Wide Web Conferences Steering Committee, 2017.

[44] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th international conference companion on world wide web*, pages 273–274, 2016.

[45] Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 273–274. International World Wide Web Conferences Steering Committee, 2016.

[46] Christine DeMars. *Item response theory*. 2010.

[47] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977.

[48] Mukund Deshpande and George Karypis. Selective markov models for predicting web page accesses. *ACM transactions on internet technology (TOIT)*, 2004.

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[50] Giorgia Di Tommaso, Stefano Faralli, Giovanni Stilo, and Paola Velardi. Wiki-MID: A Very Large Multi-Domain Interests Dataset of Twitter Users With Mappings to Wikipedia. In *International Semantic Web Conference*, 2018.

[51] Cong Ding, Yang Chen, and Xiaoming Fu. Crowd Crawling: Towards Collaborative Data Collection for Large-scale Online Social Networks. In *ACM Conference on Online Social Networks*, 2013.

[52] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015.

[53] Ananya Dubey, Seungwoo Hwang, Claudia Rangel, Carl Edward Rasmussen, Zoubin Ghahramani, and David L Wild. Clustering protein sequence and structure space with infinite gaussian mixture models. In *Biocomputing 2004*. 2003.

[54] Thi V. Duong, Hung H. Bui, Dinh Q. Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Vision and Pattern Recognition, 2005*, volume 1, pages 838–845. IEEE, 2005.

[55] Juan Echeverria and Shi Zhou. Discovery, retrieval, and analysis of the 'star wars' botnet in twitter. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 1–8, 2017.

[56] Juan Echeverria, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Gianluca Stringhini, and Shi Zhou. LOBO: Evaluation of generalization deficiencies in twitter bot classifiers. In *Proceedings of the 34th Annual Computer Security Applications Conference*, 2018.

[57] Juan Echeverria, Christoph Besel, and Shi Zhou. Discovery of the twitter bursty botnet. In *Data Science for Cyber-Security*, pages 145–159. World Scientific, 2019.

[58] Arielle Emmett. Networking News: Traditional News Outlets Turn to Social Networking Web Sites in an Effort to Build Their Online Audiences. *American Journalism Review*, 2008.

[59] Sven Engesser and Edda Humprecht. Frequency or Skillfulness: How Professional News Media Use Twitter in Five Western Countries. *Journalism studies*, 2015.

[60] Nicole Ernst, Sven Engesser, Florin Büchel, Sina Blassnig, and Frank Esser. Extreme Parties and Populism: An Analysis of Facebook and Twitter Across Six Countries. *Inf. Commun. Soc.*, 2017.

[61] Stefano Faralli, Giovanni Stilo, and Paola Velardi. Large Scale Homophily Analysis in Twitter Using a Twixonomy. In *IJCAI*, 2015.

[62] Robert Faris, Hal Roberts, Bruce Etling, Nikki Bourassa, Ethan Zuckerman, and Yochai Benkler. Partisanship, propaganda, and disinformation: Online media and the 2016 us presidential election. *Berkman Klein Center Research Publication*, 2017.

[63] Albert Feller, Matthias Kuhnert, Timm O Sprenger, and Isabell M Welpe. Divided They Tweet: The Network Structure of Political Microbloggers and Discussion Topics. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2011.

[64] Thomas S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1973.

[65] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, pages 96–104, 2016.

[66] Flavio Figueiredo, Bruno Ribeiro, Jussara M Almeida, and Christos Faloutsos. Tribeflow: Mining & predicting user trajectories. In *Proceedings of the 25th International Conference on World Wide Web*, pages 695–706. International World Wide Web Conferences Steering Committee, 2016.

[67] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

[68] John R Firth. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*, 1957.

[69] Santo Fortunato and Marc Barthelemy. Resolution Limit in Community Detection. *PNAS*, 2007.

[70] Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. An HDP-HMM for systems with state persistence. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[71] Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. A sticky hdp-hmm with application to speaker diarization. *The Annals of Applied Statistics*, pages 1020–1056, 2011.

[72] Agence France-Presse. AFP Updates Guidelines on Using Social media. *AFP Newsletter [Online]*, 2013.

[73] Guillaume Gadek, Alexandre Pauchet, Nicolas Malandain, Khaled Khelif, Laurent Vercouter, and Stéphan Brunessaux. Topical Cohesion of Communities on Twitter. *Procedia Computer Science*, 2017.

[74] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Zhao. Detecting and characterizing social spam campaigns. *ACM SIGCOMM conference on Internet measurement*, page 13, 2010.

[75] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Political discourse on social media: Echo chambers, gatekeepers, and the price of bipartisanship. In *Proceedings of the 2018 world wide web conference*, 2018.

[76] Daniel Gayo-Avello. A Meta-Analysis of State-of-the-Art Electoral Prediction From Twitter Data. *Social Science Computer Review*, 2013.

[77] Yegin Genc, Yasuaki Sakamoto, and Jeffrey V Nickerson. Discovering Context: Classifying Tweets Through a Semantic Transform Based on Wikipedia. In *FAC*, 2011.

[78] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and Combating Link Farming in the Twitter Social Network. In *Proceedings of the 21th international conference on World Wide Web*, 2012.

[79] Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, Liang Wang, and Jon Crowcroft. Of bots and humans (on twitter). In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 349–354, 2017.

[80] Christophe Giraud. *Introduction to high-dimensional statistics*. 2014.

[81] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *LREC*, 2012.

[82] Peter Haider, Luca Chiarandini, and Ulf Brefeld. Discriminative clustering for market segmentation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012.

[83] Behnam Hajian and Tony White. Modelling Influence in a Social Network: Metrics and Evaluation. In *SocialCom*, 2011.

[84] Zellig S Harris. Distributional structure. *Word*, pages 146–162, 1954.

[85] Stephen Hawkins, Daniel Yudkin, Miriam Juan-Torres, and Tim Dixon. Hidden tribes: A study of america's polarized landscape. 2019.

[86] Katherine A. Heller, Yee W. Teh, and Dilan Görür. Infinite hierarchical hidden Markov models. In *International Conference on Artificial Intelligence and Statistics*, pages 224–231, 2009.

[87] Itai Himelboim, Marc Smith, and Ben Shneiderman. Tweeting Apart: Applying Network Analysis to Detect Selective Exposure Clusters in Twitter. *Communication methods and measures*, 2013.

[88] Matthew Hindman. The myth of digital democracy. In *The Myth of Digital Democracy*. 2008.

[89] Sascha Hölig and Uwe Hasebrink. Ergebnisse für Deutschland. Reuters Institute Digital News Report 2019, 2018.

[90] Sascha Hölig and Uwe Hasebrink. Reuters institute digital news report. *Ergebnisse für Deutschland. Arbeitspapiere des Hans-Bredow-Instituts*, 2020.

[91] Hemant Ishwaran and Mahmoud Zarepour. Exact and approximate sum representations for the dirichlet process. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique*, pages 269–283, 2002.

[92] Lamjed Ben Jabeur, Lynda Tamine, and Mohand Boughanem. Active Microbloggers: Identifying Influencers, Leaders and Discussers in Microblogging Networks. In *SPIRE*, 2012.

[93] Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6): 395–405, 2012.

[94] Long Jin, Yang Chen, Tianyi Wang, Pan Hui, and Athanasios V Vasilakos. Understanding User Behavior in Online Social Networks: A Survey. *IEEE Communications Magazine*, 2013.

[95] Matthew James Johnson. *Bayesian time series models and scalable inference*. PhD thesis, Massachusetts Institute of Technology, 2014.

[96] Kenneth Joseph, Peter M. Landwehr, and Kathleen M. Carley. Two 1%s Don't Make a Whole: Comparing Simultaneous Samples from Twitter's Streaming API. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, 2014.

[97] Sneha Kudugunta and Emilio Ferrara. Deep neural networks for bot detection. *Information Sciences*, pages 312–322, 2018.

[98] Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed variational dirichlet process mixture models. In *IJCAI*, volume 7, pages 2796–2801, 2007.

[99] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a Social Network or a News Media? In *WebConf*, 2010.

[100] Sridhar Lakshmanan and Haluk Derin. Simultaneous parameter estimation and segmentation of gibbs random fields using simulated annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):799–813, 1989.

[101] Andrea Lancichinetti and Santo Fortunato. Erratum: Community Detection Algorithms: A Comparative Analysis. *Physical Review E*, 2014.

[102] Andrea Lancichinetti, Mikko Kivelä, Jari Saramäki, and Santo Fortunato. Characterizing the Community Structure of Complex Networks. *PloS one*, 2010.

[103] Dominic Lasorsa, Seth Lewis, and Avery Holton. Normalizing Twitter: Journalism practice in an emerging Communication Space. *Journalism studies*, 2012.

[104] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Computational Social Science. *Science*, 2009.

[105] Chia-ying Lee, Yu Zhang, and James R Glass. Joint learning of phonetic units and word pronunciations for asr. In *EMNLP*, pages 182–192, 2013.

[106] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The Dynamics of Viral Marketing. *ACM Transactions on the Web (TWEB)*, 2007.

[107] Yixuan Li, Oscar Martinez, Xing Chen, Yi Li, and John E. Hopcroft. In a world that counts: Clustering and detecting fake social engagement at scale. In *Proceedings of the 25th International Conference on World Wide Web*, pages 111–120, 2016.

[108] Walter Lippmann. Public opinion. 1922. 1965.

[109] Brian D Loader and Dan Mercea. Networking democracy? social media innovations and participatory politics. *Information, communication & society*, pages 757–769, 2011.

[110] Philipp Lorenz-Spreen, Lisa Oswald, Stephan Lewandowsky, and Ralph Hertwig. Digital media and democracy: a systematic review of causal and correlational evidence worldwide. 2021.

[111] Silvia Majó-Vázquez, Mariluz Congosto, Tom Nicholls, and Rasmus Kleis Nielsen. The role of suspended accounts in political discussion on social media: Analysis of the 2017 french, uk and german elections. *Social Media+ Society*, 2021.

[112] Eren Manavoglu, Dmitry Pavlov, and C Lee Giles. Probabilistic user behavior models. In *Third IEEE International Conference on Data Mining*, 2003.

[113] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a Feather: Homophily in Social Networks. *Annual review of sociology*, 2001.

[114] Panagiotis Metaxas, Eni Mustafaraj, Kily Wong, Laura Zeng, Megan O'Keefe, and Samantha Finn. What Do Retweets Indicate? Results From User Survey and Meta-Review of Research. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2015.

[115] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[116] Zachary Miller, Brian Dickinson, William Deitrick, Wei Hu, and Alex Hai Wang. Twitter spammer detection using data stream clustering. *Information Sciences*, 260:64–73, 2014.

[117] Amy Mitchell, Kenny Olmstead, Kristen Purcell, Lee Rainie, and Tom Rosenstiel. Understanding the participatory news consumer. 2010.

[118] Daichi Mochihashi and Eiichiro Sumita. The infinite markov model. *Advances in neural information processing systems*, 2007.

[119] Flaviano Morone and Hernán A Makse. Influence Maximization in Complex Networks Through Optimal Percolation. *Nature*, 2015.

[120] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen Carley. Is the Sample Good Enough? Comparing Data From Twitter's Streaming API with Twitter's Firehose. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2013.

[121] Fred Morstatter, J\{}"{u}rgen Pfeffer, and Huan Liu. When is It Biased?: Assessing the Representativeness of Twitter's Streaming API. In *WebConf Companion*, 2014.

[122] Fred Morstatter, Yunqiu Shao, Aram Galstyan, and Shanika Karunasekera. From alt-right to alt-rechts: Twitter analysis of the 2017 german federal election. In *Companion Proceedings of the The Web Conference 2018*, 2018.

[123] Jordan T Moss and Peter J O'Connor. Political correctness and the alt-right: The development of extreme political attitudes. *PloS one*, 2020.

[124] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[125] Kevin P Murphy and Mark A Paskin. Linear-time inference in hierarchical hmms. *Advances in neural information processing systems*, 2:833–840, 2002.

[126] Mark EJ Newman. Fast Algorithm for Detecting Community Structure in Networks. *Physical review E*, 2004.

[127] Nic Newmann et al. Reuters Institute Digital News Report, 2019.

[128] C Thi Nguyen. Echo Chambers and Epistemic Bubbles. *Episteme*, 2020.

[129] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, 2020.

[130] Nam T Nguyen, Dinh Q Phung, Svetha Venkatesh, and Hung Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *Computer Vision and Pattern Recognition, 2005*, volume 2, pages 955–960, 2005.

[131] Nuria Oliver, Ashutosh Garg, and Eric Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding*, 96(2):163–180, 2004.

[132] Ingram Olkin and John W Pratt. Unbiased estimation of certain correlation coefficients. *The annals of mathematical statistics*, 1958.

[133] Peter Orbanz. Lecture notes on bayesian nonparametrics. *Journal of Mathematical Psychology*, 2012.

[134] Claudia Orellana-Rodriguez, Derek Greene, and Mark Keane. Spreading One's Tweets: How Can Journalists Gain Attention for their Tweeted News? *The Journal of Web Science*, 2017.

[135] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford InfoLab, 1999.

[136] Aditya Pal and Scott Counts. Identifying Topical Authorities in Microblogs. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011.

[137] Robert E Park. The natural history of the newspaper. *American Journal of Sociology*, 1923.

[138] Thomas Paul, Daniel Puscher, and Thorsten Strufe. Improving the usability of privacy settings in facebook. *CoRR*, 2011.

[139] Steve Paulussen and Raymond A Harder. Social Media References in Newspapers: Facebook, Twitter and YouTube as Sources in Newspaper Journalism. *Journalism practice*, 2014.

[140] Bogdan Pavliy and Jonathan Lewis. The Performance of Twitter's Language Detection Algorithm and Google's Compact Language Detector on Language Detection in Ukrainian and Russian Tweets. *Bulletin of Toyama University of International Studies*, 2016.

[141] Johannes Pflugmacher, Stephan Escher, Jan Reubold, and Thorsten Strufe. The german-speaking twitter community reference data set. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020.

[142] Peter LT Pirolli and James E Pitkow. Distributions of surfers' paths through the world wide web: Empirical characterizations. *World Wide Web*, 1999.

[143] Associated Press. Social Media Guidelines for AP Employees, 2013.

[144] Jorge Lopez Puga, Martin Krzywinski, and Naomi Altman. Bayes' theorem: Incorporate new evidence to update prior information. *Nature Methods*, 2015.

[145] Steve Rathje, Jay J. Van Bavel, and Sander van der Linden. Out-group animosity drives engagement on social media. *Proceedings of the National Academy of Sciences*, 2021.

[146] Adrian Rauchfleisch and Jonas Kaiser. The false positive problem of automatic bot detection in social science research. *Berkman Klein Center Research Publication*, 2020.

[147] Jörg Reichardt and Stefan Bornholdt. Partitioning and Modularity of Graphs with Arbitrary Degree Distribution. *Physical Review E*, 2007.

[148] Jan Reubold, Ahcéne Boubekki, Thorsten Strufe, and Ulf Brefeld. Bayesian user behavior models. *Proceedings of the ECML/PKDD Workshop on New Frontiers in Mining Complex Patterns*, 2017.

[149] Jan Reubold, Ahcéne Boubekki, Thorsten Strufe, and Ulf Brefeld. Infinite mixtures of markov chains. In *International Workshop on New Frontiers in Mining Complex Patterns*. Springer, 2017.

[150] Jan Reubold, Stephan Escher, and Thorsten Strufe. The latent behavior space-a vector space for time-series data. 2019.

[151] Jan Reubold, Stephan Escher, Christian Wressnegger, and Thorsten Strufe. Protecting the public opinion: In search of a general bot identifier. 2022.

[152] Jan Ludwig Reubold, Stephan Escher, Johannes Pflugmacher, and Thorsten Strufe. Dissecting chirping patterns of invasive tweeter flocks in the german twitter forest. *Online Social Networks and Media*, 2022.

[153] Reuters. Reporting From the Internet and Using Social Media. Reuters, Thomson, 2013.

[154] Matthew Richardson and Pedro Domingos. Mining Knowledge-Sharing Sites for Viral Marketing. In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002.

[155] Fabián Riquelme and Pablo González-Cantergiani. Measuring User Influence on Twitter: A Survey. *Information Processing & Management*, 2016.

[156] Gareth O Roberts and Adrian FM Smith. Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms. *Stochastic processes and their applications*, 1994.

[157] Cristobal Romero and Sebastian Ventura. Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1):135–146, 2007.

[158] Ardavan Saeedi, Matthew Hoffman, Matthew Johnson, and Ryan Adams. The Segmented iHMM: A Simple, Efficient Hierarchical Infinite HMM. *arXiv preprint arXiv:1602.06349*, 2016.

[159] Ardavan Saeedi, Matthew Hoffman, Matthew Johnson, and Ryan Adams. The segmented ihmm: A simple, efficient hierarchical infinite hmm. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2682–2691, 2016.

[160] Sergio Salmeron-Majadas, Olga C Santos, and Jesus G Boticario. Exploring indicators from keyboard and mouse interactions to predict the user affective state. In *Educational Data Mining 2014*, 2014.

[161] Tatjana Scheffler. A German Twitter Snapshot. In *LREC*, 2014.

[162] Waltraud Schreiber, Florian Sochatzy, and Marcus Ventzke. Das multimediale schulbuch - kompetenzorientiert, individualisierbar und konstruktionstransparent. 2013.

[163] Waltraud Schreiber, Florian Sochatzy, and Marcus Ventzke. Auf dem weg zu digital-multimedialen lehr-und lernmitteln für kompetenzorientiertes inklusives unterrichten und lernen. *Online Publikation, Medienberatung NRW. Zugriff am*, 11:2017, 2014.

[164] Axel Schulz, Benedikt Schmidt, and Thorsten Strufe. Small-Scale Incident Detection based on Microposts. In *ACM Hypertext and Social Media (ht)*, 2015.

[165] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, 1978.

[166] J Sethuraman. A constructive definition of dirichlet priors. In *Statistica Sinica*, pages 693–650, 1994.

[167] Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. The spread of low-credibility content by social bots. *Nature communications*, pages 1–9, 2018.

[168] Thomas S. Stepleton, Zoubin Ghahramani, Geoffrey J. Gordon, and Tai S. Lee. The block diagonal infinite hidden markov model. In *International Conference on Artificial Intelligence and Statistics*, pages 552–559, 2009.

[169] Sebastian Stier, Arnim Bleier, Haiko Lietz, and Markus Strohmaier. Election Campaigning on Social Media: Politicians, Audiences, and the Mediation of Political Communication on Facebook and Twitter. *Political communication*, 2018.

[170] Thorsten Strufe. Profile popularity in a business-oriented online social network. In *Proceedings of the 3rd workshop on social network systems*, 2010.

[171] Venkatramanan S Subrahmanian, Amos Azaria, Skylar Durst, Vadim Kagan, Aram Galstyan, Kristina Lerman, Linhong Zhu, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. The darpa twitter bot challenge. *Computer*, pages 38–46, 2016.

[172] Erik Blaine Sudderth. *Graphical models for visual object recognition and tracking*. PhD thesis, Massachusetts Institute of Technology, 2006.

[173] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. In *Journal of the American Statistical Association Vol. 101 No. 476*, 2006.

[174] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 195–210, 2013.

[175] Thea F Van de Mortel et al. Faking It: Social Desirability Response Bias in Self-Report Research. *Australian Journal of Advanced Nursing, The*, 2008.

[176] Luis Vargas, Patrick Emami, and Patrick Traynor. On the detection of disinformation campaign activity with network analysis. In *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pages 133–146, 2020.

[177] Chris J Vargo et al. The agenda-setting power of fake news: a big data analysis of the online media landscape from 2014 to 2016. *New media & society*, pages 2028–2049, 2018.

[178] Bimal Viswanath, Muhammad Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Towards detecting anomalous user behavior in online social networks. In *Usenix Security*, volume 14, 2014.

[179] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, pages 1146–1151, 2018.

[180] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.

[181] Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y. Zhao. You are how you click: Clickstream analysis for sybil detection. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 241–256, 2013.

[182] Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y Zhao. You are how you click: Clickstream analysis for sybil detection. In *Usenix Security*, volume 14, 2013.

[183] Y Wang. Forcing a Breakdown: Establishing the Limits of Community Detection Algorithms. In *Sunbelt*, 2012.

[184] Claire Wardle and Hossein Derakhshan. Information disorder: Toward an interdisciplinary framework for research and policy making. *Council of Europe*, 2017.

[185] Audrey Watters. How Recent Changes to Twitter's Terms of Service Might Hurt Academic Research. *Read Write*, 2011.

[186] Siegfried Weischenberg, Maja Malik, and Armin Scholl. Journalismus in Deutschland 2005. *Media Perspektiven*, 2006.

[187] Robert West and Jure Leskovec. Human wayfinding in information networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 619–628. ACM, 2012.

[188] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P.N. Puttaswamy, and Ben Y. Zhao. User Interactions in Social Networks and Their Implications. In *Proceedings of the 4th ACM European Conference on Computer Systems*, EuroSys '09, 2009.

[189] Sam Wineburg, Sarah McGrew, Joel Breakstone, and Teresa Ortega. Evaluating Information: The Cornerstone of Civic Online Reasoning. *Stanford Digital Repository*, 2016.

[190] Lexing Xie, Shih-Fu Chang, Ajay Divakaran, and Huifang Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden markov models. In *Multimedia and Expo, 2003. ICME'03*, volume 3, pages III–29, 2003.

[191] Chao Yang, Robert Harkreader, Jialong Zhang, Seungwon Shin, and Guofei Gu. Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter. In *Proceedings of the 21st international conference on World Wide Web*, pages 71–80, 2012.

[192] Chao Yang, Robert Harkreader, and Guofei Gu. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Transactions on Information Forensics and Security*, 8(8):1280–1293, 2013.

[193] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. Scalable and generalizable social bot detection through data selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1096–1103, 2020.

[194] Chengjiu Yin, Noriko Uosaki, Hui-Chun Chu, Gwo-Jen Hwang, Jau-Jian Hwang, Itsuo Hatono, Etsuko Kumamoto, and Yoshiyuki Tabata. Learning behavioral pattern analysis based on students' logs in reading digital books. 12 2017.

[195] Alexander Ypma and Tom Heskes. Automatic categorization of web pages and user clustering with mixtures of hidden markov models. In *International Workshop on Mining Web Data for Discovering Usage Patterns and Profiles*, 2002.

[196] Savvas Zannettou, Michael Sirivianos, Jeremy Blackburn, and Nicolas Kourtellis. The Web of False Information: Rumors, Fake News, Hoaxes, Clickbait, and Various Other Shenanigans. *J. Data and Information Quality*, 2019.

[197] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32*, pages 9054–9065. 2019.

[198] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1513–1522. ACM, 2015.

[199] Andreas Zick, Beate Küpper, and Daniela Krause. Gespaltene Mitte–Feindselige Zustände. *Rechtsextreme Einstellungen in Deutschland*, 2016.