Assessing Influential Users in Live Streaming Social Networks

by

Isaac Jones

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved October 2019 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Ross Maciejewski
Paulo Shakarian
Nitin Agarwal

ARIZONA STATE UNIVERSITY

December 2019

ABSTRACT

Live streaming has risen to significant popularity in the recent past and largely this live streaming is a feature of existing social networks like Facebook, Instagram, and Snapchat. However, there does exist at least one social network entirely devoted to live streaming, and specifically the live streaming of video games, Twitch. This social network is unique for a number of reasons, not least because of its hyper-focus on live content and this uniqueness has challenges for social media researchers.

Despite this uniqueness, almost no scientific work has been performed on this public social network. Thus, it is unclear what user interaction features present on other social networks exist on Twitch. Investigating the interactions between users and identifying which, if any, of the common user behaviors on social network exist on Twitch is an important step in understanding how Twitch fits in to the social media ecosystem. For example, there are users that have large followings on Twitch and amass a large number of viewers, but do those users exert influence over the behavior of other user the way that popular users on Twitter do?

This task, however, will not be trivial. The same hyper-focus on live content that makes Twitch unique in the social network space invalidates many of the traditional approaches to social network analysis. Thus, new algorithms and techniques must be developed in order to tap this data source. In this thesis, a novel algorithm for finding games whose releases have made a significant impact on the network is described as well as a novel algorithm for detecting and identifying influential players of games. In addition, the Twitch network is described in detail along with the data that was collected in order to power the two previously described algorithms.

For my parents, Liz Droege and Ti Jones, without whose love and support none of this would be possible.

ACKNOWLEDGEMENTS

I would like to acknowledge the support of my committee chair Dr. Huan Liu, who gave me nothing but support when I said I wanted to focus on Twitch. I am very fortunate to have such a supportive mentor who has encouraged me to chart my own course and encouraged me to take advantages of any opportunities available to me.

I would also like to acknowledge the contributions of my other committee members in the refinement of the work presented in this thesis. The ideas presented in this thesis would not be nearly as refined and well-supported by the evidence without their contributions.

It would be impossible for me to ignore the contributions to this work by the various members of the Data Mining and Machine Learning (DMML) Lab during my time here. As a member of this lab, I have had the pleasure of learning from these highly talented researchers. In particular, I would like to thank Fred Morstatter, Phillipe Faucon, Justin Sampson, Liang Wu, and Tahora Nazer; who helped me refine many ideas over countless lunches. The contributions of other members of the DMML during my time here: Lei Tang, Xufei Wang, Salem Alelyani, Ali Abbasi, Huiji Gao, Jiliang Tang, Pritam Gundecha, Shamanth Kumar, Xia Hu, Reza Zafarani, Rob Trevino, Suhas Ranganath, Suhang Wang, Jundong Li, Ghazaleh Beigi, Kai Shu, Lu Cheng, Nur Kamrudin, Ruocheng Gao, Kaize Ding, Raha Moraffah, and Ashwin Rajadesingan were also important in refining my ideas through feedback and questions during group meetings.

During my time at ASU, I also had the opportunity to work with directly with a very talented Masters student and participate in the mentoring of his research. During my time assisting Ran Wang, I learned a lot about differing research styles as well as the importance of mentorship for up and coming researchers. In addition to Masters

students, I had the opportunity to work with a number of very clever undergraduate students in cross-listed courses, research, and as a TA. In particular; Daniel Baird, Dan Howe, and Matthew Davis deserve recognition for their contributions.

In addition to these Masters and undergraduate students, I had the pleasure of mentoring Connor Aitken through his honors thesis. Connor's work will be instrumental in determining the contours of research on Twitch for some time.

Lastly, I would like to thank my dear friends here in Arizona and elsewhere, without whose guidance and support I would have been lost. In particular, I would like to thank Katherine Frank, Carrie Bass, Scott Hughes, and Brian Gleim, who have all done much of this before and were endlessly supportive. I would also like to thank Johnny Tunstill for ensuring that I never went too long without getting some exercise and never letting me forget it when I skipped too many workouts in a row.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Twitch [1] , the premiere site for live streaming the play of video games, is a nearly-untapped source for researchers to study social behavior at a large scale. As of 2014, Twitch captured approximately 43% of the $3.8 billion "Gaming Video Content" market which has since grown to $4.6 billion [2] , yet very little research has been conducted utilizing Twitch's data. Twitch sets itself apart from sites like Facebook, Twitter, and Instagram by being built around live content. This content is also open to the public, making it easy for researchers to capture the active network through the well-documented API.

At any given time, Twitch hosts thousands of streamers playing hundreds of games for millions of viewers. Is is no surprise that social network dynamics arise naturally from these interactions. In this thesis, we focus on detecting important social behaviors and designing algorithms to identify these behaviors. In the two studies so far performed, newly released games that impact viewer behavior and influential users who act as "taste-makers" on the network are detected and identified.

Unlike other social media sites, however, these user and games are not obviously based on events and personalities in the real world. Justin Bieber, for example, does not have a Twitch stream. Nor does marketing efforts on the part of game publishing companies correlate with popularity on Twitch. In fact, some of the most popular streamers on Twitch exclusively play free-to-play games like *Fortnite* and *League of Legends*. This separation from the rest of social media ecosystem indicates that not

[1] `http://twitch.tv`

[2] According to `www.superdataresearch.com`

only will the results from Twitch analysis be unique, but that new techniques are needed to properly analyze the environment.

There is no doubt that the ability to detect patterns like this are useful to both Twitch and the gaming community. Publishers already give popular streams access to games ahead of their official release date as a way of building up excitement for a game [3] , indicating that they too believe in the power of Twitch to influence consumers. Without a systematic understanding of the way that social network forces affect an environment like Twitch, however, this behavior seems capricious and arbitrary. Thus, we propose to gain such a systematic understanding through rigorous experimentation. When complete, the contributions of this research will be as follows:

- Collection and publication of the most comprehensive, longest-duration dataset of the Twitch network.
- Analysis of the impact of new-released video games on the Twitch network as well as the features that contribute.
- Identification of influential streamers on the network and their impact.
- Understanding how the influence of a particular streamer varies over time and using this understanding to determine the likely trajectory of a streamer's influence.
- Constructing a simulator of Twitch that can be used to generate synthetic Twitch snapshots.

This experimentation is enabled by collecting the largest, most up-to-date dataset of Twitch network information currently available. In this thesis, I will describe this dataset in detail, cover the extremely sparse related work on Twitch itself and in the

[3] `goo.gl/3nACrm`

video game industry more generally, describe the experiments and conclusions which satisfy the aims listed above.

1.1 Motivation

This thesis proposes to explore the relatively unexplored Twitch network, but the experimentation and analysis of related work does not adequately describe why analysis of the Twitch network is valuable for both the scientific community and for the wider population of individuals interested in social networks and social networking trends. The simply novelty of analysis on the Twitch network is not an inherently good reason to expend such effort on the network. Here, we will discuss the reasons why we chose to analyze the Twitch network and discuss the value of our analysis of Twitch to individuals interested in working with other forms of social media.

The broad topic of social media analysis and computational social science more generally has focused primarily on 'traditional social media', which in this case means sites like Facebook, Twitter, and Instagram. These sites are largely text-based, present a user's entire history to visitors to that user's profile, and do not explicitly reward content creators for their efforts. However, new social media platforms, like Twitch, Snapchat, and TikTok, do not strictly adhere to these patterns. These sites are built around video, rather than text, and individual posts, stories, streams, and other content units often have limited lifespans after which they are nominally inaccessible or simply ignored, which is not true of traditional social media. In these sites, a user's entire post history is preserved and can be reviewed by interested parties for any reasons. As if to demonstrate this effect, one popular image macro, or meme, deals with the awkwardness of liking old photos of someone who you are romantically interested. Similarly, a modern tactic for generating negative publicity about one's rivals is to scroll through their Twitter feed and find tweets that would

now be considered offensive. Regardless of the awkwardness or fairness, respectively, of this behavior; such a thing would not be possible on emerging social media sites like Snapchat, as posts are either private or only to a user's followers/friends and are destroyed after 24 hours. There would be no capacity for this same behavior if every politician used Snapchat instead of Twitter to communicate with their base. Adopting this strategy may lead to other problems for the users, but that is out of scope of this particular discussion. In the case of Twitch, video streaming sessions are recorded and stored, but it is unusual for viewers to actually watch these recorded streams.

Twitch and other similar sites represent a new generation of social media that is built with live-streaming, video sharing, and multimedia content at its core rather than added on after the core interaction loop has been determined. Twitter's late 2017 move to 280 characters tweets can be seen as a tacit acceptable that the brevity inherent in the old 140 character limit is no longer representative of the social media landscape. Before the character limit increase, users circumvented the limit by posting screenshots of longer text blocks for their followers to read, which moved text-based Twitter into the multimedia realm. Even this, however, has not allowed Twitter to keep up with the next generation of sites that are focused on multimedia, as its user base has been leveling off and even dropping for the past few years [4] . Snapchat, on the other hand, has been adding daily active users despite a recent dip [5] and Twitch has continued to add viewers and viewing hours [6] (viewing hours are an indicator that viewers continue to be engaged with the content, not just logging on and then immediately leaving). It is clear that this new generation of social media is

[4] https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[5] https://wccftech.com/big-jump-in-monthly-active-users-sends-snap-shares-surgeing/

[6] https://www.businessofapps.com/data/twitch-statistics/

multimedia-native, popular, and persistent.

Unfortunately, the new generation of social media has been largely ignored by researchers. We will discuss this in more detail in Chapter 2, Related Work, but Twitch has received almost no attention from Computational Social Science, despite its massive popularity and the fact that the entire network is exposed via API and public-facing. Other networks, like Snapchat, are much more closed off, with content not easily accessible to the public or to interested researchers, which explains their lack of rigorous work. Analysis of the next generation of social media is necessary if social media researchers are going to keep up with the trends and forces that drive these networks. Because the new networks are so unlike the existing set of networks in terms of content, interaction patterns, attention, and format, it is unclear if the previous work on social media can be easily cross-applied to these new networks. Determining exactly this cross-applicability is one of the objectives of the work described in this thesis. By testing some of the same dynamics and behavior patterns on Twitch as on traditional social networks, we can determine if those same dynamics exists and, if they do, how difficult it is to develop algorithms that can detect and exploit those dynamics. Consider, for example, the problem of detecting or identifying bots on social media. There are a number of well-studied approaches that use Twitter data and those approaches are similar to the ones that would work on Facebook data. These approaches, however, would not necessarily work on Snapchat or Twitch for a number of reasons. On Snapchat, the most immediately apparent reason is that posts or stories are much more restricted in terms of who can see the content, unlike Twitter. In order to accomplish their goals, bot administrators would have to work much harder. This also means that the approaches for finding and identifying bots will have to be different, as the behavior of bots would necessarily be different. Other forms of behavior may be different as well. For example, are there Snapchat

influencers like there are Instagram Influencers?

The work presented in this thesis answers a number of questions about the Twitch network and its individual dynamics and operation, but perhaps most importantly it begins to answer some of the larger questions raised concerning the differences between traditional social media and the next generation of social media. These questions are larger than just the ones answered in this thesis as well. Much, but not all, of the work of computational social scientists is based on the analysis of traditional social media as a proxy for analysis of the dynamics of human society at scale. This analysis, however is deeply flawed, because it considers only traditional social media. A person's Facebook, Twitter, or Instagram feed is merely a window into that person's life, and is frequently a carefully curated window that shows the best possible view. As many such windows as computational social scientists analyze, Facebook and Twitter will never show a person's entire life and thus conclusions stemming solely from this picture will be incomplete. Twitch, though certainly biased as well, shows a different window and provides a different view of not only individual streamers and viewers, but a different window onto society at scale. Using Twitch data, motivated researchers can re-examine the conclusions of computational social scientists and help determine if those conclusions are truly meaningful and representative of society at scale or if the effects of the Facebook and Twitter platform warp and twist social interactions beyond what would naturally arise from human interaction.

In pursuit of these aims, the motivation for undertaking this thesis is two-fold. The first, understanding the differences between new social media and traditional social media, is immediate and bluntly apparent to researchers wishing to get and stay on the cutting edge of social media. Analysis and understanding of these new platforms is necessary for social scientists and the institutions that utilize those platforms to effectively reach out to that platform's users. An advertisement that works well on

Facebook, for example, may be seen as hokey and old-fashioned to the users of Reddit. The second part of our motivation is more subtle and, in a way, more more important for that subtlety. The efficacy and value of computational social science as a field is based on the fundamental assumption that the data sets analyzed are representative of the populations under analysis. If, as we posit here, the platforms themselves twist and warp behavior patterns, cross-platform verification is critical to ensuring that conclusions remain valid and useful to consumers of social science research.

Chapter 2

RELATED WORK

Despite the prevalence of video games and video gaming in the recent past and the increasingly large data sets available that cover video games, relatively little large-scale exists, especially in the domain of live streaming and cross-game analysis. Work related to the analysis of Twitch can be divided roughly into three categories; churn analysis, migration analysis, and behavior/platform analysis. We also cover the entire corpus of work which uses Twitch as its primary source of data (consisting of only three papers), and briefly summarize the area of influence and influence analysis.

It should be noted that the body of work that covers analysis of games is large and delves much more deeply into gaming that is practical to cover in related work section. Much of this research studies interesting problems and provides excellent advice for developers, administrators, and even players of games; but much of that work is omitted here as it does not relate to the topics considered in this thesis. Should the work presented here be expanded to cover other topics, like game content's effect on viewers and streamers, some of that work may become relevant.

2.1 Churn Analysis

The process by which a user stops using a particular service or playing a particular game is called *churn*. The analysis of this phenomenon is reasonably mature in social networking or product contexts, but remains somewhat cursory in the field of video gaming, possibly because data is frequently not available to researchers.

The first study of churn in video games came in 2007, as a group studied the churn of players of *EVE Online* Feng *et al.* (2007). This group had access to session

length data from the developer, CCP Games, and used this data to analyze the factors leading up to inactivity periods, or churn. The authors found that decreasing session lengths is a strong indicator that a user is about to churn.

In 2009, a group of researchers were given access to log data from *Everquest II* by the developer, Sony Online Entertainment [1] and used this access to publish a number of studies concerning player behavior in that game. Among these studies, is one analyzing churn through the lens of social influence Kawale *et al.* (2009). In this work, the researchers analyzed how a particular player churning, or quitting the game, affected those players that he or she frequently played with. They found that players who has many churned connections also tended to churn themselves. In 2010, the same group expanded their approach to other behaviors, intuiting that sequence alignment could be used to Shim and Srivastava (2010b) make better predictions than the previous methods, with mixed results. This technique attempts to use a player's action sequences to predict if and when players will enter an inactive period, or churn.

In keeping with the study of MMORPGs, Debeauvais *et al.* (2011) studied player retention in the massively popular game *World of Warcraft* in 2011. The data used in this work was collected by posting a survey on gaming websites frequented by *World of Warcraft* players. The authors studied a number of factors against the player's average hours per day spent playing, but more relevantly also studied the "Stop Rate" of game players. This stop rate describes the number of players who stopped playing the game for any length of time, even if they did not allow their subscription to lapse [2] . This analysis found that the majority of players (77%) had all stopped at some point, indicating that, especially in gaming, churn is normal and expected. Interestingly, all of the respondents were current active players which indicates that

[1] Now called Daybreak Games and no longer associated with Sony.

[2] Like many MMORPGs, *World of Warcraft* requires a monthly subscription to play. As of October 2016, subscription fess were $14.99 per month.

it is possible for developers to bring players back into the game, though this was not deeply analyzed.

Later, in 2012, another group of researchers analyzed churn rates for a set of five games, *Crysis 2*, *Medal of Honor*, *Battlefield: Bad Company 2*, *Just Cause 2*, and *Tomb Raider: Underworld* Bauckhage *et al.* (2012). Interestingly, this set contains both multi-player and single-player games, similar to the Twitch data set we use. However, all five of these games are of the same genre, so their results may not be generalizable. This work finds that the Weibull Distribution Weibull (1951) is a very good fit for distributions of play times, but attempts no predictive work.

Similar to some of the initial characterization analysis performed in Chapter 3 is the work of Sifa *et al.* (2015), which studied the amount of time users of the Steam gaming platform [3] spend playing games in their library. They found that play times follow power-law distributions similar to our findings on Twitch.

2.2 Migration Analysis

Related to the work proposed on Twitch, but not directly related to video games, is migration analysis. Especially in the social media context, this avenue of research attempts to understand the process by which and the motivations for moving from one social network to another. In Kumar *et al.* (2011), the authors authors analyze exactly these patterns. Using a set of users who have self-reported their membership across multiple networks, they study the activity of these users on different networks over two one-month time frames. The proposed work admits a different perspective than this work, first analyzing games instead of social networks, and secondly allowing a finer granularity since our snapshots are much closer together.

Closely related to the analysis of user migration is the problem of matching users

[3] http://store.steampowered.org

across social networks. While this problem is not directly relevant to Twitch data, it is easy to imagine an expansion relying on linking streamers to the accounts they use to play games for their viewers studying if, for example, more successful players of games accrue more viewers. One such approach used in Zafarani and Liu (2013) is to analyze the usernames in the network and the way these usernames are constructed. By analyzing the methods by which users tend to create usernames, pairs of usernames are matched together.

Using another approach, Zhang and Philip (2015) tackles this problem solely from a network alignment perspective. Since social network datasets are frequently anonymized and usernames can differ even if they are not, the authors assume that there are no labels for the users in the dataset and develop an optimization problem to align many networks simultaneously.

Combining both approaches, Liu *et al.* (2014) links social identities by analyzing user behavior but instead of analyzing only usernames, the authors take a long-term approach, considering actions performed on social networks. In addition, the authors include network information which leads to a co-optimization framework between both network matching and behavior matching approaches.

2.3 Behavior and Platform Analysis

On an individual level, there has been substantial work on video games and the behavior exhibited by players of games. In addition, there has been a substantial amount of research on game-related concepts like gamification. In this section, some of the work in these areas will be discussed.

One of the first major pieces of work in this area, serving as both a survey of the (then-sparse) area and inspiration for future work, is Castronova's book *Synthetic Worlds* Castronova (2008). In it, Castronova analyzes and extrapolates the trend

of synthetic worlds of video games, particularly massively multiplayer ones, tending toward greater and greater fidelity and thus receiving greater consideration as substitutes for the real world. This extends to the implications on productivity, both "lost" to synthetic worlds and replaced by labor inside synthetic worlds, economics, if and when a currency in the virtual world begins to replace "real" [4] currency, and society, as social interaction in virtual worlds normalizes and becomes interchangeable with traditional social interactions.

2.4 Currencies and Economics

Building on this work, Wang and Mainwaring (2008) studied the use of virtual currencies in China and found that QQ coins were frequently used in lieu of real money for participation in a gift economy that paralleled the gift economy in the real world. In a more formal economic study, Safferling and Lowen (2011) studied the economic forces present in the browser-based game *Kingdom of Loathing*. The authors found that many features of the game's economy paralleled that of real economies, including the conversion of real money into in-game money when the exchange rate was perceived to be favorable relative to spending time in the game to earn money. Perhaps most famously, the developer of *EVE Online*, CCP Games, maintains a small staff of economists [5] to ensure that the complex system of in-game currency, goods manufacturing, and goods consumption remains balanced. In this particular instance, substantial power was given to the economists, including the ability to intervene in the market to ensure the stability of the price of PLEX, *EVE Online*'s near-unique method of turning in-game wealth into subscription time [6] .

[4] Real is in quotes here since Castronova argues that virtual currencies are no less real than traditional currencies.

[5] http://www.ibtimes.co.uk/eve-online-meet-man-controlling-18-million-space-economy-1447437

[6] At time of writing, only *World of Warcraft* and *Wildstar* offer the same option.

2.5 Playbour

The concept of play-as-work, called *playbour*, has also been explored in subsequent work. In Yee (2006), the authors discuss the attitudes of play vs. work in video games. They re-frame the discussion of virtual work to be closer to traditional discussion about labor and work. Similarly, Zhang and Fung (2014) discusses the secondary market around video games in China, including both first-level labor (that actual play) and second-level labor, like time spend managing groups of players. The paper makes the case that many of these skills parallel real-world skills and as such the time spend performing these tasks are inseparable for traditional labor time. Rifts have even opened up in game communities about the value of labor. In Taylor *et al.* (2015), the rift between "industrial" players, those who play in order to manufacture in-game items, and "fighting" players, those who play in order to engage others in combat, in *EVE Online* is explored through user studies. The authors find that while the language from fighting players tends to be derogatory toward industrial players, industrial players are confident that their style of play is necessary for fighting players to play. They also understand that without the fighting players, there would be no market for the goods they produce. In this way, there is a feedback loop between the two types of players in the game's mechanics, but this loop is not present in the discussions. Similarly, when *World of Warcraft* introduced a tier of rewards for participation in player-versus-player (PvP) combat that equaled those for high-level player-versus-environment (PvE) play, players dedicated to the latter type of play called the rewards "welfare epics", complete with the implication that they we handed out for free. User studies conducted by the authors showed that PvE players did not feel that the skill set required for PvP play was not rigorous enough to warrant the rewards. Conversely, PvP players felt that their style of play was actually more

difficult, as other players change their tactics and strategies, while PvE content is static and requires only one, unchanging strategy.

2.6 Communication and Linguistics

There is little doubt that the rise of technology has changed the way that society at large communicates, but how Castronova's predictions have borne out in virtual worlds is not as clear. Much of the communication between players in games is in the context of "guilds" or other alliances of players for a common cause, and this has attracted attention. In Ducheneaut *et al.* (2006), the authors find that players who tend to play in groups progress through *World of Warcraft* more slowly than those who play alone and that players who are part of a guild tend to be online for longer. Interestingly, this indicates that players are willing to sacrifice optimal play in order to socialize with others. The authors speculate that this satisfies players' desire for an audience. In the same game, Thurau and Bauckhage (2010) finds that the guild system is mostly used for socialization, as most guilds are filled with low-level, low-activity characters. Guilds focused solely on high-level PvE or PvP are rare, and often fall apart possibly because the social dynamics of the group are unstable. Newer games have not invented this relationship either, as Seay *et al.* (2004) shows. Here, the usage of more types and more frequent communication is linked with persistence, both as a player of the game and as a member of the social group in *Everquest* and *Dark Age of Camelot*. In addition to simply lasting longer, the highly communicative players also report a higher sense of commitment to the group they have joined. Similarly, Ducheneaut and Moore (2004) studied social interactions in *Star Wars Galaxies*. This game is unique among those studied in that the mechanics of the game require social interaction, so engaging with other players is required for optimal play. Interestingly, they find that mandatory social interactions are often automated,

though social players that play manually often receive preferential treatment from other players.

Unsurprisingly, given the different mechanics of each game and different reward structures for players, Ahmad *et al.* (2011) finds that the models generated by Thurau and Bauckhage (2010) are not generalizable and, specifically, that *Everquest 2*'s guilds seem to operate differently. Importantly, the authors note that explicitly factoring in the level of socialization occurring within the guild is an important factor in determining the likelihood that a guild will survive. Even outside of the context of guilds, Ahmad *et al.* (2012) finds that trust in other players is an important factor for guilds and groups, and develops a model to predict if and when a two players will being to trust one another. In a rare instance of the lessons of virtual worlds being applied to real problem, Johnson *et al.* (2009) demonstrates that group formation mechanisms in guilds closely parallel group formation in real life by comparing guilds in *World of Warcraft* to street gangs in Los Angeles, going as far a building a model that formed the basis of the ones used in Thurau and Bauckhage (2010) and Ahmad *et al.* (2011).

Language in and around gaming has also received substantial attention and the particularities of this language usage has come under scrutiny. In Bergstrom (2013), the language of "Newbie Guides" for new players of *EVE Online* is found to be a gate-keeping mechanism, implicitly excluding many new members. It is, however, important to note that these guides are for members who have already joined the groups that produce these guides and no analysis was done to attempt to determine if these guides actually discourage potential new players. In a similar line of thought, Goodfellow (2015) studies the interaction between Russian-language and English-language discourse in *EVE Online* and finds that xenophobic tendencies exist in the language between both parties. While Russian-language discourse is more passively xenopho-

bic, mocking English speakers who attempt to bridge the gap, for example, English-language discourse is more actively xenophobic, coining new, derogatory terms to describe tactics discovered and employed by Russian-speaking players.

In much of this linguistic analysis, the term "metagame" is used in many different contexts with many meanings. In order to separate these terms, Carter *et al.* (2012) proposes the use of the terms "paragames" and "orthogames" to disambiguate this term. These terms have not caught on in the literature, unfortunately, so the differences between them are omitted.

Another term commonly used in the gaming community is "Theorycrafting", a term that describes a process by which new optimal strategies are invented or discovered. This has, however, forced other players to adhere to this strategy as discovered by Wenz (2013). In this way, theorycrafting serves as a way for powerful players, like guild leaders, to control the play styles of others. Though not performed by other players, the study of incentive mechanisms in gaming has received attention. Using achievements and badges has been shown to be effective in persuading Slashdot users to increase their engagement with certain mechanics of the site Anderson *et al.* (2013) and this "gamification" has its roots in similar achievement systems in video games.

2.7 Player Classification and Performance

Naturally, as games become more and more complex profiling players becomes more important to understand how players understand and interact with the game's mechanics. Similarly, games require balance in order for players to have meaningful choices. If one play style is dominant, player's agency is diminished and they lose interest Bartle (1996). As early as 1996, the profiling and balancing of player types was considered crucial to running a successful game Bartle (1996). Unlike many of the MUDs investigated by Bartle, games like *Everquest II* have fixed avatar types

and ensuring that these character classes are balanced is vital. Using behavioral profiling Shim and Srivastava (2010a) found that some such classes had advantages over others on average. They are careful to note, however, that the achievement gaps between classes shrinks significantly when playing in groups. Profiling players instead of character classes, Bell *et al.* (2013) analyzed the achievement portfolios of over 8.8 million *World of Warcraft* characters to determine that the original 4 categories of players proposed in Bartle (1996) still held true, with the exception that "hearts" had been replaced largely with "gears", players interested in production and markets. Similarly, Spronck *et al.* (2012) profiled players of the game *Fallout 3*, using data gained from the player's choices in the game's tutorial. *Fallout 3*'s tutorial contains a fictionalized aptitude test called the GOAT which purports to determine which style of play the player is most likely to pursue. The authors find that this test is only mediocre at assessing play styles. This analysis of player behavior has even moved into the nascent field of mobile gaming. In South Korea, more than ever mobile games make up the majority of hours spent with video games and Seok and DaCosta (2015) studied the personality-based factors that may predictive of gaming habits that interfere with daily life, though no strong correlations or conclusions were discovered.

Similar to having agency, it is important that players feel effective, that their actions have impact on the world. In *Everquest II*, the mentoring system was studied in Shim *et al.* (2011b) to determine if the system performed the role of increasing the performance and effectiveness of low-level players. The study found that while it did not decrease performance, there is little evidence that performance was increased for low-level players and that mentoring caused a decrease in performance for the higher-level mentor. Similarly, Shim *et al.* (2011c) studied the effectiveness of players relative to the difficulty of the challenges they faced. This study found that players

were most motivated to continue playing when they regularly faced challenges slightly above their own level. Interestingly, they did not find evidence of a "flow" state, where suitably difficult challenges induce higher performance than overly easy or overly difficult challenges.

Player performance is at its highest in the recently-emerged and increasingly popular world of eSports. Naturally, researchers have investigated these competitions in assessing the factors that affect performance. In Shim *et al.* (2011a), the authors studied the high-ranking teams of *Halo 3* players to determine the factors that predict a team's success. Unfortunately, the authors only have access to game outcomes, and find that simple metrics like kills, deaths, and assists in prior games do not predict future success well. The well-studied game *EVE Online* also has some eSports features and events, which lead to a study Carter and Gibbs (2013) of how the game's underhanded tactics and laissez-faire management attitude affected the competition. Inspired by existing work in predicting outcomes of NFL Games Boulier and Stekler (2003), researchers have also attempted to use in-game data to predict the outcome of individual eSports matches, particularly in the popular game *DOTA 2* Schubert *et al.* (2016). By tracking the outcome and advantage gained by each player encounter, they make gains in predicting the overall outcome of a match. Similarly, performance was studied in *League of Legends*, a very similar game, by Ong *et al.* (2015). They similarly found that outcome prediction was possible and found that adding some play style features for the team's players made victory predicitons more accurate.

2.8 Games as Platforms

In his book Castronova (2008), Castronova predicted that virtual worlds and games would come to be thought of as platforms on which more complex behav-

ior emerges, rather than arbiters of behavior themselves. To that effect, researchers have developed techniques to make understanding behavior on these platforms easier.

One of the most obvious techniques is visualization. In Medler and Magerko (2011), the authors describe a technique for visualizing that is game-aware, using graphical cues from the game in question to inform the visualization and make it fit with the theme of the game. Unfortunately, this requires the display to be customized for each game in question. Similarly, Cheong *et al.* (2008) describes a technique for presenting players with a summary of their accomplishments and major plot points in a game based on the content of their save file. This technique uses in-game renderings to display data, though it does require that the game be developed in a particular framework.

As with any platform, the play of games generates logs and records of the events that transpired in the game. In Shipman and Marshall (2014), player's attitudes towards the distribution of these records comes under scrutiny. Interestingly, the researchers find that players are not particularly comfortable with researchers studying their play records, even when these records are anonymized. Given the choice, players believe that a 50-year embargo should be placed on the release of such records and results.

As Twitter and Facebook receive scrutiny in social media, so too do games themselves receive scrutiny in their study as a platform. Researchers agree that players are solving extremely complex problems in the normal course of play for general games Yannakakis and Togelius (2015), but exactly how hard simpler games are is not always clear. Some simple puzzle games have automated solvers and even automated difficulty computation van Kreveld *et al.* (2015) and yet others have been shown to be NP-hard Guala *et al.* (2014). Permitting automated play of complex, competitive games is an area of active effort in AI, particularly for the game *Starcraft:*

Brood War Ontanón *et al.* (2013).

The fast, repetitive nature of games as platforms allows researchers to experiment with the mechanics of the platform and conduct experiments on linking game mechanics that are not necessarily intuitively linked. For example, Cachia *et al.* (2014) studied the effect of linking a player's weapons systems with the game's background music, though found this difficult to control. Likewise, Brown (2013) attempted to use evolutionary computation techniques to infer a player's preference in equipment drops in a RPG settings and customize drops to be more suited to that player.

When a new game debuts, there are frequently service interruptions in the first few weeks. These service interruptions are, to some extent, the result of scaling for the stable population of the game rather than the inital population, but they can also be a result of improper simulation of player load. To address this issue, Pittman and GauthierDickey (2010) studied a dataset of *World of Warcraft* players and developed a scalable virtual player that can be used to simulate play and movement patterns for appropiate load testing.

While the space of related work in games is clearly quite large, it is also clear that very little of that work is directly related to the problem at hand. No work at all has studied Twitch specifically, and the work studying churn and user migration either considers only one game or considers social media.

2.9 Twitch Work

The existing work on Twitch is extremely limited in scope and typically does not address Twitch's social nature. One piece of existing work addresses the presence of so-called "viewbots" on an unnamed live streaming network, intended to inflate a streamer's viewer numbers in order to earn or profit from revenue-sharing schemes Shah (2017). Other work address the content disparity in the chat chan-

nels of male and female streamers Nakandala *et al.* (2016). Lastly, network scientists have probed Twitch's content delivery network to understand how server loads are geographically distributed Deng *et al.* (2017).

2.10 Information Propagation

Influence and information propagation is easier to detect on social networks and social media than it is in the real world, due in part to the unification of platforms and public availability. This has lead to much research on information cascades Guo and Shakarian (2016) as researchers track the spread of ideas and information. Similarly, researchers have looked for influential users Basaras *et al.* (2013) as we have. Researchers also look at factors affecting influence in a general sense Suh *et al.* (2010). Our work stands out for its novel data set, Twitch, as well as the novel technique. Since we bound our domain to Twitch and our snapshots collect the entire network, we can guarantee that the data used by our technique complete.

Chapter 3

DATASET

In this work, the primary source of data is collected directly from Twitch, with supplementary data from GiantBomb. Twitch's data, being the more complex, will be discussed first.

3.1 Twitch Data

The Twitch social network is comprised of a tripartite network of games, streamers, and viewers. Each item in the "game" layer is a unique game or game-like object that streamers can be associated with. The games displayed are actual, published games like *League of Legends*, *Overwatch*, or *Rocket League*. Game-like objects include categories added to Twitch after it became clear that there was a demand for content not directly related to a particular game. This content includes categories of streams performing activities like *Creative* for artwork and other creative endeavors, *Gaming Talk Shows*, and *Programming* for game development or development

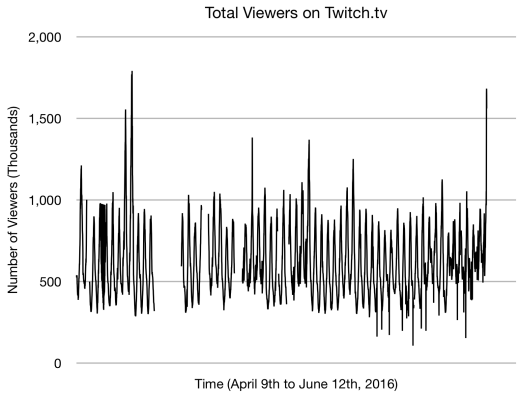

Figure 3.1: Total Viewers on Twitch.tv vs time. This graph shows a clear daily cycle of viewership on Twitch, and also indicates an increase in viewership during the weekends.

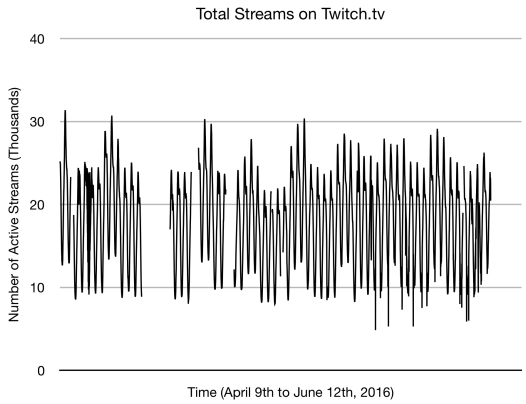

Figure 3.2: Total Streamers on Twitch.tv vs time. Similar to the previous Figure, this graph shows a strong daily cycle and indicates a weekend boost.

tutorials. Interestingly, due to an increase in popularity in South Korea, Twitch also debuted a *Social Eating* category [1] . The expansion of Twitch's primarily social content also provides an interesting view into the types of non-gaming content in which Twitch users are interested.

Connected directly to the games layer is the streamers layer. At a particular time, each streamer is linked to exactly one game. Though streamers can theoretically play more than one game, Twitch's platform does not support a streamer being associated with more than one game. Streamers are actual human beings or small groups of people sharing one account, so their decisions can be influenced by the actions of other streamers and viewers. Many streamers are rewarded financially for their effort by Twitch through their Partner Program. This is a revenue-sharing scheme that allows streamers to benefit from increased audience size (viewer numbers) through channel "subscriptions" and advertisement revenue. Therefore, streamers in the partner program are incentivized to grow their viewership. This financial motivation is also a huge incentive for less successful streamers to follow in the footsteps of successful streamers.

[1] `https://help.twitch.tv/customer/portal/articles/2483343`

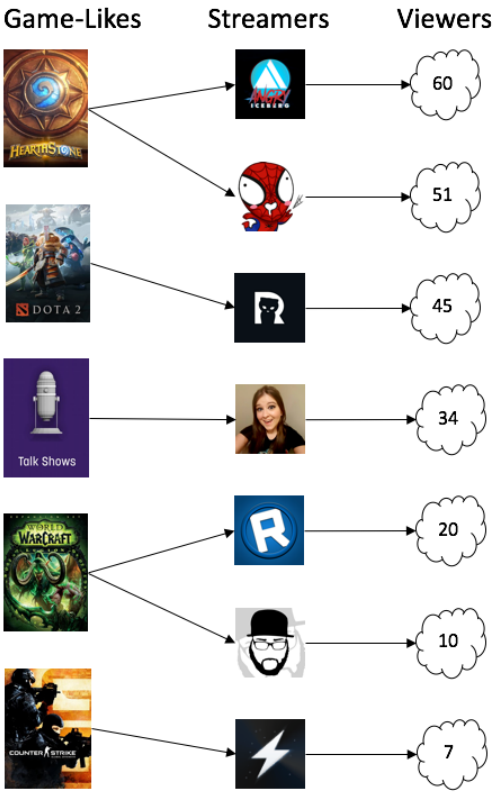

Figure 3.3: A simplified view of the Twitch.tv network depicting the relationships between games, streamers, and viewers.

Subscriptions take the form of $5 per month payments to Twitch that allow viewers to access content not available to other users. Most of the benefits for this are realized in the streamer's chat channel, a text chat that registered viewers can join in order to interact with the streamer. Subscribed users receive the ability to use a wider range of emoji in the chat and have access during a "subscriber-only" mode that allows the viewers that materially support a streamer to communicate with a streamer in a smaller group, though popular streamers may have hundreds or thousands of subscribers, so the conversation is not necessarily private. This may also be incentive for an engaged viewer to switch their support to a less popular streamer, as the viewer can make a stronger personal connection with a streamer with fewer subscribers.

The last layer is the viewers layer. While the presence of viewers drives the

popularity of streamers and of games, this is the most difficult layer to characterize in the dataset. As many viewers are anonymous, our dataset contains only aggregate counts of the viewership for each streamer and each game. If we chose to investigate the behavior of individual viewers, it would be possible to get a partial list of non-anonymous viewers by connecting to the Twitch chat. Each streamer has a connected chat channel that viewers can use to interact with the broadcasting streamer as we discussed, but many users choose not to participate in the chat and therefore either view anonymously or view without connecting to the chat. Nonetheless, the existence of viewers in large numbers and the desire for streamers to broadcast to increasingly large numbers of viewers is critical for both the continued existence of the entire platform and for the financial stability of the streamers.

As demonstrated repeatedly in the past Easley and Kleinberg (2010), power-law relationships dominate social network interactions, and the tripartite network formed by games, streamers, and viewers on Twitch is no exception. In <blah> these relationships are shown in aggregate over the two days of our dataset, the largest such dataset in the literature. The power-law relationships revealed here are not surprising given the organizational structure of the Twitch network. The "rich-get-richer" dynamics implied by power-law relationships dominate in part due to the way games and streams are presented to new users of the site. Both games to watch and, subsequently, streamers to watch are presented in descending order of current viewers, making power-law relationships unsurprising. In Figure 3.4, we present the power-law relationship between viewers and game from the Twitch network.

In the first sections of the work presented here, we focus on *influential* streamers in the middle layer, who are streamers whose choice of game to play affects the gameplay choices of other streamers. As discussed previously, there are extremely popular streamers who are capable of attracting tens of thousands of viewers to their

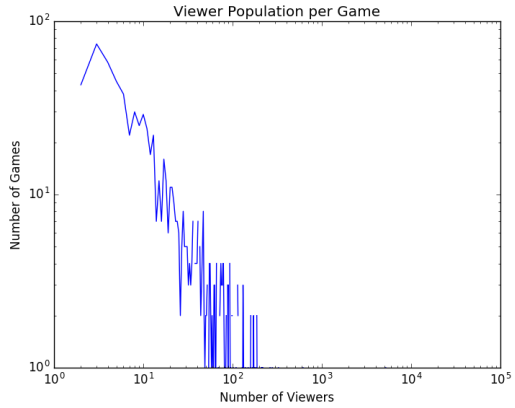

Figure 3.4: Number of games with viewer population vs. number of viewers. Note that both axes are log-scale, indicating that this trend follows a heavy-tailed distribution.

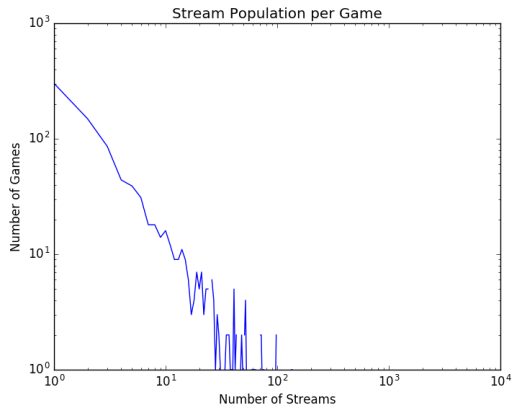

Figure 3.5: Number of games with streamer population vs number of streamers. Note that both axes are log-scale, indicating that this trend follows a heavy-tailed distribution.

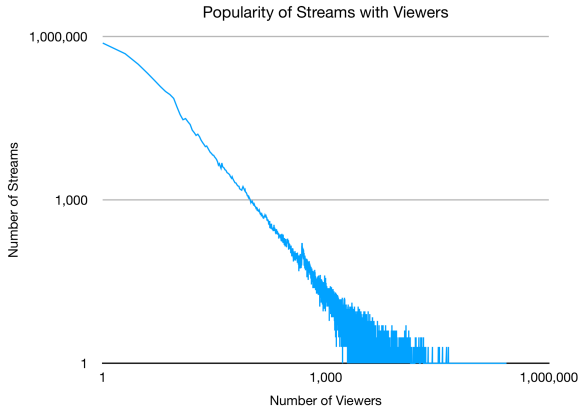

Figure 3.6: Number of viewers vs. number of streams with that viewer population. Note that both axes are log-scale, indicating that this trend follows a heavy-talied distribution.

stream on a regular basis. Many of these streamers earn a living [2] from their streaming. Though it is clear that these streamers exist, it is unclear what effect they have on the network. Do they have a suppressing effect, reducing the number of other streamers who play the same game since the new streamers have no hope of accumulating a large viewer base? Or do they have an inflationary effect, increasing the number of streamers playing the same game in the hopes of peeling off some of the huge streamer's viewers?

The latter sections will focus on the first layer, that of the games themselves. In particular, this work attempts to predict if a new game will have a significant impact on the Twitch network. Newly released games enjoying temporary popularity on the network is extremely common, as streamers use the new game as a way to attract viewers with novel content and viewers watch the new game to either consume that novel content or make determinations about their own purchasing decisions. However, not every new game sees this early surge in popularity. It is entirely possible for a game to debut on Twitch with no fanfare and make almost no impact on the network. In the latter sections, we attempt to determine what characteristics of a game affect its likelihood of making an impact on the Twitch network.

In capturing these three layers of the Twitch network, our final data set takes the form of snapshots of the network taken every 1000 seconds. In these snapshots, we capture every game played on the network along with the number of active viewers and streamers for that game, forming the entire games layer at the time of the snapshot as well as aggregate statistics for the other two layers. Similarly, we also capture the streamers layer, collecting a significant amount of information for every streamer on the network, including details like the current numbers of viewers, the total number of views that stream has ever received, the language of the broadcaster, the language of

[2] `https://nowloading.co/p/how-much-money-video-game-streamers-make/4266946`

the stream, Partner Program status, etc. Since the viewers layer can only be captured in aggregate, capturing this layer is done through the capture of the games and streamers layers since game and stream descriptions both include viewer counts for the game and stream. This collection process is done using Twitch's API, which has two relevant endpoints, one for games and one for streams. The games endpoint allows us to collect games being played on the network, which we do exhaustively as discussed previously. The streams endpoint allows us to collect information concerning active streams of a particular game. Using the results from the games endpoint, we can exhaustively collect the active streams for each of the games from the previous step. We perform this collection using a multithreaded Python crawler and store the result in a MongoDB instance for later retrieval.

3.2 GiantBomb Game Data

The GiantBomb API [3] provides an open-access, community-edited repository of data concerning individual video games that we used to gather more information about the games played and watched on Twitch. While Twitch's API provides rich information concerning its social network, it provides almost no information about the games themselves. Thus, supplemental information from the GiantBomb API is necessary.

GiantBomb's API provides a sizable amount of data necessary to make inferences using games as the independent variable. With only the data available from Twitch, the only feature would be the name of the game and its GiantBomb ID. Adding GiantBomb data provides access to the fields listed in Table 3.1. Since the data is sourced from the GiantBomb community, some games are missing fields, but this does not diminish the usefulness of the data. How the GiantBomb data is used in

[3] http://www.giantbomb.com/api

this work will be discussed in detail where it is relevant.

Aliases	Characters	Concepts
Date Added	Date Last Updated	Developers
Description	Short Description	Platforms
Franchises	Genres	Publishers
Killed Characters	Debuted Characters	Locations
Debuted Locations	Objects	Debuted Objects
Debuted Concepts	People	Debuted People
Main Image	All Images	Videos
User Reviews	Staff Reviews	Themes
Expected Release Date	Original Release Date	Re-Release Dates
Rating	Similar Games	

Table 3.1: Additional features concerning individual games available from the GiantBomb API.

Chapter 4

VIEWER MIGRATION AND INFLUENCE

Before introducing the data set and discussion the prior work in the area, we proposed that Twitch's focus on live content makes it different from traditional social media like Twitter or Facebook. If this is true, then the effect should be noticeable in some aspect of the network's operation. To test this idea, we consider an example of analyzing the effects of influence on the Twitch network in this chapter. In particular, we consider the influence of streamers.

4.1 Problem Definition

Of the three layers we discussed previously, the streamers layer is possibly the most interesting due to the complex interactions of financial motivations, viewer interest, game availability, and popularity of other streamers that all influence the decision-making process of a streamer deciding which game from their collection they are going to stream. For example, a streamer who wishes to become part of the Partner Program may choose to stream a less-enjoyed game that is more popular in order to attract viewers and thus qualify for the Partner Program. On the other hand, a streamer who is already in the Partner Program may wish to play a slightly more obscure game that he or she knows the viewers will enjoy watching in order to draw a large crowd. In either situation, the streamer starting up his or her broadcast session must consider the other streamers who are playing the same game(s). Are these other streamers more or less popular than they are? Will their stream be ranked above or below their competition? Are these other streamers so popular that they will draw viewers away? Are these other streamers obscure, allow the other's viewers to be

leeched away? All of these questions must be considered.

Previously, we described a category of streamers called influential streamers that exist in the Twitch ecosystem, influencing the choices of other streamers and viewers on the network. If these influential streamers can be identified, it would be possible for new streamers to follow these influencers, thereby gaining the benefit of having an influential streamer playing the same game as them and benefiting from the rise in that game's popularity. Defining exactly what makes a streamer influential is a difficult task due to the complex nature of the Twitch ecosystem.

Popularity is not Influence. The obvious definition of an influential streamer is a streamer whose play sessions increase the viewership of the game they are playing. This definition fits the criteria set out previously, since that streamer's choice of game affects the choices of the viewers of that game. There is, however, a subtlety missing from this definition. In this work, we wish to label streamers as influential only if their play has a net positive effect on the streamers streaming the same game that they are in addition to the positive effect on their own viewership. Using the obvious definition permits a streamer whose play increases only his or her own viewership and possibly leeches viewers from others playing the same game to be labeled as influential when that is not the intended behavior. We refer to this type of streamer as a *popular* streamer to differentiate them from the intended influential streamers.

Adjusting the definition to exclude popular users results in a modified definition: an influential streamer is a one whose play sessions increase the self-exclusive viewership of the game they are playing. In this definition, self-exclusive viewership means the total viewership of a particular game without taking that streamer's own viewers into account. This modification captures the intent of the definition, but does not punish users for acting in self-interest and increasing their own viewership, it simply does not reward them.

Influence is not Transient. In this new definition, however, there is still an issue of measurement. The most apparent way to measure viewership is to use viewer numbers. The higher the self-exclusive viewer numbers, the higher the influence. This method is preconditioned on the idea that Twitch's viewership numbers do no vary, or do not do so significantly, over the course of time. Unfortunately, this is simply not true. The total number of viewers varies enormously over the course of a day, so we cannot simply compare viewer numbers. Another way to represent a game's popularity takes a cue from the Twitch user interface itself. When selecting a game to watch, the Twitch user interface ranks games according to their current viewership, so we can use a virtual ranking to represent a game's viewership by computing where a game would be ranked if self-exclusive viewership was used instead of total viewership. This gives us a measure of viewership that is comparable across time points, since rank is meaningfully compared when viewer numbers are not. Using this measure, we can meaningfully measure the influence of a particular user by looking at the change (if any) in rank over time. In the next section, we will discuss the process of measuring the influence of a particular user.

4.2 Influence Analysis

Using the definition of determining the influence of a user, we frame the problem as a hypothesis test and refine that hypothesis using the criteria we discussed in the previous:

Hypothesis 1 *There exists some streamer, s, whose presence on the network increases the viewership of the same game s is currently streaming.*

This true hypothesis admits a corollary null hypothesis which we will test and reject in the next section:

Null Hypothesis 1 *No streamer, s, exists whose presence on the network increases the viewership of the same game s is currently streaming.*

In order to simplify discussion, we will introduce some notation. Here, s refers to a particular streamer. G refers to the set of all games being played on the network. $G(s)$ refers to the game currently being played by streamer s. In general, g refers to a particular game. V_{stream} and V_{game} refer to the set of all viewers of streams and games, respectively. $V_{stream}(s)$ refers to the number of viewers of a particular streamer s. Similarly, $V_{game}(g)$ refers to the viewership of a particular game g. Note that $V_{game}(g) = \sum_{s:G(s)=g} V_{stream}(s)$. Combining these two items, $V_{game}(G(s))$ refers to the viewership of the game being played by a particular streamer s, including that streamer's viewers.

In the hypothesis as currently written, the corner case that allows popular streams to be incorrectly labeled as influential is preserved. Thus, we must formally define the self-exclusive viewership we mentioned in the previous section. This value, abbreviated $SEV(s)$, can be computed as follows:

$$SEV(s) = V_{game}(G(s)) - V_{stream}(s) \tag{4.1}$$

This value is only meaningful for a game's viewership as a streamer's viewership exclusive of itself is always 0. Using $SEV(s)$ as above, we re-write the true and null hypothesis as follows to fit the notation:

Hypothesis 2 *There exists some streamer, s, such that $SEV(s)$ is increasing while $G(s)$ remains constant.*

Null Hypothesis 2 *No streamer, s, exists such that $SEV(s)$ is increasing while $G(s)$ remains constant.*

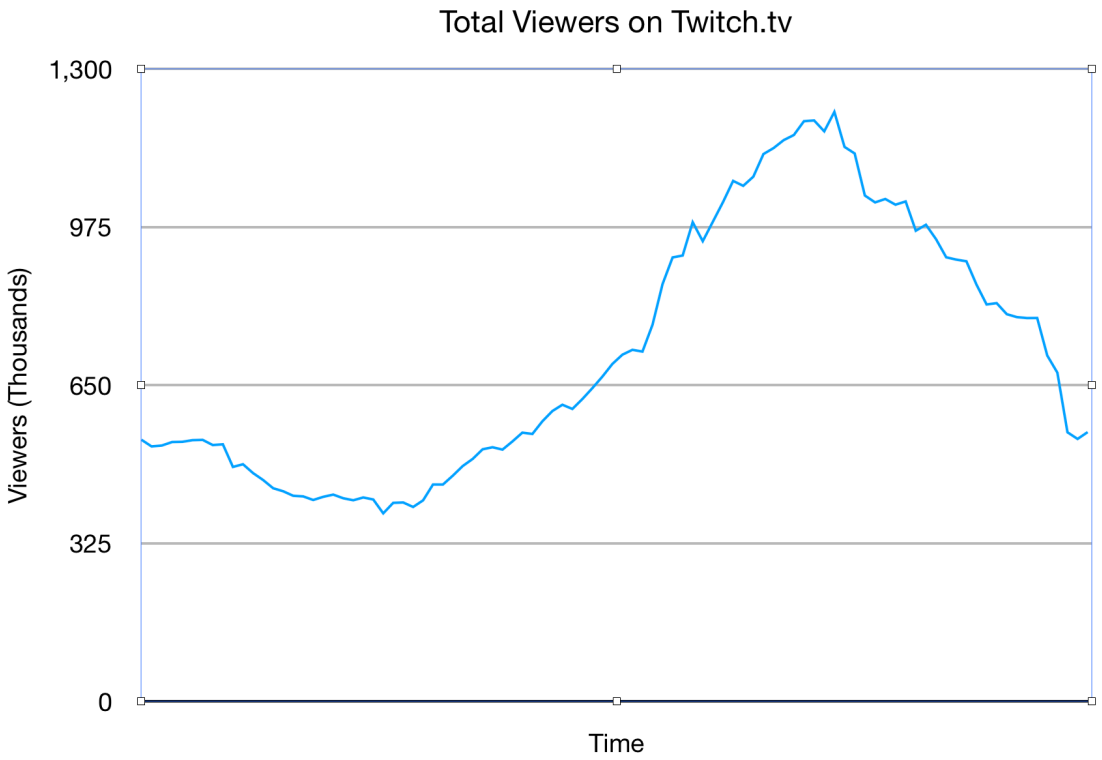

Figure 4.1: Total Viewers on Twitch over the course of 24 hours. This figure demonstrates the large fluctuation in viewership over the course of one day, necessitating the use of a metric that does not rely on raw viewer counts.

As we discussed previously, this formulation is vulnerable to temporal fluctuations in total viewership that make SEV values from two different time points not meaningfully comparable. In order to circumvent this problem, we consider the *rank* of the game on Twitch rather than the raw number of viewers. Here, the *rank* refers to the order of a game in the list presented to new viewers upon browsing to the home page. As we discussed before, this list is presented in descending order of viewership. Using the rank of a game removes the effect of temporal fluctuation in total viewership numbers, as the rank is relative to the viewership of other games which also rises and falls with the daily cycle. This also serves as a check on the significance of a change, as the effect of an individual streamer on the rank of a game is attenuated as the rank decreases (that is, becomes more popular) since each individual streamer is

a smaller proportion of the total population. This reflects the intuitive notion that it is more difficult to establish one's own influence in densely crowded streaming space than it is in a sparse space. Similarly, this method insulates from false positives in the heavy tail. The noise in the tail using $SEV(s)$ may lead to erroneous belief that a game with a very small viewership picking up one or two additional viewers causes a significant change. Using rank, this fluctuation is correctly identified as noise since the game's rank does not change much relative to its absolute value.

Input: Target stream s' and a snapshot t

$V_{game} \leftarrow$ Dictionary of game viewers, initialized to zero

foreach $s \in t$ *s.t.* $s \neq s'$ **do**
| $V_{game}(G(s)) += V_{stream}(s)$

end

Sort V_{game} by viewer count.

Return $R(G(s))$ from V_{game}

Algorithm 1: Computation algorithm for the Adjusted Game Rank of a particular stream.

To expand the notation established previously to this new way of considering games, we will refer to the rank of a game as $R(g)$. Similarly, the rank of a game played on a particular stream is $R(G(s))$. Unlike using viewers, however, there is no parallel to $V_{stream}(s)$, since we do not consider the global or network-wide rank of a stream. Lastly, in order to correctly parallel the structure used for game viewership, we also introduce the $SER(s)$, the Self-Exclusive Rank of a particular stream. This value is slightly more difficult to compute than the simple SEV we introduced in the previous section. It is also important to note that an increase in $SEV(s)$ corresponds to a decrease in $SER(s)$, as a more popular game has more viewers but a lower rank. This does change Hypothesis 2 and its corresponding Null Hypothesis to the

following:

Hypothesis 3 *There exists some streamer, s, such that $SER(s)$ is decreasing while $G(s)$ remains constant.*

Null Hypothesis 3 *No streamer, s, exists such that $SER(s)$ is decreasing while $G(s)$ remains constant.*

Figure 4.2 shows an example of a stream's SER changing over the stream's duration. This particular example shows a very long stream (about 11 hours) where the game being played drops from 51st rank at the beginning of the stream to 12th rank over the course of the stream. This is an example of a stream that would be a candidate for an influential stream since the game drops substantially in rank. In order to properly test the streams, we computed the SER of each stream at each time point for each of the $8,432,023$ sessions we derived from the dataset.

In order to test SER sequences to determine if their change in rank is statistically significant, we turn to Linear Regression. In this case, Linear Regression will determine if the rank of a stream's game is correlated with time elapsed. In Linear Regression, the statistical significance represents a comparison against a null hypothesis that the slope is equal to 0. Here, the rejection of this null hypothesis indicates that the rank is correlated with the time elapsed in a significant manner and thus the slope of the regression is meaningful. Furthermore, the direction of the linear trend, if it is significant, will determine if the correlation is positive (rank is decreasing) or negative (rank is increasing). The linear regression technique used is from Scipy Jones *et al.* (2014), as it gives the slope of the trend as well as the p-value. It is important to note that in using linear regression, we do not claim that this technique provides a good fit for the trend of the data. In fact, it is unlikely to provide a good fit. However,

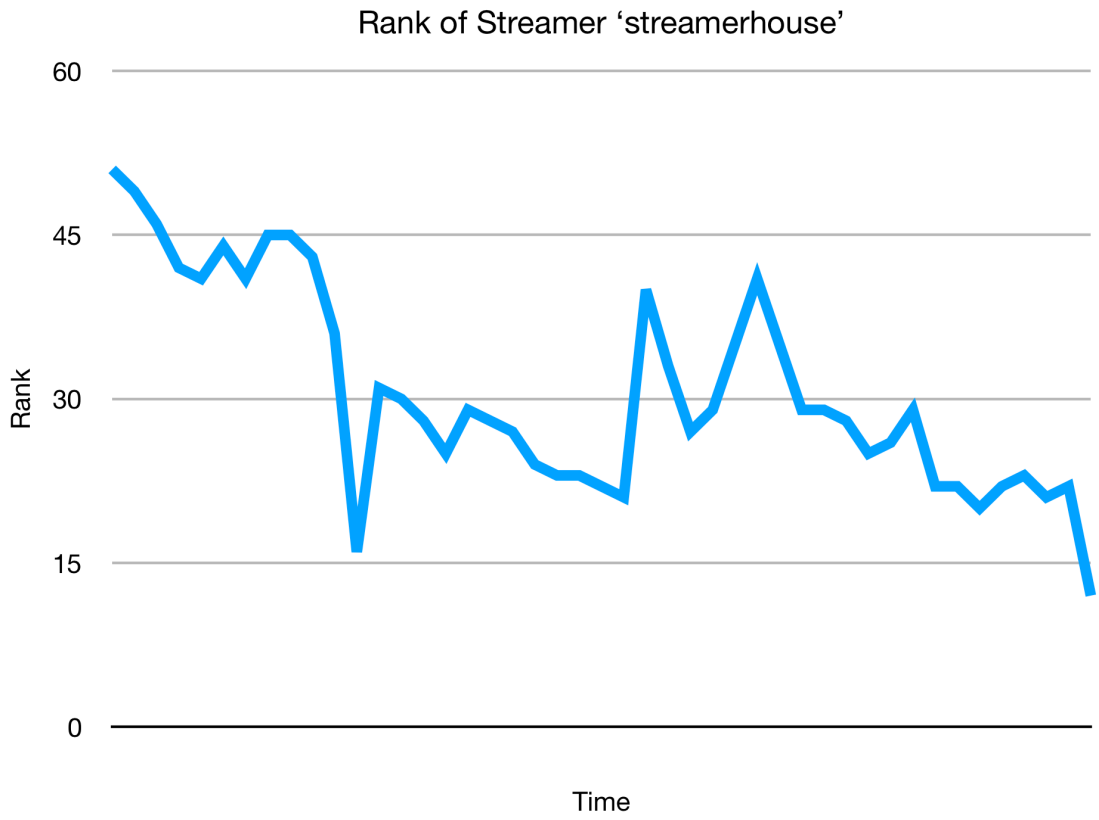

Figure 4.2: SER of a stream over its duration. This particular stream's SER is slowly decreasing over the 11-hour duration, indicating that it is possibly an influential stream.

in this work we only consider the direction of the trend and its statistical validity, so having a good fit to the data is not necessary.

Using this technique, the session depicted in Figure 4.2 has a slope of -0.5 with a p-value of $4.32e-8$, indicating that the session is representative of an influential session. This particular streamer, called *streamerhouse* is a collaborative effort between multiple individuals and frequently receives preview copies of games (according to their website, `streamer.house`), so it is reasonable that this particular stream's sessions would be influential on the network. In the next section, we will discuss the results of performing this analysis on all collected sessions.

Combining the AGR computation technique and the linear regression computation

into one algorithm yields the algorithm described in Algorithm 2.

```
Input: Ordered Snapshot Set T
D(s) ← Dictionary of streamer sessions
G'(s) ← Dictionary of streamer games
foreach t in T do
    S ← all unique streamers in t
    foreach s in S do
        if G'(s) = G(s) then
            Append AGR(s) to D(s)
        else
            Evaluate D(s) with Linear Regression
            D(s) ← AGR(s)
    end
    foreach s ∈ D(s) and ∉ S do
        Evaluate D(s) with Linear Regression
    end
end
```

Algorithm 2: Technique for finding influential sessions using Adjusted Game Rank and Linear Regression. This technique uses AGR to factor out the effects of viewership fluctuations on Twitch and the streamer's own viewers.

Using this technique, the session depicted in Figure 4.2 has a slope of -0.5 with a p-value of $4.32e-8$, indicating that the session is representative of an influential stream. This particular streamer, called *streamerhouse* is a collaborative effort between multiple individuals and frequently receives preview copies of games (according to their website, `streamer.house`), so it is reasonable that this particular stream's sessions would be influential on the network. In the next section, we will discuss the results of performing this analysis on all collected sessions.

With Negative Slope	Without Negative Slope
$1,768,510$ (54.0%)	$1,503,527$ (46.0%)

Table 4.1: Number and fraction of significant trends with a resultant negative slope and non-negative slope.

4.3 Influential Streamers and Their Impact

Using the $8,432,023$ sessions derived from the dataset and the linear regression analysis system described previously, we found that only $3,272,037$ were statistically significant. After filtering out the sessions that were statistically indistinguishable from random by linear regression, we considered the slope of the fitted lines. Table 4.1 shows the results of the slope analysis on the statistically significant sessions. The results of this analysis are somewhat surprising. We did not expect to find that the majority of the statistically significant sessions had decreasing SER. However, this does not indicate that the majority of streamers on the network are influential. Though we found more negative-slope sessions than expected, Null Hypothesis 3 is rejected, as it required us only to identify one such session.

While a particular session may have a negatively sloped SER, the quantity of such sessions indicates that every streamer responsible for any number of such sessions is not necessarily influential. For example, a 0-viewer stream could be the only stream of a particular game until a popular streamer like *streamerhouse* begins playing that game, which would cause a steep drop in SER for the 0-viewer stream. This isolated incident would not indicate that the 0-viewer stream is an influential one. However, if every time this small stream started playing a game *streamerhouse* switched to that stream's game it would be reasonable to conclude that the small stream had some influence, particularly over *streamerhouse*, however unlikely that may seem.

To address the issue of spurious streamers being incorrectly identified as influ-

Hours/Week:	10	20	30	40
Streamers	64,751	23,497	9,750	4225
Fraction	5.87%	2.13%	0.88%	0.38%

Table 4.2: Number and fraction of streamers meeting a threshold of average hours per week. Fraction indicates percentage of all observed streamers.

ential users, we use a two-pronged approach. The first part of the approach is to identify streamers who have consistently influential sessions according to the results of the linear regressions. This is done by analyzing the percentage of sessions by that streamer which are influential. A streamer whose sessions consistently increase the rank of the game they are playing clearly wields more influence that a streamers whose sessions decrease or do not affect the rank of the played game. To that end, we computed the percentage of streams that had negative slope for each streamer in our data set. Re-ranking streamers according to this strategy we found that many of the streamers that percolated to the top of this list had a very small number of sessions representing a below-average time spent streaming. In some respects, this problem is inevitable since our data set spans a little over two months of Twitch streamership.

In order to mitigate this new problem, we added an additional filtering condition, that a streamer's sessions total at least 30 hours per week. This has the benefit of eliminating "casual" streamers that stream infrequently and for short periods. In addition, limiting the requirement to 30 hours per week includes streamers who focus their streaming time to weekend sessions. In Table 4.2, we report the number of streamers meeting various hours per week criteria. By setting a restrictive, but not too restrictive threshold, we can ensure that all streamers who might realistically exert influence are captured. Allowing below full-time streamers to be part of our analysis acknowledges that influence is not determined solely by play time and that being an influential streamer with a full-time job is possible, but difficult.

Fraction of Negative Sessions (%)	Number of Streamers (%*)
> 90	66 (0.010%)
> 70	1218 (0.177%)
> 50	5895 (0.858%)
> 30	9086 (1.322%)
> 10	9707 (1.412%)
≥ 0	9750 (1.419%)

Table 4.3: Number of streamers with a minimum percentage of sessions with negative slope and statistical significance. Percentage values represent fraction of all streamers with at least one statistically significant session

After establishing our threshold for streamer activity at the 30 hours per week value, we repeated the analysis of consistently influential streamers to separate streamers into categories based on the proportion of sessions they stream that we influential. The results of this categorization can be seen in Table 4.3. This table shows us that the majority of streamers have approximately 50% of their sessions as influential sessions. Since we are looking for streamers who are consistently influential, we chose to look at the highest bracket of streamers for in-depth analysis.

Pulled from a pool of $1,102,260$ total streamers observed during the data collection period and $9,750$ streamers that fit our criteria for consideration, the approximately 66 streamers resulting from our analysis is reasonable. Manual inspection of the list of these streamers, particularly the 1218 in the second-highest category, reveals that many of these streamers are those of professional gamers and gaming-related organizations and companies. For example, the speed-runners [1] *calebhart42* and *sevens1ns* both appear in the top category, which is expected as our prior knowledge

[1]Streamers who attempt to finish a particular game in the fastest possible time.

informs us that these two streamers both exert considerable influence in the world of speed running. Similarly, the corporate stream *dreamhackcs* also appears in the top category. Dream Hack is an eSports league that covers a multitude of games, so it follows that the beginning of the *Counter Strike: GO* league season encourages other streamers to play more of the same game. Lastly, in the second highest category, the streamer *pokerstars* appears. This is a stream devoted to the play of Poker, with the namesake Poker Stars being a popular online play outlet and frequent tournament sponsor.

Similarly, we can inspect the streamers whose streams are entirely positive. These streamers are streamers who exert negative influence, in that the game they play becomes less and less popular. Accordingly, we do not expect to see any well-known streamers in this set of streamers and indeed we do not. In order to investigate this phenomenon further, we visted the Twitch profiles of the five streamers in our dataset whose sessions unanimously indicated decreasing SER. Four of these streamers, *newmultishow*, *yoko930*, *egoegogo*, and *swimstrim* appear to be regular streamers playing somewhat lesser-known games, but there is no obvious indication of why these streamers should be non-influential. In fact, three of these streamer are Twitch Partners, which may indicate that there is no strong correlation between influence as we understand it in this work and Partner status. The final user in this category, *dllgamestudio*, forms an interesting corner case that provides evidence in support of our technique. This particular user runs a "Twitch Plays" style stream of chess, which is not particularly popular on Twitch anyway. The interactivity of a "Twitch Plays" style stream would draw viewers away from a non-interactive stream and discourage new streamers from starting up a new chess stream while *dllgamestudio* is streaming. This case study supports the idea that popular streamers are not necessarily influential and vis-versa due to differences in how their actions affect other streamers.

4.4 Ablation Test

While these compelling examples of this technique's results are good arguments to its effectiveness, it would be easy to cherry-pick results from the various quadrants to demonstrate the effectiveness of the proposed technique. In order to verify that the results presented are factual and accurate, an additional test of its effectiveness is required. To this end, an ablation test is conducted, in which we consider each of the steps performed in the previously described technique individually, and discuss the effects of each step individually. in this case, the ablation test we perform also takes the form of a confusion matrix.

In the system proposed previously, we performed two major modifications in the pursuit of determining influence. The first of these modifications is to consider the *rank* of the game and the second is to consider only *self-exclusive* viewers in determining that rank. In combination, these two features are characteristic of the technique proposed in the previous sections. In order to demonstrate that the modifications have the effects claimed in the previous sections, two weeks of data were re-analyzed with the intermediary results preserved. That is, in addition to the final results of influence considering SER, influence analysis is considered using only viewership, only self-exclusive viewership, and using self-inclusive rank. Comparing these influence analysis rankings will allow us to determine the effectiveness of our system's steps individually.

Using the intermediate results from this ablation test, we re-ranked the streamers in order to compare the streamer's rankings against one another at each step of the ablation test. Since the ablation test resulted in four permutations of the two modifications, we will present the results of the analysis as the upper triangle of a $4x4$ matrix. This matrix is presented in Table 4.4. In this matrix, Kendall's Tau is

	VWR	SEV	VRA	SER
VWR	1.00	0.87	0.34	0.355
SEV		1.00	0.33	0.34
VRA			1.00	0.84
SER				1.00

Table 4.4: Agreement between ranking schemes used in the Ablation Test of the proposed method. Here, Kendall's Tau is used to compute the agreement between rankings.

used to compute the correlation between ranks. The ranks for streamers from the SER are used as the reference for determining the new position of the streamer in the vector representation of the ranking for the other three categories.

in this table, we preserve the notation previously introduced as much as possible, therefore SEV and SER have the same meanings as before. In the table, VWR indicates that the analysis is performed using the raw viewership of the game the streamer is playing and VRA indicates that the analysis is performed using the rank of the game as presented in on the network (i.e., self-viewership inclusive). This table demonstrates the importance of performing both steps of the proposed technique. With the exception of two pairs tested, each of the pairs has a correlation of approximately 0.34 according to Kendall's Tau. Thus, the steps represented in the transition from one intermediate result to the steps cause a significant perturbation in the rankings. This perturbation in the rankings is consistent with our expectations for the rankings presented in the previous sections. This table does show, however, that there are two pairs of ranking systems (VWR/SEV and VRA/SER) that have relatively high correlations. These high correlations are likely due to both of these transitions coming from the exclusion of a particular streamer's own viewers from the ranking determiners. As most of the streamers we consider are low-viewership streamers (an

effective of the heavy-tailed distribution of streamers and viewers), causing one of these low-viewership streamers to exclude their own viewers is unlikely to have much of an effect on the total viewership of the game they are playing and therefore their rank among streamers. For this reason, the relatively high correlation between these two pairs of ranking systems does not cause concern.

This ablation test supports our previous conclusions concerning our proposed method and the effect that the two proposed adjustments have on the results of the proposed technique. Through this test, we have shown that the proposed technique does substantially affect the rankings of streamers and, through this modification, measure the influence a particular streamer has on the network. In doing so, we have also provided supporting evidence to our hypotheses that streamers on the network are capable of exerting influence on one another, that influence is not directly tied to popularity, and that said influence is affected by the rank of the game that the streamer is playing.

4.5 Refinement

The work presented here covers an analysis of influence on Twitch where, we devised and implemented a technique to detect influential streamers. However, this technique is designed to apply to an existing data set collected from Twitch rather than analyzing the live Twitch environment. Rewriting the method to operate on live Twitch data would be vastly helpful in accomplishing the objective of identifying influential Twitch users, since such a technique could identify such users in an emergent fashion, which would be valuable for publishing companies looking to advertise new games on Twitch. These publishers and developers can use this technique to identify streamers who are not only influential but who play games similar to the one that the interested party is publicizing. Furthermore, using the data gleaned from the Giant-

Bomb data set on game genres, this last step could also be automated. The logical conclusion such an effort would be a system that maintains a database of streamers, their current level of influence on the network, and their genre preferences, which would allow advertisers easy access to the kinds of information needed to develop a marketing strategy for their game that utilizes Twitch's network of streamers.

Chapter 5

TYPES OF INFLUENTIAL STREAMERS

While the system described in the previous chapter is capable of finding influential streamers in the present moment (and historically influential streamers), the system does not differentiate between those streamers who are currently highly influential, those who wield some influence now but might wield more in the future, and those who are past their prime of influence. However, the question remains as to if there are streamers of these qualities on the network, and, if so, how they can be identified. In this chapter, the discussion will focus on further analysis of the streamers discussed in the previous chapters to determine if they can be segregated into categories based on their current, past, or potential future influence.

5.1 Motivation

In the previous sections, the discussion focused on finding influential streamers and this process was guided by the desire for publishers to optimize their advertising campaigns. It was posited that selecting streamers to give free copies of games or sponsor streams by that streamer would be better served by selecting influential streamers and that these streamers were not necessarily the same as the most popular streamers. In the previous chapters, it was demonstrated that the most popular streamers are not necessarily the most influential streams as measured by their ability to get others to play and watch the same game as they are playing. In this chapter, this idea will be further refined by digging deeper into the set of streamers from the dataset and attempting to determine which of those streams are good long-term investments for publishing companies. Assessing these streamers for their long-term

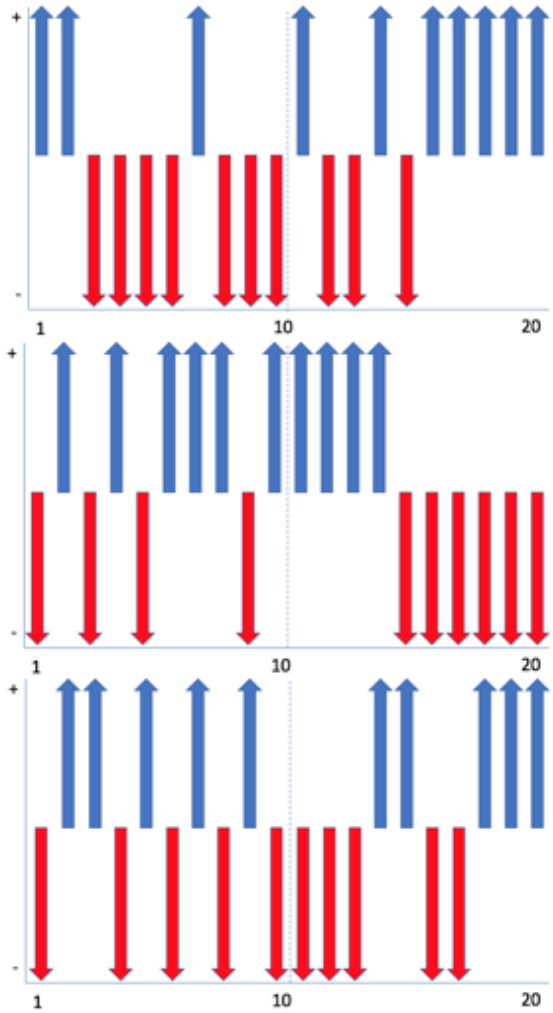

Figure 5.1: Examples of potential influence timelines. These three timelines are synthetic and for demonstration purposes, they do not represent actual timelines. The first shows a rising influencer, the second a waning one, and the last an unclear one.

ability to exert influence on the Twitch network is important for cultivating relationships between streamers and publishers. However, it is not valuable for a publisher to cultivate a relationship with a streamer who is losing their ability to influence the network. In order to perform this refinement, this section will discuss methodology for clustering streamers according to their influence timelines to determine if a simple clustering can determine if a streamer's influence is waxing, waning, or remaining steady.

In Figure 5.1, we show examples of timelines that would qualify as rising influence, falling influence, and steady influence. In each timeline, an upward arrow indicates that a particular streaming session is influential, while a downward arrow indicates that a particular session is not influential. Each timelines shows 20 streams from their theoretical streamer. The top figure shows a streamer who would be considered a rising influencer. The figure shows that of his first 10 streams, only 3 of them were influential. However, in this streamer's second 10 streams, only 3 of them are non-influential, which indicates that this streamer may be gaining some level of notoriety and is worth monitoring and possibly investing in as a publisher. In the middle timeline, the opposite effect if visible. In this case, the first 10 streams have 6 influential streams, while the latter 10 streams have only 4 influential entries. While this disparity is not as sharp as the disparity of the previous example, a publisher with a limited budget may still consider dropping this streamer from their sponsorship program, as the streamer may not wield the influence they once did. The final timeline shows a much less certain example. In this case, both the first and second halves of the timeline show 5 influential and 5 non-influential streams, though they are not distributed exactly the same. This is an example of a streamer whose influence has not changed over the course of the 20 sessions. Streamers in this category would be quite difficult to categorize in terms of their influence. If the streamer was highly influential before, perhaps having 10 influential streams in the time period preceding the graphed timeline, it would not be unreasonable to assert that the streamer is losing influence. However, if the preceding 10 streams were all non-influential, it could be just as easily claimed that the streamer is rising in influence.

This figure serves as an example of the further categorizations of influential streamers we seek to employ in this chapter. Obviously, there are far too many streamers on the Twitch network for each streamer to be analyzed individually for their influ-

ence trends or lack thereof. Therefore, we propose to automate this process using unsupervised clustering algorithms. In the next section, we will discuss our process for preparing the data used in the previous chapters for clustering.

5.2 Data Preparation

In the process of finding influential streamers, a data set was generated that consists of streamers, the streaming sessions they are responsible for, the length of those sessions, and a determination about the influence of each stream. In the previous chapter, this data was used to determine which streamers were most likely to be truly influential or not influential using two criteria; the length of time they stream, represented as a weekly average and thresholded at 30 hours per week, and the proportion of streams which were labelled influential. Using these two metrics, each potentially influential streamer was ranked. Using this influence ranking, a confusion matrix was generated that demonstrated the difference between popularity and influence in the Twitch network by analyzing some example streamers. Using this same data, this chapter will attempt to devise a clustering methodology that can distinguish between streamers who are likely to be successful in the near future and other types of streamers. In order to facilitate said clustering, each streamer's timeline will be normalized to the same length and a universal representation for influential, non-influential, and unknown influence will be applied to each streamer's timeline, allowing them to be compared against one another by a clustering algorithm.

To represent the influence of a particular session by a particular streamer, a simple convention can be adopted. An influential session is represented by a 1, a non-influential session by a -1, and an uncertain session by a 0. By way of reminder, the system for finding influential streamers used linear regression to find the slope of values of a derived measure, SER. Since the SER is a ranking, lower numbers are

better, so a downward slope is desired in this case. This makes the polarity of the SER slope opposite to that of the representation here. In addition to computing slope, the linear regression computed a p-value. In this case, this value indicated the probability of an implied null hypothesis, that the underlying distribution of SER values was in fact a straight line, rather than a line of the computed slope. Using the standard of ≤ 0.05 to reject this null hypothesis, we considered only those sessions which rejected the null hypothesis. However, these sessions are still important in determining if a streamer is likely to be influential or not. Consider a case where a streamer has exactly one influential session and a large number of these indeterminate sessions. Throwing out these indeterminate streams would result in a streamer labeled as influential, but for whom the evidence is quite weak, despite passing the 30 hours per week threshold. On the other hand, it would be similarly inaccurate to categorize this streamer as non-influential, as his or her presence does not drive others away or discourage others, as implied by non-influential assignments for streams. This streamer, like many others on the network, is most likely to simply have no impact, neither discouraging others to play or view the same game nor discouraging them from the same. Incorporating these indeterminate streams as 0 entries allows streamers like this theoretical streamer to be evaluated fairly, without either categorizing them as non-influential by assigning -1 to each such stream or mis-categorizing them as influential by throwing them out or assigning a value of 1.

In addition to representing the influence of sessions in a uniform way, each streamer's influence timeline should also be represented uniformly. In order to do so, each streamer's timeline is normalized to 100 elements by compressing their timeline. It is important to note that in this formulation of the timeline, the full length of each streamer's streams are considered, rather than just the number or ratio of influential streams. In order to construct each streamer's influence timeline, an entry in the

timeline is created for each snapshot in the session and this entry is assigned a 1, 0, or -1 value depending on the influence evaluation of the session. Appending each of these session's timelines together yields the streamer's entire influence timeline. For any streamer included in our analysis, this results in a timeline longer than 100 elements. Since each streamer has a different number of elements due to their differing time spent streaming, the timelines need to be normalized in order to make them comparable between arbitrary streamers.

In the next section, the clustering methods used in this process will be discussed. This process uses a wide variety of clustering methods, each of which has its own potential strengths and weaknesses. It is important to additionally note that all of the discussed methods are unsupervised methods, since there are no labels concerning the influence trends of each streamer.

5.3 Methods

In determining the clustering of streamers by their influence timelines or influence trends, there are many possible techniques that can be considered for clustering. In this case, the techniques available through the SciKit learn library Pedregosa *et al.* (2011) were leveraged in order to perform clustering. In particular, these techniques are: Agglomerative Clustering, DBSCAN, Birch Clustering, kMeans Clustering, and Affinity Propagation clustering. In this section, a brief discussion of each clustering technique will be provided and results for each clustering algorithm will be presented in the next section.

- Agglomerative Clustering (AC): Agglomerative Clustering is a technique that uses the distance between points to merge data points into successively larger and larger groups of points. Most Agglomerative Clustering algorithms have the advantage of keeping track of the sequence of merges, which allows for the

easy selection of the final number of clusters. Agglomerative clustering also allows for a very wide variety of merging strategies. In all of our experiments, we use the 'average' linkage strategy, which uses the centroid of each cluster to determine the distance to next point or cluster.

- DBSCAN (DB): DBSCAN clustering is a technique that takes a more global view of the feature space than other algorithms like kMeans, which consider only pairs of points or points and centroids. DBSCAN considers a dataset embedded in a feature space as areas of high and low density, and thus creates clusters from the high density areas. While techniques like kMeans can be tripped up by clusters that are not convexly shaped or clusters that are embedded within other clusters (consider a donut-shaped cluster and a separate spherical cluster in the former cluster's 'hole'), DBSCAN's focus on density rather than distance allows it to avoid this pitfall.

- kMeans Clustering (KM): kMeans clustering is a technique that attempts to find clusters by grouping data points according to their distance. By minimizing the distance between each data point and the centroid of the cluster that point is assigned to, kMeans results in dense clusters that are necessarily disjoint. This particular clustering algorithm works best when the points to be clustered together are close together in the feature space along all dimensions. This algorithm is suited to this particular problem because there is necessarily low variance in the features of influence timelines, and the difference between two streamers with similar influence patterns will be low.

- Birch Clustering (BC): Birch Clustering is an interesting form of clustering that is akin to a lossy compression function. By transforming one data point to another data point in a lossy manner, Birch clustering considers not simply

the distance between two data points, but the similarity of the information contained within each pair of data points. By transforming the question of distance into a question of information, this allows the Birch clustering algorithm to consider common sequences between data points and use those common sequences to build a tree similar to hierarchical or agglomerative clustering.

- Affinity Propagation (AP): Affinity Propagation is a technique that attempt to find cluster by sending virtual messages between data points and using these messages to determine which data points are the best exemplars for the rest of the data set. Affinity Propagation has the advantage of selecting the number of clusters based on the data set provided, rather than having to select the number of clusters individually, as with other clustering algorithms. Selecting exemplars is particularly useful for this particular problem because exemplars for each cluster are necessarily linked to individual streamers and these examples would be extremely useful for understanding each cluster.

Each of these clustering techniques necessarily results in a set of clusters, which will vary in number and composition according to the clustering technique employed. Naturally, these clusterings must be evaluated to determine their quality. In order to perform this evaluation, the Silhouette Index is used to determine the quality of the clusterings. The Silhouette Index ranges in values from -1 to 1 with values near -1 indicating an incorrect clustering, and 1 representing highly dense clusterings. The Silhouette Index is computed pointwise by the following formula:

$$\frac{b-a}{\max(b,a)} \tag{5.1}$$

In this formulation, a is the mean of the distance from the point of interest to all other points in the same class and b is the mean of the distance from the point of

Technique	Score
AC	0.1382
DB	0.1452
KM	0.0421
BC	-9.615e-5
AP	-0.1864

Table 5.1: Results of computing the Silhouette Index for various clustering methods performed on the streamer influence timeline data set.

interest to all other points outside of its class or cluster. In this chapter, the Silhouette Index values reported are found by averaging the Silhouette Index for all points in the data set.

In the next section, the results of each technique for clustering will be presented and discussed. It is important to note here that the list of techniques discussed and utilized here is not comprehensive and there may be techniques that perform better or worse on this dataset than the ones discussed here. These techniques were chosen for their ease of implementation and understanding.

5.4 Clusters of Streamers

Applying the Silhouette Index to the results of the five techniques presented in the previous section provides results that show how separated the final clusters are from one another in the clustering. In this section, we will present the Silhouette Index results from the 5 clustering techniques and then discuss the implications of these results on the original task of finding streamers who are gaining, maintaining, and losing influence.

In Table 5.1, the best results from each clustering technique are presented. Note that in the case of Affinity Propagation and DBSCAN, only one score is possible, as

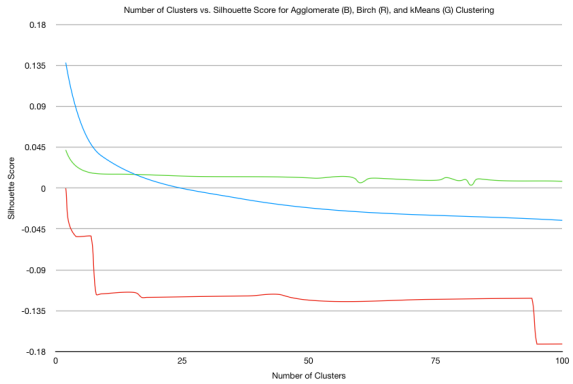

Figure 5.2: The effect of number of clusters on the final Silhouette Index score for Agglomerative (blue), Birch (red), and kMeans (green) clustering. Note that in all cases, the Silhouette Index is maximized at 2 clusters.

the number of clusters is not configurable. The final Silhouette Index scores demonstrate the true difficulty of this task, as none of these scores are particularly high. This indicates that it is indeed quite difficult to cluster streamer timelines in a way that meaningfully separates streamers. It is also valuable to note that the number of clusters (in the case of Agglomerative, Birch, and kMeans clusterings) does have an effect on the Silhouette Index results. In Figure 5.2, the effect of the number of clusters on the Silhouette Index score is presented.

This figure shows that the Silhouette Score peaks for all three clustering methods when 2 clusters are considered. This result is somewhat unexpected, as in the previous sections the universe of streamers was discussed in three broad categories; streamers with rising influence, streamers with waning influence, and streamers with steady influence. Therefore, the expectation from this experiment was that the dominant number of clusters would be at least 3, and more likely a larger number to account for differing levels of influential streamers. For example, a case can be made for almost any number of clusters, with increasingly high cluster numbers simply subdividing the existing strongly-supported clusters. It is also not hard to fabricate a case for only two clusters, with one cluster comprised of streamers with mostly influential streams

and the other comprised of mostly non-influential streams. However, in each of these cases, we would expect to see a significant deviation in the Silhouette Index which would indicate that the corresponding number of clusters is more suitable that its neighbors and no such rise can be found here.

The graph does however, reveal an interesting trend. In the graph of Birch clustering (in red), there are some clear breakpoints in the graph which show when the Silhouette Index drops substantially. There are three clear drops in the Index values at 3, 8, and 95 clusters, respectfully. It is interesting that these three breakpoints emerge in Birch clustering but no other clustering techniques. This may indicate that the true number of clusters is around one of these three values, though clearly Birch clustering is not the appropriate method to separate these clusters. The other two methods presented in Figure 5.2 show the gradual decline

The results shown here demonstrate that performing clustering using streamer timelines using the five methods discussed here is not likely to provide valuable results for individuals or groups looking to use clustering results as a means of generating a short list of streamers who could be tapped for having influential activity in the future, but do not necessarily show strong influence now. There are a number of reasons why this might be the case, however. In the next section, we will explore why this type of clustering may not have provided good results and consider possible additional approaches that could be more fruitful that the ones taken here.

5.5 Conclusions

While the approaches discussed in the previous section did not provide compelling results, it is still possible that the goal set forth for this chapter could be accomplished. The poor performance of unsupervised clustering of these streamer influence timelines could be a result of a number of different underlying factors. In particular, the poor

performance is quite reasonable if we consider the space over which the clustering was performed.

Implicitly, clustering relies on a separation between clusters in order for the clusters to provide clear, separable, and consistent clusters. In the poor Silhouette Index performance of the clustering algorithms explored herein, it is clear that either the space has no such separations or the clustering algorithms selected cannot take advantage of the separations that exist in the space. By further transforming the space over which our clustering algorithms search or selecting more appropriate clustering algorithms, it may be possible to more clearly separate the generated clusters. For example, our current formulation places equal weight on sessions that took place at the beginning of the data collection and on streams that took place at the end. Perhaps by weighting streams by their recency, it would be easier to isolate streamers who have started to gain influence. Though previously we discussed its theoretical importance, it is possible that the inclusion of streams that did not meet the significance threshold for linear regression has significantly affected the space by adding many 0 entries to influence timelines which significantly reduce the distance between even the extremes of the data. Regardless of the exact mechanism that is used to solve this particular issue, it is clear that more exploratory work is necessary to make classification of streamers into categories based on their influence potential possible.

Chapter 6

THE IMPACT OF NEW GAMES

In the previous chapters, we discussed the influence of streamers and quantified that influence longitudinally. We also tried to use the existing patterns of influence to cluster users by their influence patterns, but found that the clustering methods tried were ineffective. In addition to streamers having influence, the games played on the network also affect streaming and viewing choices. In that way, the available set of games could also be described as having influence, so any changes to that set are worth investigating.

6.1 Shifitng Viewership

The Twitch network is a dynamic network with many factors affecting the streaming and viewing population at any given time. One of the most apparent factors is the set of games available for streaming and viewing. We beleive that newly released games will cause a substantial shift in viewership patterns of the existing games. In order to determine if this is true, we must first determine if there are, in fact, substantial shifts in viewership patterns over time. Phrasing this as a hypothesis, we propose Null Hypothesis 4, a null hypothesis:

Null Hypothesis 4 *The distribution of a game g's viewership, $v(g)$, does not change over time.*

Since, as we discussed in the previous section, game viewership on Twitch is highly cyclical, it is not reasonable to analyze Twitch viewership as if it were a stream of continuous values. The data spanning say, 4:00-5:00 AM would appear

very different from the data spanning 4:00-5:00 PM. Thus, we focus our testing on cycles concomitant with the natural cycles of Twitch viewership, namely cycles with day-long or mutli-day periods. In this analysis, we use cycles of 1, 2, 3, and 7 days. While single-day and week-long cycles are self-evident, we added 2 and 3 day cycles to capture the difference between weekend and weekday viewership patterns. Thus, we revise our previous null hypothesis to that of Null Hypothesis 5.

Null Hypothesis 5 *The distribution of a game g's viewership, $v(g)$, does not change over* $1, 2, 3,$ *and* 7 *day cycles.*

In order to test this revised hypothesis, we rely on the technique of Kifer et. al. Kifer *et al.* (2004). In our implementation, we consider each game's viewership timeline a separate stream of data, and test each of these streams over all four windows specified above. To account for the invalid snapshots discussed previously, our implementation purges the test window(s) upon encountering an invalid snapshot. Using this algorithm, we can test Hypothesis 5 by comparing windows of the appropriate sizes against one another. If the algorithm detects a significant change in a game's viewership, we can reject Hypothesis 5. Otherwise, we do not have enough evidence for that particular game over that particular window size to reject the hypothesis.

The algorithm does not, however, specify a change detection method. This is intentional on the part of the original authors as it permits the usage of whichever change test is most appropriate. Since our final hypothesis only specifies the subject game and time scale as parameters and we already use those to determine stream composition and window size(s), we require a non-parametric test for change in the stream. For our streams, we chose the Kolmogorov-Smirnov (KS) Test Massey Jr (1951), as it is a reliable, non-parametric test of distribution difference. Performing the KS Test on our game streams resulted in $100,628$ distribution changes detected

1 Day	2 Day	3 Day	7 Day	Total
$51,686$	$27,059$	$17,235$	$4,648$	$100,628$

Table 6.1: Number of significant distribution changes (change events) detected for each window size tested.

with $\alpha \leq 0.05$ across the $13,951$ game streams, an average of 7 per game. Table 6.1 shows the number of distribution changes detected per window size. Henceforth, we will refer to these as *change events*.

Such a large number of changes in viewership patterns, even on the longest cycles, indicates that the null hypothesis proposed earlier is throughly rejected. That is, there is detectable variation across all four windows sizes. In order to better understand the changes in viewership distributions, Figure 6.1 shows examples of viewer distributions at the time of a change event for the game *League of Legends* over 1 and 7 days. Both of these graphs show a large increase in viewership occurring over a very short period of time. In the case of the second graph, this increase of viewership is repeated three times and then disappears. This pattern in viewership is likely caused by an eSports event occurring for League of Legends, causing many interested viewers to tune in during the competition. Interestingly, this competition is likely a European competition, as it can be clearly seen that the massive spikes occur some hours before the typical daily cycle reaches its peak.

Continuing to inspect the distribution of change events, Table 6.2 shows the ten games whose viewership streams have the most associated change events while Figure 6.3 shows the distribution of the number of changes against the number of games that have those changes. Interestingly, Table 6.2 shows that the games that have the most change events are not the most popular games. Many of these games are older games that continue to be popular in the gaming community, but do not command the attention that they once did. For example, *Final Fantasy VII* was released in

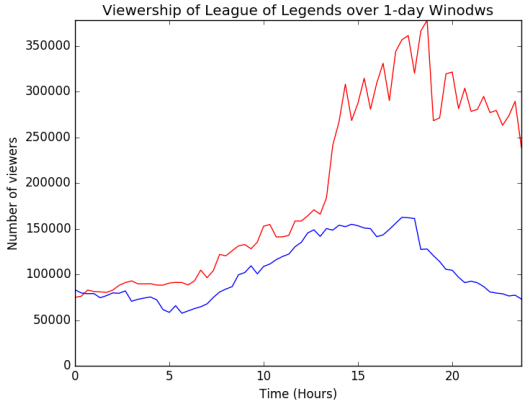

Figure 6.1: *League of Legends* viewer distribution over two 1-day periods. The KS Test detected a significant change between these two distributions.

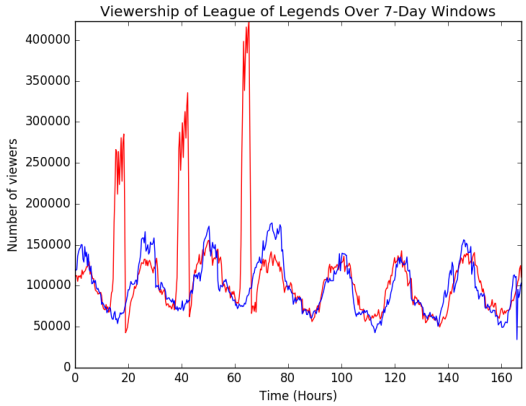

Figure 6.2: *League of Legends* viewer distribution over two 7-day periods. The KS Test detected a significant change between these two distributions.

1997 and *Diablo II: Lord of Destruction* was released in 2001. This may indicate that games like these serve to fill a gap between releases of more popular games, when the newness of newly-released games wears off for streamers and/or viewers and they return to playing and/or watching games that they are familiar with until another new game comes around. More investigation would be needed to confirm this phenomenon, but being the subject of change events both when they lose popularity and then subsequently regain it would explain the high number of change events.

Interestingly, Figure 6.3 does not follow the same power-law we would expect to see given that there were power-law relationships between the three slices of the

Game Name	Number of Changes
Heroes of the Storm	89
Final Fantasy VII	88
Sid Meier's Civilization V	87
Magic: The Gathering	87
Street Fighter: V	86
StarCraft II	86
Diablo II: Lord of Destruction	86
The Forest	86
Mortal Kombat X	86
Final Fantasy X/X-2 HD Remaster	86

Table 6.2: Number of significant distribution changes detected for each of the ten most volatile games.

tripartite graph that makes up the Twitch network. It is difficult to see intuitively why this would not also show a power-law relationship, given that some of the same popularity dynamics are at play in the decision to change games as it is to select a game, as demonstrated by Figure 3.4. More investigation is needed into the precise nature of the migration patterns between games.

Knowing that there are significant fluctuations in game viewership over time, we move on to the next step, determining if these change events are related to the debut of new games on the Twitch network. In the next section, we discuss our method of linking new games with change events on existing game streams, and the value of such a linking in predicting if games debuting in the future will cause similar impacts.

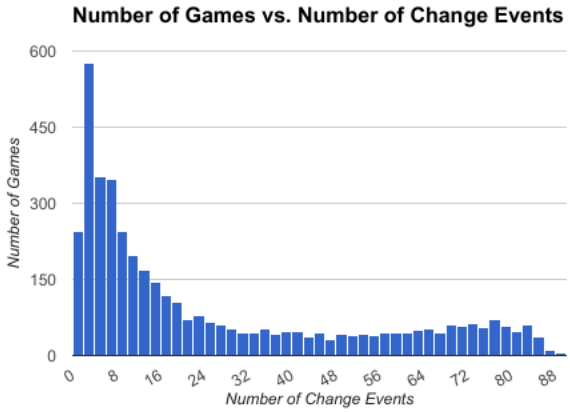

Figure 6.3: Histogram of number of change events and number of games with that number of change events. Note that unlike previous figures, this is not power-law distributed.

6.2 Video Game Debuts

Knowing that fluctuations exist in Twitch viewership distributions naturally leads us to the topic of determining the causal factors of those fluctuations. In the previous section, we speculated that changes in the set of games available to stream would be a large factor in causing fluctuations. We believe this to be the case from our manual observation of the popularity of newer games on Twitch. In addition, this makes intuitive sense: users of social networks often exhibit novelty-seeking behavior, and seeking out and viewing streams of new games fits that behavior pattern.

In order to test for the novelty-seeking behavior we beleive exists on the Twitch network, we analyzed all of our network snapshots in time order and recorded the time at which each of the nearly 14k games 'debuted' on the Twitch network, that is, the first time they appeared in a snapshot. It is important to note here that this debut time does not necessarily match the game's actual release date. Just as movie reviewers often see movies before they are released to general audiences, game reviewers receive review copies of games in advance of the street release date. The popularity of streaming websites and the massive popularity of some of the

With Change Events	Without Change Events
7,174 (51.4%)	6,777 (48.6%)

Table 6.3: Number and fraction of game debuts with and without coinciding change events.

largest streamers has lead publishers to allow some streamers to play games on stream in advance of their street release date [1] . This builds "hype" for the game and, theoretically, improves sales. However, it does mean that we cannot use a game's release date to determine when that particular game starts being available in the Twitch game pool.

After finding the 'debut' time for each game, we counted the number of change events that occurred in the 30 minutes following the game's debut. 30 minutes, we reason, is long enough that viewers interested in the new game will have switched their viewership to the new game, but not so long as to falsely attribute unrelated events to the game of interest. Note that this analysis excludes games that were streamed every day on Twitch, as no games which debuted in the first day of snapshots can possibly have change events associated with their debut, given that the shortest window size is one day. Table 6.3 counts the number of games with and without change events.

Interestingly, this analysis indicates that many game releases have a substantial impact on the Twitch viewing population. In fact, this analysis indicates that the majority of game releases are associated with change events. This does not, however, mean that viewership of the most popular games are affected. It is likely that the large population of games with change events indicates that there is heavy competition for the "tail" of the power-law distribution depicted in Figure 3.4. This revelation, in combination with the results depicted in Figure 6.3, leads us to believe that a large part of the competition for viewership occurs in the middle ranges of games. Indeed,

[1] goo.gl/3nACrm

none of the games in Table 6.2 are regularly in the top 10 games on Twitch. The fact that multiple games debut in the same snapshot (this must occur since there are $5,150$ snapshots and $13,951$ games) supports this theory, as the viewership of the most popular games is not erratic enough to account for these changes. Additionally, a large portion of game debuts linked to change events supports our theory that the set of available games influences streaming habits.

This leads us naturally to the question of prediction. Since game debuts with and without associated change events are approximately balanced, we would like to predict which games will have an impact on Twitch viewership. In order to do this, we supplemented the viewership data from Twitch with data about each game provided by GiantBomb, which provides an open-access database of metadata concerning individual video games, which we use in the next section to build predictive models.

6.3 Predicitng Novel Game Success

With mappings from game deputs to change events as well as rich data about each individual game, we would like to determine the feasibility of predicting if a game will be successful on Twitch (that is, cause at least one change event) given the data available from a service like GiantBomb. Not all games on Twitch could be matched to games on the GiantBomb service, and those games have been excluded from our analysis from the start. For example, the "Creative", "Programming", and "Talk Shows" pseudo-games do not have entries on GiantBomb, so we excluded them from our analysis. In addition, unresolvable game entries like "StarCraft II" were excluded, as that could refer to *StarCraft II: Wings of Liberty*, *StarCraft II: Heart of the Swarm*, or *Starcraft II: Legacy of the Void*, in this case the original game and its two expansion packs. Trimming games that could not be supplemented with

GiantBomb data removed only 245 games, less than 2% of the total population [2] .

The data available from Giantbomb is rich in nature by virtue of its source, being supplied by enthusiasts and the gaming community. Table 3.1 shows the features available from the GiantBomb data. In the classification techniques listed below, we use all numerical features present in a particular game's description. In the case of the description and short description features, we use the length of these descriptions. For features that pertain to characters, objects, people, locations, and reviews, we use the number of such objects. A number of additional time-based features were computed and used, including game age, difference between date added and last updated, date added and release date, original release date and expected release date, and date added and last updated. Some features, like "Main Image", were unable to be used in the classifiers described, but "All Images" was transformed into a count of images associated with the game.

All of the data provided by GiantBomb is crowd-sourced, so it is supplied by the very gaming community that Twitch also caters to. While data about games is frequently provided after release, hotly anticipated games frequently have rich data available long before the release of the game. In addition, many games on Twitch have a 24/7 viewing audience and these games are explicitly exempted from our analysis, as we discussed above. The emergence of a novel game on Twitch does not necessarily mean that game is being released to the general public, only that it is novel in our data. Therefore, while games certainly emerge on Twitch long after their true release date, our method seeks to predict if that game still holds enough interest (using GiantBomb data as a proxy).

[2] Resulting in the $13,951$ number cited early. The original set included $14,196$ games and pseudo-games.

Aliases	Characters
Concepts	Developers
Date Added	Date Last Updated
Description	Short Description
Platforms	Publishers
Franchises	Genres
Killed Characters	Debuted Characters
Locations	Debuted Objects
Debuted Locations	Objects
Debuted Concepts	People
Debuted People	Videos
Main Image	All Images
User Reviews	Staff Reviews
Expected Release Date	Original Release Date
Themes	Re-Release Dates
Rating	Similar Games

Table 6.4: Additional features concerning individual games available from the GiantBomb API.

6.4 Classification

Using the data gathered by scraping the GiantBomb database, we can accomplish two objectives simultaneously. First, we can determine if prediction of impactful games is possible at all with this data. Consistently poor performance with features that do not discriminate impactful games from those without impact would indicate that this data simply does not contain the information needed to separate these two categories. As a natural result, if the data set can differentiate between these two

group, explainable methods would be able to inform us as to which features from the diverse array of features were most important and we could use that information to inform future data collection and feature engineering work that might improve performance.

In order to accomplish these objectives, we propose to use Decision Trees and Random Forests to perform our first attempt at differentiating impactful from non-impactful games. The models we chose were implemented in sci-kit learnPedregosa *et al.* (2011).

- Decision Tree (DT): A decision tree model per Breiman *et al.* (1984), we included decision trees since the model is easy to learn and highly interpretable. In addition, decision trees easily handle categorical data, which is frequently encountered in video game contexts.

- Random Forest (RF): Similar to decision trees, random forests per Breiman (2001) train a large number of randomized decision trees and ensemble the results into a final class decision. Random forests retain a large amount of interpretability since they are simply made up of decision trees while enhancing performance through ensembling. Similar to decision trees, random forests cleanly handle categorical data.

Both of these models were limited to a depth of 5 in order to avoid overfitting, and we used 10-fold cross validation. In the next section, we will discuss the results of this classification and a follow-on experiment.

6.5 Results

Using decision tree-based models, we attempted to differentiate impactful games from non-impactful games with mixed success, as the results in Table 6.5 demonstrate.

Model	Accuracy	Precision	Recall	F1
DT	0.608	0.582	0.851	0.690
RF	0.612	0.595	0.814	0.687

Table 6.5: Results from the tree-based classifiers on the task of predicting if a game's debut will be coincident with change events.

All results presented are an average of all ten folds as discussed previously.

Interestingly, our results do not demonstrate a large difference between the Random Forest and the Decision Tree models. Low difference between these two models is interesting, since it indicates that there may not be a big difference between the different features' ability to distinguish between the two categories of games.

However, if we focus on the objective of this work, we can see that the first question we asked, is it possible to differentiate between impactful and non-impactful games, seems to be affirmative. Both of these classifiers performed better than random in their classification task, which indicates that future work may be able to continually improve the performance of classification methods with the correct feature engineering and further data collections.

In order to lay the groundwork for this task, we inspected the decision trees that were generated for each fold of the classification task to determine which feature was the most discriminative. In all ten folds, the Number of User Reviews was the most discriminative, followed by the categorical features that we expected to be present. Categorical features are quite common in this data set, as features like available platform(s), genre, and publisher(s) are all important features that would affect the environment around a game. This makes it surprising that the dominant feature turned out to be Number of User Reviews. Based on our domain knowledge, we expected one of the categorical features, particularly one like Franchise, to be the most discriminative. Despite this, the discriminative ability of a feature like User

Model	Accuracy	Precision	Recall	F1
DT	0.600	0.575	0.840	0.683
RF	0.603	0.581	0.818	0.679

Table 6.6: Results from the decision tree-based classifiers after removing the most predictive feature, the number of user reviews.

Reviews is reasonable because this feature serves as a proxy for user excitement about a game. The more excitement, or hype, surrounds a game, the more likely a user is to leave a review, positive or negative. Similarly, the more excitement or hype exists around a game, the more likely a viewer is to watch it on Twitch, even if that viewer then immediately decides not to buy the game or even watch it any longer.

However, User Reviews are a historical feature, meaning that user reviews only exist after a game is released. In order to see if a similar feature exists that also captures user excitement but does so in a way that can be used before a game is released, we re-trained the decision tree and random forest models excluding the User Reviews feature. We present the results of this effort in Table 6.6.

These results indicate that there is another feature or another set of features that provide nearly identical discriminative power to the user review feature we discussed earlier. Inspecting the actual trees generated, we see that this feature is Long Description, which we discretized by considering the length. Since the GiantBomb dataset is crowdsourced, this feature is also probably a proxy for user excitement. This is promising since the description can be partially completed before the actual release date, and the more excited the user base is about the game, the longer this description is likely to be. Similar to the other set of trees, categorical features fill in the remainder of the trees. Future work in this area should focus on determining or engineering new features that better capture user excitement around a particular game or games. While there may not be a substantial number of features from the

Model	Accuracy	Precision	Recall	F1
SVM	0.555	0.579	0.410	0.486

Table 6.7: Results from approaching the classification task as an Outlier Detection problem and using One-Class SVM.

GiantBomb dataset that capture this concept, especially in a non-historical fashion, this knowledge is valuable for researchers wishing to continue this avenue of research.

6.6 Outlier Detection

Most of the games that debut on the Twitch Network debut with little or no fanfare, and do not make a substantial impact on the network. This makes the ones that do outliers, a phenomenon we may be able to take advantage of in order to better detect those games of interest. To this effect, we repeated the analysis performed for the previous section with a classifier known for outlier detection capabilities, One-Class SVM.

While One-Class SVM does handle outlier detection well, it does lose the interpretability that led to the choice of tree-based methods in the previous sections. This is not a problem in this case, however, as our analysis of the trees satisfied our need for interpretability. In this case, we are attempting to determine if an outlier detection approach or a classical classification approach is more likely to be successful when trying to classify games on the network.

Again, we used 10-fold cross-validation on the data set to obtain training and test data sets. As before, we averaged the results from the 10 folds and present them in Table 6.7.

The results here are poorer than we expected from this approach. While perfection was certainly not expected from any approach, this particular measure performed worse than random according to F1, which was surprising. In particular, the recall

of the method suffered, which is the opposite of what would be expected from a technique specifically designed to detect outliers.

The poor performance in somewhat explainable, however, when we consider the data set under consideration and the particularities of One-Class SVM's operation. One-Class SVM essentially determines a hyperplane in the feature space that separate areas of high density and areas of low density. Since the objective is to find outliers, the areas of low density are considered outliers and the areas of high density are considered non-outliers. In the specific case of the problem we are studying, there are a couple issues with this approach. First of all, many of our features are categorical, so being 'above' or 'below' a hyperplane is not meaningful for these features. Second, the features that are not categorical tend to be counts of objects in the database, be they reviews, objects, characters, or concepts. This is also not idea for the hyperplane, as it allows the larger counts to overwhelm the small counts from sheer numbers. These factors may prohibit the classifier from developing a hyperplane that can separate the classes well.

There is another factor of our dataset that will make classification difficult under One-Class SVM and it is relevant to the density of the feature space. In our dataset, we only have approximately 14,000 games in total. Between these games, we use 32 features to classify the games into two groups, which means that our data set is extremely sparse. Recontextualizing the performance of the classifier with this sparsity, it is less surprising that it had this performance. Without appreciable differences between the areas of low and high density, the classifier was forced to choose a poor hyperplane. This density issue also gives some insight into why the recall of the classifier was so low. If the target games were in the areas of relatively high density, the classifier would quickly classify them as non-outliers, even though some other classifier could correctly differentiate.

Overall, the performance of the One-Class SVM classifier indicates that density-based outlier detection approaches are not likely to be successful in determining impactful games from GiantBomb data, and that future work should focus on the user excitement features identified by the Decision Tree techniques. There are a number of other possible options for future work in this particular research direction as well, which we will discuss in the next section.

6.7 Refinement

In this chapter, we developed and tested a system for predicting the success of a game, measured by the impact it had on the network when it first debuts. This particular strategy provided mixed results, with the prediction techniques employed able to perform only slightly better than random at the task of predicting which games would have an impact and which ones would not. The potential to refine this particular task focuses primarily on improving performance with additional methods and data sources, as any followup methods or processes must have a better data set to start.

As mentioned, it is possible that deep learning could improve upon the performance with no additional processing. Before turning to deep learning, however, feature engineering should be performed to derive some secondary features from the GiantBomb data set used to train the classifiers. For example, we do not use the text of user or professional reviews for a game in the classifiers as-is. We use the number of reviews and, as discussed in the chapter, find that the number of user reviews is extremely effective, probably as a proxy for user excitement for the game. This idea does have a flaw, however. When users strongly *dislike* a game (or product, more generally) they are also interested in leaving a review for the product. Thus, our system could be falsely classifying unpopular games as impactful because they have

a large number of user reviews. Sentiment analysis of the reviews could be used to ameliorate this issue, making it more clear which reviews are positive and negative, which would allow the system to consider these two independently when making a judgement. Such an improvement would decrease the false positive rate of the system, which would obviously yield better performance overall. There is a significant body of work concerning sentiment analysis of reviews which could be leveraged to perform this analysis.

Adding sentiment analysis is not the only potential feature engineering work possible on this review data as well. Each GiantBomb database entry also includes a number of screenshots and publicity images for the game in question, which could be analyzed using computer vision to extract more features from the images. This may be able to provide additional information about the game itself that a genre description cannot provide. For example, both *Destiny 2* and *World of Warcraft* are massively multiplayer games, but they have very different interaction mechanisms, with the former being a first-person shooter and the latter being a slower-paced, third-person, over-the-shoulder role-playing game. If features like this can be extracted from the game screenshots and other media, it may improve the ability of the classifiers to distinguish between games, even when they have the same genre in the GiantBomb data.

Chapter 7

COLLABORATIVE FILTERING

While the Twitch network is a powerful source of data for analyzing behavioral trends related to influence and novelty-seeking as we have discussed in previous chapters, it could also be source of data for Collaborative Filtering or Recommendation, a problem that has been the focus of a significant amount of research effort. Through the play of games and the viewing of games, streamers and viewers give extremely strong signals concerning their interest in a particular game. In particular, this signal is particularly powerful for viewers, who may have spent tens or hundreds of hours watching a game they do not already own. Even without this unique feature, the interest signals transmitted by Twitch have some significant differences from traditional Collaborative Filtering datasets.

In this chapter, a sample exercise of performing collaborative filtering on Twitch data will be presented and discussed, starting with a discussion of the uniqueness of Twitch data for this problem, then discussing the dataset used in the exercise, the techniques used in the exercise, the results of the exercise, and finally the lessons learned from the exercise and possible future work.

7.1 Twitch Data Applicability

In a classical collaborative filtering dataset, like products on Amazon or Yelp's restaurants, researchers rely on explicit signals from purchasers or diners, respectively, to form their data sets. These signals frequently take the form of reviews, where a customer will leave a rating for a product on the website after purchasing and using the product or dining at the restaurant. Usually, these reviews are rated from 1 to 5 stars,

with 1 star being the worst rating and 5 being the best rating. While this strategy for data collection has certainly resulted in great successes in collaborative filtering, it frequently results in missing or incomplete data. For example, this considers only users who are pleased with or upset about the product or food to leave a review. A user who purchases a doorstop on Amazon, for example, may not be motivated enough one way or the other to leave a review of that product.

In addition to the problem of reviewer motivation, reviews are falsifiable, so an unscrupulous actor may unfairly elevate their own ratings and sabotage their competition's by seeding positive reviews on their own products and negative reviews on their competition's. Amazon has attempted to ameliorate this problem by adding 'Verified Purchaser" tags to users who have been verified to purchase that product (presumably by reviewing the product from the same account that purchased the product). However, this simply reverts the problem back to one of user motivation.

An eCommerce site like Amazon could mollify this problem by considering the purchase of the product as the user signal, but in doing so loses the value of the user reviews that are so carefully cultivated on the platform. Twitch, on the other hand cuts through this particular problem by considering not explicit review signals from customers but *implicit* signals from streamers or viewers. In this case, the signal is the play time or view time of a particular game by a streamer or viewer, respectively [1] . This allows data collection on Twitch to consider a gradient of interest in a particular game far more discretely than a simple 1 to 5 rating as provided by Amazon or Yelp. This would be analogous to Yelp considering how many total hours were spent in a particular restaurant or Amazon considering how many times or for how long a particular product was used by the purchaser. Considering both the expanded

[1] It is worth noting that a streamer can also act as a viewer, and many streamers monitor their own channels during their play sessions as a way of ensuring a quality experience for their viewers.

numerical range of signals and the tighter relationship between the signal itself and the real-world use game of the product, in this case video games, it is easy to see why collaborative filtering on Twitch data has the potential to be more accurate that the looser relationships on Yelp or Amazon.

In the next section, the preprocessing steps used to transform the snapshot data described in Chapter 3 to a form suitable for Collaborative Filtering will be covered. In this chapter, our analysis consists only of the Twitch data itself, the GiantBomb data discussed previously is not yet used. Potential applications for this additional data and other potential data sets are discussed with other future work.

7.2 Data Transformations

In the previous section, the implicit signals available from streamers and viewers for collaborative filtering were discussed and compared with the signals available from other common collaborative filtering applications. In this section, the strategy for collecting those implicit signals will be discussed. In the work discussed here, we consider only streamers, as the data collection methodology does not collect information of viewers for reasons covered in Chapter 3. This limitation significantly simplifies the strategy, as it no longer requires differentiation between a user's profile as a streamer and that same user's profile as a viewer. In addition, it does not require consideration of the case where a viewer already owns a particular game, as evidenced by their streaming profile, it can be assumed that if a game appears associate with a user in our data set, that user already owns that game.

The mechanism for converting from a snapshot-based data set to one suitable for collaborative filtering is quite simple, so pseudocode of the algorithm has been omitted. It is sufficient to simply loop through each timestamp and collect a record of which game was played by each streamer at that timestamp. By simply counting

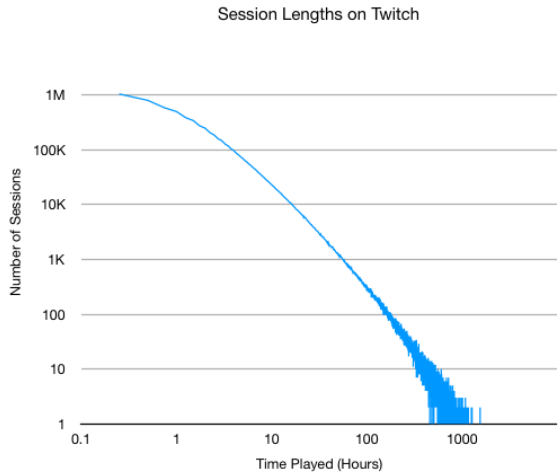

Figure 7.1: Time spent playing a game by a streamer vs number of instances of that value. Note that the Time axis is in hours, as opposed to number of observed snapshots.

the instances of each streamer-game pair, the final dataset can be reduced to a set of triples containing the streamer, the game, and an integer number representing the observation count. Since the snapshots are all taken 15 minutes apart, dividing this number by 4 yields the number of hours that streamer was observed playing that game, though we do not use this value. Figure 7.1 shows the distribution of time played per streamer-game pair. In this graph, we use the number of hours value for readability. From this graph, we can see a classic heavy-tailed distribution. The vast majority of streaming sessions are very short, though it is of note that the beginning of the graph indicates that streams of 15, 30, and 45 minutes are very close in prevalence. This is somewhat surprising, as a stream of 15 to 45 minutes does not make much sense from a streamer's perspective. It seems that this would be too short of a stream to attract a significant viewership to the stream. It is possible, however, that this represents streams that run for 15 or 30 minutes and then, having not attracted significant viewership, are terminated early to avoid damaging the streamer's reputation for streaming to larger audiences.

Using this data, another underlying question can be answered. Collaborative

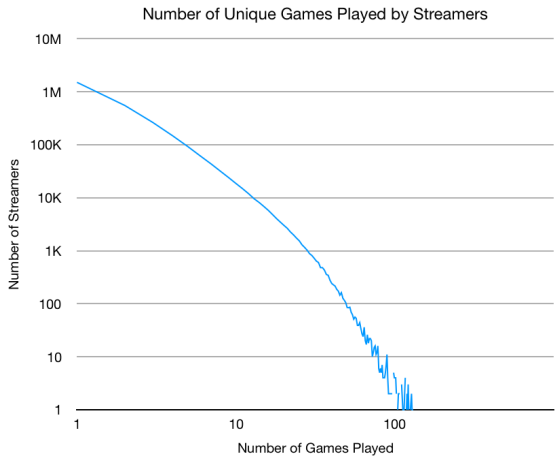

Figure 7.2: Number of games played vs number of streamers with that game diversity. Note that many streamers have only been observed to play one game. This observation is not unexpected, given the historical dynamics of the Twitch network.

filtering as a technique relies on being able to use a particular user's (in this case streamer's) signals about an item to infer preference about a different item. This process, however, is extremely difficult if users do not have signals about more than one item. In Figure 7.2 the number of games played is plotted against the number of streamers with that game diversity. The high percentage of streamers in this graph that have only been observed to play one game is not a surprise in this instance. An interesting area for future exploration would be to explore the extent of the overlap between streamers who play only one game and streamers whose streaming careers have been short in terms of total number of hours.

Before considering the techniques employed by this collaborative filtering examples, the data set generated was subject to one additional transformation. The values for total time played were normalized to the range of $[0, 1000]$. This normalization was done in order to bring the values for time played to a human-interpretable number and therefore to give the resulting predictions a more interpretable range. Since this normalization also transformed the data from integers to floating-point values, there was no loss of precision in this normalization. In the next section, we will discuss the

techniques used in this collaborative filtering example.

7.3 Techniques

While the dataset developed in the previous section uses a < user, item, rating > format to maintain data density, most collaborative filtering techniques that demonstrate good performance use a matrix representation, and it is abundantly clear from Figure 7.2 that the matrix representation of this data set will be quite sparse. Because of this sparsity, techniques must be selected that can handle the data sparsity but also handle the floating-point values for the normalized interest signals. To that end, Matrix SVD and its cousin SVD++ were selected for this particular example.

Both SVD and SVD++ can handle the extreme sparsity of the data in this problem, and as an added benefit, the close relationship between the two models gives us addition information about the problem itself. Here, the formulations for Matrix SVD and SVD++ will be given and discussed before discussing the differences between the two.

Matrix SVD can be formulated as follows:

$$r_{ui}^{hat} = \mu + b_u + b_i + q_i^T p_u \tag{7.1}$$

This model is the same as the one popularized by Simon Funk during the Netflix Prize in [TODO:YEAR]. Broadly speaking, it predicts ratings by performing the singular value decomposition and then using those basis vectors to predict a rating (the final term), factoring in a user bias (b_u) and an item bias (b_i). Since the singular value decomposition necessarily transforms the data set into a zero-mean space, the first term of this formulation adds the original mean from the dataset (μ) back to the predicted rating. SVD++ is an extension of this base model (as the name implies) and can be formulated as:

$$\hat{r_{ui}} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sigma_{J \in I_u} y_j \right) \tag{7.2}$$

In this model, the first three terms are the same, representing the original mean, the user bias, and the item bias. The final term, however, is significantly different, though through its q and p terms, it captures the same information. The new y term, however, captures new information that represents the implicit ratings we discussed earlier. In this case, the implicit ratings are the idea that the item, in this case game, was rated at all, in this case played. Since streamers have a very large number of games to choose from and many games are prominent and successful on the network at any given time, it makes sense to consider the play by a streamer, no matter how long the play session, as an affirmative choice by the streamer. This is especially reasonable when considering that games, like Amazon items or Yelp restaurants, have categories of their own and that while a streamer might not like *Final Fantasy VII* and only plays it for a few observations, this does not mean that the same streamer won't like *Final Fantasy XII*, even though fans of one are usually fans of the other and the contrapositive.

In the next section, we will discuss the application of both of these techniques to the dataset and compare and contrast the effectiveness of different learning rates and regularization parameters for the final results.

7.4 Results

The extreme sparsity of the data, as well as the significant right skew present in any heavy-tailed distribution led the experimental procedure to deviate from the default regularization parameters and learning rates. Using the default parameters, the RMSE of each model was approximately 997, which shows that the data is not suitable for analysis in this fashion. To ameliorate these issues, the learning rate was

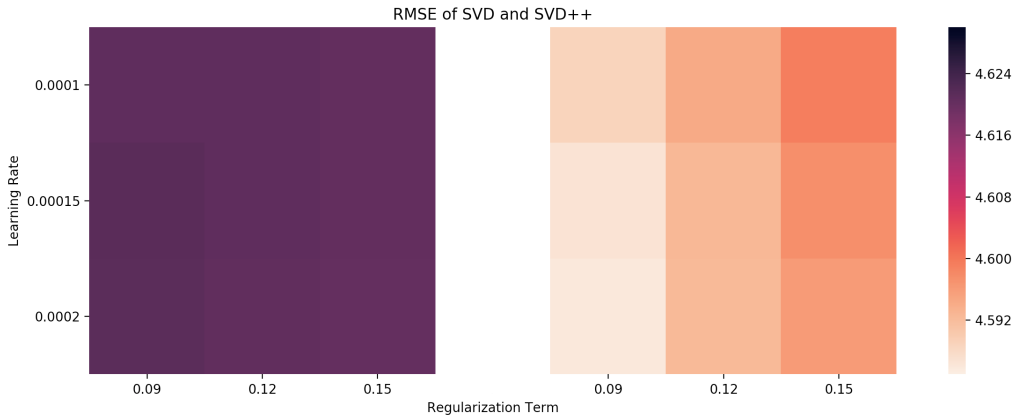

Figure 7.3: Heatmap of RMSE results for SVD (left) and SVD++ (right). Note the magnitude of difference between the two methods.

brought significantly below its default and the regularization parameters was brought above its default. This has the effect of slowing the learning rate's pull toward the higher values at the end of the heavy tailed distribution and increasing the downward pull of the regularization parameter on the final predicted values. Figure 7.3 shows a heatmap of the results for various learning rates and regularization parameters around the best ones discovered.

These results clearly show that SVD++ outperforms SVD by a significant margin for any learning rate and regularization parameter setting. Indeed, since the two graphs use the same scale, it would be easy to infer that SVD++ would continue to be superior regardless of the terms used. On closer inspection of the y-axis scale, we can see, however, that the total difference between SVD and SVD++ is approximately 0.03 points, which amounts to less than 1% of the total RMSE.

While these two methods were specifically selected to handle sparsity well, the fundamental issue remains that the data set is extremely sparse. If some of that sparsity were to be removed, would either method be able to improve their performance? In order to test this hypothesis, a secondary dataset was created that simply removed the single-game streamers that were briefly discussed in previously. Both

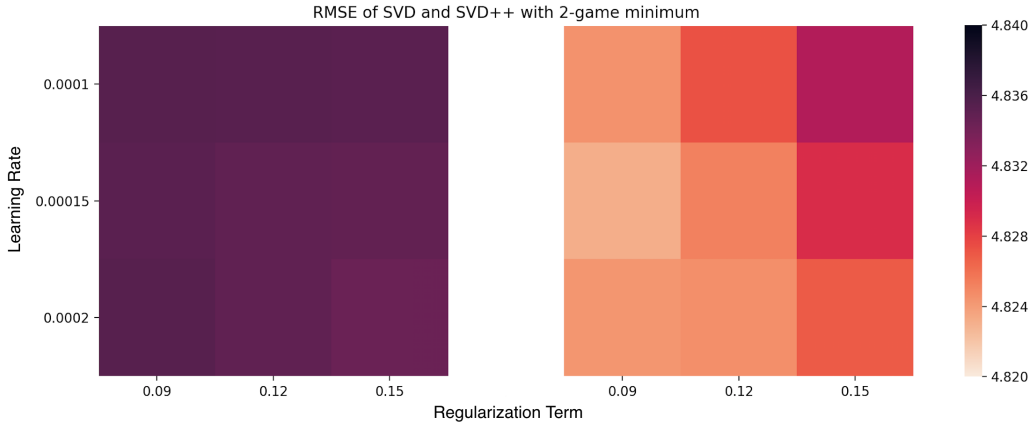

Figure 7.4: Heatmap of RMSE results for SVD (left) and SVD++ (right) after removing all streamers with only one recorded game. Note that the bottom of the scale in this image is above the top of the scale in Figure 7.3, indicating universal decrease in performance.

models were tested on this (necessarily) smaller data set, and the result can be found in Figure 7.4.

Using this smaller dataset, we can see that the performance of the SVD and the SVD++ data set are more closely aligned. However, counter to the expectation that reduced sparsity would improve performance, the performance worsened. While this might initially be surprising, it is easy to see that single-game streamers may be beneficial for predictions by elevating the scores for the popular games and decreasing the scores for unpopular games. This also implies that either single-game streamers tend to play less popular games, or that the ones that play popular games are heavily outnumbered by the multi-game streamers who play popular games as their primary game and then dabble in other, less popular games.

These results show that Twitch data is a potential source of collaborative filtering data. While the techniques for recommendation are relatively simple, the information density of the data as well as the unprecedented granularity of the signals from streamers and viewers have great potential for future improvement as well as new algorithms to take advantage of this data. In the next section we will discuss some

of the potential areas for future work.

7.5 Future Work

In this work presented in this chapter, a simple example of collaborative filtering performed on Twitch data was discussed and its results over a few data sets were presented. Should the need for improved collaborative filtering results arise, there are a few approaches that could be used to improve results. The first, and perhaps most obvious option is to use more advanced techniques for analysis. While SVD and SVD++ are both effective techniques, they are relatively simple compared to state-of-the-art techniques in the most recent literature. While state-of-the art techniques are capable of providing improvements to the performance of our model, it may also be valuable to consider designing an entirely new model to analyze Twitch data, in large part due to the differences in the Twitch data and traditional recommendation data.

The next approach is to add additional data to the model. As was hinted at previously, games have categories of their own, and the GiantBomb dataset contains a mapping of games to their genres. By adding this game information to the dataset, a collaborative filtering system would obtain a mapping that relates games to other games and provides an outside signal to reinforce the implicit genre signals given by streamers having genre preferences. Similarly, users have profiles on the Twitch network that could be used to determine a user-user similarity matrix. For example, if two users are speedrunners of similar games, as listed in their profiles, it is reasonable to assume that these two users are more likely to share game preferences.

The final approached covered here deals with additional processing on the data set used in this example. In this dataset, while normalization was performed, no attempt was made to normalize the data according to the range of play times between games

or between streamers. For example, a highly story-driven game like *Gone Home* has little replay value and is relatively short to begin with. The current example compares playtime for *Gone Home* with games like *World of Warcraft*, which is notorious for players having an enormous number of hours played. While a streamer may be equally interested in both games, the play time for *Gone Home* will be well under the mean, and interpreted as a negative score. Similarly, a streamer who streams full time (40+ hours per week) has more potential hours than a streamer who only streams one day a week (8 hours per week). If these two streamers are playing the same game, it is not reasonable to assume that the part-time streamer is less interested in the game than the full-time streamer, as our example does implicitly. To correct this issue, more effort could be expended on properly normalizing the scores in the ratings matrix to account for these inconsistencies in potential game length and in streamer availability.

Though existing collaborative filtering techniques will be valuable on this particular data set, the best technique may turn out to be one that is crafted singularly for the Twitch network and its uniqueness. Regardless of the final techniques involved, this example shows that Twitch is a potential source for complex collaborative filtering datasets that more closely parallel are real-world use case than a review data set from Amazon or Yelp. This examples shows that not only can Twitch data be used to make meaningful predictions, but that those predictions have the potential to be vastly improved by researchers dedicating time and effort to this particular problem.

Chapter 8

TWITCH SIMULATOR

In the previous chapters, we have focused on the use of actual Twitch data in the analysis of influence. As we will see, there are some disadvantages to having such a focus the actual data, which can result in a slow pace of data collection and analysis. In this chapter, we will discuss our efforts to create a simulator for Twitch data, which would greatly accelerate the process of collecting data; though many of the properties we find in real Twitch data may not be replicated.

8.1 Introduction

One of the concerns with analysis of a live-streaming social network like Twitch is linked closely with its great benefit; the live nature of the network. This live nature means that unlike traditional networks like Facebook or Twitter, historical information is unavailable, irrelevant, or both. Naturally, this presents a problem for researchers as the data collection process forces a substantial lead-up time in the research process. For example, collecting 6 months worth of data on Twitch takes 6 months of real time, where 6 months of a user's Twitter history can be accessed in seconds. To alleviate this problem, we can use synthetic data as a bridge to real data, allowing researchers and analysts to refine their techniques ahead of analyzing real data. In this chapter, a novel Twitch Simulator in proposed and its methodology is explored. First, we will discuss the supplementary analysis conducted on Twitch in support of the simulator's creation, including the analysis that was used as inputs to the simulator. Next, we will discuss the verification methodology that ensures that the outputs of the simulator are accurate to the real data collected from the

Twitch network. In order to promote clarity into the simulator's operation, we will describe the inner workings of the simulator, and finally, we will describe how to use the Twitch Simulator and how the outputs of the simulator are structured.

8.2 Supplementary Analysis

In the chapter concerning the dataset collected from Twitch, we discussed some summary statistics of the data set collected from Twitch, including determining which heavy-tailed distribution best fit the various layers of the network. These summary statistics, however, are not enough to build a robust simulator. In this section, we will discuss some additional analysis that was necessary to build the simulator, including a discussion of why we computed those statistics.

8.2.1 Stream Duration

In order to accurately simulate the Twitch network, it is important to understand how long streamers remain streaming. Is the network dominated by short, pithy streams run multiple times per day or long, elaborate streams. Understanding how stream durations are distributed is important to accurately simulating the Twitch network, but it also has a broader aim. While the basic simulator construction we describe uses only the duration distribution in order to accurately represent how long a particular streamer is online, it is possible that there are effects on streaming duration that vary from game to game. For example, an average game of *League of Legends* lasts for about 30 minutes [1] . Thus, even if a streamer wishes to stream only one game, their stream will be about half an hour. On the other hand, a game of *Fortnite* lasts about 15 minutes, and players who lose at the beginning of the match need not stay until the end of the match, unlike *League of Legends*. Therefore, a *Fortnite*

[1] `https://www.leagueofgraphs.com/rankings/game-durations`

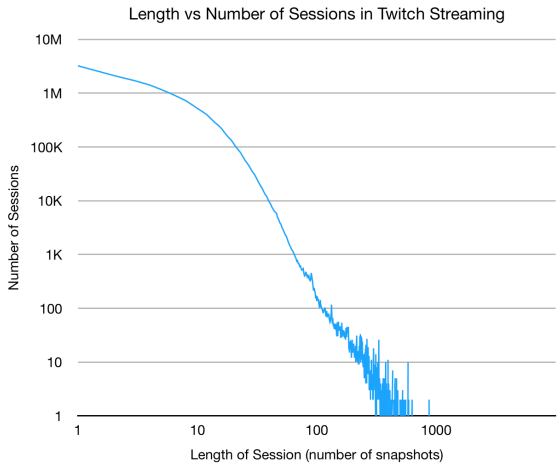

Figure 8.1: Stream length distribution over the entire data set. Note that both axes on the graph are logarithmically scaled, indicating that the distribution is heavy-tailed.

streamer streaming only one game may be streaming for as little as 5 minutes [2] .

In our baseline analysis, we found that the average stream length is 103.7 minutes long. However, this average is misleading, as stream durations follow a heavy-tailed distribution. Analysis of the distribution of lengths depicted in Figure 8.1 with the `powerlaw` package of Python Alstott *et al.* (2014) indicates that the distribution is either lognormal, powerlaw, or truncated powerlaw, but this analysis was unable to determine if any of these three are more likely than any other. Regardless of the actual distribution involved, the fact remains that this heavy-tailed distribution of streams must be accounted for in the simulator in order to accurately represent the Twitch network. However, this does not answer the question posed previously of how game choice affects stream duration. To determine this, we propose the following true hypothesis and its corresponding null hypothesis.

Hypothesis 4 *The distribution of stream durations varies depending on the choice of game.*

[2]Having a stream that intentionally lasts only 5 minutes is extremely unlikely, as such a short stream is very unlikely to capture many viewers.

Game A	Game B	KS Statistic Value
Girls' Frontline	*Cookie Clicker*	0.885
Girls' Frontline	*Blockland*	0.883
Girls' Frontline	*Eternal City*	0.882
Girls' Frontline	*Get Shorty*	0.877
Girls' Frontline	*Music*	0.873

Table 8.1: The top five pairs of games with significant distribution differences. Pairs are ordered in reverse order of KS Statistic. Interestingly, the game *Girls' Frontline* appears in every one of the top five differences, indicating that this game shows unique play patterns.

Null Hypothesis 6 *The distribution of stream durations does not vary depending on the choice of games.*

In order to test this hypothesis, we return to the stream durations depicted in Figure 8.1. Breaking these durations out into the games that contributed to the total duration, we can test the game-level distributions against one another to determine if game choice has an effect on the distribution of stream durations. In order to test these distribution, we rely on the Kolmogorov-Smirnov Test previously discussed Massey Jr (1951). As discussed previously, this test determining if two sample sets are likely to come from the same distribution. In this case, we compare the number of streams of a particular length against one another. If the KS test finds that any pair of distributions between games is significantly different (p value ¡ 0.05), the null hypothesis is considered rejected. An area of possible future work would be examining the network of games which are significantly different and those that are not to see if any patterns emerge between games that (possibly) share stream length distributions. In Table 8.1 below, we highlight the top five distribution differences, as measured by the magnitude of the KS statistic from their comparison.

The results of this analysis portray an interesting picture of the Twitch network.

While the top 5 differences shown in the table all contain the game *Girls' Frontline* as one of the two games, further investigation into the results of this test demonstrate that this particular game's differences dominate at least the top 500 differences, with only 22 exceptions in that list, none of which are in the top 100. In total, $12,372$ games and psuedo-games were tested for their distribution differences, leading to $76,539,378$ possible pairs of games. Of these possible difference, only $7,136,599$, less than 10%, rose to the level of significance. This particular finding is not surprising, since we know that the streamer/game relationship follows a power-law distribution so the majority of the games observed will have a low number of streaming sessions associated with them and therefore the length distribution of sessions for that game will be difficult to determine or compare against others. The total number of significant distribution differences is more consistent with analyzing pairwise relationships between $3,777$ games, which, while still somewhat higher than expected, is much more representative of the diversity of games on Twitch. The important conclusion from this experiment, however, is that they do exist games with significantly different distributions of stream lengths, as we expected from our domain knowledge. Matching this domain knowledge as expected is the game pair *Pillars of Eternity* and *Boderlands*, which have a k-statistic of 0.817. The former game is a slow-paced traditional RPG ala *Dungeons and Dragons* while the latter is a fast-paced Action-Shooter with light RPG elements, so their difference in distribution is expected. Future versions of the simulator should take these differences into account in order to replicate the Twitch network more accurately.

8.2.2 Viewers over Time

While correctly representing stream length is important for correctly simulating the Twitch network, it is also important to accurately represent how a streamer's

viewers evolve over the course of the stream. It would be possible to craft a simulator that does not vary a streamer's viewers during that streamer's online duration, but this is not representative of a stream's actual viewer evolution. In order for the simulator to be accurate, it must be able to account for the fact that not all streams have the same peaks and valleys in viewership. Note that this does not mean that all streams have the same viewership. Indeed, we know that stream viewerships follow a heavy-tailed distribution from our previous discussion of the data set. Understanding how the viewership various over time is important for understanding why a streamer might continue to stream or start to wrap up. For example, a streamer who is currently rising in viewership may choose to continue streaming when they would otherwise stop streaming since they have a good audience. On the contrary, a streamer who is declining in viewership may choose to end their stream early rather than continue to lose viewers. Or perhaps streamers don't care at all about how viewership numbers are changing over time and continue to stream regardless of recent viewership changes. In order to construct the simulator, we first construct a model of short-term viewership evolution that can be applied to simulated streams in order to maintain realistic viewership trends.

In order to develop this model, we first collapse or expand all streams with 2 or more observations in our data set to 100 elements. This allows viewership to be represented in a way that does not vary with the length of the stream. By using 100 points, each individual point can be thought of as a percentile. Similarly, in order to equalize actual viewership numbers, we normalize all streams to having viewership levels between 0 and 1. These viewership levels no longer reflect the actual number of viewers that stream is getting, but rather the proportion of the maximum viewers for the stream that the stream has at a given time. By normalizing streams in this way, we can meaningfully compare streams to one another. We can, for example,

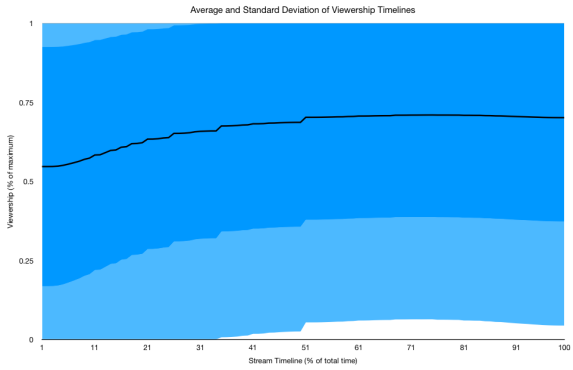

Figure 8.2: Average viewership levels relative to maximum over the course of streams normalized to 100 elements. The black line represents the mean and the two blocks of shading represent the first and second standard deviation, respectively.

compare a short, one-hour stream with a very large number of viewers to a long, multi-hour stream with a low but consistent number of viewers. In Figure 8.2, we can see the result of plotting the normalized viewership streams. In this figure, the average is depicted in a black line, the first standard deviation is depicted in blue, and the second standard deviation is depicted in light blue.

This graph is interesting and runs contrary to our expectations that the viewership with show a decline before the stream terminates. Clearly, on the average this is not the case. However, this standard deviation represented on this graph demonstrates that the distribution of these values is likely not normal, as we expected. In order to make future versions of the simulator as representative as possible to the Twitch network, further investigation should be performed to ascertain the true distribution of these relative viewership values. Previously, we speculated about the value of continuing to stream after a streamer's viewership begins to decline. Since we can easily test this hypothesis, we will perform this test here. While we do not use this information in our simulator, this question is worth investigating. Incorporating the answer to this question into future versions of the simulator may result in a more accurate simulation of the Twitch environment. To that effect, we propose the following hypothesis and it's corresponding null hypothesis:

Hypothesis 5 *Streamers tend to terminate their streams when their viewership numbers start to decline.*

Null Hypothesis 7 *Streamers do not terminate their streams when their viewership numbers start to decline.*

In order to test this hypothesis, we will analyze the behavior of each streamer individually. By grouping each stream according to the streamer, a better picture can be formed of the individual streamer's decision making process. Using each streamer's history, we can then make a fair comparison to determine if there is a relationship between decreasing viewer count and termination of the stream. Note that this test will only operate on one streamer at a time, so while our hypotheses assume that our tests cover the entire population of streamers, our tests will consider individual streamers and we will consider the null hypothesis rejected if any individual streamer is found to terminate their stream when viewership numbers dip. In order to conduct this test, we will perform a causality test on the population of streamers hosted by a given streamer's account. To perform this test, we will adapt the method of Kleinberg Kleinberg (2011). In this formulation, the target event will be the termination of the stream and the candidate event is a drop in viewership. In order to conduct this test, however, what precisely constitutes a drop in viewership is not strictly defined in our hypothesis. Therefore, we will define our candidate event, a drop in viewership, as a change in viewership between two times points that exceeds a certain relative threshold, in this case α. We can represent this formally as:

$$D(s,t) = \begin{cases} 1 & V(s,t) \leq \alpha * V(s,t-1) \\ 0 & \text{else} \end{cases} \tag{8.1}$$

α	Delay				
	1	2	3	4	5
0.9	0.267	0.245	0.234	0.224	0.211
0.85	0.238	0.215	0.206	0.197	0.186
0.8	0.199	0.177	0.169	0.161	0.153

Table 8.2: The fraction of streamers who terminate their streams in correlation with a drop in viewership. In total, $7,044,584$ sessions were tested.

In this formulation, s represents a particular streamer and t is a particular timestamp. Using this basic variables, V is the number of viewers of a particular streamer at a particular time. Note that this formulation does not consider each session of a streamer separately; if a streamer is offline, their viewership is 0. Using this formulation, D is then a binary variable which represents the previously discussed drop in viewership. Note that in this case, D will always have a value of 1 in the timestamp immediately after a streamer goes offline when their viewership drops from the final streaming viewership to a 0 offline viewership. For this reason, in determining the causal relationship between drops in viewership and stream session termination, it is important to introduce a delay between when the drop in viewership occurs and the stream termination. In order to fairly represent the possibilities for viewership change affecting stream termination, we will perform a suite of experiments that vary a number of variables in determining both a change in viewership and causality. To this end, both the α parameter that determines if a drop in viewership has occurred and the effective range of time between the candidate event and the target event will be varied, allowing for a broad experimental space. In Table 8.2, we report the results of this experiment in terms of the fraction of streamers who are believed to terminate their sessions as a result of a drop in viewership.

The results of this experiment demonstrate how heavily viewer counts factor in

to a streamer's decision-making. While many popular streamers operate on a fixed schedule, less popular streamers do not necessarily have the luxury of a tuned-in fan base that will log on to watch them play whenever they are streaming. Therefore, these streamers must begin a session whenever they can, and it is not to their advantage to be streaming when a more popular streamer leeches viewership from them. In this case, it is better to terminate your stream when your stream is not being implicitly compared to a more popular streamer's in order to retain the excitement of your current viewers for the next time you start streaming.

8.2.3 Streamer Game Choice

In our simulator, we use simulated video games to represent the actual games selected by streamers to play. The random strings generated for this purpose are obviously not representative of real games that streamers might be playing. In choosing a game for each new streamer to play, the simulator currently decides if the streamer will play a game that does not currently exist on the network or if that streamer will play a game that does. If that streamer plays a new game, a new game is randomly generated and the streamer's stream plays that game. If the streamer is playing an existing game, then a game for the streamer to play is selected from the set of extant games based on the recorded history of streamers selecting games that already exist on the network. In the current version of the simulator, this is expressed through preferential attachment. Through preferential attachment, we assume that streamers are interested in playing games for which there is already an established user base.

The standard model for preferential attachment, however, assumes that all nodes are of the same type in the preferential attachment task, which is not true of the Twitch network that the simulator is attempting to emulate. To that end, we use a modification of the standard preferential attachment process that preserves the

possibility of streamers to start streaming new games. In the standard preferential attachment model, when a new node is generated, it attaches to exactly one other node on the network with a likelihood proportional to the popularity of the target node. In our model, however, streamers and games are not directly connected with one another, so we must use a different way to use preferential attachment to link streamers and games.

The solution we have found and employ in the simulator is a simple workaround. When a new streamer joins the network, instead of generating just a new streamer, we generate both a streamer and a game. Then, we use preferential attachment to determine which game the streamer starts playing using the number of streamers already playing the game as a base for the attachment probability. To ensure that this new game has a chance of being selected (as it currently has 0 streamers), the new game is given a single 'virtual' streamer. If the new streamer does not select the new game, it is deleted.

This technique has two advantages. The first is the advantage that motivated the use of this technique; it preserves preferential attachment for streamers to games in an environment where the standard preferential attachment model does not apply. The second advantage is that, unlike a flat probability of selecting a new game, this process preserves the idea that as the number of games and streamers active on the network increase, the likelihood of a streamer selecting a novel one to play decreases. This feature is a natural result of the preferential attachment process, as more streamers extant on the network means that the single virtual streamer a new game receives when it is generated represents a smaller probability of being selected by the preferential attachment process.

In addition to this characterization of how the choice of games is made, there are other open questions concerning how game choices are made. In the gaming

world, games are categorized into genres, just like movies (though not the same set of genres). In the movie domain, moviegoers demonstrate a marked preference for particular genres. The question then, is do streamers demonstrate the same genre preference? When a streamer selects a game, are they more or less likely to play one from a genre they have played before? If a streamer, particularly one with a high viewership, continues to select games of the same genres, it is a strong indicator that his or her genre preference is a large factor in their game choices. When a streamer diversifies their game interest it also sends a strong signal, particularly if those games are popular games which implies that the streamer is chasing popularity with their game choices. Therefore, we endeavor to ascertain if streamers do in fact demonstrate genre preferences in their game choices. In order to formalize this test, we propose the following hypothesis and its corresponding null hypothesis:

Hypothesis 6 *Streamers exhibit genre preferences when selecting games to play.*

Null Hypothesis 8 *Streamers do not exhibit genre preferences when selecting games to play.*

In order to test this hypothesis, we compare the genre distributions of each individual streamer against the global genre distribution of games played on the Twitch network during the observation period. In order to compare these two distributions and determine in a statistically sound way if streamers have genre preferences, we conduct a χ^2 test for each streamer. This test will determine if a streamer's genre distribution is likely to be drawn from the background distribution of genres on the Twitch network. In Table 8.3, we report the results of performing this test on the streamers we found on the network. Since the test compares a streamer's genres against all game's genres on the network, we will use this comparison as a proxy for the global distribution of game genres. This is not likely to be a perfect proxy, as the

Reject %	Not Reject %
40.2%	59.8%

Table 8.3: Percentages of streamers who did and did not reject the null hypothesis according to the χ^2 test performed on their distributions of game genres compared with the global distribution. This test indicates that there are a sizable number of streamers who display strong genre preferences that contradict the preferences of the network as a whole.

'streamability' of games is a consideration for streamers, but due to the popularity of newer games on Twitch, it is probably representative of the genre distribution for new and 'fan-favorite' games that are likely to interest viewers. Therefore, rejection by the χ^2 test represents that a streamer has strong enough genre preferences to override this newness/popularity bias on the network.

The results of this experiment are mixed, showing that many streamers are content to play the same types of games repeatedly, giving themselves a specialty genre-wise. These are the streamers that rejected the null hypothesis of the χ^2 test. It also shows that the majority of the streamers did not reject the null hypothesis, indicating that either they do not show genre preferences or that their genre preferences are not strong enough to reject. Earlier, we speculated that this category of streamers represents streamers that follow the 'flavor of the month' game, changing to whatever games have recently been released in order to attract viewership. This test alone does not provide enough evidence to determine if this is the pattern that these streamers follow. Further experiments can and should be conducted to determine exactly how streamers with no clear genre preferences decide which game(s) to stream. While collecting data from the Twitch network will be able to find some of the rationale through a data-driven process, understanding the rationale fully will also require surveying streamers about their game selection processes. This survey process can also be adapted to answer many of the other questions posed in this section that

deal with motivations and choices by streamers. A more subjective survey process, if constructed correctly, would be used to ascertain motivational factors in game selection. These subjective motivational factors can be cross-checked with the data collected from the actual network to provide a further layer of analysis that describes how much streamer stated motivations match their actual behavior.

These supplemental experiments provide a considerable amount of extra information which can be used to increase the verisimilitude of the simulator in future iterations. This does not mean that the simulator will perfectly match the real Twitch network, however. This simulator assumes that the space of games available for streamers to choose is fixed and that it is extremely unlikely for a new game on the network to be propelled to the top of the rankings due to the way that game selection is made. This is consistent with the current state of the Twitch network, with a notable exception. Brand-new games, like *Apex Legends* or *Anthem* frequently enjoy a huge surge in popularity when they are first released, but that popularity dies down relatively quickly as the novelty of the game wears off. That said, there are some games that displace the top games and persist for a long time. For example *Apex Legends* occupies the 7th slot on the rankings by viewership at time of writing, while *Anthem* is just above the 100th position. Though there is a bias toward recent games enjoying popularity, the games that are persistently popular tend to be older games. *League of Legends*, *DOTA 2*, and *Counter Strike: Global Offensive* enjoy continuous popularity and rarely drop below the top ten spots. This indicates that there is still much to learn about the "streamability" of a game or games and that simple heuristics like recency are not good predictors of current or future popularity. In a larger sense, this indicates that this simulator, while it captures the state of the Twitch network at time of writing, will not be able to evolve with the Twitch network by capturing the trends associated with newly released games. In Chapter 6,

we attempt to build a classification system that can determine if a game is going to have an impact or not, with mixed results. In the next section, we will discuss the algorithm of the Twitch simulator and show how the simulator uses these additional data elements to create simulated snapshots of the network.

8.3 Simulator Operation

In order to successfully emulate the Twitch network with the caveats already discussed, this section will detail the operation of the Twitch simulator. This allows the simulator to be replicated by any readers who desire to do so and have access to the underlying data analyzed in Chapter chap:data and the previous

Generally speaking, the simulator is designed to adhere to the overall trends discussed in the Twitch Data chapter as closely as possible. The analysis performed in the previous section is a means to support those overall trends. Specifically, the simulator focuses on maintaining an accurate streamer count and, through the statistics collected previously, accurate game and viewer counts as well as accurate pairwise relationships between the three layers of the Twitch network. Ideally, when the simulator has any number of active streamers, with a representative set of viewers for each streamer, and with each streamer playing a game selected appropriately, the simulated network should have the same characteristics as the real network. This is because the generation mechanisms for the simulator respect the analysis, and presumably the real-world generation mechanisms that undergird the Twitch

The first value that must be determined in the simulator is the number of streamers. Fortunately, this is quite easy to determine. From the dataset discussion, we know that the number of streamers active at any particular time varies over the course of 24-hour and further 7-day cycles. By accumulating the number of streamers observed in the dataset over these weekly cycles, a target number of streamers can

be computed for each time point. Note that the simulator, unlike the dataset we collected, is more than capable of generating points between the 15-minute intervals that our data collection is limited to. Using the computed target number of streamers and the current number of active streams on the simulated network, the simulator can then create new streams so that the number of active streams matches the expected number of streams.

In creating these streams, the simulator also assigns a game to that stream/streamer. This process of assigning a game to a stream follows the game choices outlined previously, which allows the simulator to correctly mimic the distribution of games and streamers. This also has the effect of mimicking the heavy-tailed distribution of games and streamers. In the first version of the simulator, the process does not use real game names. As discussed previously, what makes a game streamable is difficult to predict and without an reasonable ability to make such predictions assigning real game names may give user of the simulator the wrong impression about the intended use case of the simulator.

Similar to the game assignment, at the time of creation the simulator also assigns a maximum viewer count, viewer trajectory, and stream length to the streamer at the time of streamer creation. These three factors are necessary for each streamer in order to properly mimic the Twitch environment, as every streamer has a viewer count that fluctuates over the course of their time streaming. Furthermore, using these metrics allows the viewer-streamer relationship to be maintained as a similar heavy-tailed distribution. In each of these three cases, the methodology for selecting these three attributes is much simpler that the methodology required to select an appropriate game for the streamer. In this case, the simulator can simply maintain a distribution of stream lengths, maximum viewers, and an abstract representation of possible trajectories. Determining each new streamer's viewership over time can

then be done by stretching the trajectory to the appropriate length and multiply the proportion of maximum viewers at each time point by the actual maximum viewers drawn from the distribution of maximum viewerships. As the simulator moves forward in time steps, the viewer count for that streamer simply becomes the next viewer count in the timeline until there are no viewer counts left, at which point the streamer "goes offline". Pseudocode for the simulator algorithm can be found in Algorithm 3.

```
while active do
    timestamp + = 1
    if Active Streamers < Expected Streamers then
        foreach expected - active do
            newStreamer.game = ChooseGame()
            newStreamer.viewers = ChooseMaxViewers() .*
             (ChooseStreamLength() X ChooseStreamTrend())
            streamers.append(newStreamer)
        end
    end
    WriteSnapshot()
    foreach streamer ∈ streamers do
        streamer.viewers.pop()
        if len(streamer.viewers) == 0 then
            del streamer
        end
    end
end
```

Algorithm 3: Pseudocode for the Twitch Simulator

In the pseudocode, the highlights discussed in the previous paragraph can be clearly seen. Contained within, the symbols .∗ and X are used to represent the

vector operations performed on the projected stream. In this case, the X symbol is used to represent the scaling operation that adjusts the length of a simulated stream from the 100-element templates to the actual length of the simulated stream. In addition, the $.*$ symbol is used to represent the scaling operation that multiplies the selected simulated stream template, which ranges from 0 to 1 representing proportion of maximum viewers to the real viewer numbers by simple element-wise multiplication. The helper function which executes after the main loop, WriteSnapshot, serves to output the state of the simulated network at that timestamp. The final loop advances the existing streamers to the next timestamp by removing the current viewership count and then offlines all of the streamers who no longer have any entries left in their simulated streams.

8.4 Results

The simulator discussed in the previous section uses data directly from the Twitch network to inform the generation process, directly simulating streamer counts, streamer-viewer relationships, stream length, and streamer-game relationships. However, the generation process contains no verification that the generated snapshots accurately represent the Twitch network. In order to perform this particular verification, we must test the generated snapshots against the real snapshots using a metric that is not directly generated by the simulator. In this case, this is the game-viewer distribution.

Since the game-viewer distribution is not directly affected by any of the generation steps of the simulator, it can be used to verify that the generation process followed by the simulator results in snapshots that accurately duplicate the real snapshots found from the Twitch Network. In this section, we will compare the game-viewer distributions of the generated snapshots against the distributions of the gathered

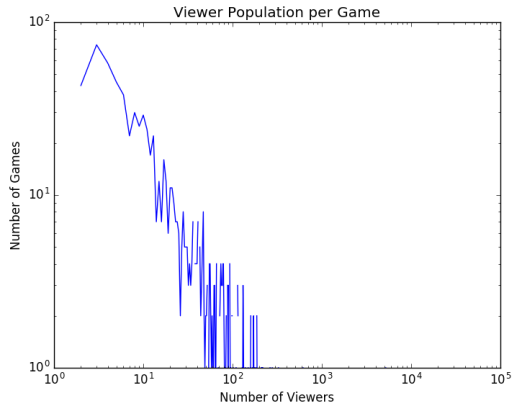

Figure 8.3: Real distribution of number of viewers vs. number of games with that viewer count.

snapshots to verify that the simulator accurately generates snapshots. Figures 8.3, 8.4, and 8.5 are three graphs that demonstrate the ability of the simulator to emulate the Twitch network. The first, Figure 8.3 is a graph of the viewer/game distribution from the Twitch data set. The second and third graphs in the figure show a distribution taken from one run of the simulator. The second graph shows the viewer/game distribution graph from the initial state of the simulator, which is randomly generated using the algorithm given above with 0 active streamers and the appropriate number of expected streamers. The third and final graph shows the viewer/game distribution after a day of simulated operation.

The simulator described in this section has much work that needs to be done in order to faithfully emulate the network, as discussed previously, though these three figures demonstrate that the current state of the simulator is strong. The preferential attachment model for initialization of the simulator is clearly effective in duplicating the state of the simulator. In comparing the two graphs of the real data and the simulated first timestep data, it is clear that the simulator may generate too many viewers on too many games compared to the real data, though this may be an artifact of the timestamp chosen to represent the real data. Overall, the viewer/game distri-

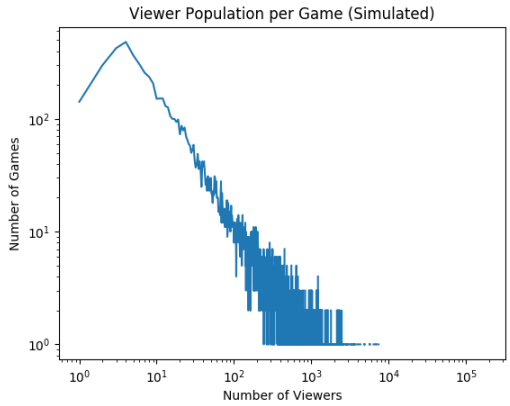

Figure 8.4: Distribution of viewers vs. games as Figure 8.3 as generated by the simulator at initialization. Note the similarities between this graph and the graph of real data.

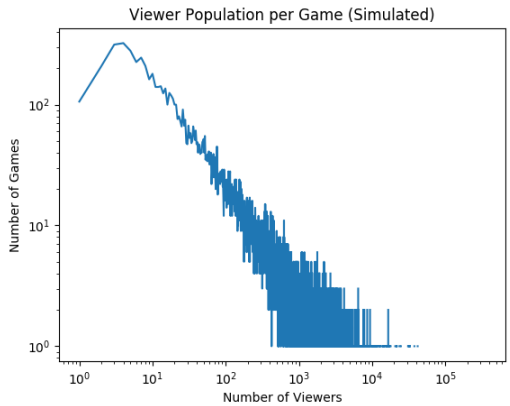

Figure 8.5: Distribution of viewers vs. games as Figure 8.3, as generated by the simulator after simulating one week of streaming. Note the similarities between this graph and the graph of real data.

bution created in the initialization step of the simulator is a reasonable estimation of the real data.

The last figure showing simulator data, Figure 8.5, demonstrates the efficacy of the simulator in ensuring that the timestamps generated by the simulator remain faithful to the Twitch network. Comparing Figure 8.5 to Figure 8.3 demonstrates that the simulator does not 'drift' off of the real data's distribution, though the distribution may have some of the same problems as the first timestamp's distribution, namely that it may overestimate how many unique games are active and the overall viewer

numbers.

In addition to closely paralleling the real Twitch data, the simulator we develop has the advantage of generating data much more quickly than collecting the same amount of data from Twitch itself. In the run of the simulator used to generate data for the graphs presented previously, an average of 235 seconds were required to generate one timestamp, including time spent initializing the simulator, which is substantial due to the size of the heavy-tailed distributions that the algorithm uses to determine a given streamer's maximum viewers and stream length. Without factoring in the initialization time, 233 seconds per timestamp are required. This is substantially faster than the 15 minutes between timestamps used in the real Twitch data. In addition, the simulator currently has room for improvement since it uses the NetworkX library to maintain a bipartite graph, which is not strictly necessary due to the simplicity of the graph and may be contributing to the generation overhead.

Throughout this chapter, we have discussed the operation of a basic Twitch simulator as well as the means by which such a simulator could be further improved. As it stands, the simulator is sufficient for analysis of viewer distributions, but lacks a large amount of the detailed simulation that would make it suitable for more detailed analysis. Implementing these improvements, especially the ones that link specific streamers to specific games and account for the various attributes of the games themselves that affect which streamers play games and for how long, would be a substantial increase in the ability of the simulator to serve as a surrogate for real Twitch data.

8.5 Simulator Uses

The simulator, in its current form, has a number of uses beyond simply generating data. In addition, as the simulator becomes more and more feature complete, the space of possible uses widens. In this section, we will discuss some of the uses for the

simulator both now and in the future as more features are added. In particular, we will discuss two use cases for which the similar can be used right now and two that will be possible after two planned additions to the simulator are made.

The first use case that is currently possible involves tracking the rise and fall of games in the simulator as a proxy for tracking them on the Twitch network. Currently, there has been very little rigorous work on tracking the mechanisms by which a game rises to popularity, whether this is on release as our work on game debuts studies or later on, perhaps as a result of a patch. The most notable example of the latter form of a rise to prominence is *Fortnite*, which rocketed to the top of the Twitch network after a Battle Royale mode was patched in similar to *PUBG*, which we have discussed in previous chapters of this thesis. Aside from being able to predict that a particular patch will presage such a rise the exact mechanism for this rise are unclear. Such a rise could be caused by a single high-viewership streamer starting to play, then attracting other streamers. It could also be caused by a large number of low-viewership streamers playing which attracts the attention of the 'big fish'. Having such an upheaval in the Twitch network is relatively rare, so it would require the continuous collection of a large amount of data joined with careful supervision of the real network in order to determine exactly when such an event takes place. However, with the simulator, we could generate data and look for this pattern much more quickly that we could in the real network. This would be an interesting case study to check for exactly such a pattern and then attempt to use real-world data to verify whatever conclusion was reached using simulated data.

In addition to tracking the rise of newly patched or newly popular games on the network, the simulator could also be used to indirectly verify one of the avenues of research we propose and discuss earlier in this thesis. This particular avenue is the impact detection and prediction work discussed in Chapter 6. While the current

version of the simulator accurately replicates the distributions between the three layers of the network in an overall sense, it does not at any point attempt to model removing viewers from a particular streamer or game in order to advantage a newly-emerging game that might be of more interest to these theoretical viewers. This makes the simulator a perfect test bed for verifying one of our underlying assumptions about the Twitch network in the context of impact detection; that new, more interesting games leech viewers from existing games. To this effect, we could generate a large amount of data from the simulator and then repeat our impact analysis using this newly generated data. Since our simulator explicitly does not remove viewers as stated previously, this should manifest in the simulator as a lack of, or a small number of, impact events unlike what we found on the Twitch network. This application is a valuable sanity check on our work and something like this could be repeated with other work performed on the Twitch network.

Another such sanity check that is possible, but not with the current version of the simulator, is verification of our definition of influence on Twitch. Our current model of influence relies on the idea that the existence, or lack thereof, of other streamers influences the decisions a particular streamer will make. As in the previous example, the simulator does not reflect this idea. The choice of game to play for a particular streamer is based solely on the number of active viewers of that game in a preferential attachment fashion. With this consideration, the simulator should show no effect of influence on the network, so if we repeat the influence analysis performed in Chapter 4, potentially influential sessions should be distributed across streamers at random. Before performing this analysis, however, there is an additional feature that needs to be added to the current iteration of the simulator. The simulator currently generates streamer IDs at random when adding a new streamer to the pool rather than considering various data points about streamers, like schedules, time

zones, or historical start times. This means that regardless of the true nature of this influence analysis, the potentially influential sessions would be randomly distributed across simulated streamers anyway. Further work is necessary to develop a system for codifying schedules and meaningfully assigning streamer IDs when they are added to the pool of active streamers.

In addition to the rise of new or updated games as discussed previously, another phenomenon that seems to defy common wisdom and is not repeated on other network is the persistent dominance of particular games, like *Fortnite* discussed before. Unlike other networks, where memes and topics are constantly shifting in response to external factors, even when new games are released, the most popular games suffer only a temporary setback or often no setback at all. Simultaneously, work done earlier in this chapter shows that most streamers exhibit strong genre preferences, and that those preferences often defy the overall trend of Twitch. This leaves us with the question of how the top games remain at the top when they are outside of streamers' preferred genres. In the current version of the simulator, we do not consider genre when either creating a new game or assigning a new streamer to a game. However, if that consideration were to be added, with the appropriate weighting to ensure that viewership and genre preferences are in balance, we could come to a deeper understanding of how these top games continue to dominate. In particular, we could consider if this dominance is the result of streamers ignoring genre preferences, or if it is simply the case that the most popular streamers do not have genre preferences that defy this trend.

While the current iteration of the simulator appears simple, and is in some ways, we believe that this does not diminish the ability of the simulator to answer questions about the Twitch network, particularly those that rely on contrasting the actual network against the simple models that are believed to underpin the real network.

In addition, as the simulator becomes more complex, and the tested and verified behavior of streamers and viewers continues to be factored into the simulator's model, the possible options for continued usage of the simulator grows. As more and more researchers consider and are convinced of the value of analysis of the Twitch network for the reasons we have laid out previously, we believe that the value of simulator will continue to grow, as it will provide a backbone against which researcher can test their models for veracity. In doing so, they will also contribute to the simulator, as those same behavior patterns can be codified into the simulator and in turn increase its veracity.

Chapter 9

CONCLUSIONS

The work conducted for and explored in this thesis provides groundwork for continued research on Twitch as a social network through direct and indirect comparisons to existing social networks as well as demonstrating the value of conducting research on Twitch independently. In this chapter, the high-level outcomes of the work done here will be discussed and directions for future research will be presented and evaluated.

9.1 Contributions and Advancements

Primarily, the work presented has covered the Twitch live-streaming social network and has performed a variety of different types on analysis on a dataset collected from the network. Specifically, analyses of influence, success prediction, and collaborative filtering were conducted on the network. In the cases of collaborative filtering and success prediction, the techniques utilized in the analysis were relatively simple, off-the-shelf techniques available in many machine learning packages. Rather than a weakness of the analysis, we consider this a strength, as even these relatively simple techniques demonstrated that there is value in conducting this type of analysis on the Twitch network. Indeed, both of these analyses indicated that there is some potential for more advanced, Twitch-specific version of these techniques to provide better results on data collected on the network. More detail about this potential can be found in the next section. In the case of the influence analysis, we designed a custom technique to determine the influence of streamers on a more fundamental level than simply investigating popularity. The lack of definite results from analyzing influencer

types or modalities also demonstrates the possibility for further investigative work. As before, in the next section, we will discuss possible future directions for continued analysis of Twitch influencers.

Synthesizing the knowledge gained from the three avenues explored in this thesis demonstrates why further exploration of Twitch is necessary in general. From these three directions, we can start to see the differences in user behavior between Twitch and the other major social networks. While individual users still behave in similar ways, as demonstrated by the effectiveness of collaborative filtering on the dataset of users and games played, the network as a whole behaves in ways that are difficult to predict with respect to game choices, as demonstrated by the predictive analysis task. This is consistent with what we would expect to see, given what is known about other social networks. For example, on Twitter and Facebook, collaborative filtering and other recommendation techniques have been successful in recommending content to users of the networks. This is analogous to predicting new games for streamers to stream or for viewers to watch, the streamer side of which is explored in Chapter 7. Similarly, predicting which content will 'go viral' on Facebook and Twitter is quite difficult. The techniques we explored have a similar difficulty in predicting which newly released games will 'go viral' on Twitch. From our domain knowledge, we know that there are games that have this effect on the network, like *Fortnite* or *Red Dead Redemption 2*. Both of these games drew enormous amounts of attention when they were released and *Fortnite* has maintained its position at or near the top of the concurrent viewership rankings.

The emergent and continued popularity of *Fortnite* presents a unique challenge, as its rise did not begin until the game was expanded and the now-enormously popular Battle Royale mode was added (and this mode was made free-to-play). This challenge is unique, and indeed continuing, as the method described in Chapter 6 is designed to

capture the impact of *new* games, as opposed to updated ones. *Fortnite*, for example, was released as a base-building, cooperative survival game and still contains that mode [1] . When *Playerunknown's Battle Grounds* (*PUBG*) rose in popularity, the Battle Royale mode was added to the early access and later made free-to-play. By maintaining a free-to-play model, *Fortnite* quickly surpassed *PUBG*, which has an up-front cost. This substantially increased the accessibility of *Fortnite* and allowed it to surpass *PUBG*. While *Fortnite* and other Battle Royale games remain highly popular, this phenomenon is not in scope for this discussion and is a good topic for future work. From the perspective of our popularity prediction technique however, *Fortnite* would have already made its debut on the network and would therefore be ineligible for future consideration. This weakness in our method should also be corrected, allowing for reconsideration of games that have gotten significant updates. Unfortunately, this would require a new data source to determine if and when a game receives an update which substantially changes the game (like adding a new game mode). The GiantBomb dataset discussed in Chapter 3 does have a *Date Last Updated* field, which would appear to solve this problem, but this field lists the most recent update regardless of the relative size of the update. This means that keying off of this field would cause the technique to reprocess games any time an update of any magnitude is published, not just major updates as we proposed earlier.

While these two facets both parallel existing social networks, the third aspect explored in Chapter 4 shows that Twitch has important differences from existing social networks. In this chapter, we found that the effects of influence, as opposed to popularity, operate substantially differently on Twitch than they do on traditional social networks. In this case, we define influence as the ability to induce others to change their behavior and popularity as the ability to attract viewers to one's own

[1] `https://en.wikipedia.org/wiki/Fortnite`

content (in this case stream) and find that the two are not strongly linked. Indeed, it appears that in many cases there is an inversion; extremely popular users are not influential, possibly because they crowd out the available attention space and discourage other streams from playing the same game. There does exist, however, a sweet spot in the middle where moderately popular streamers are also influential and their broadcasts are accompanied by an increase in overall popularity of the streamed game. In our work, we identify a number of such influential streamers and consider why they would be considered influential. This is in contrast to studies of influence on traditional social media, which is strongly linked with popularity. This could be due to the difference between attention mechanisms on Twitch and on traditional social media. In the case of traditional social media, users transmit thoughts and ideas through short, easily-consumed posts and a site visitor's session involves consuming many of these posts from a wide variety of different users. Consider a Twitter user's feed, which consists of a number of posts organized in approximately chronological order coming from a wide variety of other users that the first user follows. This means that it is relatively easy for a user to consume all of the tweets that someone they follow publishes, and conversely means that someone tweeting can be assured that their tweet will have reach proportional to their number of followers. Therefore, the number of people possibly affected by that tweet is proportional to their number of followers. However, this is not necessarily true on Twitch. On Twitch, users essentially watch a single streamer per viewing session. Since each viewer is limited to one streamer or a small set of streamers, a streamer cannot assume that all potential viewers interested in their stream will see it, since streamers are directly competing for attention. In contrast, the ability of twitter users to view a large number of tweets means that there is not as much competition for attention.

While there are certainly some similarities between Twitch and traditional so-

cial media, as demonstrated by the predictive power of collaborative filtering, some behaviors and interactions dynamics are substantially different between the two platforms. Though the difference between popularity and influence on Twitch may seem trivial, it is important for understanding the dynamics of human behavior. If these two features of human behavior operate differently in the different contexts of Twitch and traditional social media, what other well-studied features operate differently? Do communities form and grow as readily on Twitch as on traditional social media? Or do the limitations of attention on Twitch lead to small communities with very little overlap? In addition, which of the popularity and influence models is truer to the way that society self-organizes offline? It seems reasonable to assume that different scenarios in the real world lean toward one or the other differently, but an investigation must be done in order to find out. Such an investigation must be carried out in person or through surveys, further increasing the difficult of performing such an experiment. In the next section, we will discuss some of the possible future directions for research on Twitch or stemming from the research conducted here.

9.2 Future Research Directions

In each chapter here, we presented and briefly discussed some possible avenues for future research directions resulting from the work we conducted. In this section, we will focus on entirely new areas of research that are inspired by the availability of Twitch data. In doing so, we will cover both extensions and applications of existing techniques and methods to the Twitch networks and completely new areas of research that are enabled through access to the Twitch data. The new areas proposed in this section cover a number of recent phenomena that have been documented and studied anecdotally but, to our knowledge, have not be studied in a systematic, empirical manner.

The first area of interest is in community detection. From our domain knowledge, we know that there are groups of streamers who support each other's streams and advertise for their fellow streamers through the 'hosting' mechanic of Twitch. This hosting mechanism allows a particular streamer to 'signal boost' another streamer by rebroadcasting the second streamer's content on their channel [2] . If we were able to collect a data set of streamers who host other streamers and the streamers they host, it would be possible to create a hosting network of streamer-streamer interactions and perform community detection on this network to determine if strong communities exist on the network. In addition to this linking mechanism, there exists a feature on Twitch, called 'teams', that allows streamers to self-declare that they are part of a group of streamers. This feature could be utilized to provide an additional data source for the community detection task. While the teams feature and the hosting feature likely have significant overlap, as streamers with a team are more likely to host teammates than non-teammates, it is also likely that there are streamers who host outside of their team, if only because they have no active teammates. The results of this community detection work could be used to provide information about whether or not groups of streamers working together have a better change of rising above the noise of the network than streamers working alone. Similarly, it could be used to connect individual streamers to group of streamers working together in order to encourage that streamer to continue streaming. Implicit in this last task is the assumption that streamers who are members of teams or groups are more likely to continue streaming, either in the short term, individual sessions or long-term streaming career. Community detection could also be used to recommend new streamers to viewers, under the assumption that individual members of a group of

[2]This does not affect our data collection, since any viewers are included in the boosted streamer's viewer count and the Twitch API consider the hosting streamer offline.

streamers are likely to be interested in and broadcast similar games and a viewer of one of the group is therefore likely to be interested in the games that another member of the group is broadcasting.

In order to conduct the work described in this thesis, we collected a large dataset concerning the Twitch network. One data source we did not explore directly from the network was information about a particular streamer's viewers. Partly, this was because the work we were planning to perform did not require this information. However, we also chose not to collect the data due to the additional technical complexity of collecting that information. Due to the structure of the Twitch API at the time of data collection, collecting information about each streamer's viewership would have required at least one additional API call for each streamer in the snapshot, which would have been prohibitively slow and prevented us from collecting snapshots every 15 minutes, as we currently do. With additional time and resources, however, the collection technique could be modified to increase parallelization and collect individual viewership data. By doing so, we could expand the work of many of these sections to include analysis from a viewer perspective in addition to the streamer perspective we currently analyze. For example, by analyzing viewership records, we may be able to generate some insights on which streamer behaviors attract repeat viewers, attract new viewers, and possible even discourage viewership. This would be a valuable resource for streamers who are just getting started on the Twitch network, as these streamers would be able to reference which behaviors to avoid or continue in order to maximize their viewership. We could also expand the collaborative filtering work we performed on the streamer/game data set to a viewer/game data set. Performing this work would provide an interesting perspective due to the possible differences in recommendations between streamers and viewers. In previous discussions, we mentioned that different types of streamers may turn out to have different

recommendations when their motivations for streaming are implicitly or explicitly taken into account. This may be true of streamers and viewers as well. For example, a viewer may be interested in new game content so that the viewer can observe the game being played before they make the decision to purchase the game. In this case, learning the genre preferences of the viewer is important, as a viewer who prefers RPGs may not be interesting in a first-person shooter game at all. Similarly, it is important to have additional game information, since this hypothetical has price as a factor. A free-to-play game, for example, is not a valuable recommendation as the viewer does not need to get a preview of a game he or she can simply download for free.

In addition to verifying assumptions or theories about what actually attracts viewers to a particular streamer's channel using individual viewership data, collecting data about the individual viewers of a stream, even if that data is limited to usernames, would allow us to formulate and test hypotheses as to what makes viewers leave a particular stream and join another one, especially when that migration occurs during one of the streamer's streaming sessions. Determining the factors that lead viewers to abandon one streamer for the other may help interested streamers retain viewership or, somewhat more nefariously, help streamers attract viewers from other streamer's streams. This migration of viewers is important for understanding exactly how the various factors we have discussed affect viewership. While tracking trends at an aggregate level is certainly possible and may provide some insight into the broad movements on the Twitch network, individual viewer information is necessary to gain a better understanding of the trends and forces on Twitch that affect specific groups of viewers who have specific goals from their viewing experience. Having the viewership history of individual streamers would make the process of discovering these groups and determining what motivates their viewing behavior much easier.

While stream viewer listings would be helpful for determining viewer behavior, there is another source of data we do not collect in the current iteration of the data collection algorithm. That data is the text chat data from the chat channels attached to each streamer's broadcast. This data has been analyzed by some prior work on Twitch, but it has not been linked with viewership data or viewership over time as we analyze in the work presented here. In the context of the work already presented here, we could use chat data for additional evidence that the streamers we believe are influential are actually influential. Theoretically, an influential streamer would increase engagement with the game they are playing along with all of the other ways we measure engagement on Twitch. In the current iteration of the technique, this means numbers of streamers and number of viewers. However, we could also measure engagement by considering the number of chat channel messages sent in streams playing the game in question. A measurable increase in chat messages across channels is evidence that the streamer is actually driving engagement among real viewers as opposed to viewbots. Analyzing chat channel activity may also be a way for Twitch to combat viewbots, or at least streamers that use viewbots. With the enormous number of streamers on Twitch, there is sufficient representation for different levels of viewership to establish a baseline for chat channel activity at a particular level of viewership. Using this baseline, a streamer's chat channel activity can be used to try to determine if that streamer is using viewbots by measuring channel activity against the baseline. If a particular streamer fails to meet or exceed that baseline repeatedly, his or her channel may be employing viewbots to artificially inflate their view count. Streamers use viewbots due to the incentive structure of Twitch, the potential for analysis of which we will discuss later. These two examples are a demonstration of what could be done with just chat channel message frequency data to enhance an exiting research area and open a new one. More could be performed with more

structured, detailed analysis of chat channel content as well.

Using the message frequency is not the only way that chat channel data could be utilized to learn more about the Twitch Network. Prior work has already compared the chat channel content of male and female streamers to determine if there are any significant difference, and found that there were. The chat channel content can also be used to enhance some of our existing work. In addition to considering message frequency when determining influential streamers, we can also consider message content. As we have discussed previously, an influential streamer is essentially a representative for the game they play. Therefore, an influential streamer's chat should also be focused on the discussion of the game's content. We could adapt the techniques of the previously mentioned prior work to analyze the chat channel content as further supporting evidence that a streamer we believe is influential is actually influential. There are other possibilities for analysis of chat content as well. Sentiment analysis could be used to determine if a streamer's chat is actually enjoying a game or not. This would be useful for two different reasons; first it would help determine if the viewers are interested in the channel for the streamer or the game. Determining what attracts viewers to a particular stream would be valuable for longer-term analysis of the Twitch network and this is one step in that direction. Second, this could be used as a feedback mechanism to a game developer. If there are aspects or segments of a game where the chat content becomes overwhelmingly negative, this may be an indication that this is an area of the game that needs to be streamlined or improved. The developers can then patch the game to ameliorate these issues. Perhaps most interestingly, Twitch chat gives viewers access to a larger palette of emotes than normal ASCII emotes, and it can often be difficult to tell what these emotes mean, especially since partnered streamers can create custom emotes for their stream. Sentiment analysis may be able to tease out the general meaning of these

emotes and present a guide to new viewers and users about the uses of each emoji. This could be especially valuable in the context of evolving meanings of emoji, as seen on other social networks.

While viewership of games varies for a number of reasons, we theorized that one of the reasons viewers may choose to watch a channel is that they are interested in getting more information about a game in advance of purchasing it in order to help make a purchasing decision. If this is true, then new games that have an entry price should have more viewers than new games that are free-to-play. This would be relatively easily testable with a dataset of Twitch viewership levels and a listing of the release dates of new games along with their costs. Note that this last piece of information is not available in the GiantBomb data set, so we would need to find or generate that data on our own. While the GiantBomb data does have release dates, it does not have pricing information in the data set. If it is true that viewers are more likely to watch games with entry costs, then this would be important information for u-and-coming streamers who have some disposable income or are willing to purchase new games as investments to their streams.

While purchasing new games as they become available can be considered an investment for a particular streamer in order to improve viewership, there are other, more obvious investments that streamers make with the intention of improving their streaming experience for their viewers. These can range from commissioning graphics for their stream overlays to hardware investments in video cameras, microphones, and even entirely new computers to ensure that their streams run at as high a quality as possible. While Twitch encourages streamers to provide a high-quality experience for their viewers, they do not provide any financial incentives (as far as the authors are aware) or assistance to those streamers who spend considerable amounts of money to improve their streams. Instead, the return on this investment is implicit through

the promise of attracting more viewers to their channels and earning greater revenue through ad views and channel subscriptions. In this way, Twitch fits in to the so-called "gig economy" of Uber, Lyft, YouTube, and other services where users are not guaranteed that their investments will pay off in any way. Since Twitch data is much more available than data on Uber, Lyft, and other such services, it would be possible to conduct a study of how this gig economy model affects Twitch streamers and if, in general, the investments in streaming equipment does represent a significant value add for the streamers that make such investments or if they are net losses for the streamers. More broadly, this type of "gig economy" research is necessary as more and more potential participants read news articles about highly successful "Instagram Influencers" and try to follow in their footsteps. Well-researched, peer-reviewed science is a necessary part of determining if the gig economy is sustainable in the long term or if regulation or legislation is necessary to prevent the exploitation of gig economy participants by massive tech companies.

One of the interesting outcomes of Twitch's dominance of the market is that almost all gaming-related events have a Twitch tie-in of some sort. For example, the Electronic Entertainment Expo (E3) is the preeminent conference for video games and advancements in gaming. New games, new consoles, new hardware, and many other developments are announced at E3 due to the amount of media attention directed at the conference during its duration. During the 2019 conference, which took place in early June, there was a substantial virtual presence on Twitch at well. In fact, the Twitch broadcasts ran nearly the entire duration of the conference [3] . In addition to the domain-specific conferences and trade shows, other events occur on Twitch that are worthy of discussion. For many years, charity fundraising events called *Awesome Games Done Quick* and *Desert Bus for Hope* were created and persist primarily for a

[3] `https://i.imgur.com/Xrigbn0.png`

streaming audience. Though these events are scheduled well in advance, many events occur which are not predictable but certainly are interesting for many of Twitch's viewers. The *Twitch Plays Pokemon* stream we discussed earlier falls into this category, as it was not planned and became enormously popular, but there are other, more recent examples like the charity stream of textitDonkey Kong 64 from the streamer H.Bomberguy, which because a viral sensation and raised over $\$340,000$, attracting celebrity guests up to and including Alexandria Ocasio-Cortez [4] . Currently, events like these spread by word-of-mouth and can be difficult to find out about if one is not monitoring Twitch continuously. To better increase the visibility of these events to Twitch viewers who are interested in both the pre-planned and emergent events, an event detection algorithm should be developed that would allow both Twitch and the viewing community to keep up with the events occurring on Twitch and to join ones already in progress if possible. This theoretical system could use the examples of the pre-scheduled events like E3 as training data to determine what the effect is on the network when an event like this occurs and then use those observations to detect emergent events like H.Bomberman's stream when they do occur.

9.3 Twitch and Social Media

The results of the analyses performed in this thesis paint a picture of the Twitch network specifically, covering its overall structure, some aspects of streamer dynamics, the relationships between game competing for attention, and the relationships between streamers and the games they play. The work performed here has wider applications to other forms of social media as well, as discussed in the motivation section of the introduction. In this section, we will review some of those motivating factors and discuss how dealing with those factors in our analysis can and should be

[4] `https://bit.ly/324biaH`

applied to analysis of other new and emerging social media platforms in the future.

First and foremost, the Twitch network is hyper-focused on live content. As we discussed in the dataset chapter, almost all of the users on the network are actively streaming or viewing content that is currently being broadcast. Though a streamer's gameplay sessions are recorded, it is unusual for viewers interested in Twitch content to watch these recorded streams. While Twitch is almost certainly an outlier in this case, the focus on live content on Twitch is indicative of a larger trend in social media and beyond towards exactly this kind of live content. Facebook has a live-streaming service called Facebook Live, Instagram has video feeds and IGTV broadcasts, and new forms of social media like TikTok and Snapchat are entirely live-focused. Snapchat's video content goes even further than others, with video or image messages that expire immediately after viewing with very limited ability to rewatch.

As social media continues to evolve and, most likely, continue its trend toward live content with little considerations for history, the lessons learned from the work that went into this thesis will continue to be important. Researchers must consider that focus on live content will continue to accelerate the pace of life on social media and factor that into future algorithmic advancements. The work of this thesis, especially the attempt to categorize streamers according to their influence potential, demonstrates this point acutely. Future research performed on emerging social platforms and even on Twitch, must account for this in the work and begin work under the assumption that capturing historical data is either impossible (in the case of Snapchat), without significant value (in the case of Twitch), or both. This will require a shift in the way many researchers think about their research, as it will be necessary to begin capturing data very early in the research process. In addition, it will change the way data is collected. Rather than collecting a specific dataset to match the set of

hypotheses researchers wish to test, data collection will have to be extremely general in nature in order to account for the different directions the research could take. It should be noted that general purpose data collection, even for research purposes, has significant privacy implications that are out of scope for this particular thesis but are no less important.

In addition to the importance of collecting and analyzing the live data, the work conducted in this thesis highlighted the importance of multimedia data. Twitch as a platform revolves almost entirely around video content. Similarly, Snapchat relies heavily on image and video content. Even Instagram, which we frequently refer to in our set of traditional social media, is built around image sharing. New and emerging social platforms must involve image or video content at their core, or they will fail to engage new users who now expect to be able to post and share this kind of content on the network. Similarly, it is important for researchers to be able to analyze that content. The work presented in this thesis is hampered by its lack of analysis of the actual image and video content presented on the Twitch network and analysis of the video content appears in nearly every one of our Next Steps sections. Analyzing this content is not easy and it is necessary for a deep understanding of the kinds of conversations that are occurring on the new platforms.

To this effect, researchers wishing to steep themselves in the worlds of new and emerging social media must be able to analyze image and video content. As ridiculous as it may seem, the use of image macros and short video clips has become ubiquitous among users of Twitter, Tumblr, Reddit, and even Instagram and often these are presented with little or no text superimposed on the image. Or, if text is superimposed, understanding of the underlying image and its meaning is necessary to understand the content. This does not mean that researchers need to understand and manually analyze every image or video clip, but it does mean that researchers need

to understand how multimedia well enough to analyze and figure out the meaning as necessary. It also means that any text accompanying an image macro or video clip cannot necessarily be taken at its face value. Often, the text accompanying an image or in any text comments to that image could be simply playing into the joke of the image and would be couched in the language of the original image.

The single most important lesson learned from the work discussed in this thesis is that modern users of social media, especially a younger generation of users, no longer have a single social media site with which they spend a majority of their time. Previous studies of social media often contain an underlying assumption that we discussed briefly in the Motivation section of the Introduction, namely that if a researcher collects a user's Facebook or Twitter profile they have a record of all or a vast majority of that user's online behavior intended for public consumption. Twitch, on the other hand, is highly specialized to games and gaming, and yet the streamers and viewers on the site inevitably have detailed YouTube, Twitter, and Instagram profiles as well. Many streamers cross-link these profiles and the savviest streamers understand that their audiences engage differently with the content on each network and tailor them appropriately. In order to get a full or even representative picture of that streamer's life, it would be necessary to collect and rectify data from a wide variety of sources.

The fracturing of the social media space is not going to go away either. New social media platforms will continue to emerge and those sites will grow more and more specialized. As these platforms emerge, collecting the entirety of a user's online presence will become more and more difficult and this is a challenge that researchers will have to deal with accordingly. Long gone are the days when a user's MySpace and/or Facebook profiles were their online presence. Now, most social media users have accounts on half a dozen or more sites, and are variously active on each one.

Researchers can no longer assume that their Twitter data set or their Facebook data set contains enough information and should seek to supplement it with additional data, particularly if that research is interested in the behavior of younger generations of users.

In all, while not directly stated in the text of the individual chapters, the work done in this thesis provides groundwork which should influence the content of research on any and all forms of social media, especially those on new forms of social media. In addition to its research implications, most notably that the operations of different social networks vary enough that even what seems like fundamental results should be verified, we discussed in this section the perceived trends of social media in general. These trends push social media platforms inevitably toward the types of multimedia we discussed previously, be it image sharing, video sharing, or something altogether novel that we do not currently anticipate. It is also clear that new social media platforms will need to differentiate themselves somehow and the dominant trends seems to indicate that a platform can do so by operating in a niche that is not currently fulfilled by existing platforms. Likely, new platforms will continue to crop up in this vein, and each one will present new challenges for researchers interested in that platform, and those challenges will vary.

REFERENCES

Ahmad, M. A., Z. Borbora, C. Shen, J. Srivastava and D. Williams, *Guild play in mMOGs: Rethinking common group dynamics models* (Springer, 2011).

Ahmad, M. A., Z. Borbora, J. Srivastava and N. Contractor, "Love all, trust a few: Link prediction for trust and psycho-social factors in mmos", in "Social Computing, Behavioral-Cultural Modeling and Prediction", pp. 123–130 (Springer, 2012).

Alstott, J., E. Bullmore and D. Plenz, "powerlaw: a python package for analysis of heavy-tailed distributions", PloS one **9**, 1, e85777 (2014).

Anderson, A., D. Huttenlocher, J. Kleinberg and J. Leskovec, "Steering user behavior with badges", in "Proceedings of the 22nd international conference on World Wide Web", pp. 95–106 (International World Wide Web Conferences Steering Committee, 2013).

Bartle, R., "Hearts, clubs, diamonds, spades: Players who suit muds", Journal of MUD research **1**, 1, 19 (1996).

Basaras, P., D. Katsaros and L. Tassiulas, "Detecting influential spreaders in complex, dynamic networks", Computer , 4, 24–29 (2013).

Bauckhage, C., K. Kersting, R. Sifa, C. Thurau, A. Drachen and A. Canossa, "How players lose interest in playing a game: An empirical study based on distributions of total playing times", in "2012 IEEE CIG", pp. 139–146 (IEEE, 2012).

Bell, J., S. Sheth and G. Kaiser, "A large-scale, longitudinal study of user profiles in world of warcraft", in "Proceedings of the 22nd international conference on World Wide Web companion", pp. 1175–1184 (2013).

Bergstrom, K., "Eve online newbie guides: Helpful information or gatekeeping mechanisms at work?", in "Selected Papers of the 14th Association of Internet Researchers Conference. Denver, CO", (2013).

Boulier, B. L. and H. O. Stekler, "Predicting the outcomes of national football league games", International Journal of Forecasting **19**, 2, 257–270 (2003).

Breiman, L., "Random forests", Machine learning **45**, 1, 5–32 (2001).

Breiman, L., J. Friedman, C. J. Stone and R. A. Olshen, *Classification and regression trees* (CRC press, 1984).

Brown, J. A., "Evolved weapons for rpg drop systems", in "Computational Intelligence in Games (CIG), 2013 IEEE Conference on", pp. 1–2 (IEEE, 2013).

Cachia, W., L. Aquilina, H. P. Martinez and G. N. Yannakakis, "Procedural generation of music-guided weapons", in "2014 IEEE Conference on Computational Intelligence and Games", pp. 1–2 (IEEE, 2014).

Carter, M., M. Gibbs and M. Harrop, "Metagames, paragames and orthogames: A new vocabulary", in "Proceedings of the international conference on the foundations of digital games", pp. 11–17 (ACM, 2012).

Carter, M. and M. R. Gibbs, "esports in eve online: Skullduggery, fair play and acceptability in an unbounded competition.", in "FDG", pp. 47–54 (2013).

Castronova, E., *Synthetic worlds: The business and culture of online games* (University of Chicago press, 2008).

Cheong, Y.-G., A. Jhala, B.-C. Bae and R. M. Young, "Automatically generating summary visualizations from game logs.", in "AIIDE", pp. 167–172 (2008).

Debeauvais, T., B. Nardi, D. J. Schiano, N. Ducheneaut and N. Yee, "If you build it they might stay: Retention mechanisms in world of warcraft", in "6th International Conference on Foundations of Digital Games", pp. 180–187 (ACM, 2011).

Deng, J., G. Tyson, F. Cuadrado and S. Uhlig, "Internet scale user-generated live video streaming: The twitch case", in "International Conference on Passive and Active Network Measurement", pp. 60–71 (Springer, 2017).

Ducheneaut, N. and R. J. Moore, "The social side of gaming: a study of interaction patterns in a massively multiplayer online game", in "ACM conference on Computer supported cooperative work", pp. 360–369 (ACM, 2004).

Ducheneaut, N., N. Yee, E. Nickell and R. J. Moore, "Alone together?: exploring the social dynamics of massively multiplayer online games", in "Proceedings of the SIGCHI Conference", pp. 407–416 (ACM, 2006).

Easley, D. and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world* (Cambridge University Press, 2010).

Feng, W.-c., D. Brandt and D. Saha, "A long-term study of a popular mmorpg", in "Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games", pp. 19–24 (ACM, 2007).

Goodfellow, C., "Russian overlords, vodka, and logoffski russian-and english-language discourse about anti-russian xenophobia in the eve online community", Games and Culture **10**, 4, 343–364 (2015).

Guala, L., S. Leucci and E. Natale, "Bejeweled, candy crush and other match-three games are (np-) hard", arXiv preprint arXiv:1403.5830 (2014).

Guo, R. and P. Shakarian, "A comparison of methods for cascade prediction", arXiv preprint arXiv:1606.05730 (2016).

Johnson, N. F., C. Xu, Z. Zhao, N. Ducheneaut, N. Yee, G. Tita and P. M. Hui, "Human group formation in online guilds and offline gangs driven by a common team dynamic", Physical Review E **79**, 6, 066117 (2009).

Jones, E., T. Oliphant and P. Peterson, "{SciPy}: open source scientific tools for {Python}", (2014).

Kawale, J., A. Pal and J. Srivastava, "Churn prediction in mmorpgs: A social influence based approach", in "Computational Science and Engineering, 2009. CSE'09. International Conference on", vol. 4, pp. 423–428 (IEEE, 2009).

Kifer, D., S. Ben-David and J. Gehrke, "Detecting change in data streams", in "Proceedings of the Thirtieth international conference on Very large data bases-Volume 30", pp. 180–191 (VLDB Endowment, 2004).

Kleinberg, S., "A logic for causal inference in time series with discrete and continuous variables", in "IJCAI Proceedings-International Joint Conference on Artificial Intelligence", vol. 22, p. 943 (2011).

Kumar, S., R. Zafarani and H. Liu, "Understanding user migration patterns in social media.", in "AAAI", (2011).

Liu, S., S. Wang, F. Zhu, J. Zhang and R. Krishnan, "Hydra: Large-scale social identity linkage via heterogeneous behavior modeling", in "Proceedings of the 2014 ACM SIGMOD", pp. 51–62 (ACM, 2014).

Manevitz, L. M. and M. Yousef, "One-class svms for document classification", Journal of Machine Learning Research **2**, Dec, 139–154 (2001).

Massey Jr, F. J., "The kolmogorov-smirnov test for goodness of fit", Journal of the American statistical Association **46**, 253, 68–78 (1951).

Medler, B. and B. Magerko, "Analytics of play: Using information visualization and gameplay practices for visualizing video game data", Parsons Journal for Information Mapping **3**, 1, 1–12 (2011).

Nakandala, S., G. L. Ciampaglia, N. M. Su and Y.-Y. Ahn, "Gendered conversation in a social game-streaming platform", arXiv preprint arXiv:1611.06459 (2016).

Ong, H. Y., S. Deolalikar and M. Peng, "Player behavior and optimal team composition for online multiplayer games", arXiv preprint arXiv:1503.02230 (2015).

Ontanón, S., G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill and M. Preuss, "A survey of real-time strategy game ai research and competition in starcraft", IEEE Transactions on Computational Intelligence and AI in games **5**, 4, 293–311 (2013).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python", Journal of Machine Learning Research **12**, Oct, 2825–2830 (2011).

Pittman, D. and C. GauthierDickey, "Characterizing virtual populations in massively multiplayer online role-playing games", in "International Conference on Multimedia Modeling", pp. 87–97 (Springer, 2010).

Safferling, C. and A. Lowen, "Economics in the kingdom of loathing: analysis of virtual market data", (2011).

Schubert, M., A. Drachen and T. Mahlmann, "Esports analytics through encounter detection", in "MIT Sloan Sports Analytics Conference", (MIT Sloan, 2016).

Seay, A. F., W. J. Jerome, K. S. Lee and R. E. Kraut, "Project massive: a study of online gaming communities", in "CHI'04 extended abstracts", pp. 1421–1424 (ACM, 2004).

Seok, S. and B. DaCosta, "Predicting video game behavior an investigation of the relationship between personality and mobile game play", Games and Culture p. 1555412014565640 (2015).

Shah, N., "Flock: Combating astroturfing on livestreaming platforms", in "Proceedings of the 26th International Conference on World Wide Web", pp. 1083–1091 (International World Wide Web Conferences Steering Committee, 2017).

Shim, K. J., K.-W. Hsu, S. Damania, C. DeLong and J. Srivastava, "An exploratory study of player and team performance in multiplayer first-person-shooter games", in "Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on", pp. 617–620 (IEEE, 2011a).

Shim, K. J., K.-W. Hsu and J. Srivastava, "Effects of mentoring on player performance in massively multiplayer online role-playing games (mmorpgs)", in "2011 International Conference on Advances in Social Networks Analysis and Mining", pp. 561–562 (IEEE, 2011b).

Shim, K. J., K.-W. Hsu and J. Srivastava, "An exploratory study of player performance, motivation, and enjoyment in massively multiplayer online role-playing games", in "Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on", pp. 135–140 (IEEE, 2011c).

Shim, K. J. and J. Srivastava, "Behavioral profiles of character types in everquest ii", in "Computational Intelligence and Games (CIG), 2010 IEEE Symposium on", pp. 186–194 (IEEE, 2010a).

Shim, K. J. and J. Srivastava, "Sequence alignment based analysis of player behavior in massively multiplayer online role-playing games (mmorpgs)", in "Data Mining Workshops (ICDMW), 2010 IEEE International Conference on", pp. 997–1004 (IEEE, 2010b).

Shipman, F. M. and C. C. Marshall, "Creating and sharing records of multiplayer online game play: practices and attitudes", in "Eighth International AAAI Conference on Weblogs and Social Media", (2014).

Sifa, R., A. Drachen and C. Bauckhage, "Large-scale cross-game player behavior analysis on steam", Borderlands (2015).

Spronck, P., I. Balemans and G. Van Lankveld, "Player profiling with fallout 3.", in "AIIDE", (2012).

Suh, B., L. Hong, P. Pirolli and E. H. Chi, "Want to be retweeted? large scale analytics on factors impacting retweet in twitter network", in "Social Computing (SocialCom), 2010 IEEE Second International Conference on", pp. 177–184 (IEEE, 2010).

Taylor, N., K. Bergstrom, J. Jenson and S. de Castell, "Alienated playbour relations of production in eve online", Games and Culture p. 1555412014565507 (2015).

Thurau, C. and C. Bauckhage, "Analyzing the evolution of social groups in world of warcraft®", in "Proceedings of the 2010 IEEE CIG", pp. 170–177 (IEEE, 2010).

van Kreveld, M., M. Löffler and P. Mutser, "Automated puzzle difficulty estimation", in "2015 IEEE Conference on Computational Intelligence and Games (CIG)", pp. 415–422 (IEEE, 2015).

Wang, Y. and S. D. Mainwaring, "Human-currency interaction: learning from virtual currency use in china", in "Proceedings of the SIGCHI Conference on Human Factors in Computing Systems", pp. 25–28 (ACM, 2008).

Weibull, W., "Wide applicability", Journal of applied mechanics **103**, 293–297 (1951).

Wenz, K., "Theorycrafting: Knowledge production and surveillance", Information, Communication & Society **16**, 2, 178–193 (2013).

Yannakakis, G. N. and J. Togelius, "A panorama of artificial and computational intelligence in games", IEEE Transactions on Computational Intelligence and AI in Games **7**, 4, 317–335 (2015).

Yee, N., "The labor of fun how video games blur the boundaries of work and play", Games and Culture **1**, 1, 68–71 (2006).

Zafarani, R. and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach", in "Proceedings of the 19th ACM SIGKDD", pp. 41–49 (ACM, 2013).

Zhang, J. and S. Y. Philip, "Multiple anonymized social networks alignment", in "2015 IEEE ICDM", pp. 599–608 (IEEE, 2015).

Zhang, L. and A. Y. Fung, "Working as playing? consumer labor, guild and the secondary industry of online gaming in china", New Media & Society **16**, 1, 38–54 (2014).