Linking Models for Collective Attention in Social Media

Swapnil Mishra

A thesis submitted for the degree of
Doctor of Philosophy of
The Australian National University

November 2019

Except where otherwise indicated, this thesis is my own original work.

Swapnil Mishra
3 November 2019

to my family and friends

Acknowledgments

I would like to express my sincere gratitude towards everyone who made this thesis possible.

Firstly, I would like to express my deepest gratitude to my primary supervisor Prof. Lexing Xie for her guidance, support, and encouragement throughout this thesis. I have learned a lot from her about what research is, and the patience she has shown towards me is genuinely exceptional. I am grateful to Lexing for not only advising me as a supervisor but also as someone who lent support during my hard times. I could not have imagined a better supervisor.

I would especially like to thank my co-supervisor Dr. Marian-Andrei Rizoiu for helping me to grow as a researcher. I truly admire his passion, drive, and ability to express complicated things in the most straightforward way possible. Andrei has continuously challenged me and has been a great source of motivation and advice. I sincerely appreciate all the efforts he has taken to shape me as a person.

I am thankful to ANU, NICTA and CSIRO Data61 for providing financial and technical support for my research.

I am grateful to all the members of the ANU Computational Media Lab for all the good times we had both during and outside research. I express my gratitude towards friends I made in Canberra and Pune. My colleagues and friends made my time during the thesis a memorable one.

I can not thank my wife Meenakshi Dubey, enough for her sacrifice, support, advice, and love throughout this journey. She is the reason I smile despite when chips are down. I would like to thank my in-laws, Ramesh Dubey, Saroj Dubey, Deepak Dubey, and Sandeep Dubey, for showing their love, support, and faith in me. I would like to thank my brother, Dr. Harsh Mishra, for his unconditional support and love.

Finally, and most importantly, I would like to thank my parents, Dr. Vijay Shankar Mishra and Lily Mishra. I could not have done this without their guidance, encouragement, selfless love and support.

Abstract

Social networks are ubiquitous in the modern world for propagating and acquiring information. Thus, understanding and predicting the popularity of online information is an important problem in social media analysis. Considerable progress has been made recently in data-driven predictions, and in linking popularity to various external factors. Most of the work on popularity prediction and understanding is either based on learning a variety of features from full network data or using generative processes to model the event time data. However, there exists no prior work that connects or compares models across different paradigms. Accordingly, this thesis focuses on developing and connecting models to predict and understand popularity across different paradigms and settings.

To this aim, we first bridge gaps between feature-driven and generative models with the help of a hybrid model and a new performance benchmark. We model each social diffusion with a marked Hawkes self-exciting point process, and we estimate from data the content virality, memory decay, and user influence. Next, we learn a predictive layer for popularity prediction using a collection of cascade histories. We show the Hawkes process with a predictive overlay outperforms recent feature-driven and generative approaches on both existing tweets data and our new dataset. We also show that a feature-driven method based on a basic set of user features and event time summary statistics performs competitively in both classification and regression tasks and that adding point process information to the feature set further improves predictions. A common benchmark dataset for popularity prediction helps us to utilize both feature-driven, and generative paradigms to better predict and understand online popularity. As the first proposed work that links models across feature-driven and generative models, our work has influenced subsequent works on online popularity since its publication in 2016.

Secondly, we identify that the existing methods for popularity modeling and prediction typically focus on a single source of external influence. However, for many types of online content such as YouTube videos or news articles, attention is driven by multiple heterogeneous sources simultaneously - e.g., microblogs and traditional media coverage. Correspondingly, we propose RNN-MAS, a recurrent neural network for modeling asynchronous streams. It is a sequence generator that connects multiple streams of different granularity via joint inference. We show

that RNN-MAS not only outperforms the current state-of-the-art YouTube popularity prediction system, but it also captures complex dynamics, such as the seasonal trends of unseen influence. Further, to increase the explainability and interpretability of our model, we propose two new metrics: the *promotion score* quantifies the gain in popularity from one unit of promotion for a YouTube video; the *loudness level* captures the effects of a particular user tweeting about the video. We use the loudness level to compare the effects of a video being promoted by a single highly-followed user (in the top 1% most followed users) against the same video being promoted by a group of mid-followed users. We show that results depend on the type of content being promoted: superusers are more successful in promoting Howto and Gaming videos, whereas the cohort of regular users is more influential for Activism videos.

Additionally, we apply the RNN-MAS model to the problem of predicting the popularity of an item before being published. We train a single model for a group of videos to learn possible evolution dynamics of a given video from the historical data of the videos in the same group. A novel simulation strategy based on the proposed metrics enables us to simulate a representative promotion for the video, and predict the achieved popularity before it is published. Experiments on our large scale YouTube dataset show that our proposed method outperforms non-trivial baselines.

By and large, this thesis proposes models for popularity modeling and prediction that are the first of their kind, and it links models across various paradigms and data availability. This work provides accurate and explainable popularity predictions, as well as computational tools for content producers and marketers to allocate resources for promotion campaigns. In addition to these contributions, this work may contribute to a more comprehensive understanding of popularity prediction and understanding models across different classes or types.

Publications, Software and Data

The majority of the thesis has been published in conference proceedings and a book chapter. Software and data developed as a part of the thesis and these publications is provided to form basis of future work.

Publications

S Mishra. "Bridging Models for Popularity Prediction on Social Media." *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. ACM, 2019.* (Doctoral Symposium)
`https://dl.acm.org/citation.cfm?id=3291598`

S Mishra, MA Rizoiu, L Xie. "Modeling Popularity in Asynchronous Social Media Streams with Recurrent Networks." *Twelfth International AAAI Conference on Web and Social Media. 2018.* (Full Paper, Acceptance Rate: 16%)
`https://arxiv.org/abs/1804.02101`

MA Rizoiu, **S Mishra**, Q Kong, M Carman, L Xie. "SIR-Hawkes: on the Relationship Between Epidemic Models and Hawkes Point Processes." *Proceedings of the 2018 World Wide Web Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2018.* (Full Paper, Acceptance Rate: 15%)
`https://arxiv.org/abs/1711.01679`

MA Rizoiu, Y Lee, **S Mishra**, L Xie. "Hawkes Processes for Events in Social Media." *Frontiers of Multimedia Research. Association for Computing Machinery and Morgan & Claypool, 2017.* (Invited Book Chapter)
`https://arxiv.org/abs/1708.06401`

S Mishra, MA Rizoiu, L Xie. "Feature driven and point process approaches for popularity prediction." *Proceedings of the 25th ACM International on Conference on Information and Knowledge*

Management. ACM, 2016. (Full Paper, Acceptance Rate: 17%)
`https://arxiv.org/pdf/1608.04862.pdf`

Software

Featuredriven-hawkes: source code
`https://github.com/s-mishra/featuredriven-hawkes`

RNN-MAS: source code
`https://github.com/computationalmedia/rnn-mas`

HawkesN: source code
`https://github.com/computationalmedia/sir-hawkes`

Data

NEWS- 49.7 million tweets on top 10 english news outlets
`https://tinyurl.com/y4xkl43t`
(Chapter 3 (Section 3.1))

ACTIVE'14: 13,738 tweeted youtube videos with tweets
`https://tinyurl.com/yy8g2zqa`
(Chapter 3 (Section 3.2.2))

ACTIVE-3YR: 389,612 tweeted youtube videos with tweets
`https://tinyurl.com/y2unrca9`
(Chapter 3 (Section 3.3.1))

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the modern world, web content, and especially online social networks, has changed the ways people connect and get information. The amount of information produced is enormous and increasing exponentially day by day, [Turner et al., 2014] reports data we create would grow to be 44 zettabytes in 2020. The ubiquity of online social networks has made the world more connected than ever before [Bhagat et al., 2016]. Shorter distances and ease of access have made social media an important venue for information sharing.

Individuals interact with items on social media by sharing, posting, commenting, liking or voting them. Thus, when individuals interact with items, they create a piece of information that can be accessed by his/her friends/followers on the network; with these individuals also being able to pass it to their friends in turn. Hence these interactions lead to information spreading from person to person, evoking a chain of interactions usually referred to as *information cascades* or just *cascades*.

Naturally, an important step is to quantify the spread of these cascades. One of the ways to quantify the spread of a cascade is to measure its popularity. Popularity can be measured as the amount of attention gathered by an item in the network. Recent work has shown that the amount of attention gathered by item is skewed [Clauset et al., 2009; Goel et al., 2012], where most of the attention is distributed among very few items, and most of the items gather only a negligible amount of attention. Hence, in the presence of information overload and of the asymmetric concentration of attention, it becomes even more essential to understand and predict the popularity of cascades.

While modeling popularity is essential, it is also a very challenging task. As noted by researchers, the popularity of an item is inherently very unpredictable and pointing out the factors responsible for evolution are difficult [Salganik et al., 2006; Martin et al., 2016]. Another difficulty is, in general, quantifying the factors, that makes content popular, like content quality or user preference. Other factors like interactions of the physical, online and social world are in itself quite intricate and are the subject of an active area of research among various disciplines [Hirshleifer and Hong Teoh, 2003; Christakis and Fowler, 2007; Keeling and Eames, 2005]. Last but not the least, a significant difficulty in modeling popularity in the current digitized world is developing appropriate models and inference algorithms that can scale to the rising amount of data. Such tasks of modeling and predicting popularity motivates the work done in this thesis.

Insights into popularity dynamics can help content producers to prioritize production and schedule promotions better, and help content providers to allocate resources for hosting and advertising. Furthermore, this would provide with more insights into understanding collective behavior by compiling seemingly incoherent individual responses to a single piece of information.

1.1 Motivation: Gaps in Popularity Prediction

In recent years, a series of work has been proposed for modeling popularity in social media. In general, the existing work can be divided into various categories based on: i) the type of approach (paradigms) used for modeling, and ii) the granularity of data available. Nonetheless, there exists no work compares and links models across various categories, therefore limiting their applicability.

There are two main approaches for modeling popularity: *feature-driven* methods and *generative* methods. *Feature-driven* methods [Berger and Milkman, 2012; Szabo and Huberman, 2010; Kaltenbrunner et al., 2007; Cheng et al., 2014] rely on extracting an exhaustive set of features describing the evolution of a cascade and then using these features as part of any classical machine learning prediction framework. For example, [Bakshy et al., 2011; Martin et al., 2016] found past user success – the ability of a user to start large cascades - as the most important feature to predict tweet popularity. Early activity has been shown predictive for total popularity [Pinto et al., 2013; Szabo and Huberman, 2010]. When predicting whether a Facebook cascade will double in size, [Cheng et al., 2014] found that temporal features – like the resharing rate or the acquired volume of views – are the most informative features. Whereas, *generative* models [Crane and Sornette, 2008; Shen et al., 2014; Zhao et al., 2015; Rizoiu et al., 2017] take a bottom-up approach for modeling a cascade. They make assumptions about the diffusion process and define a model for individual actions in the cascade. In their seminal work, Crane and Sornette [Crane and Sornette, 2008] showed how a Hawkes point process can account for popularity bursts and decays. Afterward, more sophisticated models have been proposed to model and simulate popularity in microblogs [Shen et al., 2014; Zhao et al., 2015; Yu et al., 2017] and videos [Ding et al., 2015] by incorporating various specific assumptions related to the network under study. These two classes of approaches are seemingly seen as disjointed and incomparable in the research community. Correspondingly, this thesis seeks to answer the research question: **Can we unify feature-driven and generative models?**

In real-world information cascades, based on the availability and type of data, popularity prediction methods can be classified into two main classes: models for *event-series* data and *volume-series* data. Models for event-series data [Du et al., 2016; Shen et al., 2014; Zhao et al., 2015] have access to each event in a cascade and events occur at random time intervals, for example, data from Twitter cascades. However, in second case models for volume-series data [Szabo and Huberman, 2010; Pinto et al., 2013; Rizoiu et al., 2017] have access to the volume of events at regular intervals, instead of individual events in a cascade. Each of these models specializes in a distinct data type, but it is common to observe data of different types for the same online item, for example in YouTube videos or news articles, attention is driven by multiple heterogeneous sources simultaneously – e.g., microblogs or traditional media coverage. It is desirable to develop a model that accounts for multiple heterogeneous series. Hence, in the thesis we seek to answer the research question: **can we model multiple heterogeneous asynchronous streams of data driving content popularity?**

Furthermore, state-of-the-art popularity models provide black-box predictions [Zhao et al., 2015; Martin et al., 2016; Mishra et al., 2016]. In practice, one often demands simulations on various *what-if* scenarios, such as to quantify the effect of a unit amount of promotions, to capture seasonality or the response to outliers, to name a few. Lastly, the influence users have on popularity has been subject to constant debate in this research area. The view that one or a few influential champions can make or break a cascade [Budak et al., 2011] contrasts with the view that popularity mainly results from a large number of moderately influential users [Bakshy et al., 2011]. It is desirable to have one model on which the future effect of different users can be comparably studied. As a result, in the thesis we ask the question **can we design metrics for explainability and interpretability of state-of-the-art prediction models.**

In summary, there is a dearth of work in the field of popularity modeling that helps us to *link (bridge)* models across paradigms, settings and data availability. Hence, the focus of the thesis is in exploring these missing links and proposing models that are more unified, general and interpretable than the existing work.

1.2 Key Contributions of Thesis

The main contribution of this thesis lies in developing models for understanding and predicting popularity that links models across various paradigms, settings and data availability. The major contributions of this thesis are:

1. **Bridging the gap between feature-driven and generative models.** We propose a two-stage model for predicting the popularity of retweets cascades, i.e., the final size of the cascade. In the first stage, we capture the evolution dynamics of a retweet cascade with the help of a generative model based on the self-exciting Hawkes processes. Then, we utilize a feature-driven predictive layer on top of the Hawkes model that learns the final prediction from a group of cascades. Furthermore, we also identify a set of basic user and temporal features that can be used by a feature-driven model to achieve competitive results when compared to previous state-of-the-art (SOTA) prediction algorithms. This two-stage model helps us to retain the predictive efficacy of feature-driven models as well as have the explanatory power of generative models for describing the diffusion dynamics.

2. **Novel point-process models.** Most of the formulations of the Hawkes process utilized in social networks for modeling cascades have an implicit assumption about the availability of infinite population for a cascade to grow. However, this is far from reality as all social networks have a finite underlying population. Accordingly, we propose an extension, HawkesN, for the Hawkes process that accounts for underlying population size and modulates the intensity of the Hawkes process according to the available size left for the cascade to grow. We systematically show that HawkesN generalizes better than the state-of-the-art Hawkes model. In our next extension, we tackle the problem of specifying the shape for the kernel in the Hawkes Process beforehand. To mitigate this problem, we propose a Hawkes formulation based on recurrent neural networks (RNN), named RPP, that directly learn the shape of the kernel from the data. RPP learns the intensity of the Hawkes process by embedding the history inside the hidden state of the RNN. We show that RPP can learn true intensity with our evaluations on simulated data. Finally, we show that RPP has better performance on real-world datasets than parametric alternatives. RPP helps us to link the point process models with recurrent neural networks.

3. **Modeling Multiple Asynchronous Streams.** We study the problem of predicting and understanding the popularity of an item under the influence of multiple promotions that

are asynchronous with each other, and propose a joint model, RNN-MAS, for multiple streams based on the recurrent neural networks. Explicitly, we predict daily views of a YouTube video by joint modeling the daily shares of the video on YouTube with a series of tweets on Twitter mentioning the video. The majority of the previous work on multiple asynchronous streams modeled data by aggregating streams to make them homogeneous. However, in aggregating all of these interactions, we lose valuable fine-grained information. Hence, our joint modeling technique utilizes the full information available to predict better and capture the evolutionary dynamics. We show from our experiments, that RNN-MAS can better capture complex dynamics like seasonality.

4. **Metrics for Explainability.** Undoubtedly formulating point process with neural networks increases our prediction prowesses, but they usually come at the cost of a reduction in interpretability and explainability of point process models. To tackle this disadvantage, we derive two new metrics for RNN-MAS to quantify average response to the unit promotion, and the relative influence among users of different fame. For calculating our metrics, we propose a general simulation strategy that not only helps us to estimate the influence of users across different networks, but it also helps us to generate a representative promotion for a video from the learned model. We find that users of the same fame on Twitter have disproportionate influence in promoting content on YouTube. Finally, we apply RNN-MAS to predict the popularity of a YouTube video before being published by utilizing the simulation strategy in conjunction with the idea of learning a single model for a group of videos.

5. **Large-scale Datasets.** We curated three large-scale datasets, i) News, ii) Active'14, and Active-3Yr, for facilitating our work in linking models across various paradigms and data availability. We used News for creating the first common benchmark, to the best of our knowledge, for predicting the popularity of retweets with feature-driven and generative models. We show that features extracted from News for common benchmarking are neither exclusive to Twitter nor expensive to curate. Hence, increasing their broader applicability for popularity prediction and understanding. We further curate two YouTube datasets, Active'14 and Active-3Yr, that are first of their kind, by combining information about an online item (video) from two different networks (YouTube and Twitter). They enable us to link models for predicting the popularity of an online item under the influence of multiple external sources. We also show, how a dataset like Active-3Yr which

spans over three years can be used to study the longitudinal effect of data on popularity prediction. Finally, we utilize Active-3Yr for predicting the popularity of an online item before getting published, which is not possible on small datasets like Active'14.

1.3 Thesis Organization

The rest of the thesis is outlined as follows. In Chapter 2, we review various existing techniques for modeling popularity and provide a background of the Hawkes Process and Recurrent Neural Networks (RNN). In particular, we detail a taxonomy for existing popularity prediction techniques. We then review the essential concepts related to the Hawkes Process like self-excitation, conditional intensity, and branching structure. We end the chapter with a brief introduction of RNN.

In Chapter 3 we present three new datasets for popularity prediction. We first present News a domain-specific Twitter dataset of tweets related to news articles from the period April 2015 to July 2015. It is the first dataset that can be utilized for benchmarking both feature-driven and generative models for popularity prediction. We also identify a set of features that can be built on News to achieve state-of-the-art popularity prediction results. Next, we present two YouTube videos datasets, Active'14 and Active-3Yr. These datasets are unique for their availability of cross-platform information available, daily aggregated views and shares from YouTube and an associated series of individual tweets about the video from Twitter. Both datasets help us to study the evolution of popularity across different networks, YouTube and Twitter.

In Chapter 4 we bridge the gap between feature-driven and generative models for popularity prediction. In particular, for retweet cascades in the News dataset we predict the size of a retweet cascade after observing it for a given period. In this work, we first introduce a new generative model, a two-layered approach, built on the intuitive *self-exciting Hawkes process* [Hawkes, 1971]. Three key factors in information diffusion are built into the proposed model: the social influence of users, the length of "social memory" and the inherent tweet quality. We use a predictive layer on top of the generative model to make final predictions. Our second proposed approach, the feature-based method, uses the features we identified in Chapter 3 for the News dataset. This results in a competitive feature-based predictor, which consistently outperforms the current state-of-the-art popularity prediction model [Zhao et al., 2015]. We show that the same set of features can be employed in both regression and classification tasks. We further propose a hybrid model, which uses the data from the identified features and parameters from the fitted generative model, and show additional performance improvement.

We further generalize our self-exciting model in Chapter 5. We propose two new models in this chapter. The first generalized model is named *HawkesN*, and it explicitly accounts for the limited population size in the network (which Hawkes cannot capture). We present experimental evaluation on three real-world datasets to show the superiority of HawkesN over Hawkes in modeling event sequences that are longer. To the best our knowledge, HawkesN is the first self-exciting model used for modeling social cascades that can account for the limited population in the network. Next, in the chapter we propose *Recurrent Point Process (RPP)* another generalization of Hawkes model developed in Chapter 4 based on RNN [Elman, 1990; Graves, 2013; Sutskever et al., 2014]. It learns a general representation of the dependency over past events to predict future events with the help of a RNN. We show that the proposed model is effective and yields results better than its parametric alternative both on simulated and real-world datasets.

Chapter 6 aims to explain and predict the popularity of an online item under the influence of multiple external sources, in different temporal resolutions – such as both promotion events (e.g., tweets) and volumes (e.g., number of shares per day). In particular, we propose RNN-MAS (Recurrent Neural Networks for Multiple Asynchronous Streams), a flexible class of models learnable from social cascades that can describe heterogeneous information streams, explain predictions, and compare user effects for both individuals and groups. We illustrate the effectiveness of the proposed model for predicting the popularity of YouTube videos under the influence of both tweeting events and sharing volumes. We also propose several new ways to interpret and simulate popularity and implement them for RNN-MAS. The first is a *unit promotion response* metric, that measures the gain in popularity per unit of promotion. Measured at different times and promotion scales, it can describe the time-varying and nonlinear effect of online promotions. The second measure, *unseen response*, captures the effect of unobserved external influence. Since neural networks are a flexible function approximator, we show that this measure can successfully capture seasonal effects. To understand the influence of users, we compute a new metric, *loudness level*. It is used to quantify the popularity gain from highly influential users and moderately influential groups of users for each video. We observe the disproportionate influence of highly influential users depending upon the content that is being promoted by herself/himself.

In Chapter 7 we extend RNN-MAS model to account for multiple videos. In this work, we learn a single model for a group of videos; groups can be based on various attributes like channel or category of a video. Creating models based on groups helps us to tackle the problem of cold-start predictions.

Finally, in Chapter 8, we summarize our work presented in this thesis, and present a number of interesting future directions.

Chapter 2

Related Work

Our work in the thesis relates to several active areas of research for understanding and predicting popularity. First, in Section 2.1, we review studies related to the measurement of popularity in online social networks. Section 2.2 reviews various popularity modeling and prediction techniques used in literature. In Section 2.3 we present work on influence estimation. Lastly, in Section 2.4, we review the basics of recurrent neural networks (RNN) and their application in popularity prediction.

2.1 Measurement Studies on Popularity in Online Social Networks

In the early days of online media, most of the work for understanding popularity centered around measurement studies. YouTube is one of the first and most widely measured online media for understanding popularity [Cha et al., 2007; Gill et al., 2007]. In their pioneering study, [Cha et al., 2007] studied the distribution of views for videos on YouTube. They found that the distribution of views follows a power-law with an exponential cutoff, and the exact shape of the distribution varies as per different categories. Authors also found that very few old videos show a sudden jump in their popularity. Another study by [Gill et al., 2007] looked at the videos watched inside the campus of the University of Calgary, and correlated it with the most popular videos on YouTube, at the time of observation. They found that the viewing pattern for videos varies significantly by the time-of-day and day-of-week. In a subsequent work, [Yu, 2015] collected a large collection of tweeted videos that has gained a minimum amount of attention. Similar to the earlier work, they showed the long-tail distribution of views and found that most of the videos receive their popularity early. Furthermore, they find that for a certain group of videos, the popularity on YouTube is related to the tweeting activity about video on Twitter.

Similar to YouTube, other OSNs like Facebook, Twitter, and Flickr have also been studied in popularity measurement studies [Kwak et al., 2010; Cha et al., 2008; Nazir et al., 2008]. [Cha et al., 2008, 2009] studied Flickr to measure and understand the evolution of the popularity of photos. They found that the social influence is an essential factor responsible for the popularity of photos on Flickr. Moreover, in their study most of the photos that make way to a user are not more than 2-hop distance away; hence, cascades in Flickr are highly localized. [Nazir et al., 2008] studied the Facebook platform for analyzing the popularity of applications. They found that on Facebook, once an item becomes popular, it stays popular, confirming the phenomenon of 'rich-gets-richer' on these networks. In a seminal study of Twitter, [Kwak et al., 2010] studied the popularity of tweets. They found out that most of the content popular on Twitter relates to 'News.' While studying retweeting behavior, they found three interesting observations. First, on average, any tweet that has been retweeted once will reach an audience of 1,000 users irrespective of the number of followers of the original user. Secondly, they observed that once a tweet is retweeted, its second retweet comes almost instantly. Lastly, they observed that most of the interactions in Twitter are not reciprocal; users treat Twitter as an information-sharing network rather than a social network. Their observations show that information diffuses fast on Twitter,, and to a certain point, the original tweet is not an important factor defining its popularity. In their study of Twitter, [Wu et al., 2011] found that although the attention of users is more fragmented on Twitter than traditional media, still most of it is concentrated within a small elite population, celebrities, and media producers. Furthermore, they observed that within the elite users, the network is very homophilous, with users following other users of the same kind. Lastly, they found that the lifespan of an item on Twitter is dependent upon the type of content, for example, news content dies very fast whereas content by bloggers stays longer.

In summary, we have reviewed previous measurement studies on online popularity on various social networks. Our work in Chapter 3 uses the popularity scale over time inspired by these large-scale measurements to analyze our YouTube datasets, Active'14, and Active-3Yr. We then use this scale to measure the performance of our popularity prediction models in Chapter 6 and 7. Furthermore, a common observation among these studies about content deciding the shape and lifespan of popularity motivates our analysis of content specific trends in YouTube datasets in Chapter 3, and we use them for learning a single model for a group of videos in Chapter 7.

2.2 Popularity Modeling and Prediction

Measurement studies improve our understanding of online popularity and lay the foundation for future work. A natural next step after gaining insights is to model and predict the popularity of content. In popularity modeling literature, techniques can be divided into two main classes: i) feature-driven methods, and ii) generative methods. Feature-driven methods rely on extracting an exhaustive set of features that can be used with classical machine learning algorithms to predict popularity. Whereas, generative methods model the evolution of popularity from first principles, generally by utilizing the timing information from cascades.

The rest of our discussion in the section is structured as follows: in Section 2.2.1 we present feature-driven methods for popularity prediction. Section 2.2.2 first introduces some general background of point processes and then discusses the generative models used in online popularity literature.

2.2.1 Feature-driven Methods

Feature-driven approaches treat popularity as a non-decomposable process and take a bottom-up approach. The most important task is identifying a set of informative features that can best represent the content and achieve high prediction accuracies. The intuition behind the aforementioned class of models is that learning algorithms can somehow identify and capture latent dependency between popularity and an extensive set of features.

Feature-driven methods formulate the problem of predicting popularity in two ways: regression and classification. Next, we give details about the two settings along with examples from previous work.

Regression vs. Classification

The regression problem is to predict the exact popularity of an item up to some time t, where $t = \infty$ is equivalent to predicting the final popularity of the online item. For example, predicting the number of views for a YouTube video [Szabo and Huberman, 2010; Pinto et al., 2013], predicting the final size of a URL cascade on twitter Bakshy et al. [2011]; Martin et al. [2016].

However, at times, predicting the exact value is needless, and we only need to segregate popular items for unpopular ones. In this setup, instead of predicting the exact size, we predict whether the popularity of a particular item will cross a predefined threshold value or not. For instance, predicting whether a cascade will double its size or not [Cheng et al., 2014], whether an item will have 10 million views [Shamma et al., 2011], or be among the top 5% of the most popular items [Yu et al., 2014]. Classification setting is a relatively easier setup [Bandari et al., 2012].

Features

Most of the work in feature-driven methods are scattered around finding and constructing as many informative features as possible with human expertise and domain knowledge. The type of features used can be divided into four main categories: content, user, temporal and structural. We detail each of the four categories below.

Content features. The most basic feature responsible for the propagation of diffusion is the content itself. Content features are readily available in most of the scenarios as they do not depend on the network under study.

On Twitter, tweet content is used to derive features like the number of URLs, mentions or hashtags in the tweet [Tsur and Rappoport, 2012; Suh et al., 2010]. [Wu et al., 2018b] uses freebase topics describing a YouTube along with the category of the video as one of their features. Recent studies have generally identified content at best to be a very weak predictor when compared to other features like temporal, user or structural [Cheng et al., 2014; Martin et al., 2016].

User features. They relate to all the users who are part of the cascade. In Twitter, the most straight forward user feature is their number of friends or followers. [Petrovic et al., 2011] found out that the features of the author who started the tweet are more important than the features of the tweet itself (content). [Bakshy et al., 2011; Martin et al., 2016] found the past success of a user to be an informative feature. [Cheng et al., 2014] constructed a variety of user features on facebook like whether the user is a page or person, age, gender, time since on FB. Not all user features are available on all platforms.

Temporal features. Temporal features hope to capture the unfolding of a cascade. [Cheng et al., 2014] reported them to be the best performing features among everyone. Surprisingly [Szabo and Huberman, 2010; Pinto et al., 2013] found the early popularity, most straightforward of all temporal features, to be a robust and powerful predictor for predicting the final volume of views for a YouTube video. One point to note is that we can create temporal features without having access to the underlying network.

Structural features. They relate to the shape and status of the underlying network on which diffusion unfolds. They are mostly extracted by building a user's friendship graph and then extracting graph level attributes from it. [Cheng et al., 2014; Romero et al., 2013] reported structural features to be better than the user and content features. However, they also observed that structural features are not as useful as temporal features. Structural features are costly to create both time and space-wise as we need to extract extra information generally not available through public API. Hence, our work in the thesis focuses on predicting the popularity of cascades without considering the underlying network structure.

In summary, there has been much work in the space of predicting popularity with feature-driven methods. However, there still seems to be a disconnect within the community in identifying features that make predictions more accurate across different settings, i.e., regression and classification. Another unclear aspect is the set of features that are transferable between different online social networks. Moreover, many of these features are constructed on proprietary datasets. Hence, there is little understanding about the wider applicability of features. In Chapter 3, we curate the News dataset from Twitter public API and construct a set of features that can be built on any free public online social network. Our work in Chapter 4 evaluates both regression and classification tasks over News dataset to understand the performance of features across different settings.

2.2.2 Generative Models for Popularity and Point Processes

Many different kinds of generative models are used for modeling online social networks. For example, linear-threshold model [Bass, 1969; Granovetter, 1978], independent cascade model [Goldenberg et al., 2001], point process model [Crane and Sornette, 2008] and epidemic models [Leskovec et al., 2007]. However, for modeling and predicting online popularity most of the work is centered around point process models. We next review some basics of the point processes, before discussing the application of the generative models in modeling and predicting popularity.

Definition of Point Process

A point process is a random process whose realization is a collection of event times T_i lying on a non-negative real line, where T_i can be seen as the arrival time for the i^{th} event. Another equivalent representation of the point process is specifying them as a counting process, $N(t)$, which counts the number of events till time t, i.e.

$$N_t := \sum_{i \geq 1} \mathbb{1}_{\{t \geq T_i\}} \tag{2.1}$$

where $N_0 = 0$. The counting process takes a jump of 1 unit at each event time T_i.

Intensity Function

Event times in a point process are random variables. They can be completely characterized by the conditional intensity function, defined as

$$\lambda(t|\mathcal{H}_t) = \lim_{dt \to 0} \frac{\mathbb{P}\{N_{t+dt} - N_t = 1|\mathcal{H}_t\}}{dt} \tag{2.2}$$

where $\mathcal{H}_t$, is the history of the process upto time t, i.e., $\mathcal{H}_t = \{T_1, T_2, T_3, \ldots, T_{N_t}\}$.

For convenience, in rest of the thesis, we use the shorthand notation $\lambda(t) = \lambda(t|\mathcal{H}_t)$, with an implicit assumption of history before time t. From Equation (2.2), we can see that conditional probability of observing an event between time $[t, t + dt\ )$ under the condition that no other event has happened between T_{N_t} and t is $\lambda(t)dt$. The conditional intensity function can also be seen as the instantaneous rate of events per unit time.

Poisson Process

A (homogeneous) Poisson process is the simplest type of the point process where its conditional intensity function is given by

$$\lambda(t) = \mu \tag{2.3}$$

For Poisson process, the intensity at point of time t is constant and independent of its history, $\mathcal{H}_t$. Also, inter-arrival time, $\tau_i = T_i - T_{i-1}$, for any i is exponentially distributed with mean $\frac{1}{\mu}$.

Non-homogeneous Poisson Process

A generalized form of Poisson process is the non-homogeneous Poisson process, where the event intensity $\lambda(t)$ is not constant but a function of time t

$$\lambda(t) = \mu(t) \tag{2.4}$$

Non-homogeneous Poisson process provides greater flexibility than a Poisson process for modeling scenarios where we need to vary the event intensity with time - for example, capturing the arrival of cars during peak and non-peak hours in a parking lot. Note, similar to a Poisson process, in its current definition, a non-homogeneous Poisson process is also independent of the history, $\mathcal{H}_t$.

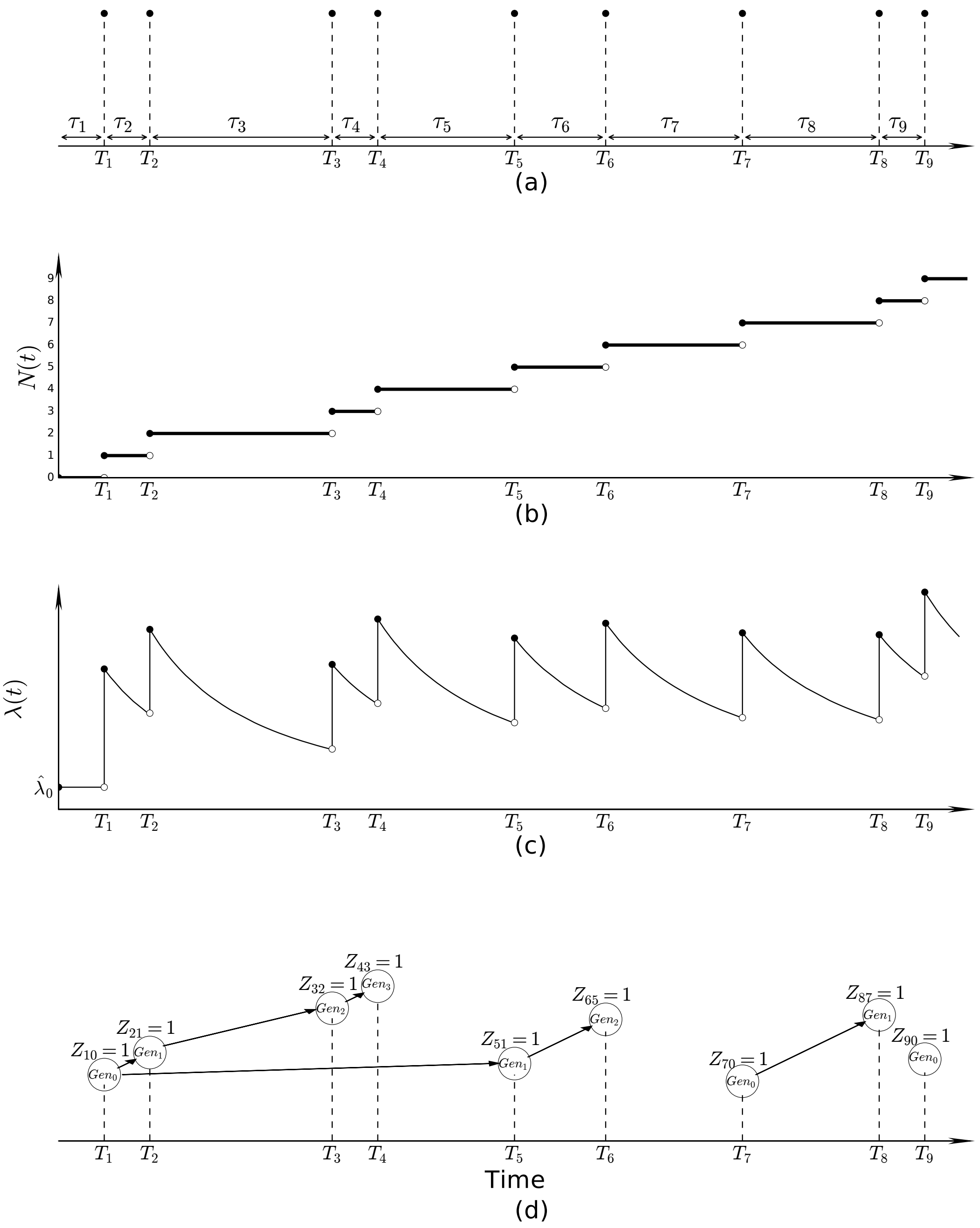


Figure 2.1: Hawkes process with an exponential decay kernel. (a) The first nine event times are shown. T_i represent event times, while τ_i represent inter-arrival times. (b) Counting process over time, N_t increases by one unit at each event time T_i. (c) Intensity function over time. Note how each event provokes a jump, followed by an exponential decay. Later decays unfold on top of the tail of earlier decays, resulting in apparently different decay rates. (d) The latent or unobserved branching structure of the Hawkes process. Every circle represents one event having occurred at T_i, the arrows represent the root-offspring relation. $\mathcal{G}en_i$ specifies the generation of the event, with $i = 0$ for immigrants or $i > 0$ for the offspring. Z_{ij} are random variables, such that $Z_{i0} = 1$ if event i is an immigrant, and $Z_{ij} = 1$ if event i is an offspring of event j.

Self Exciting Point Processes

A self-exciting point process is a point process where each event increases the probability of future events [Delay and Vere-Jones, 2003], more specifically, with each event the conditional intensity of the process increases. It can be seen as an extension of the non-homogeneous Poisson process where intensity not only varies with time t but also depends upon the history $\mathcal{H}_t$.

In particular, a well known and highly used self-exciting process is known as the Hawkes process [Hawkes, 1971]. For a Hawkes process, the *conditional intensity*, $\lambda(t)$, at a time point t is affected by all the events in the history through a triggering kernel, $\phi\left(\tau\right)$, as follows:

$$\lambda^*(t) = \mu(t) + \sum_{t_i<t} \phi(t - T_i) \ , \tag{2.5}$$

where $\mu(t)$ is the arrival rate of new events due to external factors. The self-excitation in the Hawkes process is due to the second term in the Equation (2.5), where the intensity at time point t has a contribution from a previous event at time T_i through the triggering kernel $\phi(t - T_i)$. Therefore, with each new event the instantaneous intensity rises suddenly, and slowly decreases with time assuming the triggering kernel $\phi(t)$ is monotonically decreasing.

Throughout literature, three families of functions have been mainly used to model triggering kernels with Hawkes point-processes [Rodriguez et al., 2011]: power-law functions $\phi^p(\tau) = (\tau + c)^{-(1+\theta)}$, used in geophysics [Helmstetter and Sornette, 2002] and social networks [Crane and Sornette, 2008] ; exponential functions $\phi^e(\tau) = e^{-\theta\tau}$, used in geophysics [Hawkes, 1971; Hawkes and Oakes, 1974] and financial data [Filimonov and Sornette, 2015]; Rayleigh functions $\phi^r(\tau) = \tau e^{-\frac{1}{2}\theta\tau^2}$, used in epidemiology [Wallinga and Teunis, 2004].

Hawkes process can also be seen as a Poisson cluster process, as shown in [Hawkes and Oakes, 1974]. [Hawkes and Oakes, 1974] define events in the Hawkes process to be of two types: offsprings and immigrants. Offsprings are the events that arrive in the system as a result of influence from the existing events whereas, immigrants are the events that arrive independently into the system. Each offspring can be attached to an existing event, and hence forming a cluster. This cluster representation is called the branching structure for a Hawkes process. Each immigrant in the system creates its own cluster and arrives into the system with intensity $\mu(t)$.

We illustrate concepts related to the Hawkes process in Figure 2.1. Figure 2.1(a) shows a realisation of first 9 events of a Hawkes process with exponential kernel and the Figure 2.1(b) shows the corresponding counting function $N(t)$ associated with those 9 events and Figure 2.1(c) plots the instantaneous event rate (conditional intensity) of the process. Figure 2.1(d) shows the underlying branching structure associated with the Hawkes process. In the figure, each event is denoted by a circle at event time T_i and arrows between them denote the parent-offspring relationship between events. In order to represent a cluster of events, we denote each event with a random variable Z_{ij}, where $Z_{i0} = 1$ if event i is an immigrant, and $Z_{ij} = 1$ if event i is an offspring of event j. Also, each event is associated with a generation, i.e. $\mathcal{G}en_k$ denotes the k-th generation. We introduce the random variables Z_{ij}, where $Z_{i0} = 1$ if event i is an immigrant, and $Z_{ij} = 1$ if event i is an offspring of event j. The text in each circle denotes the generation to which the event belongs to, i.e. $\mathcal{G}en_k$ denotes the k-th generation. Immigrants are labeled as $\mathcal{G}en_0$, while generations $\mathcal{G}en_k$, $k > 0$ denote their offsprings.

For example T_2 and T_5 are immediate offsprings of T_1, i.e. mathematically denoted as $Z_{21} = 1$, $Z_{51} = 1$ and $Z_{10} = 1$. As per the cluster representation, all the events associated with an event as its direct or indirect offspring forms a cluster, hence we have three clusters in Figure 2.1(d). The first cluster denotes events related to T_1 and has events T_2, T_3, T_4, T_5 and T_6 in it. Similarly, second cluster consists of events T_7 and T_8, and finally event T_9 is a cluster by itself.

Generative Models for Popularity

Generative models are used for studying online popularity as they unearth, and can account for the underlying mechanisms that generate popularity. [Crane and Sornette, 2008] in their seminal work, utilized point processes for studying popularity bursts and decays on a large scale YouTube dataset of about 5 million videos. They established the presence of self-excitation behavior in YouTube popularity, and they characterized the shapes of bursts and decays in YouTube into four different categories. In a follow-up, work [Crane et al., 2008] authors show that different shapes of popularity evolution can be used to differentiate viral, quality and junk videos. [Matsubara et al., 2012] extended the earlier work by proposing the SPIKEM model. Their model can be seen as a combination of the epidemic models and the Hawkes model. Their model could account for six different shapes of popularity evolution as found in empirical study by [Yang and Leskovec, 2011], an improvement over four shapes as identified by [Crane and Sornette, 2008]. After the seminal work of [Crane and Sornette, 2008], more sophisticated models have been proposed to model and simulate popularity in microblogs [Yu et al., 2017] and videos [Ding et al., 2015]. These approaches successfully account for the social phenomena which modulate online diffusion: the "rich-get-richer" phenomenon and social contagion.

Certain generative models are built explicitly for prediction purposes. [Shen et al., 2014] employ reinforced Poisson processes for predicting the popularity of tweets on Twitter. They model three phenomena: fitness of an item, a temporal relaxation function, and a reinforcement mechanism. Their reinforcement mechanism models the popularity as a step function with pre-defined prior which is proportional to current popularity, making this model somewhat similar to self-exciting processes. Later on, [Gao et al., 2015] extends the model by [Shen et al., 2014] to any step function instead of a pre-defined prior. More recently, [Zhao et al., 2015] utilized the Hawkes process to predict the popularity of retweets. Their model SEISMIC, is a double stochastic process, one accounting for infectiousness and the other one for the arrival time of events. [Kobayashi and Lambiotte, 2016] extended SEISMIC for predicting popularity as a function of time, instead of just predicting the final size. However, their method does not scale to large cascades. Recently, [Rizoiu et al., 2017] extended the Hawkes Process to account for the volume process associated with an event series. Their work can model a Hawkes process in the presence of only aggregated data, for example for a YouTube video we just have aggregated data of daily views not the exact event times of each view. They describe the volumes of attention over fixed time intervals (e.g., daily).

Generative models are typically designed to explain the popularity, not predict it. Most of the methods presented earlier introduce multiple regularization and correction factors in order to make meaningful predictions. This is sub-optimal both for prediction and for interpretation. Moreover, no work exists that compares or links popularity prediction models across the feature-driven and generative model. Hence, in our work in Chapter 4 we develop a two-layer Hawkes model that uses a predictive layer on top to make predictions, giving us the best performances of both techniques.

2.3 Influence Estimation

One related area of research to popularity is influence estimation. A perennial question with researchers, from the days marketing existed, is identifying a group or single user who should be targeted to spread information in a network. According to one of the earliest models by [Katz and Lazarsfeld, 1955], a small set of "influentials" people, act as a bridge between mass media and the general public for the flow of information. In order to spread information, we need to identify these people and make them adopt the desired information. Whereas, as per the new theory by [Watts and Dodds, 2007], it is not these "influentials" but the group of ordinary people and acceptability within the network that spreads information. Hence, as per the new theory in order to spread information, we need to target this group of ordinary people to diffuse the idea into the network. With the advent of online social networks verifying and testing these different ideas became feasible as now we can observe the network over which diffusion occurs.

One of the earliest work was done by [Cha et al., 2010], they measured the role of Twitter users in spreading tweets with news URLs. They found that rarely retweets are driven by the number of followers of a user. They also found that the influence users wielded on their peers in the network is neither lost nor gained accidentally.; users need to work consistently to gain the confidence of other users. In a recent study, [Bakshy et al., 2011] measured the influence of a user as the amount of attention a URL gets, which was first posted by the given user. They observed that, although highly followed users are influential, but so are a group of users with average influence. They argue, in real-world scenarios, this might be more cost-effective.

Our work in Chapter 6 tackles the same problem of calculating the influence of users. However, the novelty of the work lies in comparing the influence of users across network boundaries as we compute the total amount of views a user or group of users on Twitter can gather for a video on YouTube.

We note there are influence maximization studies in literature developed based on game theoretical models [Kempe et al., 2003; Gomez-Rodriguez et al., 2016], which are different from influence estimation. The task in influence maximization is to find the best set of nodes that will spread the information among the most number of users. In contrast, the task in influence estimation is to find the number of users a piece of information will spread given a specific starting node.

2.4 Recurrent Neural Networks

Recurrent neural networks (RNN) [Elman, 1990] have been shown as an effective sequence model in a wide range of applications, such as text [Sutskever et al., 2014], image [Vinyals et al., 2015], video [Jain et al., 2016] and time-series data [Chandra and Zhang, 2012; Lin et al., 1996]. In the thesis, we build on RNNs to represent point process data with neural networks. We next present a brief overview of RNN and their use in modeling the Hawkes process in literature.

Recurrent Neural Networks (RNN) [Elman, 1990] are standard sequence models where the same feed-forward structure is replicated at each time step. They have additional connections from the output of the previous time step to the input of the current time step – therefore creating a recurrent structure. Their hidden state vector $\mathbf{h_t}$ can be defined recursively as:

$$\mathbf{h_t} = f\left(\mathbf{x_t}, \mathbf{h_{t-1}}\right)$$

where f is the feed forward network, $\mathbf{x_t}$ is the current input, $\mathbf{h_{t-1}}$ is the output from previous time step. Long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997; Graves, 2013] units are essentially recurrent networks with additional gated structure, defined as:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{V}_i\mathbf{c}_{t-1} + \mathbf{b}_i\right) \\
\mathbf{f}_t &= \sigma\left(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{V}_f\mathbf{c}_{t-1} + \mathbf{b}_f\right) \\
\mathbf{c}_t &= \mathbf{f}_t\mathbf{c}_{t-1} \odot \tanh\left(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{c}_t + \mathbf{b}_o\right) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right)
\end{aligned} \tag{2.6}$$

where $\mathbf{x}_t$ is input at time t, σ is the logistic sigmoid function and $\odot$ denotes element-wise multiplication. The notations $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$ and $\mathbf{h}_t$ stand for the input, forget, cell-state, output and hidden state at time t. We use the following short-hand notation for the LSTM further in the thesis:

$$(\mathbf{h}_t, \mathbf{c}_t) = LSTM(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \tag{2.7}$$

LSTM and its variants have been successfully used for modeling time series and predicting sequence, due to their ability to capture the effects of past data in their hidden state [Hochreiter and Schmidhuber, 1997; Graves, 2013; Chung et al., 2014].

RNNs are utilized by [Du et al., 2016; Xiao et al., 2017; Wang et al., 2017a; Mei and Eisner, 2017] to built point process models for predicting popularity of events. Whereas, recent work by [Cao et al., 2017; Li et al., 2017a] developed an end to end cascade prediction model based on RNNs.

In comparison with these models, our model in Chapter 6 and Chapter 7 predicts popularity using two asynchronous streams of data about a single event. Although HIP [Rizoiu and Xie, 2017], [Roy et al., 2013], [Castillo et al., 2014] and [Abisheva et al., 2014] have utilized multiple streams of data for predicting the popularity, but they aggregate data in different streams rather than using the more fine-grained information. Work by [Wang et al., 2017b], and [Xiao et al., 2017] combines asynchronous streams. However, they aggregate exogenous individual events at fixed intervals rather than considering a separate exogenous volume stream.

2.5 Summary

Our work in modeling popularity on social media builds on a broad base of work in feature-driven and point process based approaches. In particular, we identify a set of features that can be built on free public available datasets and show competitive results across different problem settings of regression and classification. We also build generative methods for modeling popularity based on Hawkes Processes and present a unifying view across feature-driven and generative methods. Our work further represents event data with the help of RNNs and develops the first known algorithm for handling multiple asynchronous streams of data under a joint model. This piece of work builds the gap between event series and volume data modeling with joint inference. We also review some specific literature in some chapters individually as they relate to a very specific way of modeling social data handled in the chapter.

Chapter 3

Datasets

In this chapter we describe the details of three datasets we curated for our analysis in the thesis. The datasets presented in the chapter helps us to understand and answer the questions we raised in Chapter 1 related to links between feature-driven and generative models, and linking models for events and volume data. In Section 3.1, we present a domain specific Twitter news dataset, NEWS. The salient feature of NEWS dataset is the presence of both meta-information and the time stamps attached to a tweet, which helps to explore connections between generative and feature-driven models for popularity in Chapter 4. In Section 3.2.2 and Section 3.3 we present two datasets related to videos uploaded on YouTube. Section 3.2.2 presents ACTIVE'14, an enhanced version of the YouTube dataset Active developed by [Rizoiu et al., 2017]. In Section 3.3.1 we present ACTIVE-3YR, a YouTube dataset spanning over three years, 2015 to 2017. Both YouTube based datasets are the first datasets in the literature that combine two social platforms, YouTube and Twitter. They have volume data about daily views and shares gathered by a video, along with information about individual tweets associated with the video. The presence of volume and individual events information helps us to build models linking volume and events data in Chapter 6 and Chapter 7. Table 3.1 lists all the datasets curated as a part of this thesis.

Table 3.1: Datasets curated in the thesis for modeling popularity in social media.

Type of Data	Dataset Name	Chapter Used	Publications
Twitter (News)	NEWS	Chapter 4,5	[Mishra et al., 2016; Rizoiu et al., 2018]
YouTube+Twitter	ACTIVE'14	Chapter 6,7	[Mishra et al., 2018]
YouTube+Twitter	ACTIVE-3YR	Chapter 7	under preparation

3.1 News Dataset

Many of the datasets used by prior work for popularity prediction were not made public because of user agreement terms and they are difficult to replicate. The barriers for having an open benchmark include proprietary insider information (e.g. complete Facebook network measurements [Cheng et al., 2014]), privileged API access (e.g., Twitter firehose [Zhao et al., 2015]) or contain restricted data from non-English sources [Ding et al., 2015; Yu et al., 2017]. Furthermore, we argue that predicting popularity on a topic or domain- specific dataset is as relevant as that on the firehose feed – content producers and consumers are often interested in content within a source (e.g. NYTimes) or a topic (e.g. technology). Additionally, there is always a new content in news due to their relation with contiguously developing events that typically have lifespans of 24 hours or 7 days. Hence, we conjecture it is interesting to look at articles that capture attention beyond this expected lifespan of articles.

We construct a domain specific dataset of tweets related with news articles, using only free access APIs. We use the *free* Twitter Streaming API[1] for collecting the tweets related to news articles, over a period of four months from April 2015 to July 2015. In order to have a diversity in our coverage of news articles we track the official handles of ten most popular news outlets as per 'Alexa'[2], as shown in Table 3.2:

Table 3.2: List of news media outlets captured for creating NEWS dataset.

Name	Twitter Handle	URL snippet	Twitter User Id
NewYork Times	@nytimes	nytimes com, http nyti ms	807095
Associated Press	@AP	ap org, http apne ws	51241574
CNN	@CNN	cnn com, http cnn it	759251
Washingtonpost	@washingtonpost	washingtonpost com, http wapo st	2467791
Reuters	@Reuters	reuters com, http reut rs	1652541
Yahoo News	@YahooNews	news yahoo com, http yhoo it	7309052
Guardian	@guardian	theguardian com, http gu com	87818409
BBC	@BBC	bbc com	19701628
Huffingtonpost	@HuffingtonPost	huffingtonpost com, http huff to	14511951
Google News	@googlenews	news google com, news goo gl	33584794

We also capture all tweets that either contain i) mention of the above users or ii) commonly used shortened URL used for any of the above stated ten news websites. We follow a two stage strategy where we first follow the Twitter handle of the news media outlet. In the second stage, we track the mention of Twitter user name or URL for the media outlets. All user handles, ids

[1] `https://dev.twitter.com/streaming/overview`

[2] `http://www.alexa.com/topsites/category;1/Top/News`

and URLs tracked are shown in Table 3.2. This collection scheme leads to a total of 49,735,271 tweets in the dataset.

3.1.1 Retweet Cascades

We present our dataset as a retweet cascade, where we combine all the retweets of a single tweet as a single cascade. Each tweet in the cascade is represented as a tuple, (m_i, t_i), where m_i is the number of followers of the user who created the tweet and t_i is the difference in time in seconds of the retweet from the original tweet. In our representation we assume the first original tweet occurs at time zero, i.e $t_0 = 0$. The above stated representation of our dataset naturally presents dataset as a time series of events that are related to each other via first event in the series. Our representation of cascades is inspired from the dataset, Tweet-1Mo, released by [Zhao et al., 2015].

We note that, there are multiple cascades about the same news article in our dataset as cascades are represented as retweets of an original tweet. Despite these multiple cascades talk about the same article, their dynamics are still very different. For example in Figure 3.1, we have two retweet cascades talking about a single news article published in NewYork Times covering news about death of actor Leonard Nimoy who played 'Mr. Spock' in Star Trek movies. The initial dynamics of both cascades looks very different where the first cascade attracts events with high number of followers whereas the second cascade attracts much more events than first cascade with less number of followers. We set out to capture this complex interplay in event timings and number of followers of events in the models presented in Chapter 4 and Chapter 7.

Beyond follower and event timings. As mentioned earlier representation of cascades in the News is inspired by the Tweet-1Mo. However, the News dataset contains several additional user information, deemed to be useful for predicting the popularity of online items in earlier work by [Martin et al., 2016; Cheng et al., 2014]: for each user in a retweet cascade, we record her number of friends, the number of posted statuses and the account creation time. We restricted ourselves to the user features mentioned above as these features are inexpensive to create due to their availability in every tweet retrieved from the free Twitter Streaming API. Hence, requiring no further queries to retrieve the desired data. Whereas other user features, like the list of followers or friends of a user, are expensive to create as they need additional queries to the API for retrieval. Furthermore, for highly followed users due to the restrictions in the API, a large number of queries are required to retrieve the full list. Consequently, the retrieval becomes

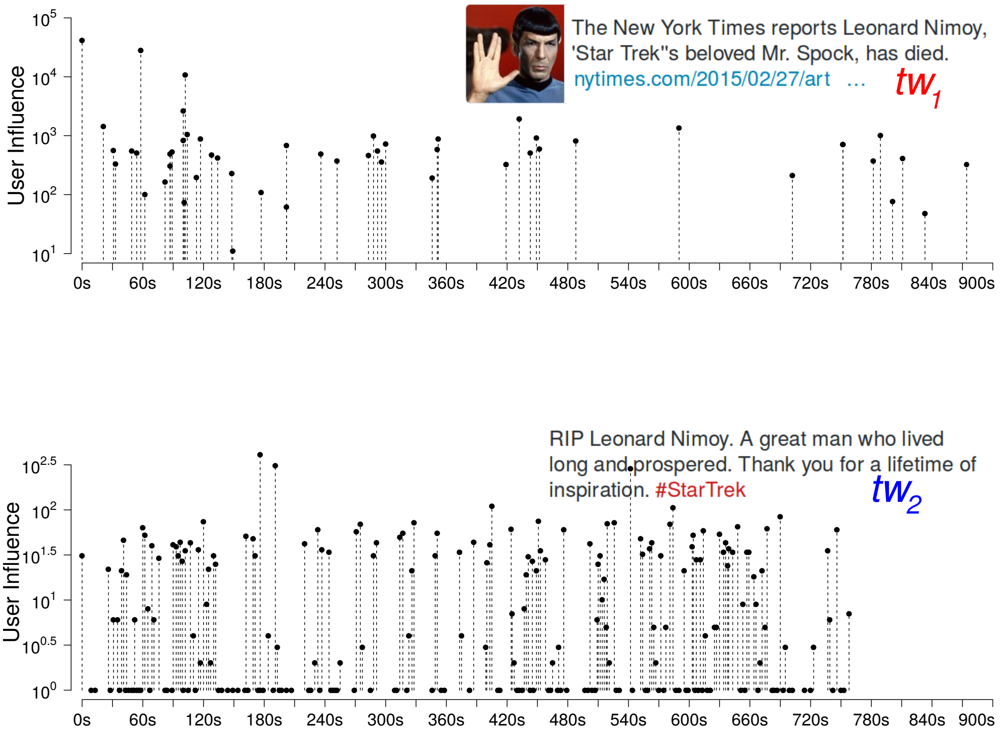


Figure 3.1: Two retweet cascades announcing the death of "Mr. Spock". The underlying diffusion processes of tw_1 and tw_2 are very different: tw_1 lasts for 2.5 hours and attracts the attention of highly influential users, especially music related channels, however tw_2 diffuses faster and ends in just 12 minutes. x-axis: time in seconds; y-axis: number of followers each tweet can reach.

unfeasible as it quickly uses up the free access quota.

The user information from the News dataset is used for constructing features described in Section 3.1.2. Correspondingly, facilitating common benchmarking on both feature-driven and generative methods on a single dataset, which was not possible on Tweet-1Mo, previous benchmark dataset for tweet popularity prediction, due to the lack of the available information. Most of the cascades in the News dataset are of length one or two and, for compatibility with the Tweet-1Mo, we present the statistics for cascades of length over 50: 110,954 cascades, mean cascade length is 158, and median length is 90.

3.1.2 Features for popularity prediction

We have additional information in our News dataset (when compared to Tweet-1Mo), which helps us to build features describing each cascade using four different types of information: a) basic user features, b) temporal features, c) volume features and d) past user success. We have restricted ourselves to above stated features as they have been reported to be most informative in recent literature [Bakshy et al., 2011; Cheng et al., 2014; Martin et al., 2016; Pinto et al., 2013; Szabo and Huberman, 2010] and require only free access to the Twitter data source. In particular, [Martin et al., 2016] compare a wide range of features, including user, content and user-content interplay features, and they show that basic user features together with past user success features account for 87% of the prediction performance. [Cheng et al., 2014] studied the problem of predicting whether a cascade will double its size or not in the case of Facebook photos. They show that temporal patterns of the observed adoption are the best performing features. They report temporal features can achieve performance scores within 0.025 of those achieved when using all features in combination for measuing accuracy. Lastly, [Szabo and Huberman, 2010] and later [Pinto et al., 2013] show that early popularity is predictive for total popularity. All the features are built for hold-out predictive setting, i.e we build features for the seen or observed period during the unfolding of a cascade and use it for predicting popularity of the content. The extensive set of features we build on News dataset are listed below.

Basic User Features. Basic user statistics capture the social influence of a user [Bakshy et al., 2011; Cheng et al., 2014; Martin et al., 2016]. When observing the prefix of a cascade, we use the five point summary (min, median, max, 25-th and 75-th percentile) to represent the distribution for a feature:

- Number of Followers: count of user followers;
- Number of Friends: count of friends for a user;
- Number of Status: count of statuses posted by user;
- User Time: Time when user account was created.

Temporal Features. We capture the temporal dynamics of a cascade using the following features, as computed in [Cheng et al., 2014]:

- First Half Rate: mean waiting time between retweets, during the first half of the observed cascade;
- Second Half Rate: mean waiting time between retweets, during the second half of the observed cascade;

- Waiting Time Distribution: five point summary of waiting times between retweets, in the observed cascade;
- Exposure Distribution: five point summary of the distribution of number of followers for all users before the last retweet, in the observed cascade.

temporal features helps us to capture how is a cascade unfolding with respect to time in a network. They help us to implicitly capture the rate of growth(virality) of a content in the network.

Volume. Number of retweets in the observed part of the cascade, which captures early popularity. Volume helps us to capture the raw total amount of attention gathered by the content till an observed period [Pinto et al., 2013; Szabo and Huberman, 2010].

Past User Success. Past user success helps us to capture the previous success of an user in promoting a content [Bakshy et al., 2011; Martin et al., 2016]. We measure it using the average size of the cascades started by a given user in past. For an observed cascade, we use the five point summary of the distribution of past user success for all previously known users who participate in the cascade. We use the News historical data from April to June to construct the past user success feature for the feature-driven approach. As this feature can only be constructed for users who have started a cascade earlier, we consider only those users who initiated at least 2 cascades in the past as active users. Non-active users have a past user success of 0.

News dataset helps us in two main ways i) motivating use of event timings and user influence(number of followers) for predicting popularity and ii) creating a strong basis for future benchmarking in Chapter 4 for predicting popularity of events streams of data based on both generative and feature-driven models. This benchmarking is first of its kind in predicting popularity on online media where you can compare different class of models against each other.

3.2 YouTube Dataset

In this section we present two YouTube videos dataset i) Active'14 an extension of Active dataset released by Rizoiu *et al.* [Rizoiu et al., 2017] and ii) Active-3Yr a new YouTube video dataset spanning over three years 2015-2017. The uniqueness of our datasets over any other YouTube videos dataset is the presence of individual tweets about each video along with the existing information of daily views and shares of the video on YouTube. Active'14 and Active-3Yr are the first datasets to combine evolution of popularity of an online item on both YouTube and

Table 3.3: Number of videos per category in Active'14 dataset. Music is the most dominant category with almost 25% of videos and Howto & Style being the least dominant category with around 1.3% of videos. [Rizoiu et al., 2017]

Category	No.of Videos	Category	No.of Videos
Comedy	865	Music	3549
Education	298	News & Politics	1722
Entertainment	2422	Nonprofits & Activism	333
Film & Animation	664	People & Blogs	1947
Gaming	882	Science & Technology	262
Howto & Style	180	Sports	614
Total:	13,738		

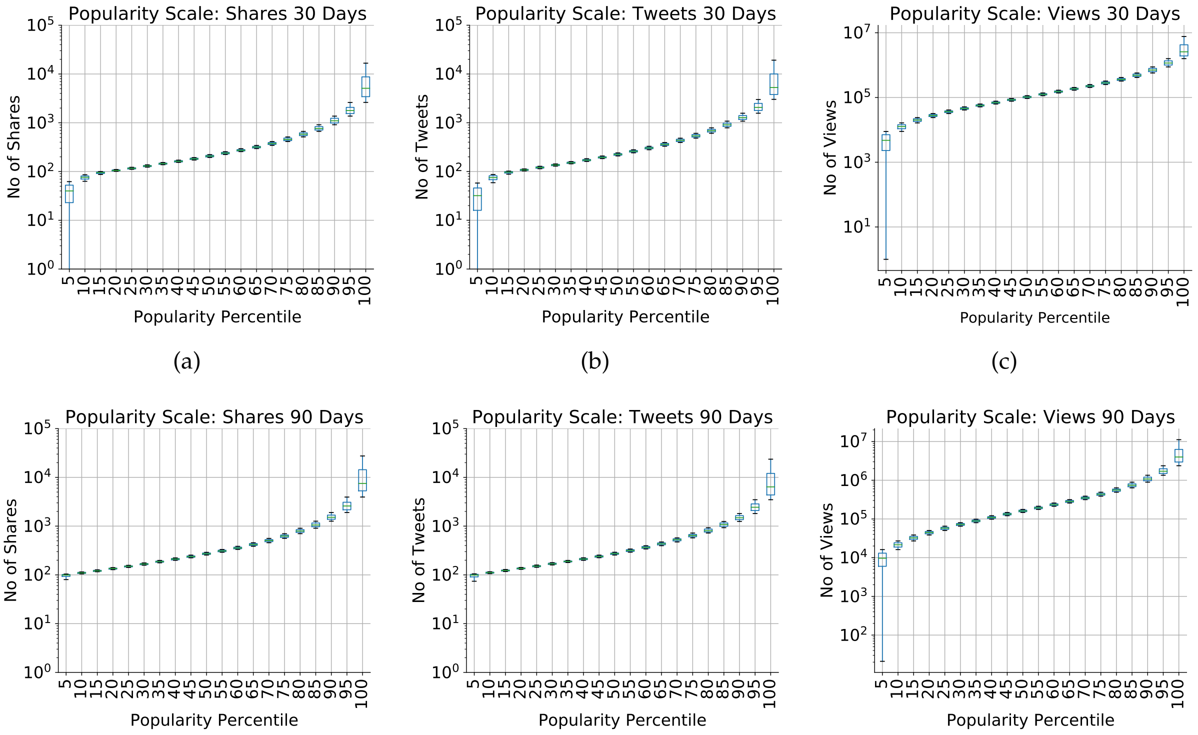
Twitter. They allow us to study the problem of popularity across platforms and at different granularities, daily volumes and individual events, in Chapter 6 and Chapter 7.

3.2.1 Active Dataset [Rizoiu et al., 2017]

The YouTube videos dataset released by [Rizoiu et al., 2017] contains 13,738 videos. All videos in this dataset were uploaded between 2014-05-29 to 2014-08-09 and have at least 100 views, 100 shares, and 100 tweets. They query the Twitter API for tweets containing a reference to YouTube videos. Extracted tweets have a field `expanded_url` in them, which contains a link to the YouTube videos. Hence, a list of unique YouTube videos ids is constructed from the tweets. Finally, they set up a YouTube crawler to scrap data from YouTube website using the curated list of unique video ids. The crawler extracts information about the daily views, the daily shares, channel (attribute identifying the user who uploaded the video) and an owner assigned category of the tweeted videos. Only information about the first 120 days for a video is kept for prediction setup. Table 3.3 shows the distribution of videos as per different categories. Four most dominant categories, 'Music', 'Entertainment','People & Blogs' and 'News & Politics', account for almost 70% of videos in the dataset.

3.2.2 Active'14 Dataset

Active'14 is an enhanced version of Active. We augment each video in Active with corresponding individual tweets data about the video, emitted during the same period. There are in total 30.2 million tweets in the dataset. On average each video is tweeted 2151 times, and the median number of tweets for a video is 327. We follow the procedure proposed by [Rizoiu et al., 2017] to group videos by their popularity bin. Popularity bin is calculated by transforming the ranking of videos as per the total view (share/tweet) counts into a percentage scale where the video with highest count is mapped to 100%, and video with the lowest count to 0%. We group videos in bins of 5% each, each bin has approximately 686 videos in it.



(a) (b) (c)

(d) (e) (f)

Figure 3.2: Top Row: Popularity scale for videos in Active'14 dataset for shares, tweets and views for first 30 days. Bottom row: shows the same data after first 90 days. We can see that distribution of shares and tweets for videos in same popularity bins are almost identical for both set of days. Also moving from 30 days to 90 days amount of change in shares and tweets is negligible for larger percentiles whereas views shows a slightly increasing trends. We note that, all the videos in Active'14 set receive at least 100 shares, 100 tweets and 100 views over their lifespan of 120 days. Hence, the rise in lower percentiles over time is an expected phenomenon.

Figure 3.2 shows the box-plot of video view, share, and tweet counts (in log-scale) of each bin for 30 and 90 days. From Figure 3.2(a), (b), (d) and (e) we observe the popularity scale for tweets and shares are very similar across different days. The shape of the evolution of popularity bins for views remains almost constant when moving from 30 days to 90 days, with a slight increase shown by the videos in the highest popularity bin(last box-plot in Figure 3.2(c) and (f)).

In order to augment tweets to the Active data, each tweet in the dataset about the video is represented as a tuple (m_i, t_i), where m_i is the number of followers of the user who created the tweet and t_i is the difference in time in seconds of the tweet from the first tweet about the video in the dataset. We note, the representation of tweet cascade in here is similar to one in Section 3.1 for the News dataset. However, in the News dataset, we create retweet cascades whereas in here tweets are organized as cascades corresponding to each video under consideration. Hence in this representation, each video has exactly one tweet series corresponding to it. The augmentation of individual tweets data in Active'14 dataset enriches it to have both the micro event information of tweets about a video along with macro information about shares and views(aggregated number of shares, and views per day). We utilize the Active'14 dataset in Chapter 6 to predict the popularity of a YouTube video under promotion from shares in YouTube and tweets on Twitter.

3.3 Active-3Yr **Dataset**

Active-3Yr dataset is a longitudinal YouTube dataset spanning across three years 2015, 2016 and 2017. In total there are 389,632 videos in this dataset where each video has at least 100 shares, 100 tweets and 100 views within first 120 days of its upload. All videos in this dataset are uploaded on YouTube between 2015-01-01 to 2017-09-01.

In Section 3.3.1 we will detail the steps used for creating the dataset and follow it up with a discussion on various trends observed in this longitudinal dataset in Section 3.3.2.

3.3.1 Active-3Yr **Curation**

We first curated a dataset similar to Active'14 but for year 2017, using YouTube insight tool developed by Wu *et al.* [Wu et al., 2018b]. The raw number of videos in this dataset after applying our constraints of having at least 100 shares, 100 tweets and 100 views within first 120 days of its upload is 456,324 videos. We now extract a list of all unique channels from this dataset, to get 89,404 unique channels. We further filter the list to contain only those channels that have uploaded a video in our Active'14 dataset, leaving us with 3,093 unique channels that were active in period 2014-05-29 to 2017-12-31.

Finally, we retrieve all the videos uploaded bwtween 2015-01-01 to 2017-12-31 by all the channels in our filtered list using the YouTube insight tool [Wu et al., 2018b], giving us in total of 936,793 videos. We again filter videos as per our minimum requirements stated above to end with a list of 389,612 videos.

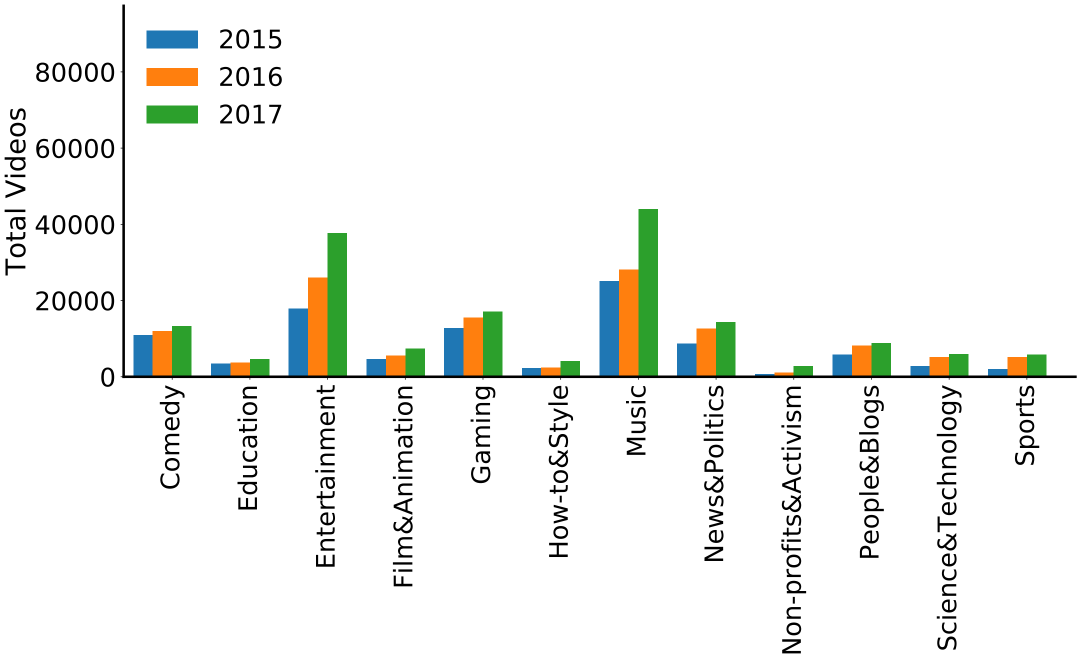


Figure 3.3: Distribution of videos in Active-3Yr dataset per category for each given year (2015, 2016, 2017) in dataset.

3.3.2 Active-3Yr Trends

In this section we present our observations about distribution of videos as per categories and channels. Next we present our observations on how popularity scale has evolved between years for videos and finally end our discussion with presenting individual example of a YouTube channel 'MBCkpop' that shows very surprising trends.

Distribution of videos as per categories and channels. Figure 3.3 shows the distribution of videos as per different categories for every given year (2015, 2016 and 2017) in the dataset, while Table 3.4 presents the same distribution aggregated over all the years for different categories. Similar to the Active'14 dataset, 'Music' is the most dominant category in the Active-3Yr dataset, accounting for almost 25% of videos. However, categories like 'Gaming' and 'Comedy' have an increased presence in Active-3Yr in comparison to Active'14 (refer Table 3.3), whereas we observe a drop in the share of videos belonging to the category 'People & Blogs'. This observation helps us to conclude that even if much content is produced in categories like People & Blogs', the life of channels in these categories are relatively smaller than the channels in categories 'Gaming' and 'Comedy'.

Table 3.4: Number of videos per category in Active-3Yr dataset. Music is the most dominant category with almost 25% of videos and Nonprofits & Activism being the least dominant category with around 1.2% of videos.

Category	No.of Videos	Category	No.of Videos
Comedy	36311	Music	97332
Education	11878	News & Politics	35783
Entertainment	81762	Nonprofits & Activism	4691
Film & Animation	17668	People & Blogs	22963
Gaming	45407	Science & Technology	13905
Howto & Style	8802	Sports	13110
Total:	389,612		

Table 3.5: Number of videos per channel in Active-3Yr dataset for 10 channels with highest number of videos posted.

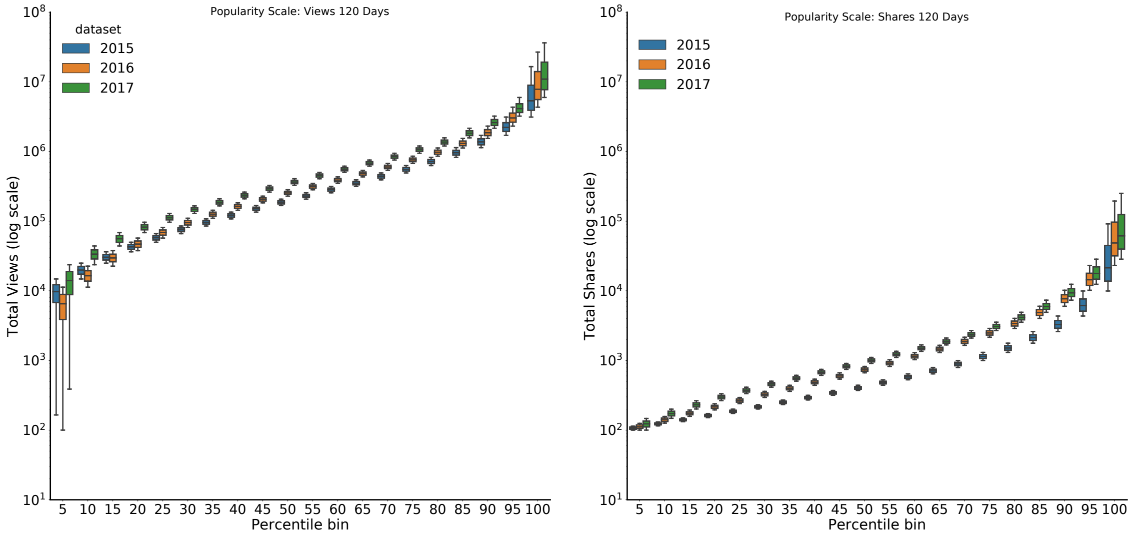
Channel Name	No.of Videos	Channel Name	No.of Videos
M2	1788	'need to get russian'	1182
Gameplayrj	1708	KBSKpop	1173
The Young Turks	1693	The Next News Network	1137
SMTOWN	1390	Mnet K-POP	1100
rezendeevil	1281	MBCkpop	1072

Table 3.5 presents the 10 most productive channels, channels with highest number of videos, in our dataset. Expectedly most of these channels are from the category 'Music', but an interesting observation is that half of them M2, KBSKpop, Mnet K-POP, SMTOWN, and MBCkpop produce videos of a specific genre known as K-pop.[3]

Popularity scale over time. Figure 3.4 shows the evolution for popularity in Active-3Yr dataset for years 2015, 2016 and 2017. The increase in dynamic range of total views over consecutive years for same percentile bin is very small, as seen from the last box-plot for the three years in Figure 3.4(a). However, the increase for shares is much more significant (Figure 3.4(a)). The following observation tells us that although over the years the distribution of views have remain same but the amount of promotions a video gets has increased. Hence, the driving ability of a certain amount volume of promotion to gather views has decreased over the years. Note as described in Section 3.2.2 the trends for tweets are very similar to the trends for shares.

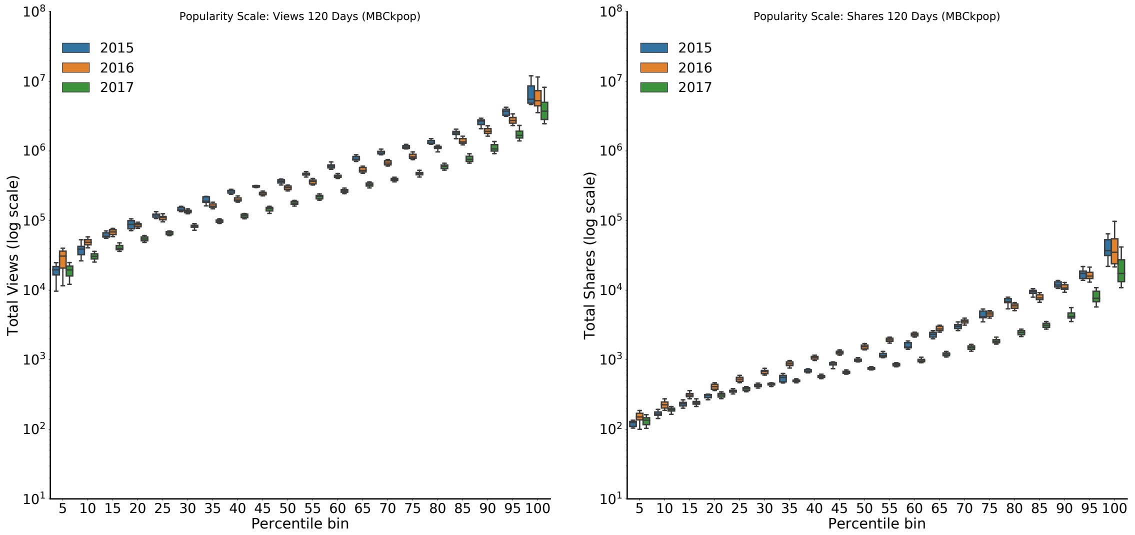
Individual Example. Figure 3.5 shows the evolution of the popularity scale for both views and shares of videos from channel MBCkpop in the Active-3Yr dataset, which is 10^{th} most

[3] https://en.wikipedia.org/wiki/K-pop



(a) (b)

Figure 3.4: Figure (a) shows the evolution of popularity scale for total views of videos in Active-3Yr. Figure (b) shows the evolution of popularity for total shares for same set of videos.



(a) (b)

Figure 3.5: Figure (a) shows the evolution of popularity scale for total views of videos in Active-3Yr for channel 'MBCkpop'. Figure (b) shows the evolution of popularity for total shares for same set of videos.

productive channel in our dataset. To our surprise, we observe the opposite trend for videos from this channel when compared against the general trend we presented above for categories. Both views and shares decrease for the same percentile bin when we move ahead in time from

2015 to 2017. On further analysis, we find similar trends for some other popular K-pop channels too. We note, our observation does not imply that the popularity of the K-pop music genre is under decline, as it might be a case of attention being distributed evenly across various channels rather than a few selected ones. Hence, any study on understanding the evolution of popularity for a group of videos should consider the channel to be a better indicator than the category. Similar observations have been made in earlier work by [Martin et al., 2016] that user features are more important than content features.

Active-3Yr dataset is used in Chapter 7 to learn a single model for a group of videos. The observations related to unfolding of popularity scale over a prolonged time and for various different kind of videos in Active-3Yr forms our basis for developing shared models. The three year span of dataset helps us to analyze the longitudinal effects of data for predicting popularity.

3.4 Summary

We introduced three new datasets in this chapter namely: News, Active'14 and Active-3Yr. The observations made in this chapter helped us at various stages in our research to build intuitions and hypothesis for our models. News dataset is used to create the first benchmark for comparing predictive results for popularity prediction in Twitter across generative and feature-driven models, in Chapter 4. Similarly Active'14 opens up the avenue for predicting popularity by utilizing data from more than one network in Chapter 6. Finally Active-3Yr dataset spanning over three years is used in Chapter 7to study longitudinal behavior of our models alongside opening opportunities to do cold start prediction on videos based on specific attributes.

Chapter 4

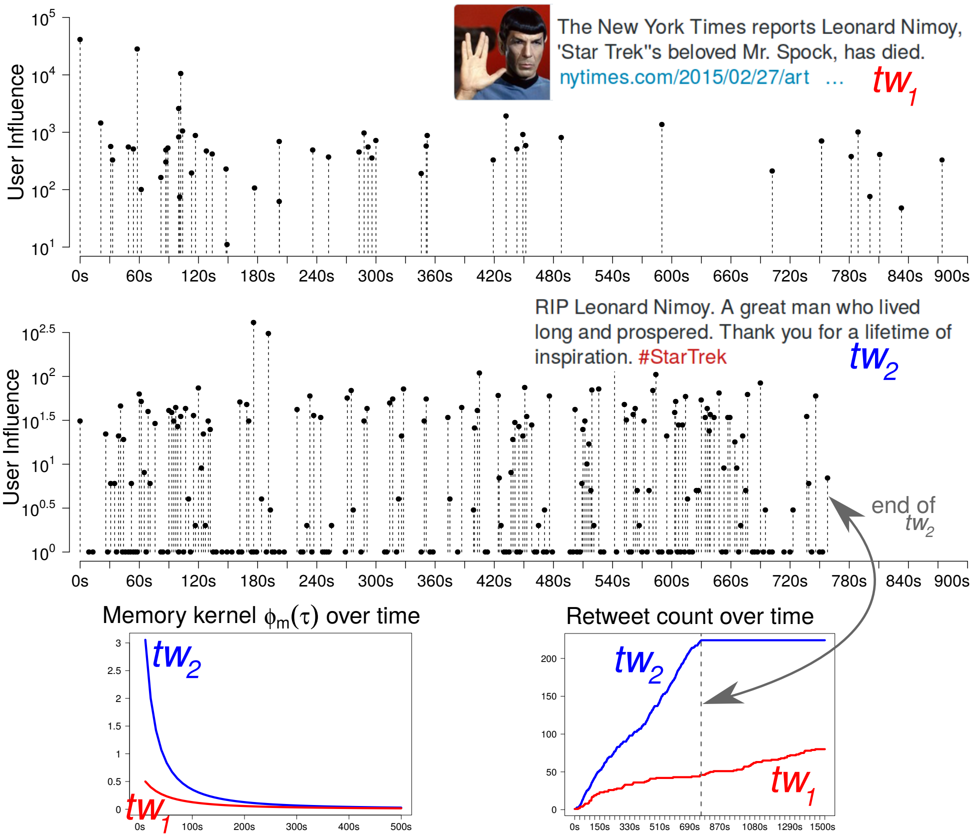
Bridging Point Process and Feature-Driven Approaches

In this chapter, we will present our work on bridging point process and feature-driven methods for modeling information diffusion cascades. In particular, we address the problem of predicting the size of a retweet cascade after observing it for a given period. In Section 4.3 we demonstrate how to use the Hawkes process for modeling retweet diffusions by constructing a special kernel that accounts for various social effects like preferential attachment, social memory, and content quality. Next, in Section 4.4 we present a way to utilize the Hawkes processes for predicting the final size of a cascade and bridge the gap between feature driven and point process based popularity prediction model by systematically incorporating a prediction layer on top of Hawkes Process for prediction. Finally, we leverage the advantages of both feature-driven and generative models by combining them to build a hybrid model in Section 4.5.2. Our Hybrid model gives us explicit power to model important factors responsible for social interactions along with the predictive power of feature-driven models based on historical data. Our work in Section 4.3 for modeling tweet cascades with Hawkes Process inspires the generalized models we develop in Chapter 5 and Chapter 6.

4.1 Motivation

When presented with the task of predicting popularity, a clear gap appears between the two main classes of approaches: the feature-driven approaches and the generative approaches. *Feature-driven approaches* [Asur and Huberman, 2010; Cheng et al., 2014; Martin et al., 2016; Pinto et al., 2013] summarize network, user, and event history information into an extensive set of features, and they use machine learning approaches to predict future popularity. *Generative* *methods* [Ding et al., 2015; Shen et al., 2014; Yu et al., 2017; Zhao et al., 2015] leverage fine-grained timing information in the event series, however they make stronger assumptions about the diffusion process. A thorough understanding of strengths and weaknesses of the two classes is currently missing, especially since the generative methods are usually designed for explaining the mechanisms that generate attention, and not optimized for prediction. Another gap arises between the different problem settings. Prior work has address problems like predicting the volume of attention [Shen et al., 2014; Zhao et al., 2015], predicting if a cascade will double in size [Cheng et al., 2014], whether an item will have 10 million views [Shamma et al., 2011], or be among the top 5% most popular [Yu et al., 2014]. However, the understanding of how the different approaches and the different features generalize over the various problem settings is incomplete. In addition, a practical bottleneck is the availability of certain types of these features, such as network data and corresponding features – for example the number of users exposed in the diffusion up to a time *t* or potential number of reachable users. A third gap in the current available work is the consensus and understanding about which features are informative for feature-driven models and the lack of benchmark datasets on which new methods can be devised.

In this chapter, we address the above challenges in the context of predicting the final size of retweet cascades. We build two predictive approaches, one generative and one feature-driven, and we show how to combine them into a hybrid predictor to further increase performances. The generative method is a two-layered approach, built on the intuitive *Hawkes self-exciting process* [Hawkes, 1971] model. Self-exciting point processes are a class of stochastic processes, in which the occurrence of past events makes the occurrence of future events more probable. Three key factors in information diffusion are built into the proposed model: the social influence of users, the length of "social memory" and the inherent tweet quality. We use a predictive layer on top of the generative model to make final predictions. This helps us to take into account other cascades and mitigate limitations of model assumptions and parameter estimation. Our second proposed approach, the feature-based method, uses the features that we identified in Section 3.1.2. This results in a competitive feature-based predictor, which consistently outperforms the current state-of-the-art popularity prediction model [Zhao et al., 2015] – reducing the mean absolute relative error by more than 200% when compared to the latter, after observing the retweet series for 10 minutes. We show that the same set of features can be employed in both regression and classification tasks. We further experiment with a hybrid predictor, which uses both data features and measures issued from the generative model, and we show additional



The New York Times reports Leonard Nimoy, 'Star Trek''s beloved Mr. Spock, has died. nytimes.com/2015/02/27/art …

RIP Leonard Nimoy. A great man who lived long and prospered. Thank you for a lifetime of inspiration. #StarTrek

Figure 4.1: Two retweet cascades announcing the death of “Mr. Spock”. The underlying diffusion processes of tw_1 and tw_2 are very different: tw_1 lasts for 2.5 hours. It attracts the attention of highly influential users, especially music related channels, however the generative kernel of tw_2 amplifies more than that of tw_1. tw_2 diffuses faster and ends in just 12 minutes, but the final sizes for the two cascades are similar (224 retweets for tw_1 and 219 for tw_2). We model and interpret these cascades using our generative model in Sec. 4.5.4. x-axis: time in seconds; y-axis: number of followers each tweet can reach; $\phi_m(\tau)$: memory kernel of Equation (4.3) plotted with $m = 1000$.

performance improvement.

We conduct experiments on the News dataset that we have introduced in Section 3.1. Fig. 4.1 illustrates two of the retweet cascades in this dataset, related to the passing of the actor Leonard Nimoy (“Mr. Spock” of “Star Trek”) . Even though the two cascades achieve similar popularity, our generative model is capable of differentiating between their very different diffusion dynamics. For example, the memory kernel $\phi_m(\tau)$ of tw_2 – i.e. the contribution of each event to the spawning of future events – amplifies more (shown by a larger area under the $\phi_m(\tau)$), but is forgotten faster (the entire diffusion ends in just 12 minutes).

The three main contributions of this chapter are:

- **Bridging the popularity prediction problem space:** we address three types of gaps: problem settings and task (e.g. regression vs. classification), type of approach (feature-based vs. generative) and the generalization of the features across feature-driven approaches.

- **Comparative understanding of feature-driven and generative models:** we propose a generative method, based on Hawkes self-exciting processes, which features the advantages of generative methods – e.g. interpretable results – and which is adapted for predicting the final cascade popularity by using an additional predictive layer; we compare approaches to popularity prediction from the two main classes, i.e. feature-based and generative, and we show that combining them into a hybrid approach increases performances.

- **Benchmarking Dataset:** we perform extensive experiments on the NEWS dataset from Chapter 3 for predicting popularity of retweet cascades. We show that the identified set of features in Section 3.1.2 of Chapter 3 performs comparable to the previous state-of-the-art model for both regression and classification settings. This new dataset allows comprehensive performance benchmarks on features about users and temporal activity.

4.2 Problem statement

In this work, we aim to model the retweeting dynamics of a tweet and predict its future popularity, i.e., total number of retweets for the tweet. In Twitter terminology a retweet is defined as the re-sharing of an original tweet by a user via the dedicated Twitter interface. We sort the original tweet along with its retweets according to their post time in ascending order, forming an information cascade. Each tweet is denoted by an event time t_i, and magnitude m_i which denotes the number of followers of the user posting the tweet. Specially, we use $t_0 = 0$ to denote the post time of the original tweet. Consequently t_i is the time difference between the post time of the retweet from the original tweet. Our aim is to predict the final size of the information cascade, N_∞, after observing a cascade up to time T. Table 4.1 summarizes the terminology, along with parameters of our model and their interpretations.

Table 4.1: Summary of notations. Top: Quantities for estimating popularity using point processes. Bottom: Parameters of the marked Hawkes process.

Notation	Interpretation
m_i	event magnitude. Social influence of the user emitting the tweet.
t_i	event time. Tweet timestamp.
$\lambda(t)$	conditional intensity – or event rate – of a non-homogeneous Poisson Process.
$\phi_m(\tau)$	triggering kernel. Contribution of event (m, t) to the total event rate, calculated at time $t + \tau$.
n^*	branching factor, mean number of offspring events spawned by a parent event. For $n^* < 1$ the cascade dies out (subcritical regime), for $n^* > 1$ the cascade explodes exponentially (supercritical regime).
T	time extent of the observed series of events in the beginning of a retweet cascade. $T = max(t_i)$.
n	number of events in the observed series of events in the beginning of a retweet cascade. $n = max(i)$.
N_∞	expected total number of events in a given retweet cascade, derived from the generative model.
N_{real}	the real number of events in a given retweet cascade, or total popularity.
ω	predictive factor learned using a Random Forest regressor.
$\hat{N}_\infty$	total popularity prediction made using the output of the predictive layer.

Parameter	Interpretation
θ	the power-law exponent, describing how fast an event is *forgotten*. $\theta > 0$.
κ	tweet quality. High quality tweets are more likely to generate more retweets. $\kappa > 0$.
c	temporal shift cutoff term so that $\phi_m(\tau)$ stays bounded when $\tau \simeq 0$. $c > 0$.
β	user influence power-law warping effect in $b(m)$. $\beta > 0$. $0 < \beta < \alpha - 1$ for having a defined branching factor n^*.
α	exponent of the user influence power-law distribution. Fitted to the `#followers` distribution: $\alpha = 2.016$.

4.3 Retweeting as a marked Hawkes process

The cascade of retweets along with the original tweet can be seen as a *word of mouth* diffusion, users share content, and other users consume the shared content and at times re-share it with users connected to them. Visibly, each retweet increases the likelihood of future retweets, therefore the retweeting process is self-exciting and can be modeled using a marked Hawkes process [Hawkes, 1971].

In Section 2.2.2, we presented how a Hawkes process can be seen as a branching process (see Figure 2.1). Utilizing the terminology from the aforementioned branching process, a retweet cascade can be seen as a point process with one immigrant, i.e., the initial tweet, and the rest of the events being its offsprings, i.e., retweets. As per Section 2.2.2 the event intensity function in a Hawkes process is given as:

$$\lambda(t) = \lambda_0(t) + \sum_{T_i < t} \phi_{m_i}(t - T_i) \ . \tag{4.1}$$

$\lambda_0(t)$ is the rate at which immigrants arrive into the systems. In case of retweets cascade as the original tweets is the only immigrant we have $\lambda_0(t) = 0, \forall t > 0$. Hence for retweet cascade the intensity function is given as:

$$\lambda(t) = \sum_{T_i < t} \phi_{m_i}(t - T_i) \ . \tag{4.2}$$

As we formulate our model as *marked Hawkes process* we model the mark or magnitude of each event with the user influence of each tweet. The initial original tweet has magnitude m_0 at event time $t_0 = 0$. Each subsequent retweet has the magnitude m_i at the event time t_i.

4.3.1 Social power-law kernel

Each event (m_i, t_i) effects the event intensity through the triggering kernel $\phi_{m_i}(t_i)$. We construct a power-law kernel [Helmstetter and Sornette, 2002] where we separate the effects of the magnitude (m_i) and of the event time (t_i):

$$\phi_{m_i}(t_i) = \kappa m_i^{\beta} (t_i + c)^{-(1+\theta)} \ . \tag{4.3}$$

κ describes the *virality* – or quality – of the tweet content and it scales the subsequent retweet rate; β introduces a warping effect for the user influence and it is related to the observed long-tail distribution of user influence in social networks; and $1 + \theta$ ($\theta > 0$) is the power-law exponent, describing how fast an event is *forgotten*, the parameter $c > 0$ is a temporal shift term to keep $\phi_{m_i}(t_i)$ bounded when $t_i \simeq 0$; here κm_i^{β} accounts for the magnitude of influence, and the power-law kernel [Crane and Sornette, 2008] $(t_i + c)^{-(1+\theta)}$ models the memory over time. We approximate user influence m using the number of followers obtained from Twitter API.

We can very well construct an exponential kernel $\phi^e(t_i) = e^{-\theta t_i}$, used in financial data [Filimonov and Sornette, 2015] or Rayleigh kernel $\phi^r(t_i) = e^{-\frac{1}{2}\theta t_i^2}$, used in epidemiology [Wallinga and Teunis, 2004]. We experimented with all three kernels, and found that the power-law kernel consistently outperform the other two at prediction performances. Consequently, we only discuss detailed results from the power law kernel in the rest of this chapter.

4.3.2 The branching factor

One key quantity that describes the Hawkes processes is its branching factor n^*, defined as the expected number direct offspring spawned by a single event. The branching factor n^* intuitively describes the amount of events to appear in the process, or informally, *virality* in the social media context. In addition, the branching factor gives an indication about whether the cluster of offspring associated with an immigrant is an infinite set. For $n^* < 1$, the process in a *subcritical regime*: the total number of events in any cluster is bounded as each event is associated with a finite cluster of offsprings, both in number and time extent. When $n^* > 1$, the process is in a so-called *supercritical regime* with $\lambda(t)$ increasing and the total number of events in each cluster being unbounded. We compute the branching factor of the marked Hawkes process constructed in Section 4.3.1 by taking expectations over both event times and event marks.

$$n^* = \int_1^\infty \int_0^\infty P(m)\phi_m(\tau)d\tau dm \ . \tag{4.4}$$

We assume that the event marks m_i are *i.i.d.* samples from a power law distribution of social influence [Kwak et al., 2010]: $P(m) = (\alpha - 1)\, m^{-\alpha}$. α is an exponent which controls the heavy tail of the distribution and it is estimated from a large sample of tweets. We extract the number of followers m for a large sample of users from our dataset described in Section 3.1 and fit it to a power-law distribution following the method detailed in [Clauset et al., 2009]. We obtain $\alpha = 2.016$ and we use it throughout the experiments. Substituting Equation (4.3) and $P(m)$ into (4.4), we obtain the closed-form expression of the branching factor:

$$n^* = \kappa \frac{\alpha - 1}{\alpha - \beta - 1} \frac{1}{\theta c^\theta}, \text{ for } \beta < \alpha - 1 \text{ and } \theta > 0 \ . \tag{4.5}$$

Not only the branching factor n^* is an intuitive descriptor of the *virality* of a cascade, it is also used for predicting the total popularity in Section 4.4 (illustrated in Figure 4.2) and as a constraint in model estimation in Section 4.3.3.

4.3.3 Estimating the Hawkes process

The log-likelihood function. Our marked Hawkes process is completely defined by its four parameters $\{\kappa, \beta, c, \theta\}$. In this section, we describe a maximum-likelihood estimation procedure of the model from a set of observed events upto time T, consisting of $\{(m_i, t_i), i = 0, \ldots, n\}$ for time $t < T$. The log-likelihood function of a point-process described by Equation (4.2) and (4.3) is written as ([Delay and Vere-Jones, 2003], Ch. 7.2):

$$\begin{aligned}\mathcal{L}(\kappa, \beta, c, \theta) &= \log P(\{(m_i, t_i), i = 1, \ldots, n\}) \\ &= \sum_{i=1}^{n} \log\left(\lambda\left(t_i\right)\right) - \int_0^T \lambda(\tau) \mathrm{d}\tau \end{aligned} \tag{4.6}$$

We start by computing the integral on the right hand side of Equation (4.6), denoted by Λ:

$$\begin{aligned}\Lambda &= \int_0^T \lambda(\tau)\mathrm{d}\tau \\ \overset{cf.\ 4.1}{=}\ & \int_0^{t_1} \overbrace{\sum_{t=0}^{0} \phi_{m_i}\left(t - t_i\right)}^{=0} \mathrm{d}t + \\ & \int_{t_1}^{t_2} \phi_{m_0}\left(t - t_0\right)\mathrm{d}t + \int_{t_1}^{t_2} \phi_{m_1}\left(t - t_1\right)\mathrm{d}t + \\ & \int_{t_2}^{t_3} \phi_{m_0}\left(t - t_0\right)\mathrm{d}t + \int_{t_2}^{t_3} \phi_{m_1}\left(t - t_1\right)\mathrm{d}t + \\ & \qquad\qquad\qquad \int_{t_2}^{t_3} \phi_{m_2}\left(t - t_2\right)\mathrm{d}t + \\ & \vdots \\ \Rightarrow \Lambda &= \sum_{i=1}^{n} \int_{t_i}^{T} \phi_{m_i}\left(t - t_i\right)\mathrm{d}t \end{aligned} \tag{4.7}$$

Introducing Equation (4.3) into Equation (4.7), we obtain:

$$\begin{aligned}\Lambda &= \sum_{i=1}^{n} \int_{t_i}^{T} \kappa\left(m_i\right)^{\beta} \frac{1}{\left(t - t_i + c\right)^{1+\theta}} \mathrm{d}t \\ &= \sum_{i=1}^{n} -\kappa\left(m_i\right)^{\beta} \frac{\left(t + c - t_i\right)^{-\theta}}{\theta} \Bigg|_{t_i}^{T} \\ \Rightarrow \Lambda &= \kappa \sum_{i=1}^{n} \left(m_i\right)^{\beta} \left[\frac{1}{\theta c^{\theta}} - \frac{\left(T + c - t_i\right)^{-\theta}}{\theta}\right] \end{aligned} \tag{4.8}$$

Hence Equation (4.6) can be written as,

$$\mathcal{L}(\kappa, \beta, c, \theta) = \sum_{i=2}^{n} \log \kappa + \sum_{i=2}^{n} \log \left(\sum_{t_i > t_j} \frac{(m_j)^{\beta}}{(t_i - t_j + c)^{1+\theta}} \right) - \\ - \kappa \sum_{i=1}^{n} (m_i)^{\beta} \left[\frac{1}{\theta c^{\theta}} - \frac{(T + c - t_i)^{-\theta}}{\theta} \right] \tag{4.9}$$

The first two terms in Equation (4.9) are from the likelihood computed using the event rate $\lambda(t)$, the last term is a normalizing factor from integrating the event rate over the observation window $[0, T]$.

Finding the maximum-likelihood solution. Equation (4.9) is a non-linear objective that need to be maximized. There are a few natural constraints for each of model parameter, namely: $\theta > 0$, $\kappa > 0$, $c > 0$, and $0 < \beta < \alpha - 1$ for the branching factor to be meaningful (and positive). Furthermore, while the supercritical regimes $n^* > 1$ are mathematically valid, it will lead to a prediction of infinite cascade size – a clearly unrealistic outcome. We further incorporate $n^* < 1$ as a non-linear constraint for the maximum likelihood estimation. Ipopt [Wächter and Biegler, 2006], the large-scale interior point solver can be used to handles both non-linear objectives and non-linear constraints. For efficiency and precision, it needs to be supplied with pre-programmed gradient functions.

Computation of partial derivatives for optimization

We present the partial derivatives calculation for Equation (4.9), used by the Ipopt function optimization procedure for fitting the parameters of the Hawkes self-exciting model to a given cascade.

Differentiating Equation (4.9) w.r.t κ we obtain:

$$\frac{\partial \mathcal{L}}{\partial \kappa} = \frac{n-1}{\kappa} - \sum_{i=1}^{n} (m_i)^{\beta} \left[\frac{1}{\theta c^{\theta}} - \frac{1}{(T + c - t_i)^{\theta} \theta} \right] \tag{4.10}$$

For differentiating $\mathcal{L}(\kappa, \beta, c, \theta)$ w.r.t β, c and θ, we first consider the 2^{nd} second term in the right hand side of Equation (4.9):

$$\sum_{i=2}^{n} \log \left(\sum_{t_i > t_j} \frac{(m_j)^{\beta}}{(t_i - t_j + c)^{1+\theta}} \right) = \\ \sum_{i=2}^{n} \log \left[(m_{i-1})^{\beta} (t_i - t_{i-1} + c)^{-(1+\theta)} + \right. \\ \left. \ldots + (m_1)^{\beta} (t_i - t_1 + c)^{-(1+\theta)} \right] \tag{4.11}$$

Using the logarithmic derivation rule and Equation (4.11) in Eq (4.9), we obtain the partial derivative w.r.t β:

$$\frac{\partial \mathcal{L}}{\partial \beta} = \sum_{i=2}^{n} \frac{\sum_{t_i > t_j} (t_i - t_j + c)^{-(1+\theta)} (m_j)^{\beta} \log (m_j)}{\sum_{t_i > t_j} (t_i - t_j + c)^{-(1+\theta)} (m_j)^{\beta}} \\ - \kappa \sum_{i=1}^{n} (m_i)^{\beta} \, log\, (m_i) \left[\frac{1}{\theta c^{\theta}} - \frac{1}{(T + c - t_i)^{\theta} \theta} \right] \tag{4.12}$$

Similarly, we compute the partial derivatives w.r.t c and θ:

$$\frac{\partial \mathcal{L}}{\partial c} = \sum_{i=2}^{n} \frac{\sum_{t_i > t_j} -(1+\theta) (m_j)^{\beta} \; (t_i - t_j + c)^{-(2+\theta)}}{\sum_{t_i > t_j} (t_i - t_j + c)^{-(1+\theta)} (m_j)^{\beta}} \\ - \kappa \sum_{i=1}^{n} (m_i)^{\beta} \left[\frac{1}{(T + c - t_i)^{1+\theta}} - \frac{1}{c^{1+\theta}} \right] \tag{4.13}$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{i=2}^{n} \frac{\sum_{t_i > t_j} - \log(t_i - t_j + c) \, (t_i - t_j + c)^{-(1+\theta)} (m_j)^{\beta}}{\sum_{t_i > t_j} (t_i - t_j + c)^{-(1+\theta)} (m_j)^{\beta}} \\ - \kappa \sum_{i=1}^{n} \frac{(m_i)^{\beta}}{\theta^2} \left[(\theta \log (T + c - t_i) + 1) (T + c - t_i)^{-\theta} - \right. \\ \left. (\theta \log (c) + 1) \, c^{-\theta} \right] \tag{4.14}$$

Existing literature [Mohler et al., 2011; Ogata, 1999; Delay and Vere-Jones, 2003] warns about three possible problems that can arise when using maximum likelihood estimates with Hawkes processes: edge effects, squared computational complexity and local minima. In this application, since we always observe a cluster of events generated by an immigrant, we do not have *edge effects*, i.e, missing events early in time. The *computational complexity* of calculating the log-likelihood and its gradients is $O(n^2)$, or quadratic with respective to the number of observed events. In practice, we use three techniques to make computation more efficient: vectorization in the R programming language, storing and reusing parts of the calculation, and data-parallel execution across a large number of cascades. With these techniques, we estimated tens of thousands of moderately-sized retweet cascades containing hundreds of events in reasonable amount of time. Lastly, the problem of *local minima* can be addressed using multiple random initializations.

4.4 Predicting Popularity

In this section, we first derive the number of expected future events in a cascade (Section 4.4.1). Next, in Section 4.4.2 we describe a predictive layer tuned for estimating the total size, using information from other cascades in history.

4.4.1 The expected number of future events

Once we have observed a retweet cascade until time T and fitted the parameters of the point process, we can can simulate a possible continuation of the cascade, until it dies out – assuming $n^* < 1$. But this is only one of the many simulations and hence can't be used to predict a unique (constant) size for the cascade. In order to predict a unique non-stochastic size of a cascade, we derive the expected number of events in the cascade, over all possible continuations. We group events based on the generation of their parents (i.e. the preceding event they are triggered by), and estimate the number of events in each generation.

We denote as $Generation_1$ the set of simulated events spawned by the observed events (shown in red in Figure 4.2). Similarly, $Generation_2$ (in green in Figure 4.2) groups events spawned by events in $Generation_1$, and so on. Let A_i be the expected number of events in $Generation_i$. The expected number of total events in the cascade, N_∞, is defined as:

$$N_\infty = n + \sum_{i=1}^{\infty} A_i \ , \tag{4.15}$$

where n is the number of observed events. For generations after the first one, the best estimate about $A_i, i \geq 2$ is constructed using the average number of offspring events n^* and the number

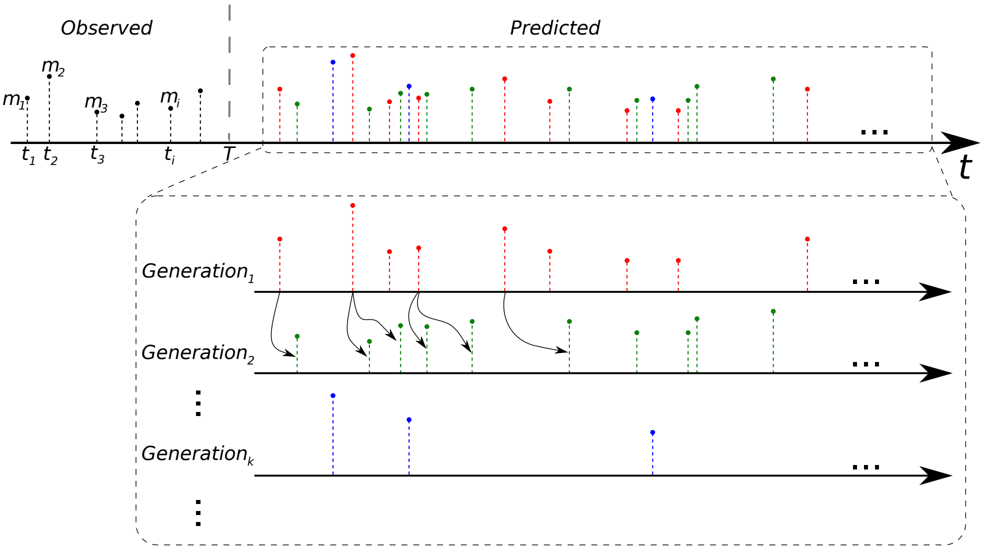


Figure 4.2: Illustration of the rationale behind popularity prediction. The model parameters are estimated starting from a series of observed events (m_i, t_i). Part of one possible unfolding of the diffusion cascade is simulated, using the event rate defined by Eq. (4.3) and simulated by thinning [Ogata, 1999]. The event generations are shown: events in *Generation*$_1$ are shown in red color, *Generation*$_2$ in green and *Generation*$_k$ in blue. Note there is no theoretical limit to the number of generations or to the extent of time until the cascade dies out. Some of the parent-offspring relations between events in consecutive generations are shown.

of events in the previous generation, i.e. $A_i = A_{i-1}n^*$. Assuming A_1 known, we derive:

$$A_i = A_{i-1}\, n^* = A_{i-2}\, \left(n^*\right)^2 = \ldots\ = A_1\, \left(n^*\right)^{i-1}, i > 1 \tag{4.16}$$

Therefore, the second term in (4.15) is the sum of a converging geometric progression (assuming $n^* < 1$):

$$\sum_{i=1}^{\infty} A_i = \frac{A_1}{1 - n^*} \text{ where } n^* < 1 \tag{4.17}$$

A_1 could also be calculated in a fashion similar to Eq. (4.16). However, a more precise estimation can be obtained, given that magnitudes of the observed events – parents for events in *Generation*$_1$ – are known:

$$\begin{aligned} A_1 &= \int_T^{\infty} \lambda(\tau) \mathrm{d}\tau = \int_T^{\infty} \sum_{t > t_i} \phi_{m_i}\left(t - t_i\right) \mathrm{d}t \\ &= \kappa \sum_{i=1}^{n} \frac{{m_i}^{\beta}}{\theta\left(T + c - t_i\right)^{\theta}} \end{aligned} \tag{4.18}$$

We obtain the estimate of the total number of events in the cascade by introducing (4.18) and (4.17) into (4.15):

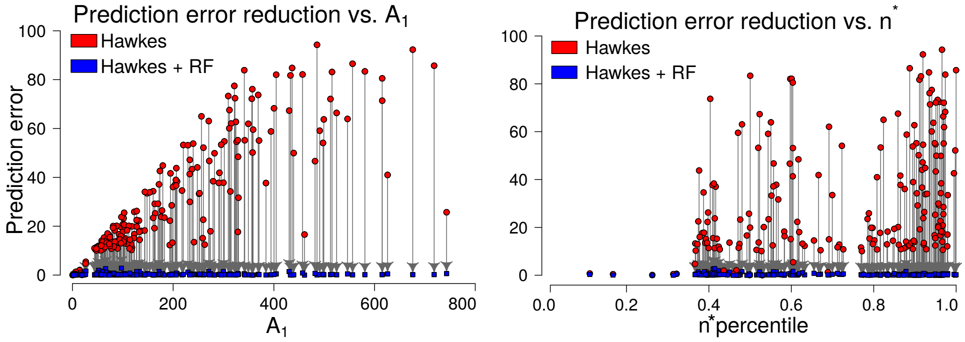
$$N_\infty = n + \frac{\kappa}{(1-n^*)} \left(\sum_{i=1}^{n} \frac{m_i{}^\beta}{\theta \left(T + c - t_i\right)^\theta} \right), n^* < 1 \tag{4.19}$$

4.4.2 Predicting total popularity

Using generative models for prediction is often sub-optimal, often due to simplifying assumptions of the generative process. Point processes in general, are optimized for explaining observed event history and not optimized for prediction. Zhao *et al.* [Zhao et al., 2015], for example, recently deployed a number of heuristic correction factors to discount the initial burst and account for a long-term decay. We consider that the Hawkes point process can benefit from systematically learning a predictor, to better account for data noise and limiting model assumptions. First, it is well know that using the number of followers as user influences m_i is at best an approximation [Cha et al., 2010] and often does not scale with retweeting propensity. Second, the network in which the cascades happen can be nonhomogeneous over the lifetime of the diffusion, e.g. users who respond early may have a inherently shorter memory than those responding late. Last, point processes cannot generate a meaningful prediction for supercritical cascades, and the model estimation can be prone to local minima. In this section, we leverage a collection of previously observed cascade histories to fine tune popularity prediction. This predictive layer accounts for the limitations above, and places point process estimates in a comparable framework as the feature-driven approaches.

Prediction setup. Each retweet cascade is described using four features $\{c, \theta, A_1, n^*\}$. These four features were found to be the most correlated with $\epsilon = (N_{real} - N_\infty)^2$, the error made when predicting popularity using N_∞, among all parameters, and derived quantities of the Hawkes model. Furthermore, n^* and θ are expressed as percentiles over the range observed in the training set, to account for their non-linear relation with ϵ. Note that parameters κ and β are not used: the information in κ is already included in A_1, whereas β was found non-correlated with ϵ.

In the classification task detailed in Section 4.5.3, a Random Forest classifier is trained to output the selected class. In the popularity prediction task – Section 4.5.1 and 4.5.2 – we train a Random Forest regression algorithm to output a scaling factor ω for the expected number of future events in a given retweet cascade.



(a)

(b)

Figure 4.3: Reduction of prediction error for a subset of cascades. For each cascade, the error made when predicting final popularity using the theoretical N_∞ is shown using red circles, and with blue squares the error made when using the predictive layer $\hat{N}_\infty$. Each gray arrow shows the reduction of error for a single cascade, and pairs together a red circle with a blue square. The error reduction is show in relation to A_1 (a) and n^* (b).

The final corrected popularity prediction is computed as:

$$\hat{N}_\infty = n + \omega \left(\frac{A_1}{1 - n^*} \right) \quad . \tag{4.20}$$

Effects of the predictive layer. The immediate effect of introducing the predictive layer is the reduction of the ARE prediction error, measured as detailed in Section 4.5.1. Fig 4.3 illustrates this reduction on a subset of the cascades in the News dataset – described in Section 3.1 – and its relation with A_1 and n^*. The length of the arrows is proportional with the error reduction. The range the errors made without the predictive layer increases linearly with A_1. Consequently the error reduction when using $\hat{N}_\infty$ is generally higher when A_1 is higher (Figure 4.3a). The relation between n^* and the prediction error is non-linear, therefore in Figure 4.3b the horizontal axis shows the percentile values of n^*. The variation of the error reduction has two maxima, one around 40% and another one around 95%. Figure 4.3 can be summarized as follows: the error reduction increases linearly with the expected number of events in $Generation_1$, but exhibits two maxima in relation with the expected number of events in $Generation_i, i \geq 2$. Another interesting observation is that N_∞ tends to over-estimate cascade sizes, in other words cascades tend to achieve less popularity than expected when observed in isolation. This aligns with the hypothesis that content items compete with each other for a finite amount of human attention [Miritello et al., 2013].

Table 4.2: (left hand side) Mean Average Relative Error (ARE) ± standard deviation obtained using Hawkes and Seismic on the Tweet-1Mo and News datasets. (right hand side) Number of cascades for which prediction failed.

		Prediction error: different time lengths			Number of failed cascades		
Dataset	Approach	5 minutes	10 minutes	1 hour	5 minutes	10 minutes	1 hour
Tweet-1Mo	Seismic	2.61 ±55.80	0.70 ±15.58	0.51 ±10.81	507	164	71
	Hawkes	0.36 ±0.52	0.33 ±0.41	0.30 ±0.38	302	105	58
News	Seismic	11.13 ±282.96	0.84 ±11.77	0.33 ±0.92	1022	155	123
	Hawkes	0.42 ±6.83	0.25 ±0.60	0.22 ±1.16	149	45	37

4.5 Experiments

We first present in Section 4.5.1 a thorough comparison between the Hawkes model and Seismic, the previous state-of-the-art generative model on popularity prediction, using two datasets. In Section 4.5.2 and Section 4.5.3 we compare the feature-driven classifier to generative approaches. Section 4.5.1 and Section 4.5.2 tackle regression tasks, i.e., predicting total cascade size after observing it for a certain time, while Section 4.5.3 tackles classification tasks, i.e., whether or not the cascade size will double from what has been observed.

4.5.1 Cascade size: two generative methods

We compare our point processes model, Hawkes, with Seismic [Zhao et al., 2015] as baseline on two datasets, Tweet-1Mo and News. Note that Tweet-1Mo is the dataset on which [Zhao et al., 2015] reported results, while News dataset supports extracting a richer set of features that is not available in Tweet-1Mo. In addition, [Zhao et al., 2015] showed that Seismic compared favorably against a number of baselines: from auto-regressive and generative models such as dynamic Poisson model [Agarwal et al., 2009; Crane and Sornette, 2008] and reinforced Poisson model [Shen et al., 2014], to linear regression and its variants [Szabo and Huberman, 2010].

Experimental setup. We run our experiments on the Tweet-1Mo dataset and a subset of News: 20,093 cascades from the month of July, which have the total length of at least 50. Both algorithms observe the initial part of each cascade for a limited extent of time and predict its final popularity. We estimate separate sets of parameters for Hawkes and Seismic after observing the cascades for 5 minutes, 10 minutes and 1 hour, respectively. After the fitting is finished, we train the Random Forest regressor in the predictive layer of the Hawkes model. Performance of this regression layer is reported using ten fold cross-validation. 40% of cascades are used for training, 60% for testing. All experimental protocol choices – minimum cascade length

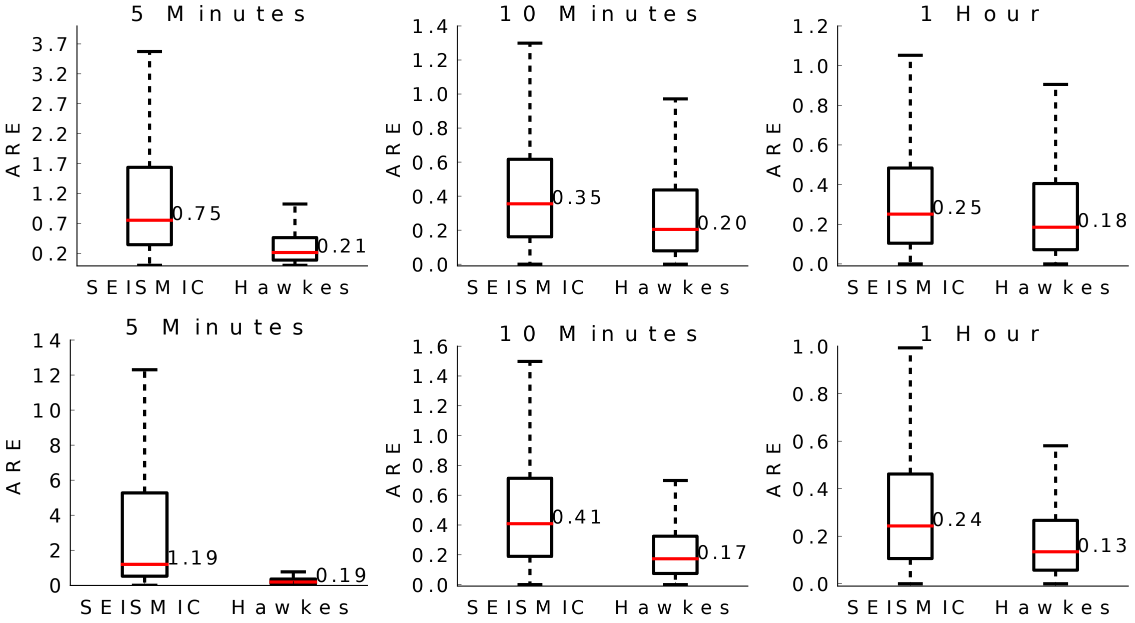


Figure 4.4: Distribution of Absolute Relative Error (ARE) on the Tweet-1Mo dataset (top row) and on the News (bottom row), for Seismic and Hawkes. A part of the diffusion cascade is observed before making the predictions: 5 min (left column), 10 min (middle column) and 1 hour (right column). The red line and the numeric annotations denotes the median values of the distributions.

threshold, random training/testing split and length of time prefixes – are made to mimic closely the original experimental setup [Zhao et al., 2015].

$$ARE^w = \frac{\left|\hat{N}^w_\infty - N^w_{real}\right|}{N^w_{real}} \quad ,$$

where $\hat{N}_\infty$ and N_{real} are the predicted and observed popularities of cascade w.

Results. The prediction results are summarized in Figure 4.4 as boxplots of ARE for each combination (dataset, approach, observed time), while Table 4.2 shows the mean ARE ± stan-

Table 4.3: Mean ARE ± standard deviation for Seismic, Hawkes, Feature-Driven and Hybrid models for the News July'15 dataset.

Approach	5 minutes	10 minutes	1 hour
Seismic	15.16 ±375.08	0.71 ±4.89	0.32 ±0.40
Feat.-Driven	0.25 ±0.18	0.22 ±0.17	0.17 ±0.14
Hawkes	0.27 ±1.83	0.22 ±0.80	0.17 ±0.36
Hybrid	0.17 ±0.16	0.15 ±0.14	0.11 ±0.12

dard deviation. Note that generative models do not output a prediction for all cascades. For example, when branching factor $n^* \geq 1$ for HAWKES the estimate for cascade size is infinite, similar failures also occur with SEISMIC when the infectiousness parameter $p \geq \frac{1}{n^*}$ ([Zhao et al., 2015] Eq(7)). Here we report the number of "failed" cascades in Table 4.2, and only report the average ARE on cascades that both approach produce a prediction. As shown in right hand side of the Table 4.2, we observe that HAWKES is able to produce meaningful predictions for more cascades than SEISMIC. For example, on NEWS data with 5 minutes, we output prediction for 873 (or 0.7%) more cascades than SEISMIC.

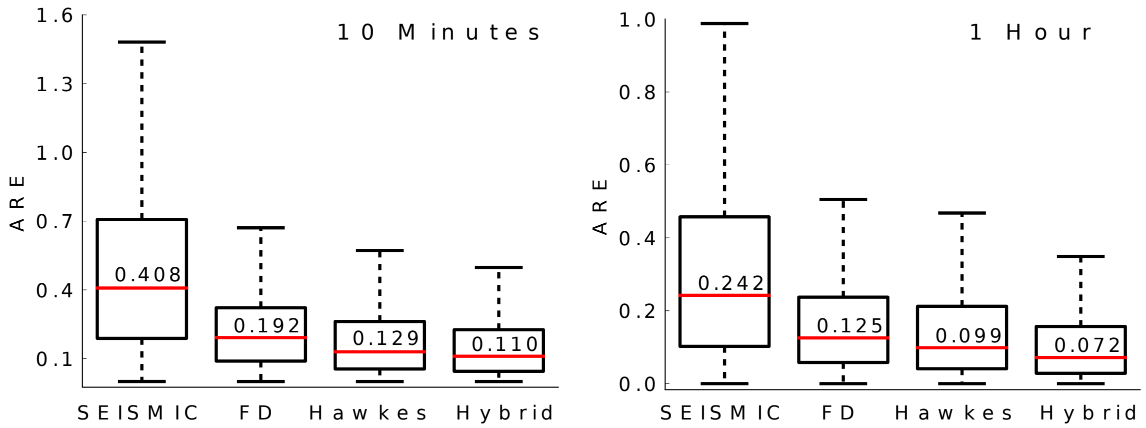
We can see that HAWKES consistently outperforms SEISMIC, as it achieves lower average ARE on both datasets. Mean ARE for HAWKES is at least 40% better than SEISMIC on TWEET-1MO and 33% on NEWS. We observe that NEWS dataset after 5 minutes of observation seems the most difficult to predict: HAWKES produces a mean ARE of 0.42, while the mean ARE for SEISMIC is 11.13. Figure 4.4 (top row) shows that the median of HAWKES is at least 25% better than SEISMIC for any time prefix, suggesting that our approach predicts better for most of the cascades in TWEET-1MO. Similar conclusions can be drawn from the bottom row of Figure 4.4 (NEWS data): the median ARE of HAWKES is at most 0.19 at 5 minutes, while the best of SEISMIC models is a median ARE of .24 at 1 hour.

We are glad to see that the HAWKES model achieves better predictive performance than SEISMIC, by exploiting full flexibility in its non-linear parameters, and a predictive layer that systematically optimizes for future prediction using information from other cascades.

4.5.2 Cascade size: generative vs. feature-driven

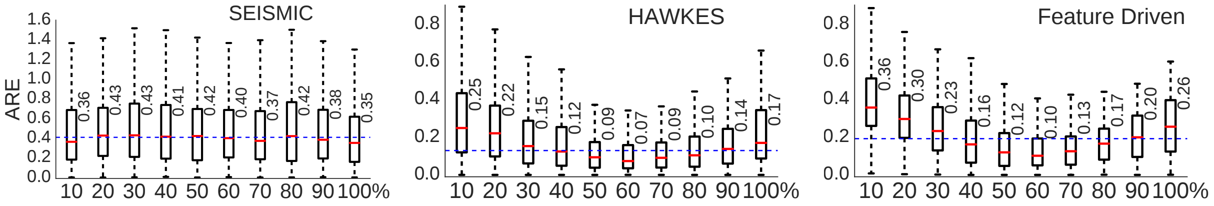
In this section we compare the performances of feature-driven and generative approaches on the NEWS dataset only, as the TWEET-1MO does not contain the necessary data to construct the features detailed in Section 3.1.2.

Experimental setup. Similar to the setup in the previous Sec 4.5.1, we observe cascades for 5 minutes, 10 minutes and 1 hour and fit the HAWKES and SEISMIC models for each cascade. The train-test split is different, in order to replicate closer the experimental setup in Martin *et al.* [Martin et al., 2016]: the data from first half of July (1-15) is used for training and the data from second half of July (16-31) for testing. We use the NEWS historical data from April to June to construct the past user success feature for the feature-driven approach. We consider only those users who initiated at least 2 cascades in the past as active users. Non-active users have a past user success of 0.



(a) (b)

Figure 4.5: Distribution of ARE on the News dataset, split in time for July, for Seismic, Feature-Driven, Hawkes and Hybrid, after observing 10 minutes (a) and 1 hour (b). The red line and the numeric annotations denote median value.



(a) (b) (c)

Figure 4.6: ARE distribution over popularity percentiles, on News dataset, after observing cascades for 10 minutes. The blue line denotes the overall median ARE shown in Figure 4.5a. Note that the y-axes are not on the same scale for all three graphs.

We also construct a Hybrid approach, which leverages the features of the Hawkes model – i.e. $\{c, \theta, A_1, n^*\}$ – together with the features detailed in Section 3.1.2. We use a Random Forest regressor for predicting the final volume size after tuning the predictor on the training set with 10 fold cross-validation.

Results. Similar to the previous subsection, we report in Table 4.3 the mean ARE ± standard deviation and in Figure 4.5 the box plots of ARE for 10 minutes and 1 hour. In according with the results of Section 4.5.1, Hawkes surpasses Seismic for all considered setups. Note that these results are numerically different from those in Sec 4.5.1, because the underlying test dataset consist only the subset from second half (16-31) of July 2015. Surprisingly, even the Feature-Driven model outperforms Seismic for all time prefixes, by at least 40%. In both cases, the differences are statistically significant ($p - val < 0.001$) for 10 minutes and 1 hour. The Hawkes

Table 4.4: Statistical testing results for SEISMIC, HAWKES, FEATURE-DRIVEN and HYBRID models for the NEWS dataset, indicating the algorithm with statistical significant better results. Win is decided when an algorithm has majority of lower error values than the competing algorithm and p-value confirms the results as statistically significant, else we mark it a draw in case of no statistical difference.

Approach	5 minutes	10 minutes	1 hour
SEISMIC vs. FEAT.-DRIVEN	Draw	FEAT.-DRIVEN	FEAT.-DRIVEN
SEISMIC vs. HAWKES	Draw	HAWKES	HAWKES
SEISMIC vs. HYBRID	Draw	HYBRID	HYBRID
FEAT.-DRIVEN vs. HAWKES	Draw	Draw	Draw
FEAT.-DRIVEN vs. HYBRID	HYBRID	HYBRID	HYBRID
HAWKES vs. HYBRID	HYBRID	HYBRID	HYBRID

model exhibits similar mean ARE as FEATURE-DRIVEN and higher standard deviation. However, its boxplot in Figure 4.5 shows lower median and quartiles. This indicates that HAWKES predict better than FEATURE-DRIVEN for most cascades, but the error distribution of HAWKES is skewed by outliers. HYBRID performs the best, with similar boxplot summarization as HAWKES, but with a 42% better mean APE – 0.17 to 0.11 for 1 hour. Results for statistical significance are presented in Table 4.4 for SEISMIC, HAWKES, FEATURE-DRIVEN and HYBRID models for the NEWS dataset, we use a strict cut-off p-value, 0.01, as our sample size is in order of thousands. As seen from Table 4.4 the HYBRID model shows statistical significant improvement over all other approaches almost every time. The results for HAWKES and FEATURE-DRIVEN are in general better than SEISMIC, but statistically insignificant with respect to each other. This result shows that HAWKES and FEATURE-DRIVEN approaches are complimentary, likely because they summarize the cascade information differently, and the errors are uncorrelated.

ARE distribution over cascade popularity. Figure 4.6 presents the ARE distribution over cascade popularity as boxplots for each popularity decile. Both HAWKES and FEATURE-DRIVEN predict better for most of the cascades with the middle popularity. For the extreme ends, performance is affected, because for lower popularity percentiles even a small error is amplified to a large ARE and cascades in higher popularity bins are inherently more difficult to predict. The prediction performance of SEISMIC seems consistent across popularity bins, but always worse than HAWKES or FEATURE-DRIVEN, as already showed by the aggregated results in Figure 4.5a. We also tried to explain prediction performance breaking down by NEWS sources, but we did not see any notable patterns.

Table 4.5: Accuracy ± standard deviation when predicting whether a cascade will double its size or not.

Approach	25 tweets	50 tweets
Random Guess	0.52	0.53
HawkesC	0.66 ±0.013	0.70 ±0.009
Feature-Driven	0.79 ±0.009	0.81 ±0.011
Hybrid	0.79 ±0.008	0.82 ±0.013

4.5.3 Will this cascade double

The learning problem in this section is to classify whether an observed cascade will double its volume or not. The experimental setup follows that of Cheng *et al.* [Cheng et al., 2014], in which cascades are observed for a fixed number of retweets, instead of for a fixed extend of time in Section 4.5.1 and 4.5.2. The proposed Hawkes model outputs directly the total popularity as a cascade size and uses the predictive layer to correct the theoretical estimate. As a consequence, it cannot be used in a classification setup. Instead, we construct a classifier HawkesC, in which a Random Forest Classifier is trained on the same features as the predictive layer of Hawkes and used to output a binary decision.

Experimental setup. We use the July subset of the News dataset, filtering to only cascades of length greater than or equal to 25. We observe the cascades until two different initial retweets, 25 retweets and 50 retweets, and we predict whether they will reach 50 and 100 retweets, respectively. We perform a 40%-60% stratified train-test split. Feature-Driven approach uses the features mentioned in Section 3.1.2, except for *volume* which is constant for all cascades. Hybrid approach combines features from both Hawkes and Feature-Driven. The random train-test split is repeated 10 times and we report mean accuracy and standard deviation.

Results. Table 4.5 summarizes the classification performances. A random guess (majority class) would output an accuracy of 52% and 53% for an observed length of 25 and 50 respectively. The generative-based classifier HawkesC improves substantially over the baseline of random guess, however Feature-Driven has the best prediction accuracy.

Interestingly, combining the generative features with the data-driven features did not lead to a significant improvement, likely due to Feature-Driven already has stronger results than HawkesC.

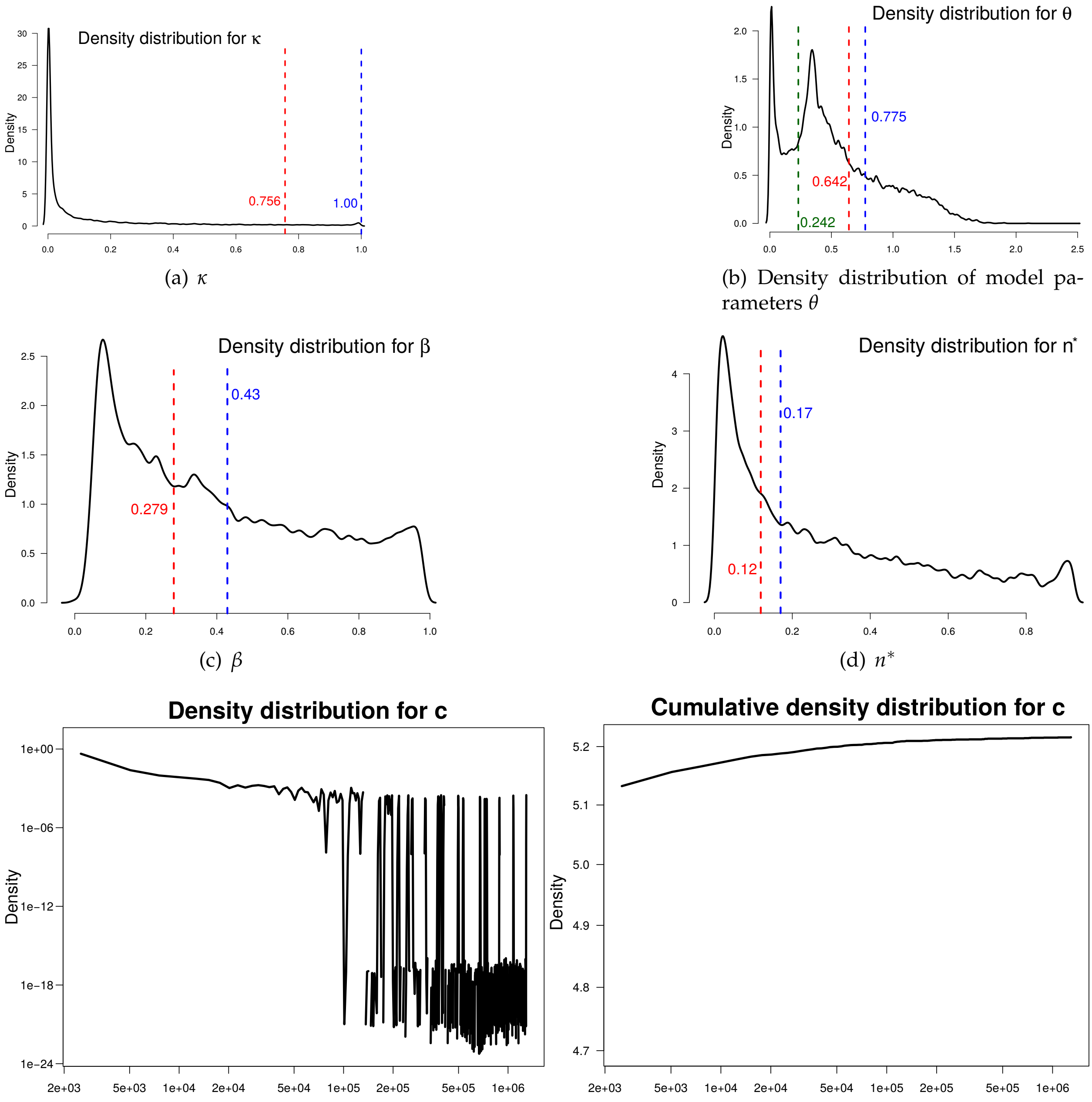
Admittedly, predicting whether a cascade will double in size is an easier problem than forecasting its final volume. Overall, the results suggests that the predictions are robust, and the performance are comparable to earlier results reported by Cheng *et al.* [Cheng et al., 2014] (0.81 accuracy after seeing the first 50 shares).

4.5.4 Interpreting the generative model

Individual cascades: fast vs slow. We first examine the model parameters for the two cascades illustrated in Figure 4.1. These two cascades are about the same event and have similar observed popularities (224 vs. 219) through very different diffusion speeds at a complex interplay between power-law memory $(\tau + c)^{-(1+\theta)}$ and user influences. For tw_1, the maximum-likelihood estimate of parameters are $\{\kappa = 0.75, \beta = 0.27, c = 58.46, \theta = 0.64\}$ with a corresponding $n^* = 0.12$; for tw_2, $\{\kappa = 1, \beta = 0.42, c = 27.77, \theta = 0.77\}$, and $n^* = 0.16$ (also shown in Figure 4.7). The two diffusions unfold in different ways: tw_1 has lower value of κ and higher waiting time c, hence its memory kernel $\phi(\tau)$ has lower values than that of tw_2 and a slower decay. However, tw_1 attracts the attention of some very well-followed music accounts. The original poster (`@screencrushnews`) has 12,122 followers, and among the retweeters `@TasteOfCountry` (country music) has 193,081 followers, `@Loudwire` (rock) had 110,824 followers, `@UltClassicRock` (classic rock) has 99,074 followers and `@PopCrush` (pop music) has 114,050 followers. The resulting cascade reached 1/4 its size after half an hour, and the final tweet was sent after 4 days. In contrast, tw_2's memory kernel has higher values, but faster decay. The most influential user in tw_2 has 412 followers, and the entire diffusion lasted for only 12.5 minutes.

Interpreting a collection of cascades. We use a large subset of 17,146 cascades of the NEWS dataset, described in Section 3.1. For each cascade we fit the parameters of its generative model $\{\kappa, \beta, c, \theta\}$. Figure 4.7 shows the density distribution of parameters κ, θ, $beta$, n^* and c. As one would expect, most cascades have a low value of κ, and reflect the long-tailed distribution of content quality. Parameter θ – which controls for the speed of the decay of the memory kernel $\phi_m(\tau)$ – has a more surprising distribution. Recent work [Zhao et al., 2015] on point processes with power-law kernels argued that a unique value of $\theta = 0.242$ is sufficient to explain the general behavior cascades. We find θ to vary in the interval $(0, 2.8]$ and showing two maxima at 0.01 and 0.34. This observation indicates that learning individual memory exponents θ allows us to better describe the diffusion dynamics of each cascade.

The density of β shows two local maxima: a maximum of density around 0.079 and a smaller peak around $\beta \sim \alpha - 1$ resulting from the optimizer terminating at the boundary of the non-linear constraint $n^* < 1$. n^* – descriptor of the virality of a cascade – shows a maximum around $n = 0.022$, and a smaller local maximum around 1 as optimization has a non-linear constraint



(a) κ

(b) Density distribution of model parameters θ

(c) β

(d) n^*

(e) c in log-log scale

(f) cumulative distribution of c in log-log

Figure 4.7: Distribution of parameter κ (a), θ (b) , β (c), n^* (d), c in log-log scale (e) and cumulative distribution of c in log-log (f), on a sample of 17,146 cascades from the NEWS dataset. The dashed vertical lines show fitted parameter values for the two retweet cascades illustrated in Figure 4.1: tw_1 in red and tw_2 in blue. We note that SEISMIC [Zhao et al., 2015] uses a fixed value of $\theta = 0.242$ for all cascades (denoted in green).

of $n^* < 1$. The low value of n^* signifies a low virality for most of the cascades. Parameter c is a cutoff parameter, its purpose is to keep the kernel function $\phi_m(\tau)$ bounded when $\tau \sim 0$. It controls the *reactivity* of the initial diffusion: small values lead to the first retweets being posted early after the initial tweet; large values introduce long waiting times between the first tweet and subsequent retweets. Figure 4.7(e) shows its long-tail density distribution in log-log scale and Fig 4.7(f) shows its cumulative density distribution. The large majority of cascades are very reactive, with a median value of $c = 111$ seconds.

4.6 Conclusion

In this chapter we proposed a Hawkes self-exciting model, which intuitively aligns with the social factors responsible for diffusion of cascades: social influence of users, social memory and inherent content quality. We systematically construct a predictive layer, which helps optimize predict from other cascades. We perform extensive evaluation on two large datasets: a benchmark dataset constructed by Zhao *et al.* [Zhao et al., 2015] and a domain-specific dataset curated on news tweets defined in Section 3.1.2.

We also proposed common benchmark that allows researchers and practitioners to compare feature-driven and generative approaches on different variants of popularity prediction problems. We compare the feature-driven approaches and generative approaches in two tasks: i) predicting the total size of retweeting cascades and ii) predicting whether cascades will double their size. Both our proposed generative HAWKES method and our FEATURE-DRIVEN method outperform the current state of the art predictor. Performances are further improved when combining both approaches into HYBRID, which makes us argue that popularity modeling should leverage the best that both worlds have to provide.

Despite the state of the art performance by our models, we still – (a) can not account for the limited size of community over which a cascades diffuses and (b) assume restrictive parametric form for the generative process. In our next Chapter 5 we formulate two advanced Hawkes models to tackle the specified problems.

The work presented in the chapter, evolved the field for popularity prediction by advocating use of feature-driven methodologies with generative models. For example, later work by [Li et al., 2017a] and [Cao et al., 2017] proposed a RNN based approach for predicting popularity by combining features proposed in the chapter with the generative RNN model. [Wu et al., 2018a] took the approach of bridging models further by using a GAN for popularity prediction, where the discriminator is a feature driven model inspired by features presented in the chapter and the generator is Hawkes model described for retweets in Section 4.3.1. Furthermore, work by [Chen et al., 2017], [Lu et al., 2018], [Li et al., 2017b] and [Chen et al., 2019] focused on formulating more sophisticated Hawkes models for modeling popularity in social media.

Chapter 5

Generalized Point Process Models

In Chapter 4, we presented a self-exciting Hawkes model (generative model) to capture the dynamics of a social cascade, and we showed that combining it with feature-driven methods provides the best performances of both generative and discriminative techniques. In this Chapter, we further enhance the Hawkes process to model mainly two important practical aspects of a social cascade – i) the limited population size to which a cascade can grow over time, and ii) the unknown underlying distribution of influence between events responsible for generating data. We propose two new models HawkesN in Section 5.1 accounting for limited size, and Recurrent Point Process (RPP) in Section 5.2 to tackle the restrictive expressive power of parametric generative processes.

5.1 HawkesN: Finite population model

In this Section, we will propose a finite population model, HawkesN, and we detail its parameter learning procedure. Finally, we show the effectiveness of HawkesN against the classical Hawkes model proposed in Chapter 4 on three real-world datasets.

5.1.1 Motivation

The Hawkes self-exciting point processes [Hawkes and Oakes, 1974] have shown good performance in modeling event sequences, as detailed in Chapter 4. But considering the nature of the spread of information on social media, some features make modeling with Hawkes processes susceptible. For instance, the Hawkes process has the property that the expected number of events triggered by a single event is static; i.e., it remains the same throughout the lifetime of the cascade. In the specific case of the model developed in Chapter 4, we extend the basic Hawkes model to make the expected number of events depend on the influence(magnitude) of the user. The proposed extension follows the ETAS [Ogata, 1999] model applied for modeling earthquakes. However, the stated change still does not account for the number of previous occurring events. In case of information cascades on social media, this assumption seems questionable as we only have a limited number of population to reach/affect. Early in the onset of any social diffusion, we would expect the rate of transmission to be much higher than the later part of the cascade, due to the decrease in population that can now be affected, and because some potential candidates have already been exposed. Thus, we introduce a new type of point process model, HawkesN, where the point process is still self-exciting with a component scaling down the excitement with an increase in the size of the cascade.

5.1.2 HawkesN: proposed finite population model

We generalize the Hawkes model [Hawkes, 1971] to account for finite population sizes. Intuitively cascades do not only follow the self-exciting word of mouth diffusions, but the size of the network or population over which they spread also limits them. We introduce the finite population size N so that we modulate the event rate at time t by the available population. To the best of our knowledge, no prior work on modeling social processes using Hawkes models had accounted for a finite underlying population.

The event rate function in this modified model, HawkesN, is defined as:

$$\lambda^H(t) = \left(1 - \frac{N_t}{N}\right) \left[\mu + \sum_{t_j < t} \phi(t - t_j)\right], \tag{5.1}$$

where $\phi(t - t_j)$ can be the same kernel function used with Hawkes, and N_t is the counting process associated with the point process. Both $\lambda^H(t)$ and N_t are right-continuous functions. The term $1 - \frac{N_t}{N}$ scales the event rate at time t with the proportion of the events which can still occur after time t.

When $t = 0$, we have $\lambda^H(t) = \mu$. When $N_t = N$, we have $\lambda^H(t) = 0$, i.e., there will be no more new events when the pool of users who can act is exhausted.

When $N \to \infty$, Eq. (5.1) simplifies to Eq. (2.5). In other words, the Hawkes process is a special case of HawkesN with an infinite population.

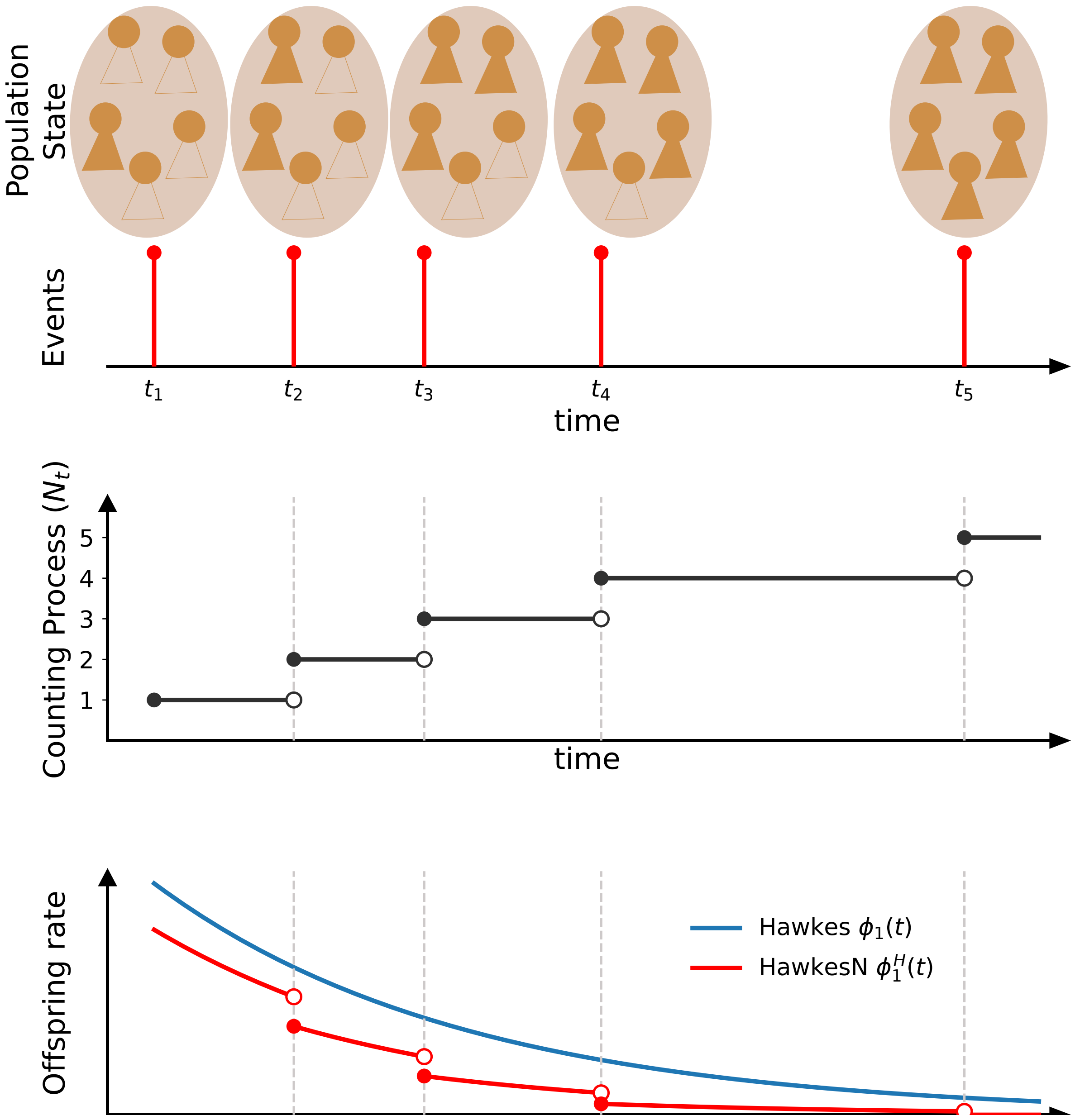


Figure 5.1: An example diffusion illustrating the finite population effects in self-exciting process. *(top panel)* An event refers to the j^{th} user taking an action at time t_j (e.g. posting a tweet). The state of the user population is shown at each time t_j: users *fully filled* have performed the past observed actions; users *partially filled* are yet to perform any action. *(middle panel)* The counting process N_t increases by one with each event; *(lower panel)* The *offspring rate* $\phi_1(t)$ – the rate of events generated by this first event at time t_1, modeled by Hawkes and by HawkesN (denoted by $\phi_1^H(t)$).

Fig. 5.1 illustrates the HawkesN process for an information diffusion in a population of five users ($N = 5$). Each user takes action at most once, represented as event time $t_j, j = 1..5$. The corresponding counting process N_t is shown in the middle plot. Events $t_2..t_5$ are considered to have been triggered by event t_1. The bottom panel compares the *offspring rates* – the rate of events generated by the first event at t_1 – for Hawkes (denoted as $\phi_1(t)$) and HawkesN (denoted as $\phi_1^H(t)$). In HawkesN, the population modulates the event rate by decreasing it after each new event, and the event rate becomes zero after t_5.

The Hawkes process does not take into account the population size, i.e., it will have $\phi(t) > 0$ in Eq. 2.5 even after t_5.

We use the exponential kernel function for HawkesN:

$$\phi(\tau) = \kappa m^\beta \theta e^{-\theta\tau} \tag{5.2}$$

κ is a scaling factor, m is the local user influence, β introduces a warping effect for the local user influence, and θ is the parameter of the exponential function which models the decay of *social memory*. The exponential kernel is a common choice in literature [Mishra et al., 2016; Zarezade et al., 2017; Zhao et al., 2015; Shen et al., 2014; Bao et al., 2015; Ding et al., 2015; Gao et al., 2015]. Other kernels have been used with Hawkes, including power-law functions [Helmstetter and Sornette, 2002; Crane and Sornette, 2008; Mishra et al., 2016; Kobayashi and Lambiotte, 2016] and Rayleigh functions [Wallinga and Teunis, 2004]. In our recent work, we extend HawkesN theory to show the equivalence of HawkesN with SIR in [Rizoiu et al., 2018].

Branching factor. We define the branching factor of HawkesN as the expected number of offspring events directly spawned *by the first event of the process*. For large values of N and fast decaying kernel functions $\phi(t)$, we can approximate $\frac{N_t}{N} \approx 0$ and therefore the branching factor for HawkesN is:

$$n^* \approx \int_1^\infty \int_0^\infty p(m)\kappa m^\beta \theta e^{-\theta\tau} d\tau dm = \kappa \frac{\alpha - 1}{\alpha - \beta - 1} \tag{5.3}$$

where $p(m)$ is the distribution of local influence that [Mishra et al., 2016] studied on a large sample of tweets, and found to be a power-law of exponent $\alpha = 2.016$. Although the branching factor of the Hawkes model in Section 4.4 and HawkesN are numerically identical, they represent different things. For Hawkes, the branching factor controls the size of the cascade whereas for HawkesN it indicates growth rate of the cascade as the final size of a cascade is determined/limited by N.

5.1.3 Fitting HawkesN to data

In epidemiology, the size of population N is usually considered a fixed known parameter – e.g., the number of people in a community. For online diffusions, it could be possible to estimate N from past diffusions by using the historical data of cascades which are similar to the one being estimated. However, in this section, we analyze the case when N is not known beforehand and needs to be estimated from the observed data.

5.1.3.1 The likelihood of HawkesN

Let $\{t_1, t_2, \dots, t_n\}$ be a set of event times assumed to have been generated from a HawkesN process described in Sec. 5.1.2 and each event time has a mark/magnitude m_i attached to it. When modeling diffusion cascades, it is typically assumed that every event apart from t_1 is a reaction to the first event, i.e., the background intensity is zero $\mu = 0, \forall t > 0$ [Mishra et al., 2016]. We estimate the remaining HawkesN parameters $\{\kappa, \beta, \theta, N\}$ by maximizing the log-likelihood function of the point process.

$$\mathcal{L}(\kappa, \beta, c, \theta) = \sum_{j=1}^{n} \log\left(\lambda^H\left(t_j^-\right)\right) - \int_0^{t_n} \lambda^H(\tau)\mathrm{d}\tau. \tag{5.4}$$

We further detail the integral term:

$$\begin{aligned}
\int_0^{t_n} \lambda^H(\tau)\mathrm{d}\tau &= \int_0^{t_n} \left(1 - \frac{N_{t^-}}{N}\right) \sum_{t_j < t} \phi(t - t_j)dt \\
&= \sum_{j=1}^{n-1} \int_{t_j}^{t_n} \left(1 - \frac{N_{t^-}}{N}\right) \phi(t - t_j)dt \\
&= \sum_{j=1}^{n-1} \sum_{l=j}^{n-1} \frac{N-l}{N} \int_{t_l}^{t_{l+1}} \phi(t - t_j)dt \\
\overset{cf.\ (5.2)}{} &= \kappa \sum_{j=1}^{n-1} (m_j)^\beta \sum_{l=j}^{n-1} \frac{N-l}{N} \left[e^{-\theta(t_l - t_j)} - e^{-\theta(t_{l+1} - t_j)}\right].
\end{aligned} \tag{5.5}$$

Eq. (5.4) is a non-linear objective and there are a few natural constraints for each of the model parameters, namely: $\theta > 0$, $\kappa > 0$, and $0 < \beta < \alpha - 1$ for the branching factor to be defined. We use the mathematical modeling language AMPL [Fourer et al., 1987], which offers a complete set of modeling tools, including automatic gradient computation and support for a large number of solvers. We choose as solver Ipopt [Wächter and Biegler, 2006], the state of the art optimizer for non-linear objectives.

Implementation details AMPL supports a comprehensive set of solvers including solvers for linear programming, quadratic programming and non-linear programming [Fourer et al., 1987]. This link[1] gives a full list of solvers for AMPL.

Solvers Applied in Implementation. We used two solvers in our fitting procedure:

- **LGO**: a *global optimizer* for non-linear problems, which is capable of finding approximate solutions when the problems have multiple local optimal solutions ([Pintér, 2013]). It is also one of the default solvers provided by AMPL.

- **IPOPT**: an open-source large-scale *local optimizer* for non-linear programming, which is released in 2006 [Wächter and Biegler, 2006].

Local solvers rely on improving an existing solution, employing complex techniques to avoid getting stuck in local minima. They require an initial point from which to start exploring the space of solutions. Global solvers attempt to search for the optimal solution in the entire space of solutions (one solution would be, for example, to divide the solution space into hyper-squares and apply local optimization in each one of them). Global solvers tend to find solutions which are not too far from the optimal, but they lack the precision of specialized local solvers In summary: local solvers achieve solutions very close to the optimal, but run the risk of getting stuck in horrible local optima; global solvers achieve imprecise solutions close to the optimal.

Optimization implementation setup. Our optimization setup is constructed to account for the weaknesses of each class of solvers. A classical solution to the problem of local optima with local solvers is to repeat the function optimization multiple times, from different starting points. We generate 8 random sets of initial parameters, within the definition range of parameters, and we use the IPOPT solver using each of these as the initial point. We also combine the global and the local solver: we use LGO to search in the space of solutions for an approximate solution, which we feed into IPOPT as an initial point for further optimization. Lastly, we run IPOPT without any initial parameters, leveraging IPOPT's internal strategy for choosing the starting point based on the parameters' range of definition. After completing these 10 rounds of optimization, we select the solution with the maximum training log likelihood values. We note the strategy of using a global solver like LGO with IPOPT has much higher time complexity than just running an extra run of IPOPT with a randomized initialization point. However, our experiments show that the combination of LGO with IPOPT yields the best results for fitting.

[1] `http://www.ampl.com/solvers.html`

Table 5.1: Datasets profiling: number of cascades and number of tweets tweets; min, mean and median cascade size.

	#cascades	#tweets	Min.	Mean	Median
ActiveRT [Rizoiu et al., 2017]	41,411	8,142,892	20	197	41
Seismic [Zhao et al., 2015]	166,076	34,784,488	50	209	111
News [Mishra et al., 2016]	20,093	3,252,549	50	162	90

5.1.4 Experiments

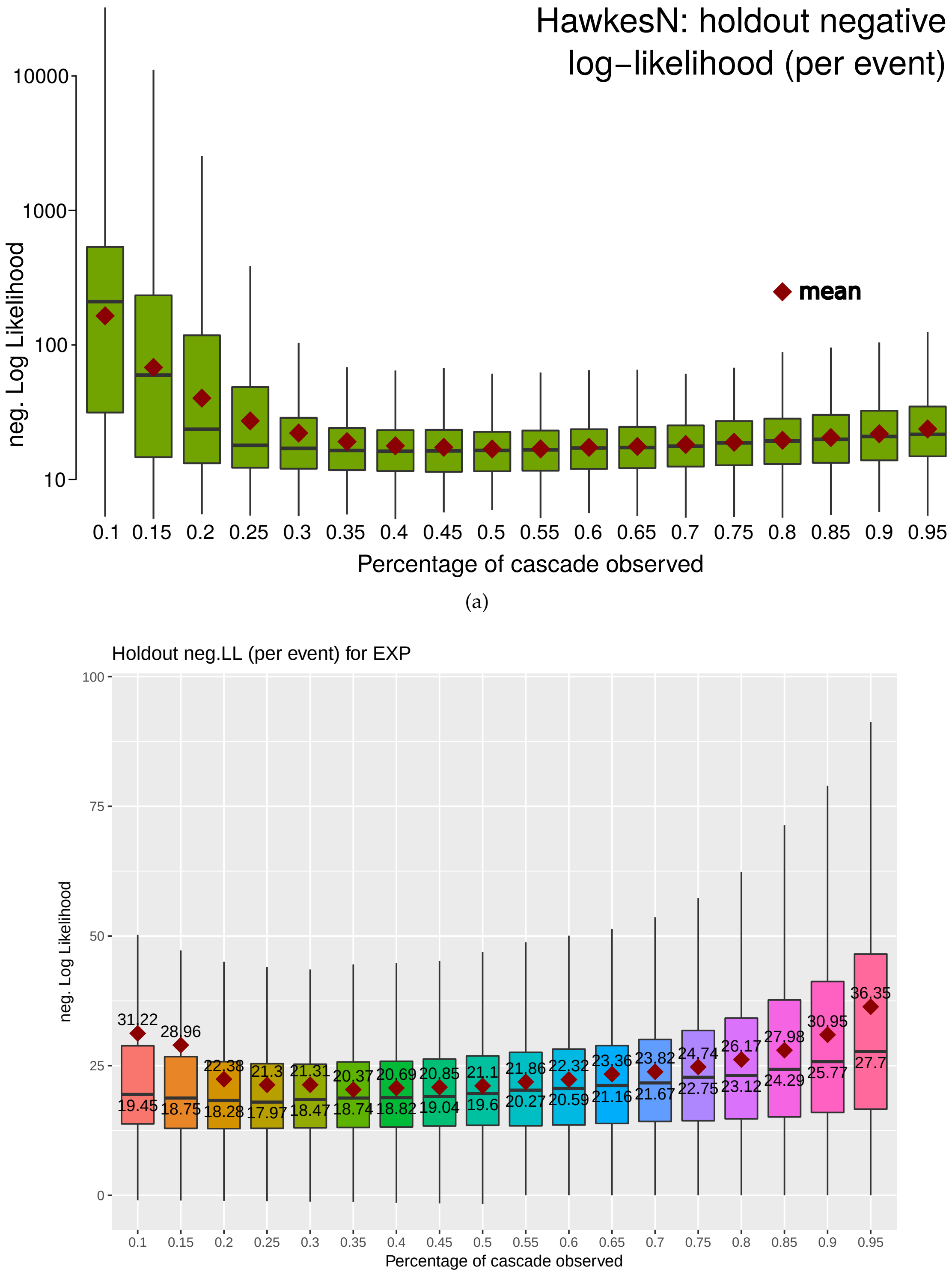
5.1.4.1 Datasets

We use three datasets of retweet diffusion cascades in Twitter, i) News [Mishra et al., 2016](Section 3.1), Seismic [Zhao et al., 2015] and ActiveRT [Rizoiu et al., 2017]. For each tweet in each cascade, we have information about the time offset of the retweet and the number of followers of the user posting the retweet. The ActiveRT dataset was collected by [Rizoiu et al., 2017] for 6 months in 2014. It contains more than 41k retweet cascades related to more than 13k YouTube videos, each cascade containing at least 20 tweets.

The Seismic dataset was collected by [Zhao et al., 2015]. It contains a sample of all tweets during a month (i.e., using the firehose Twitter API restricted access), further filtered so that the length of each cascade is greater than 50. The News dataset, already presented in Chapter 3, was collected over a period of four months in 2015 [Mishra et al., 2016]. It has tweets containing links to news articles, curated by tracking the official Twitter handles of popular news outlets, such as NewYork Times, or CNN. Each cascade contains at least 50 tweets. Table 5.1 summarizes these datasets.

5.1.4.2 Generalization to unobserved data

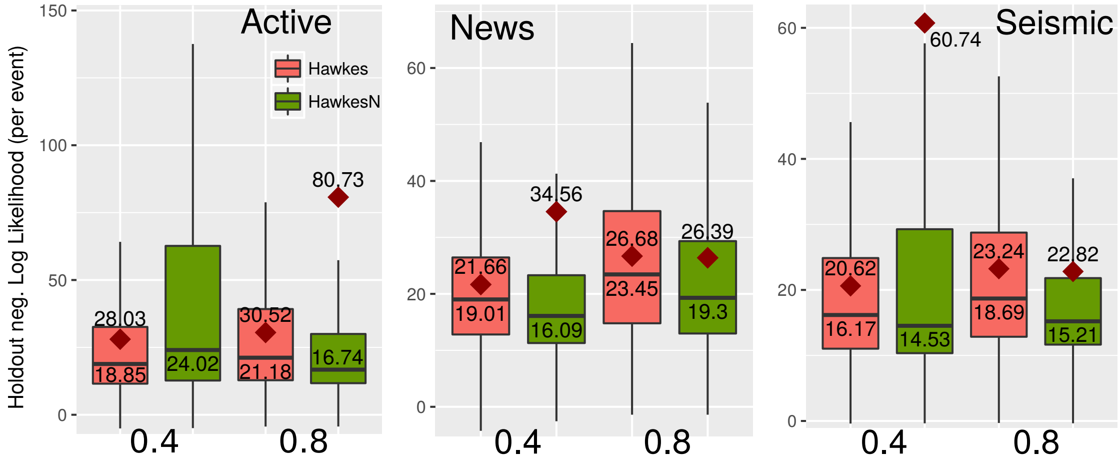
We empirically validate HawkesN by studying how it generalizes to unseen data. We compare HawkesN with the Hawkes model for information diffusion, proposed by [Mishra et al., 2016] and presented in Chapter 4 of this thesis. We adopt the setup in [Zhao et al., 2015; Shen et al., 2014; Bao et al., 2015; Ding et al., 2015; Gao et al., 2015; Rizoiu et al., 2017]: the first few events in diffusion are observed and used to fit the models. Hawkes is fitted as described in [Mishra et al., 2016] and in Section 4.3.3, and HawkesN is fitted as described in Sec. 5.1.3.1. For the latter, population size N is also fitted from data. We measure the holdout likelihood, i.e., the likelihood of the events in the unobserved period. The lower the negative holdout likelihood, the better the



(a)

(b)

Figure 5.2: Performances of HawkesN explaining unobserved data, using holdout negative log likelihood. The performance over all cascades in a dataset are summarized using boxplots, lower is better *(a) and (b)*. The percentage of observed events in each cascade used to train HawkesN and Hawkes respectively is varied between 10% and 95%. We use 1000 cascades randomly sampled from News.



(a) (b) (c)

Figure 5.3: *(a), (b) and (c)* The performances of HawkesN and Hawkes on all cascades of ActiveRT, Seismic and News, for Hawkes and HawkesN, when observing 40% and 80% of each cascade.

model generalizes to unseen data. We report the *per event* holdout negative likelihood, to render the results comparable across holdout sets containing different numbers of events. We chose to observe a given proportion of each cascade, to render the results comparable across cascades of different lengths.

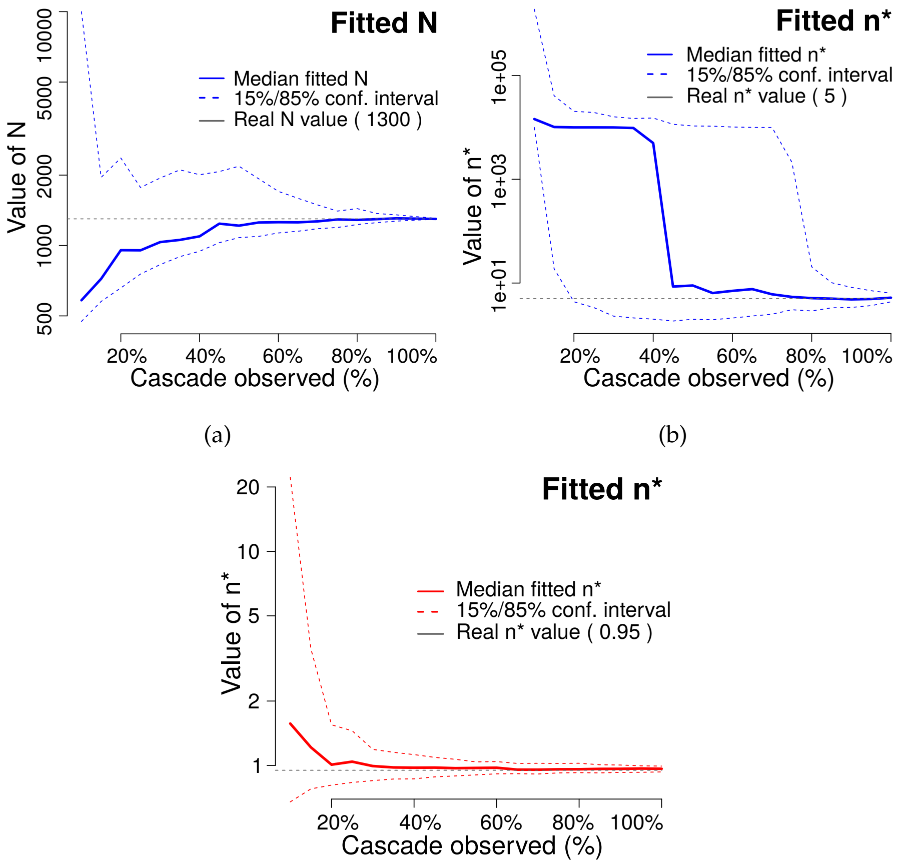
Figure 5.2(a) shows the generalization performances of HawkesN, when varying the percentage of observed events from 10% to 95%. We observe a high variance of performance when observing less than 40% of each cascade. The basic Hawkes model shows less variance at lower percentages as shown in Figure 5.2(b).

Plots (a) to (c) in Fig. 5.3 show the generalization performance of Hawkes and HawkesN, on the three datasets (ActiveRT, News, Seismic), for the observed percentages of 40% and 80%. Visibly, HawkesN has a consistently lower median value for the negative log-likelihood than Hawkes for higher observed percentages. The mean negative log-likelihood values are comparable for HawkesN and Hawkes on News and Seismic. On ActiveRT the mean of HawkesN is higher – likely due to YouTube videos behaving differently, with some old ones (e.g., Music) still being shared.

For higher observed percentages, the mean negative log-likelihood improves for HawkesN, and it degrades for Hawkes. The increase in performance with the increase in observed data indicates that the modulation factor $\left(1 - \frac{N_t}{N}\right)$ (in Eq. 5.1) helps improve the likelihood, and HawkesN scales better with the increase in the amount of the observed data.

5.1.4.3 Robustness of fit – additional graphics

As we could see from the high variance of estimates by HawkesN for smaller prefixes, a key question regarding the HawkesN process in the context of modeling information diffusion is the number of events in each cascade that need to be observed for an accurate estimation of the parameters. It is particularly important when the maximum number of events N is not known in advance and needs to be estimated from data.



(a)

(b)

(c)

Figure 5.4: Robustness of estimating the population size N and the branching factor n^* for HawkesN. One set of parameters for each model was simulated 100 times and fitted on increasingly longer prefixes of each simulation. One value for N and $n*$ is obtained for each fit and the median and the 15%/85% percentile values are shown. *(c)* Reliability of fit for the basic Hawkes model.

We answer this question with the help of simulated data. Starting from a set of parameters, we simulate 100 realizations. We fit HawkesN on increasing prefixes of each realization. In

Fig. 5.4, plots (a) and (b) shows the graphics for the branching factor and parameter N for HawkesN, respectively. For calibration, we perform the same exercise for the basic Hawkes Process, and we present the graphic for its branching factor in Fig. 5.4(c). The basic Hawkes requires observing less than 30% of the length of the cascade to make reliable estimates. Our proposed HawkesN model is more sensitive to the amount of available information and requires observing more than 40% of the cascade before the median n^* and N estimates approach the true values.

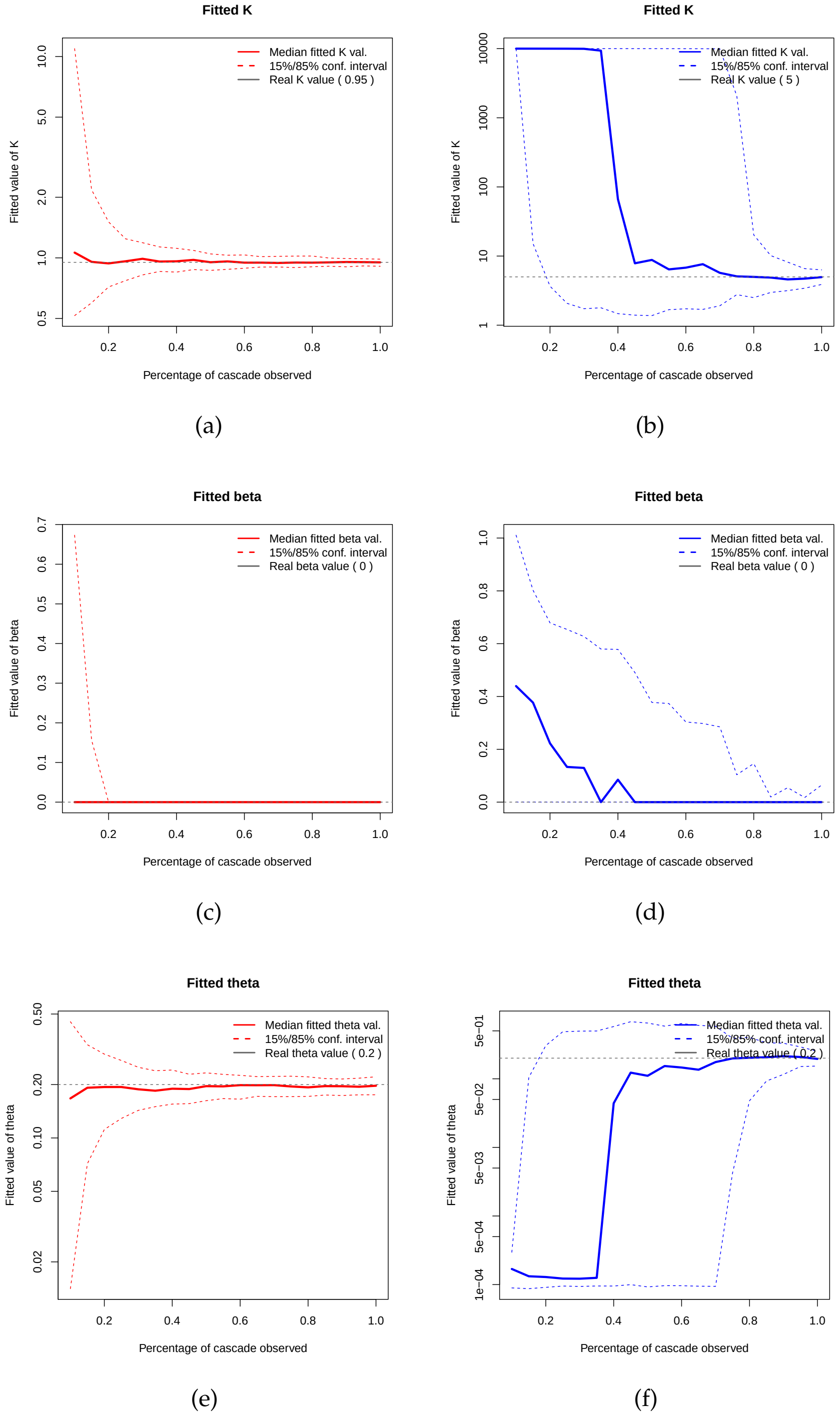
Fig. 5.5 shows the robustness of fit for parameters κ, β and θ for Hawkes *(a), (c) and (d)*, and HawkesN *(b), (d), and (f)*. The results for the other model parameters are analogous to the results for the branching factor and N, indicating HawkesN requires more data to estimate the mean parameters in comparison to Hawkes.

5.2 Recurrent point process

In this section, we first give our motivation for a recurrent neural network based point process model(Section 5.2.1). Next, we discuss our new model architecture and its learning procedure in Section 5.2.2 and 5.2.3, respectively. Finally, in Section 5.2.4 we presents results to prove the efficacy of our models by comparing them against current state-of-the-art models.

5.2.1 Motivation

Recently there are many machine learning based models for scalable point process modeling [Xiao et al., 2016; Farajtabar et al., 2015; Du et al., 2015]. The progress of event modeling in this direction is in part to the advanced mathematical reformulations and optimization techniques *e.g.*, [Lewis and Mohler, 2011; Zhou et al., 2013; Farajtabar et al., 2015], along with novel parametric forms for the conditional intensity function [Sornette and Helmstetter, 2003; Shen et al., 2014; Zhao et al., 2015; Mishra et al., 2016; Xu et al., 2016] as carefully designed by researchers' prior knowledge to capture the characters of the dataset in their study. In addition to prediction tasks, the model can uncover the hidden structure of information diffusion network or influence patterns. Although these models provide opportunities for interpretability, one major limitation of the parametric forms of point processes is due to their specialized and restricted expression capability for arbitrary and complex event data which tends to be oversimplified or even infeasible for capturing the problem complexity in real applications. Moreover, it is prone to the risk of under-fitting/over-fitting due to model misspecification.



(a) (b)

(c) (d)

(e) (f)

Figure 5.5: Robustness of estimating parameters κ, β and θ for Hawkes *(a), (c) and (d)* and HawkesN *(b), (d), and (f)*. One set of parameters for each model was simulated 100 times and fitted on increasingly longer prefixes of each simulation. One value for parameter is obtained for each fit and the median and the 15%/85% percentile values are shown.

In this section, we develop a point process representation with a non-parametric intensity function. We formulate the conditional intensity of a point process as a nonlinear mapping of the past events data. We hope this nonlinear mapping is flexible and complex enough for various real-world applications, *e.g.*, social media analysis, financial predictions, seismology, and many other situations. To overcome the disadvantages associated with the explicit parametric form of intensity function we use a neural network for modeling this non-linear nonparametric model.

Recurrent Neural Networks (RNN) [Elman, 1990] are common sequence models where the same feed-forward structure is replicated at each time step. They have additional connections from the output of previous the time step to the input of the current time step – therefore creating a recurrent structure. Their hidden state vector $\mathbf{h_t}$ can be defined recursively as:

$$\mathbf{h_t} = f\left(\mathbf{x_t}, \mathbf{h_{t-1}}\right)$$

where f is the feed forward network, $\mathbf{x_t}$ is the current input, $\mathbf{h_{t-1}}$ is the output from previous time step. Long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997; Graves, 2013] units are essentially recurrent networks with additional gated structure, defined as:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{V}_i\mathbf{c}_{t-1} + \mathbf{b}_i\right) \\
\mathbf{f}_t &= \sigma\left(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{V}_f\mathbf{c}_{t-1} + \mathbf{b}_f\right) \\
\mathbf{c}_t &= \mathbf{f}_t\mathbf{c}_{t-1} \odot \tanh\left(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_c\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{V}_o\mathbf{c}_t + \mathbf{b}_o\right) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right)
\end{aligned} \tag{5.6}$$

where $\mathbf{x}_t$ is input at time t, σ is the logistic sigmoid function and $\odot$ denotes element-wise multiplication. The notations $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$ and $\mathbf{h}_t$ stand for the input, forget, cell-state, output and hidden state at time t. We use the following short-hand notation for the LSTM set of equations in Equation (5.6):

$$(\mathbf{h}_t, \mathbf{c}_t) = LSTM(\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \tag{5.7}$$

LSTM and its variants have been successfully used for modeling time series and predicting sequence, due to their ability to capture the effects of past data in their hidden state [Hochreiter and Schmidhuber, 1997; Graves, 2013; Chung et al., 2014; Jain et al., 2016].

We build on this intuition to construct LSTM based point process models in Section 5.2.2.

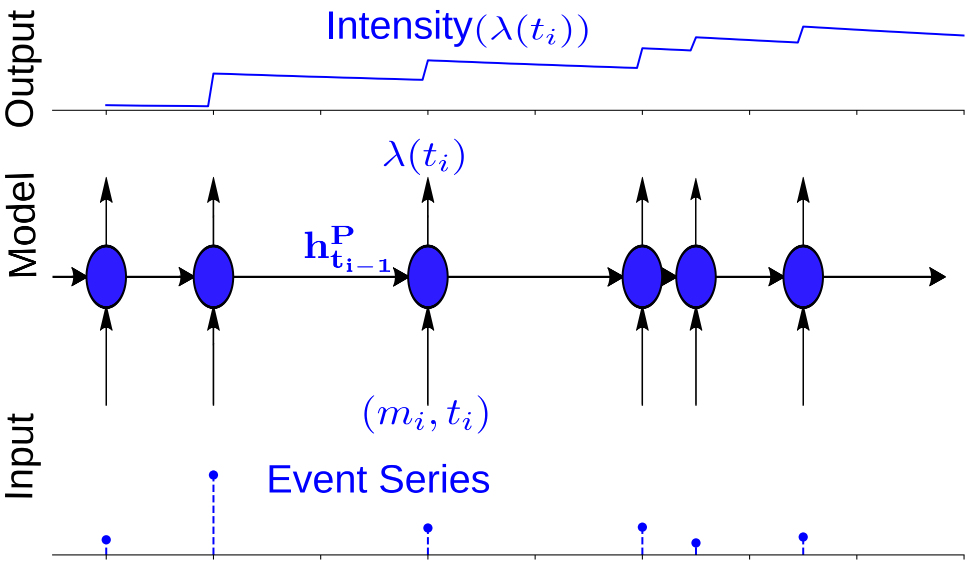

Output
Intensity$(\lambda(t_i))$
Model
$\lambda(t_i)$
$\mathbf{h}^{\mathbf{P}}_{\mathbf{t_{i-1}}}$
(m_i, t_i)
Input
Event Series

Figure 5.6: Recurrent models for events (RPP) Top row: output series; middle: block diagram for recurrent network, each circle refers to an LSTM unit; bottom: input series.

5.2.2 Recurrent Point Process (RPP)

We present a model to frame events data as point processes in Figure 5.6, where the influence of history on future events is learned as parameters of LSTM unit.

The equation for event rate at time t for the model is written as:

$$\lambda(t) = \exp\left(W_o^P \mathbf{h}_\mathbf{t}^\mathbf{P} + \alpha(t - t_i)\right) \tag{5.8}$$

such that $\mathbf{h}_\mathbf{t}$ is :

$$\mathbf{h}_\mathbf{t}^\mathbf{P} = LSTM\left((m_i, t_i), \mathbf{h}_{\mathbf{t_i}}^\mathbf{P}, \mathbf{c}_{\mathbf{t_i}}\right) \tag{5.9}$$

where t_i is the time for last event before time t, m_i is the magnitude of last event. In point process literature, the magnitude of an event is generally referred to as mark of an event. Mark represents attributes of the event, e.g. magnitude of the earthquake, or the type of a trade - bond or stock. In our work for modeling tweets we consider the mark as the magnitude (#followers) of user tweeting. $\mathbf{h}_{\mathbf{t_i}}$ and $\mathbf{c}_{\mathbf{t_i}}$ denotes hidden and cell state respectively at time t_i, and α is a scalar that modulates effect of current time on intensity.

Our model is similar to the one proposed by [Du et al., 2016]. In their work, they assume discrete values of marks as they model different event types rather than magnitude. We generalize their model to work for continuous marks by modeling mark generation separately from

Table 5.2: Summary of notations for RPP model

	Symbol	Quantity
Event	$e_i = (m_i, t_i)$	i^{th} event at time t_i of magnitude m_i
Data	$\lambda(t_i)$	Intensity at i^{th} event

history, as done in [Rizoiu et al., 2017; Mishra et al., 2016].

5.2.3 Model Learning

For fitting our model to a sequence of events $S = \{e_i\}$ where, e_i is the i^{th} event such that m_i and t_i stands for magnitude and time of the i^{th} event, we maximize the log-likelihood of all parameters in RPP, e.g., $\Theta^{RPP} = \{W_o^P, \mathbf{h_t^P}, \alpha\}$, for the observed sequence. Using Equation (5.8) and Equation (5.9), we can write log-likelihood for sequence as:

$$\begin{aligned} LL\left(\Theta^{RPP} \mid S\right) &= \sum_{i=1}^{|S|} log\left(\lambda\left(t_i\right)\right) - \int_0^{t_{|S|}} \lambda\left(\tau\right) d\tau \\ {}^{cf.\ (5.8)} &= \sum_{i=1}^{|S|} log\left(exp\left(W_o^P \mathbf{h_{t_i}^P} + \alpha\tau_i\right)\right) - \int_0^{t_{|S|}} exp\left(W_o^P \mathbf{h_t^P} + \alpha\tau\right) d\tau \\ \implies LL\left(\Theta^{RPP} \mid S\right) &= \sum_{i=1}^{|S|} \left[W_o^P \mathbf{h_{t_i}^P} + \alpha\tau_i + \frac{1}{\alpha}\exp\left(W_o^P \mathbf{h_{t_i}^P}\right)\right. \\ &\qquad \left. - \frac{1}{\alpha}\exp\left(W_o^P \mathbf{h_{t_i}^P} + \alpha\tau_i\right)\right] \end{aligned} \tag{5.10}$$

For predicting the time for the next event in sequence we can utilize the relationship between conditional density function of time, $f\left(t\right)$, and conditional intensity function (event rate), $\lambda\left(t\right)$ [Delay and Vere-Jones, 2003]:

$$f\left(t\right) = \lambda\left(t\right)\exp\left(-\int_{t_i}^{t} \lambda\left(\tau\right) d\tau\right) \tag{5.11}$$

In order to predict the next time step, $\hat{t}_{i+1}$, where t_i is event time for the last observed event, we take the expectation in Equation (5.11) as follows:

$$\implies \hat{t}_{i+1} = \int_{t_i}^{\infty} t \cdot f\left(t\right) dt \tag{5.12}$$

5.2.4 Experiments

Overview of methods We compare several approaches for predicting either the next event time in a series of events or the likelihood of a series of future (holdout) events:

- **RMTPP:** State of the art model based on recurrent neural networks for modeling events data with the help of point processes, proposed by [Du et al., 2016].
- **RPP:** Our model detailed in Section 5.2.2, modeling event data with LSTM.
- **PL:** Hawkes model proposed in Chapter 4 for modeling information diffusion. It utilizes a power-law kernel for modeling Hawkes process as detailed in Section 4.3.1. Showed state of the art performance for predicting popularity in [Mishra et al., 2016].
- **Exp:** Hawkes model where an exponential kernel is used, instead of a power-law kernel as done above in PL.
- **Seismic:** Seminal model based on self-exciting processes for predicting event data in social networks, proposed by [Zhao et al., 2015].

Dataset

We use two different datasets in our experiments.

RMTPP-Syn: synthetic data simulated with the same parameters as [Du et al., 2016]. It has 100,000 events from an unmarked Hawkes process [Hawkes and Oakes, 1974]. The conditional intensity for simulated data is given by:

$$\lambda\left(t\right) = \lambda_o + \alpha \sum_{t_i<t} exp\left(-\frac{t-t_i}{\sigma}\right)$$

where $\lambda_o = 0.2$, $\alpha = 0.8$ and $\sigma = 1.0$. We use 90% of events for training and rest for testing.

Tweet1MO: a real-world tweets dataset released by [Zhao et al., 2015], containing all tweets between October 7 to November 7, 2011.

Table 5.3 lists two prediction tasks, next event time and hold-out likelihood, against competing methods for the task. We present results for the next event time prediction on RMTPP-Syn as Tweet1MO has continuous marks that RMTPP can not model. Seismic cannot predict the next event time; hence we compare performance for hold-out likelihood on Tweet1MO.

We predict for the next event time for RMTPP, PL, Exp, and RPP using Equation (5.12). Figure 5.7 reports error in prediction as absolute error. We report the error for an oracle process

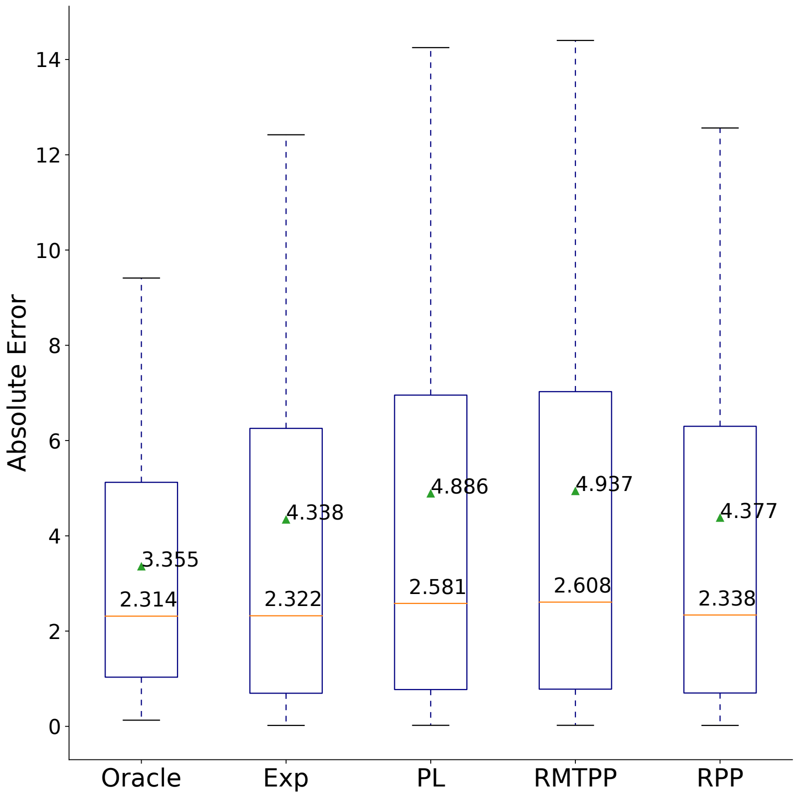


Figure 5.7: Error in estimated next event time for various models on RMTPP-Syn. RPP has better performance than state-of-the-art recurrent method RMTPP and performance comparable to Exp/Pl shows, RPP can capture real intensity without specification of true parametric form.

Table 5.3: Summary of methods and prediction tasks for event data modeling.

	Prediction Type	
Method	Next Event (RMTPP-Syn)	hold-out likelihood (Tweet1MO)
RMTPP	✓	✗
Seismic	✗	✓
PL	✓	✓
EXP	✓	✓
RPP	✓	✓

Table 5.4: Results showing mean(median) of negative log-likelihood per event for holdout set for Tweet1MO after observing a part of the cascade.

Observed Fraction	PL	Seismic	RPP
0.40	8.54(7.87)	7.62(6.98)	7.46(6.47)
0.50	8.38(7.96)	7.44(6.44)	7.26(6.39)
0.80	9.22(9.29)	8.02(7.3)	6.97(6.14)

that is set to the ground truth parameters generating data. Error for the oracle process is the difference between the expected values and simulated values. We observe that RPP has a lower mean and median error than RMTPP. RPP shows a similar mean and median as EXP, indicating RPP is flexible enough to capture the influence of past events on the future.

We compare the negative log-likelihood on hold-out data for RPP and alternatives. We vary the fraction of observed events to report results on Tweet1MO in Table 5.4, lower is better.

We observe that RPP performs better than PL and Seismic for all observed fractions. More importantly, the performance of RPP increases consistently with an increase in the observed fraction, indicating better generalization towards data.

5.3 Conclusion

We presented an extension to the Hawkes model, HawkesN, to account for the finite population. We showed through experiments that our proposed model helps us to get better generalizations and explainability over cascades of longer length. We also showed that the finite population model is a generalization over the Hawkes model when we assume an infinite population available for diffusion to grow. Furthermore, in our later work [Rizoiu et al., 2018], we establish a novel connection between HawkesN and Susceptible-Infected-Recovered (SIR) models; generally, these models are considered to have roots in different paradigms. Namely, we show that the rate of events in HawkesN is equivalent to the rate of new infections in SIR after marginalizing the recovery events.

We also presented another extension to Hawkes processes, RPP, which models the intensity of the Hawkes process with the help of LSTM (recurrent neural networks). In RPP instead of assuming a parametric shape for the influence of events on each other, we learn it directly from the historical data. Through empirical evaluations, we show significantly improved prediction and generalization ability of our models over many non-trivial baselines and state-of-the-art recurrent neural network based RMTPP [Du et al., 2016] model. In Chapter 6, we utilize RPP to build a novel model that can handle multiple heterogeneous unaligned streams of information for modeling popularity.

Chapter 6

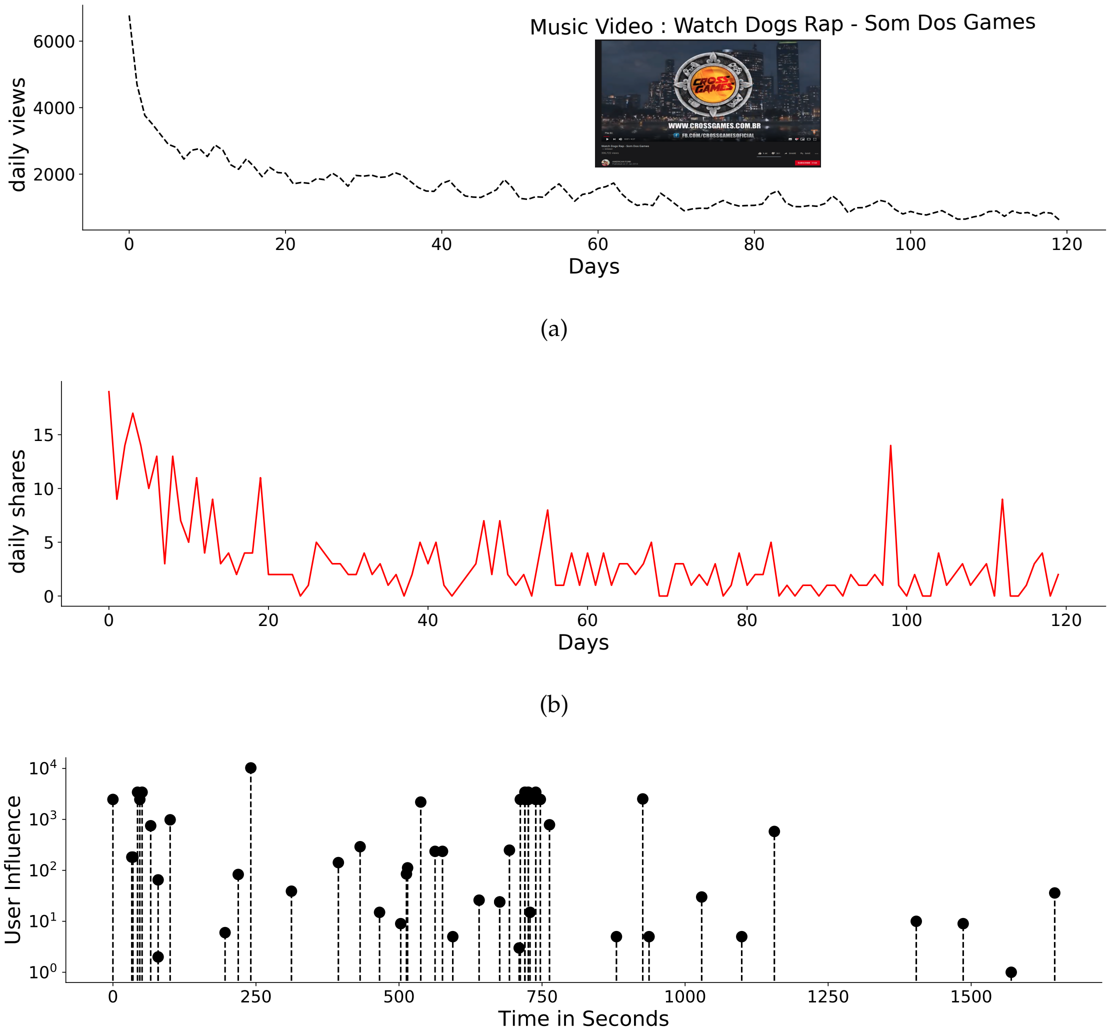
Reccurent Models for Multiple Asynchronous Streams

In Chapter 4 and Chapter 5 we presented models for predicting popularity for an event stream when individual actions are available. However, for many types of online content such a YouTube videos we have access to only aggregated data over a specific period (daily or hourly) instead of individual asynchronous actions due to restrictions in user privacy and data volume. Moreover, for online content such as YouTube videos or news articles, attention is driven by multiple heterogeneous sources simultaneously – e.g. microblogs or traditional media coverage. Here, in Section 6.4 we propose RNN-MAS, a recurrent neural network for modeling asynchronous streams. We further define two new metrics in Section 6.5: the *promotion score* quantifies the gain in popularity from one unit of promotion for a Youtube video; the *loudness level* captures the effects of a particular user tweeting about the video. Finally, in Section 6.6 and Section 6.7 we present the results for RNN-MAS model. Section 6.6 describes the results for predictive evaluation against the baselines. Whereas, in Section 6.7 we utilize two above mentioned metrics to understand evolution of popularity and compare the effects of a video being promoted by a single highly-followed user (in the top 1% most followed users) against being promoted by a group of mid-followed users.

6.1 Introduction

Successful recent models of popularity fall into two categories. The first describes individual user actions, or discrete events in continuous time (e.g. tweets) [Du et al., 2016, 2013; Mishra et al., 2016; Shen et al., 2014; Yu et al., 2017; Zhao et al., 2015]. The second is based on aggregate metrics of user actions [Cheng et al., 2014; Martin et al., 2016] or aggregated event volumes (e.g.

the number of daily views) [Szabo and Huberman, 2010; Pinto et al., 2013; Rizoiu et al., 2017; Yu et al., 2015].



(a)

(b)

(c)

Figure 6.1: Three different types of data streams associated with the YouTube video *Watch Dogs Rap - Som Dos Games*. (a) and (b) shows the daily views and daily shares gathered by the video, respectively. (c) shows a snapshot of individual tweets talking about the same video in its first 25 minutes after being uploaded.

Each of these models specialize in a distinct data type, but it is common to observe data of different types for the same online item. For example, Figure 6.1 shows three different type of data streams associated with a YouTube video titled *Watch Dogs Rap - Som Dos Games*. Figure 6.1(a) and Figure 6.1(b) shows daily views and daily shares respectively, received by the video in its first 120 days once uploaded on YouTube. Whereas, Figure 6.1(c) shows individual tweets received by the video in its first 25 minutes. Hence in oder to accurately capture the popularity of a video it is desirable to develop a model that accounts for multiple heterogeneous series.

Furthermore, many popularity models provide black-box predictions [Zhao et al., 2015; Martin et al., 2016; Mishra et al., 2016]. In practice one often demands simulations on various *what-if* scenarios, such as to quantify the effect of a unit amount of promotions, to capture seasonality or the response to outliers, to name a few. Lastly, the influence users have on popularity has been subject to constant debate in this research area. The view that one or a few influential champions can make or break a cascade [Budak et al., 2011] contrasts with the view that popularity largely results from a large number of moderately influential users [Bakshy et al., 2011]. It is desirable to have one model on which the future effect of different users can be comparably studied.

In this chapter we explain and predict the popularity of an online item under the influence of multiple external sources, in different temporal resolutions – such as both promotion events (e.g. tweets) and volumes (e.g. number of shares per day). In particular, we propose RNN-MAS (Recurrent Neural Networks for Multiple Asynchronous Streams), a flexible class of models learnable from social cascades that can describe heterogeneous information streams, explain predictions, and compare user effects for both individuals and groups. Recurrent neural network is an effective tool for sequence modeling in natural language and multiple other domains [Elman, 1990; Graves, 2013; Sutskever et al., 2014]. We link multiple recurrent neural networks by allowing them to exchange information across different asynchronous streams. This is an extension to the recent RMTPP [Du et al., 2016] and RPP developed in Section 5.2, for a single social event sequence. We illustrate the effectiveness of this model for predicting popularity of YouTube videos under the influence of both tweeting events and sharing volumes – improving state of the art prediction by 17%.

We propose several new ways to interpret and simulate popularity, and implement them for RNN-MAS. The first is a *unit promotion response* metric, that measures the gain in popularity per unit of promotion. Measured at different times and promotion scales, it can describe the time-varying and nonlinear effect of online promotions. The second measure, *unseen response*, captures the effect of unobserved external influence. Since neural nets is a flexible function approximator, we show that this measure can successfully capture seasonal effects. To understand the influence of users, we compute a new metric, *loudness level*, as the log-ratio of marginal gain from users. It is used to quantify the popularity gain from powerful users and moderately influential groups of users for each video. We observe that superusers are more effective than a cohort of regular users for only a minority (37%) of *Nonprofit and activism* videos, whereas superusers dominate in other video categories such as *HowTo & style and Gaming*.

The main contributions of this chapter include:

- RNN-MAS, a new and flexible model that links multiple asynchronous streams for predicting online popularity.
- New measures, *unseen response* and *promotion score*, to quantify content virality and explain different factors that affect popularity.
- A method for quantifying Twitter user influence on disseminating content on YouTube, proposing a new *loudness* metric and a set of observations across diverse content types on the relative influence of superusers versus everyday users.

6.2 Background

Our proposal in the chapter lies at the intersection of two distinct bodies of literature: Hawkes intensity process (HIP) [Rizoiu et al., 2017] and Recurrent neural networks [Elman, 1990; Hochreiter and Schmidhuber, 1997; Graves, 2013]. In this section we will briefly discuss HIP, and for basics of Recurrent Neural Networks please refer Section 5.2.1 in Chapter 5.

Hawkes Intensity Processes (HIP)

HIP was proposed by [Rizoiu et al., 2017], to model the evolution of popularity under a continuous external promotion. HIP extended Hawkes process [Hawkes, 1971] to model attention at the collective-level, by taking expectation over individual events, denoted as $\xi[d]$. It describes the volume of attention series $\xi[d]$ as a self-consistent equation:

$$\xi[d] = u[d] + \alpha s[d] + C \sum_{\tau=1}^{d} \xi[d-\tau](\tau + c)^{-(1+\theta)} \tag{6.1}$$

where $s\,[d]$ and $u\,[d]$ are respectively the external promotion and the unseen influence series on d^{th} day; α is the sensitivity to exogenous promotions; θ modulated the power-law memory kernel, C scales with content quality and c is a threshold parameter to keep the power-law kernel bounded when $\tau \simeq 0$. Equation (6.1) describes the volume volumes of attention over fixed time intervals (e.g. daily), hence can be used to model popularity when only aggregated data is available for online content like views and shares of a YouTube video; unlike Hawkes Processes that require individual events for model estimation.

Still HIP suffers from two drawbacks when modeling popularity. First, analogous to the Hawkes process, HIP also requires specifying a parametric kernel (which is assumed to be Power-Law in the original work). Second, HIP models volumes of data over fixed time intervals, even when detailed information about individual external promotion events $s[d]$, is available – e.g. which and when an user tweeted a video. The models we propose in Section 6.4.1 and Section 6.4.2 tackle these challenges via joint inference of the volume series and point process series.

6.3 Problem Statement

In this chapter, we aim to model the popularity of YouTube videos. The popularity of a video is measured as the total number of views gathered by the video on day d, denoted as V_d. We assume the popularity of a video to be externally driven by promotions on other platforms [Rizoiu et al., 2017]. We capture the external promotions in two stages: (i) as daily shares received by a video on YouTube, and (ii) as tweets received by a video on Twitter. We note collecting an extensive set of external promotions is practically impossible both due to size and distributed nature of Internet. Daily shares of a video from YouTube platform partially accounts for the external promotions for a video on different platforms and tweets provides us full coverage of promotion around a video on Twitter. We note, our Active'14 dataset has access to both external promotions as discussed in Section 3.2.2. In our work, we denote total daily promotions as $\mathbf{s}_d$ for total number of shares ("and" or "or" tweets) on day d. The individual tweets are denoted as per the terminology presented in Table 4.1. Our aim is to predict the total number of views gathered by a video in a period between days d_i and d_j, after observing the video for a specific period of $[d_0, d_{i-1}]$ days. Table 6.1 summarizes the terminology used in the chapter.

6.4 RNN for Volume Prediction

In the following section we first introduce a recurrent model for a volume series in Section 6.4.1. Next in Section 6.4.2, we present a joint model for volumes and event series by combining

Table 6.1: Summary of notations for RNN models

	Symbol	Quantity
Event	$e_i = (m_i, t_i)$	i^{th} event at time t_i of magnitude m_i
Data	$\lambda(t_i)$	Intensity at i^{th} event
Volume	s_d	Exogenous stimulus at fixed interval d
Data	V_d	Attention Volume at fixed interval d

volume models in Section 6.4.1 with event series model (RPP) developed in Section 5.2. Finally in Section 6.4.3 we present our learning approach for the aforementioned models.

6.4.1 Volume RNN

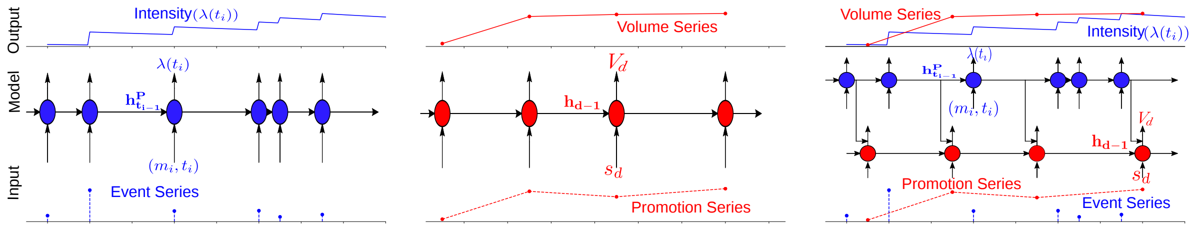
Intuitively, the Equation (6.1) for HIP can be seen as a generalization of an autoregressive model, with a history size as long as the volume series being modeled. We can describe the Equation (6.1) as a linear system where current value of volume intensity $\xi[d]$ is a convolution of its past values ($\xi[d_i] \forall d_i < t$) with a memory kernel. As the attention volume modelled in Equation (6.1) is a discretization over fixed intervals, the problem of predicting volume can be seen as a sequence prediction problem, where the next element in a sequence is predicted based on the history of the all the elements in the sequence. Hence to mitigate the problem of assuming fixed parametric kernel in HIP, we formulate a LSTM based model in Figure 6.2(b), for predicting attention volume series as follows, using LSTM notation introduced in Equation (5.7):

$$\hat{V}_d = W_o^{Vol} \mathbf{h_d} \tag{6.2}$$

such that,

$$\mathbf{h_d} = LSTM(\mathbf{s_d}, \mathbf{h_{d-1}}, \mathbf{c_{d-1}}) \tag{6.3}$$

where $\hat{V}_d$ is the predicted attention volume at time d (we $\xi[d]$ in HIP denotes same quantity) , $\mathbf{h}_d$ is the hidden state at time d and input to the system is $\mathbf{s}_d$ at time d, where s_d is the exogenous stimuli for the attention volume at time d. We note unlike the continuous time points in RPP model proposed in Section 5.2, the time points for Volume RNN are discrete, hence Volume RNN is more closer to classical sequence models used in literature [Elman, 1990; Hochreiter and Schmidhuber, 1997; Graves, 2013; Chung et al., 2014] than RPP. The LSTM based formulation allows us to learn any linear or non-linear convolution of history instead of a pre-defined Power-Law decay as done in HIP.



(a) (b) (c)

Figure 6.2: Recurrent models for events and volumes. (a) RPP (Figure 5.6), (b) VolRNN-S/TS and (c) RNN-MAS. Top row: output series; middle: block diagram for recurrent network, each circle refers to an LSTM unit; bottom: input series.

6.4.2 RNN-MAS for asynchronous streams

In real world scenarios, we have multiple streams of data promoting a single attention series. For example, being tweeted and being shared both lead to more views on a YouTube video. In this section we present a combined model to take into account the volume promotions and individual events for predicting an attention volume series.

RNN-MAS model is shown in the Figure 6.2(c). It has two components: a volume RNN model as shown in Figure 6.2(b) predicting the desired attention volume series at fixed time intervals by taking into account the promotional volume series; a second component as RPP model(Section 5.2) as shown in Figure 6.2(a), responsible for modeling the individual promotion series. Intuitively we would like RPP to modulate the response of volume RNN, to make better predictions for attention series. We achieve it by combining the hidden states for individual models just before making the prediction for attention series.

We combine individual models as follows:

$$\hat{V}_d = W_o^{MAS}\left[\mathbf{h_d}, \mathbf{h_d^P}\right] \tag{6.4}$$

where, $\mathbf{h_d}$ and $\mathbf{h_d^P}$ are the hidden state for volume RNN and RPP on day d, calculated as per Equation (6.3) and Equation (5.9) respectively, shown with arrows coming from RPP model towards the volume RNN model at regular intervals in Figure 6.2(c).

We note another plausible setup for joint modeling is attention volumes affecting the event rate of RPP. As our main concern is predicting popularity under promotion we do not utilize this setting.

6.4.3 Model Learning

We use stochastic gradient descent (SGD) together with the Adam optimizer [Kingma and Ba, 2014] to train our models.

Volume RNN: For training Volume RNN, we use mean square loss (*MSE*) as the loss function and each fixed interval data is considered as separate time steps of the given variables. This can be seen as modeling a time-series regression with LSTM. The MSE for our model over the observed length of Days D is given as:

$$MSE(\hat{V}) = \frac{1}{D}\sum_{d=1}^{D}\left(W_o^{Vol}\mathbf{h_d} - V_d\right)^2 \tag{6.5}$$

where V_d is the real volume at d^{th} day.

RNN-MAS: For training the joint model, we first train both RPP and volume RNN models independently. Once we have trained both models independently,we calculate the hidden state of RPP model for day d $\left(\mathbf{h_d^P}\right)$ using Equation (5.9), and the hidden state for volume RNN at day d $(\mathbf{h_d})$ is calculated using Equation (6.3). In next step we concatenate the hidden states from the two independent models as per Equation (6.4). The concatenated states are used to learn the weights in RNN-MAS using MSE for the attention volume series, given as:

$$MSE(\hat{V}) = \frac{1}{D}\sum_{d=1}^{D}\left(W_o^{MAS}\left[\mathbf{h_d}, \mathbf{h_d^P}\right] - V_d\right)^2 \tag{6.6}$$

where V_d is the real volume at d^{th} day.

We construct a random set of 500 videos from Active'14 described in Section 3.2.2. A single RNN-MAS model is learned for all the 500 videos. Finally, for each video we learn an individual model that uses the jointly learned parameters as the initialization point, instead of randomly initializing the individual model. We observe faster convergence and better prediction results for the model initialized with the learned parameters when compared to running with random initialization.

6.5 Popularity Metrics

RNN-MAS describes popularity under influence of heterogeneous streams of shares and tweets. However, with the introduction of LSTM, this model loses the ability to directly interpret and analyze results that a typical parametric linear system like HIP provides [Rizoiu and Xie, 2017]. Hence, to tackle the problem of interpretability and examine various *what-if* scenarios we propose two metrics based on RNN-MAS to quantify average response to unit promotion, and the relative influence among users of different *fame*. We also utilize the model to estimate a response series to unseen influence for a video.

Simulation

We can use our learned volume prediction models to simulate a series of views $V_d^{P(\cdot)}$ from $d = 1$ to $d = D$ days, where $\hat{V}_d^{P(\cdot)}$ stands for views on d^{th} day under a promotion function $P(\cdot)$. For computing views we evolve hidden states as per the Equation (5.9) and Equation (6.3) for our models, where the input parameter is defined by the value of function $P(\cdot)$ at various time steps. At the end of each time step, views are computed as per Equation (6.4) and Equation (6.2) for RNN-MAS and Volume RNN respectively. We define, $\nu\,(P(\cdot))$, the cumulative views generated by promotion function $P(\cdot)$ from $d = 1$ to $d = D$ days as:

$$\nu\,(P(\cdot)) = \sum_{d=1}^{D} \hat{V}_d^{P(\cdot)} \tag{6.7}$$

For all our simulations we choose $D = 10,000$ days, as this represents ≈ 27 years, longer than lifetime of any YouTube video.

6.5.1 Response to unseen influence

Despite taking into account multiple sources of external influence, there are influence signals that our models, RNN-MAS and volume RNN, do not capture explicitly. Examples include seasonality, or discussions in forums that are known to be an important factor for *gaming* videos — widely promoted on forum `www.minecraftforum.net`.

For understanding this response to unseen influence we set the promotion function $P(\cdot) = 0$ in our simulation setup described in Section 6.5 and obtain a series of $V_d^{P(\cdot)=0}$, denoted by $\nu(0)$.

The response series to unseen influence here generalizes previous definitions. In HIP [Rizoiu et al., 2017] external influence $u[t]$, is assumed to be an initial impulse plus a constant (Equation 5 in Section 2.4). In this work the shape of response series to latent promotions is unconstrained.

In Section 6.7.1 we present various case studies to illustrate how the response series to unseen influence for our models is able to capture richer temporal trends, such as seasonality.

6.5.2 Response to unit promotion.

In linear time variant (LTI) systems such as HIP, the total gain per unit promotion, ν, is well defined. It is calculated by computing the impulse response [Rizoiu and Xie, 2017] of the system. In our models, RNN-MAS and volume RNN, the notion of impulse response does not readily apply as it is a non-linear time variant system.

We compute the average response to unit impulse in three steps: compute response of our models to p units of promotion to volume RNN at $d = 0$ days; subtract response to unseen influences, calculated as per Section 6.5.1; normalize it by p. Response to p units of promotion is calculated using function $P(d) = p\mathbb{1}[d]$ in simulation setup of Section 6.5, where $\mathbb{1}[d]$ takes a value of 1 at $d = 0$, and 0 otherwise. The average response to unit promotion from promotion $P(d) = p\mathbb{1}[d]$, $\varrho\,(p)$ named as promotion score, is calculated as:

$$\varrho\,(p) = \frac{\nu(p\mathbb{1}[d]) - \nu(0)}{p} \tag{6.8}$$

In this definition, promotion score can be negative, capturing cases where being promoted decreases popularity (e.g. spam or paid tweets). Also note that $\varrho(p)$ is a function of p, i.e. for different simulation parameter p, the average response to unit promotion is different. This allows us to describe phenomena of diminishing returns in may real-world marketing scenarios [Jones, 1990]. We examine the effect of p in Section 6.7.2 and found that $\varrho(p)$ tend to decrease as p grows.

6.5.3 Loudness level

Using simulation metrics similar to ϑ and $\varrho(p)$, we can measure the influence of an individual or a group of users by the responses generated by their tweeting events.

Loudness level is a relative measure of total influence for one or a group of users. It is measured in decibels (dB). It is computed as log-ratio of the response of the target group $\psi(S)$ versus that of a comparison value ψ_0.

We compute $\psi(S)$ in two steps: calculate $\nu(P(\cdot))$ such that $P(\cdot) = \{e_i\}$ in simulation setup of Section 6.5, where $\{e_i\}$ is the set of tweets generated by user group S; subtract response to unseen influences, calculated as per Section 6.5.1. It can be written as:

$$\begin{aligned} \psi(S) &= \nu(\{e_i\}) - \nu(0) \\ \implies \psi_{dB}(S) &= \log_{10}\left(\frac{\psi(S)}{\psi_0}\right) dB \end{aligned} \tag{6.9}$$

where ψ_0, the comparison value, is set to 1 for experiments in Section 6.7.3.

As loudness level accounts for all activity generated by a user group, it can be used to compare the relative effects of promotion between users of different *fame* in Twitter.

6.6 Predictive Evaluation

In this section, we evaluate the performances of predicting the attention that videos receive in the future using RNN-MAS and its variants, against a number of baselines.

Overview of methods

We use the following approaches to predict the series of volume of attention that a video receives during the next time-frames:

- **HIP [Rizoiu et al., 2017]:** State of the art system for predicting views of video by using daily shares as promotions, outperforms linear regression baselines [Pinto et al., 2013] and [Szabo and Huberman, 2010].
- **VoRNN-S:** Our model detailed in Section 6.4.1 which like HIP, uses only daily shares for prediction.
- **VoRNN-TS:** A version of our model detailed in Section 6.4.1, using both daily shares and tweets as promotions, where tweets are aggregated daily to make it synchronous to views and shares series.
- **RNN-MAS:** Our dual RNN model described in Section 6.4.2 that combines daily shares with the event series of tweets, modeled as point process by RPP, to make predictions.

6.6.1 Volume Prediction

Setup. We evaluate our volume prediction approaches in the same temporal holdout setup employed by [Rizoiu et al., 2017], on the Active'14 dataset. Four data streams are available for each video in this dataset: The views volume series, the shares volume series, the tweets volume series, and the tweet event data series. In Table 6.2, we list all models used in our experiments along with the input streams each model utilizes for predicting views series of video. First, we train each model using the streams data in the first 90 days, for each video. Next, we forecast the views series during the next 30 days (from day 91 to 120) assuming known the promotions streams. Finally, we compute the popularity gain by summing up the views series during the test period. We evaluate performances using the Absolute Percentile Error (APE). APE is calculated by mapping the total views gathered to a popularity scale where the lowest number of views corresponds to 0% and highest number of views corresponds to 100%. Next, we normalize the popularity scale between 0 and 1, and compute the absolute error in the predicted percentile, computation follows the definition of [Rizoiu et al., 2017]. Note that out of the approaches in Table 6.2, RNN-MAS is the only model that can leverage the tweet event data stream for forecasting the futures views series.

Table 6.2: Summary of methods used in the volume evaluation, and type of promotion data that each approach can use.

	Type of Promotion		
Method	Shares (daily)	Tweets (daily)	Tweets (events)
HIP	✓	✓[1]	✗
VoRNN-S	✓	✗	✗
VoRNN-TS	✓	✓	✗
RNN-MAS	✓	✓[2]	✓

Results. Figure 6.3 shows the results for Active'14. We observe that VoRNN-TS, outperforms HIP by 7% and 3% for median and mean error, respectively. RNN-MAS performs best with an improvement of 11% and 17% for median and mean error respectively over HIP.

We observe an increase in performance of RNN-MAS when compared with VoRNN-S whereas for VoRNN-TS performance decreases, despite both models utilizing additional tweets data over shares in VoRNN-S. We are modeling this extra information in VoRNN-TS as volume of tweets wherein in RNN-MAS we utilize the timing (and magnitude) of individual tweets. Results

[1] Not used as performance decreases

[2] Not used, theoretically it can

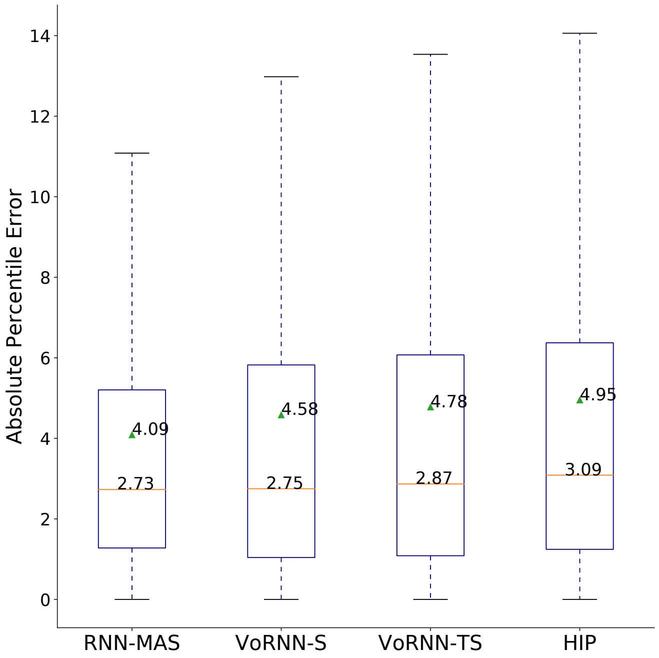


Figure 6.3: Absolute Percentile Error(APE) for different approaches on Active'14 dataset. All approaches use shares as promotion, VoRNN-TS and RNN-MAS use extra information about tweets.

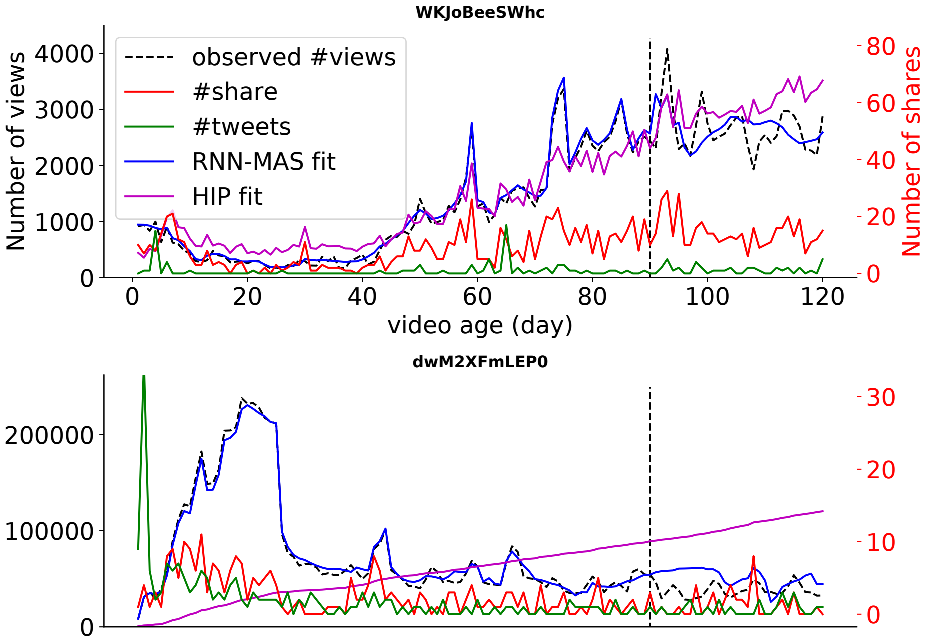


Figure 6.4: Two sample fittings where error(APE) for RNN-MAS is 2.01 and 0.13 and error for HIP is 2.3 and 5.1 for videos 'WKJoBeeSWhc' and 'dwM2XFmLEP0' respectively.

show that timing of events appears to contain information useful for predicting views, hence necessitating a model to jointly model event and volume data.

Figure 6.4 shows two sample fittings, for videos *WKJoBeeSWhc* and *dwM2XFmLEP0* where error(APE) for RNN-MAS is 2.01 and 0.13 and error for HIP is 2.3 and 5.1 respectively. The vertical dashed line divides data into training and testing set of 90 and 30 days respectively. The dashed line shows the real view series where blues line shows the fit by RNN-MAS and magenta shows the fit for HIP. For video *WKJoBeeSWhc* you could see two different phases, one before 90 days where view series (shown with dashed black line) is showing an inherent upward trend and one after 90 days where this inherent tendency has died. Fitting for RNN-MAS (shown in blue) is able to capture this change wherein HIP (shown in magenta) fails. For video *dwM2XFmLEP0* performance of HIP is even worse as it is not able to model the view series in accordance with shares (shown in red) and RNN-MAS ably models the response. One explanation is that RNN-MAS captures latent influence better, as shown in Section 6.7.1.

6.7 Explaining Popularity

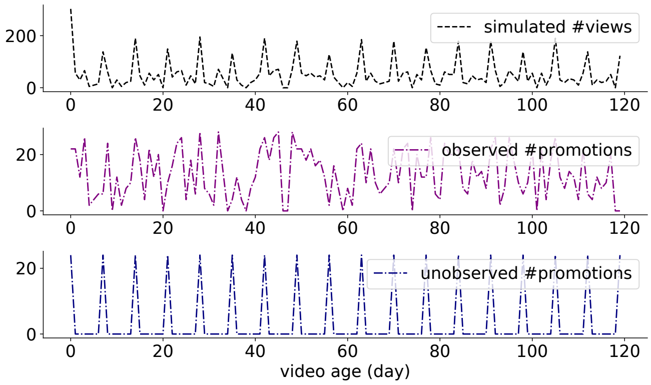
In this section, we exemplify how the response series to unseen influence, and how the metrics *promotion score* and *loudness level* described in Section 6.5 can be used to analyze the popularity of online videos.

6.7.1 Understanding unseen influence

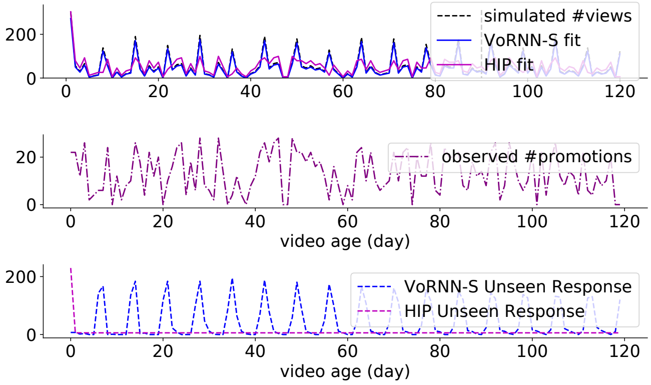
We present case studies on real and simulated data to illustrate the complex dynamics captured by the response series to unseen influence for our models when compared to HIP.

Synthetic data. We investigate on synthetic data the ability of volume RNN models and HIP to uncover seasonal trends. We simulate two sources of promotion for 120 days: the first series has a weekly cyclic component, with a maximum amplitude of 24 units of promotion (shown in the bottom row of Figure 6.5(a)); the second series contains random promotions with the same maximum amplitude (shown in the middle row of Figure 6.5(a)). Using these two promotion series, we simulate the view series using HIP (shown in the top row of Figure 6.5(a)). Next, we use the fitting procedure described in Section 6.6.1 to train the HIP and the VoRNN-S models on the simulated view series, using only the random promotion as the external stimulus series.

The top row of Figure 6.5(b) shows the fitting results for both models. Visibly, the series fitted by VoRNN-S (blue line) follows the simulated views (dashed black lines) more closely



(a)

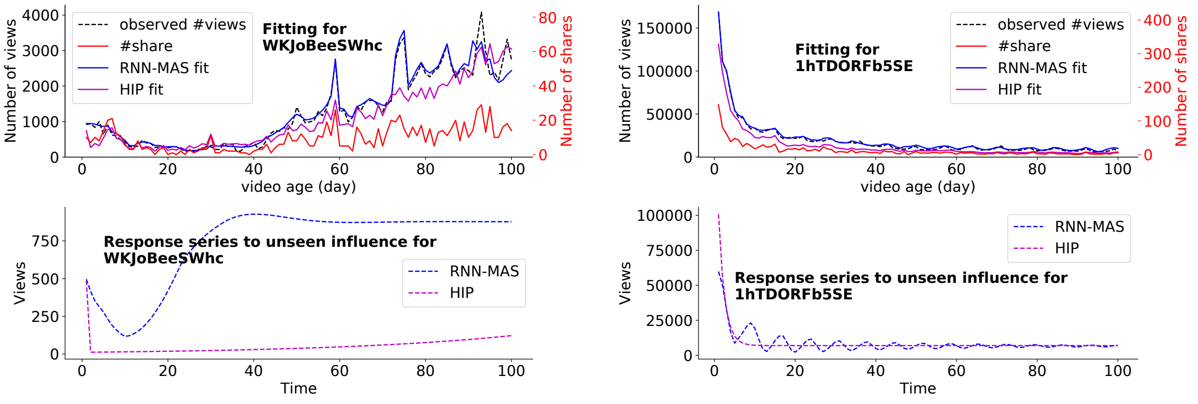


(b)

Figure 6.5: **(a)** Simulating a views series with known unobserved influence. A view series (top row) simulated by HIP with two sources of external stimuli: a series of random promotions (middle row) and a cyclic promotion series with weekly spikes (bottom row). **(b)** Retrieving the reaction to unknown influence through fitting. The simulated views series in (a, top row) is fitted using VoRNN-S and HIP (shown in the top row) using only the observed influence series (middle row). The reaction to unobserved influence (bottom row) is shown for HIP (magenta) and VoRNN-S (blue).

than the series fitted by HIP (magenta line). The bottom row of Figure 6.5(b) shows the response to unseen promotions, as estimated by VoRNN-S (in blue) and HIP (magenta). We see that VoRNN-S uncovers a cyclic response compatible in phase with the unobserved promotion series. The response of HIP shows a sharp drop followed by a near-zero constant response, due to its simplistic external influence modeling.

Real data fittings. We show sample fittings for two videos *WKJoBeeSWhc* and *1hTDORFb5SE* in the top row of Figure 6.6(a) and Figure 6.6(b) respectively. The bottom row of both figures shows the response series to unseen influence for RNN-MAS and HIP. Visibly, RNN-MAS can capture complex dynamics that HIP cannot explain: video *WKJoBeeSWhc* (Figure 6.6(a)) exhibits a delayed response, while *1hTDORFb5SE* (Figure 6.6(b)) features a seasonal trend. Part of the fitting performance gain of RNN-MAS can be attributed to its modeling of the response to unseen influence.



(a)

(b)

Figure 6.6: **(a)(b)** Fittings for the attention series (top row) and the response to unseen influence (bottom row) using RNN-MAS and HIP, for two sample videos *WKJoBeeSWhc* (a) and *1hTDORFb5SE* (b).

6.7.2 Average unit promotion score

HIP was used in [Rizoiu and Xie, 2017] to quantify the total amount of attention generated by a single unit of promotion – dubbed the *virality score* ν – for an online promotion application. In Figure 6.7, we compare the promotion score $\varrho(p)$ of our models against ν for all the videos in the Active'14. The x-axis shows the promotion score $\varrho(p)$ with $p = 5$, split into 20 percentile bins. The y-axis shows ν in percentiles, summarized using boxplots for each bin. Overall, we observe a strong agreement between $\varrho(p)$ and ν – the boxplots are placed around the main diagonal – signifying that RNN-MAS can capture online promotability as well as HIP does. Table 6.3 also shows strong correlation between predicted values of $\varrho(p)$ and ν for different values of $p = 1, 5, 10$.

Furthermore, we observe a decrease in the value of $\varrho(p)$ and in correlation between ν and $\varrho(p)$, as p increases (Table 6.3): the median values are $\varrho(1) = 263$ views, $\varrho(5) = 260$ views, $\varrho(10) = 259$ views. This suggests that, unlike HIP, our models can capture the complex phenomenon of diminishing returns in promotion [Jones, 1990].

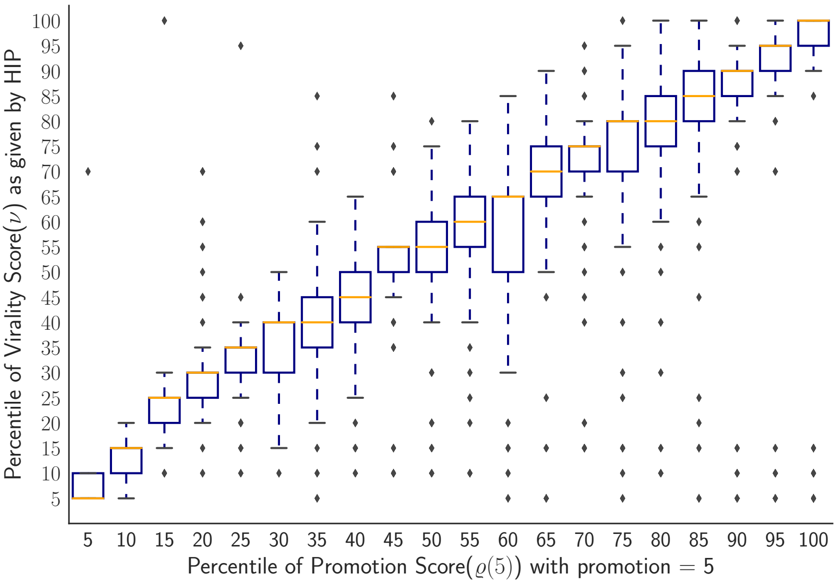


Figure 6.7: Comparison of HIP's virality score ν (x-axis) against the promotion response of RNN-MAS $\varrho(p = 5)$ (y-axis)

Table 6.3: Table showing correlation of promotion $\hat{\varrho}\,(p)$, with different values of $p = 1, 5, 10$ with virality-score of HIP. Here we find the correlation between the relative rank of promotion/virality.

	$\hat{\varrho}\,(1)$	$\hat{\varrho}\,(5)$	$\hat{\varrho}\,(10)$	virality HIP (ν)
$\hat{\varrho}\,(1)$	1	0.976	0.962	0.852
$\hat{\varrho}\,(5)$		1	0.989	0.801
$\hat{\varrho}\,(10)$			1	0.764
virality HIP (ν)				1

6.7.3 User Influence

Setup. Bakshy et al [Bakshy et al., 2011] showed that for promoting content on Twitter, the most reliable and cost-effective method is to utilize a group of individuals who have average or even less number of followers. We evaluate the influence of a super user against a cohort of small users. Figure 6.8 shows the rank versus the number of followers in Active'14. We define a *super* user as having 21,368 or more followers, or user at the lower end of the top 1% of all users; a *small* user as the median user with 120 followers. The cohort has 178 *small* users, the sum total of their followers is comparable to a super user. This is based on the assumption that followers of these small users do not overlap, i.e., they represent independent initiations from different part of the network.

For each video and each user group we generate a series of tweets from the trained RPP

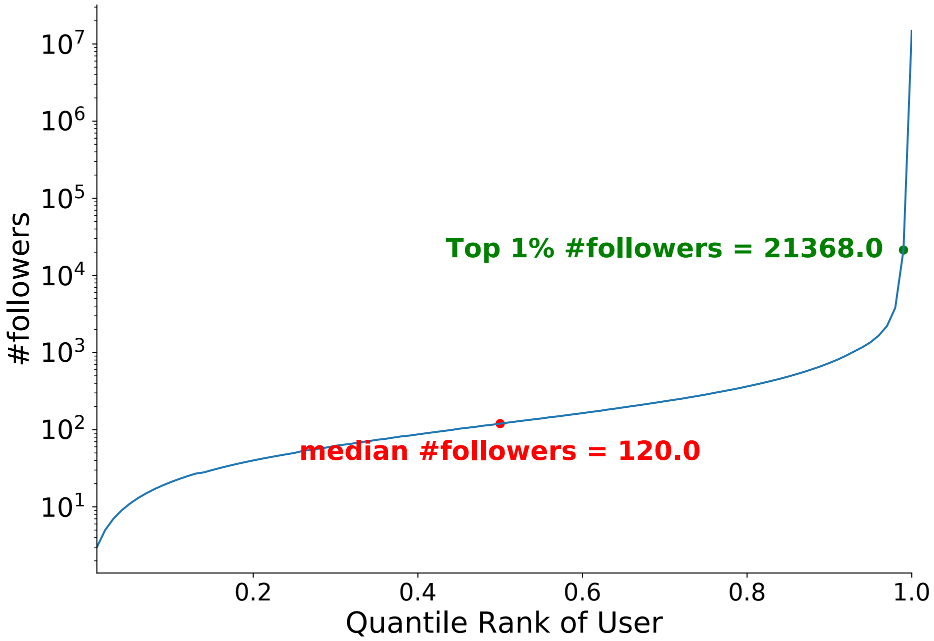
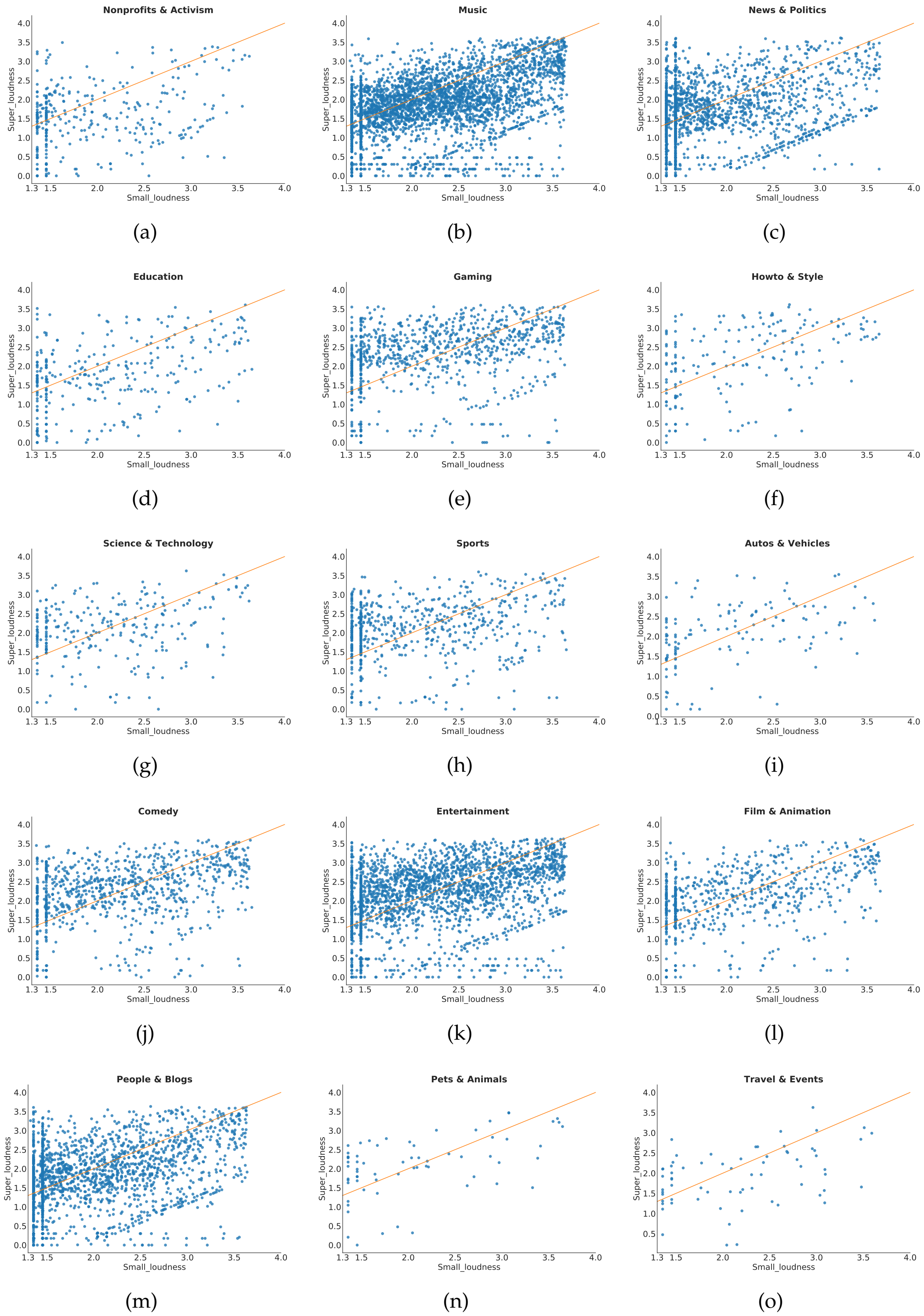


Figure 6.8: (a) Quantile rank versus number of followers of an user on ACTIVE'14. The median user has 120 followers whereas a user with 21368 followers is in top 1%.

model of the video. This series of tweets act as promotions by each user group. Now as sampling from RPP model is stochastic [Ogata, 1999], we generate 20 different simulations for each setup of super and cohort of median users for each of the video. Now each of this generated simulation is used as promotion for the trained model for the video as described in Section 6.5.3. The effect for tweet series promotion is captured by metric ψ_{dB} for each group. Now for purpose of comparison we take the mean value of ψ_{dB} over the 20 simulated tweet series promotions.

Results. Figure 6.9 shows the value of ψ_{dB} for super user against a cohort of small users on all videos in ACTIVE'14. We observe the promotion capability of super user varies differently among different type of content being promoted. For categories like *Nonprofits & Activism* (37%), *Music* (35%) and *News & Politics* (43%) of videos (above the green line) have ψ_{dB} of super user greater than ψ_{dB} of cohort of small users. Whereas for categories like *HowTo & Style* (50%), *Science & Technology* (54%) and *Gaming* (56%) of videos have ψ_{dB} of super user greater than ψ_{dB} of cohort of small users. This shows disproportionate influence of super user in promoting content of different categories.

We present an example video (*6Ga0V1mrZDo*) from *Nonprofits & Activism* category in Figure 6.10, where $\psi_{dB}(super)$ is 3.3 and $\psi_{dB}(small)$ is 1.3. The video here is uploaded by *PETA*, talking about deaths of pigs in slaughter houses. In Figure 6.10, not all peaks in tweets corre-



(a) (b) (c)

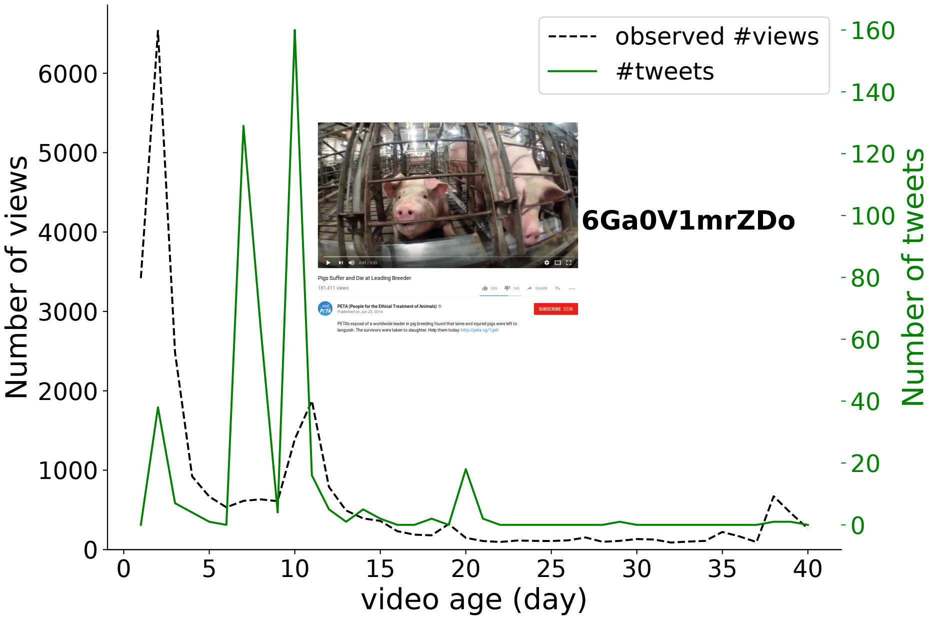
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 6.9: $\psi_{db}(super)$ versus $\psi_{db}(small)$ scatter plot for all videos in ACTIVE'14. See Section 6.7.3 for discussions.



Pigs Suffer and Die at Leading Breeder

PETA (People for the Ethical Treatment of Animals)
Published on Jun 25, 2014

PETA's exposé of a worldwide leader in pig breeding found that lame and injured pigs were left to languish. The survivors were taken to slaughter. Help them today: http://peta.vg/1geh

Figure 6.10: View and tweet series for video *6Ga0V1mrZDo*,with $\psi_{db}(super)$ = 3.3 and $\psi_{db}(small)$ = 1.3. See Sec 6.7.3 for discussions.

sponds towards the increase in views of the video. On examining the data we find that peaks where views and tweets correlates are the points where a super user is part of the tweets. In first peak around day 3, there is a user with 494851 followers. In the third peak around day 12, there is a user with 23561 followers. At two other tweeting peaks, there are cohorts of small users (all with less than 4000 followers) tweeting about the video, but no view count pikes. Presence of super users around peaks in view series corroborates the loudness estimate for this video.

6.8 Conclusion

This chapter proposed RNN-MAS, a model for predicting popularity under the influence of multiple heterogeneous asynchronous streams – namely tweets and volumes of shares for a YouTube video. With this model, we demonstrated superior performance on forecasting popularity on a large-scale YouTube video collection. We further design two new measures, to explain the viral potential of videos, another to uncover latent influences including seasonal trends. One important application of such model is to compare effects of different kinds of promotions. We propose a new metric, dubbed *loudness*, to quantify the relative effectiveness of user groups. We show that superusers and grassroot are effective in different content types.

We extend the work presented here in Chapter 7, to allow different videos (online items) to share parts of the model, and eventually make predictions at publish time of the video (cold-start predictions).

Other interesting direction for future work include modeling temporal dynamics drifts or finite population effects with RNN-MAS like model.

Chapter 7

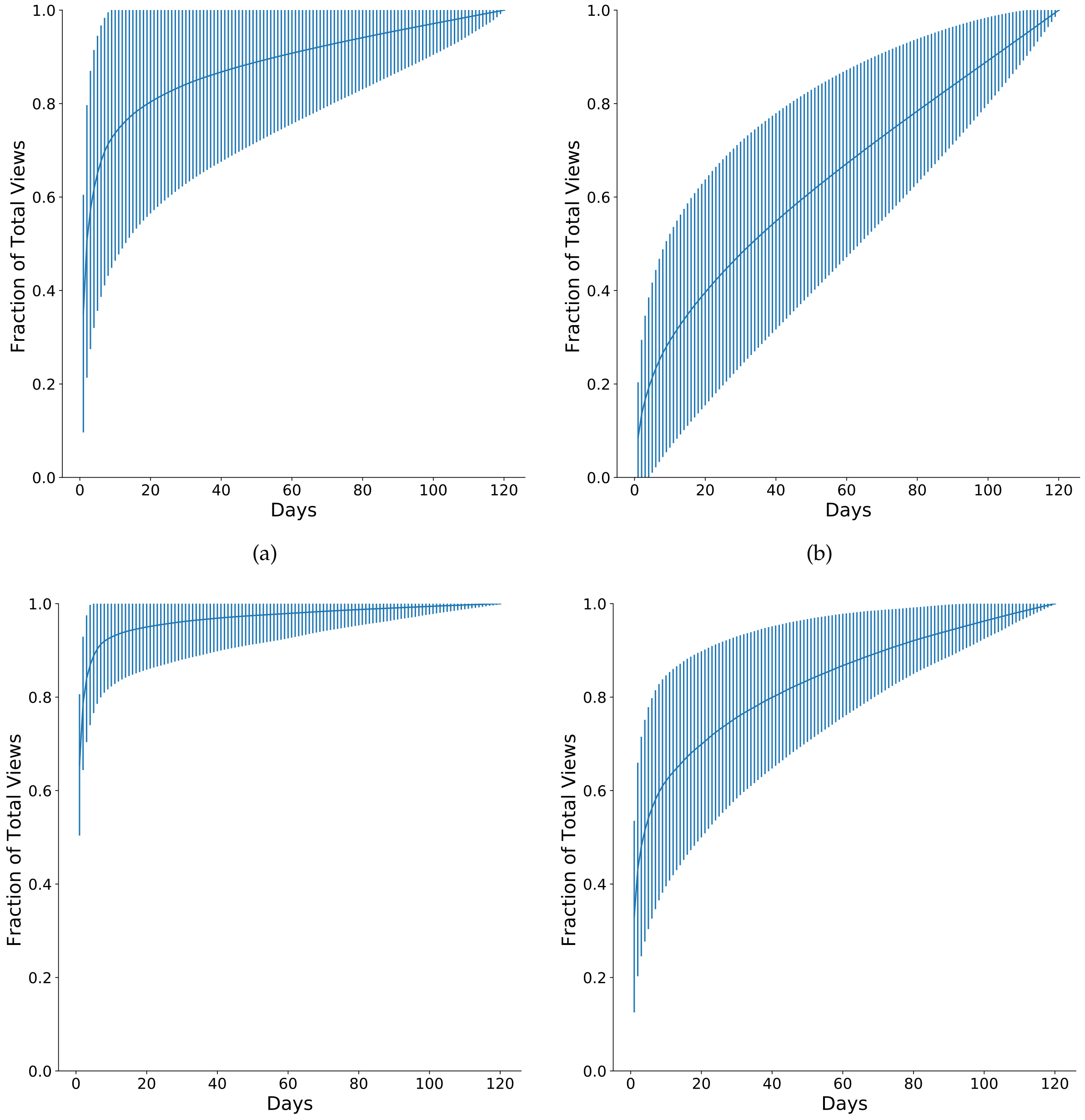
Popularity Models for a Group of Videos

In Chapter 6, we showed the efficacy of RNN-MAS in predicting views of a YouTube video under multiple promotions, i.e., shares on YouTube and tweets on Twitter. In this chapter, we propose an extension to the RNN-MAS model which uses a single model for predicting popularity of a group of videos. Section 7.1 presents the motivation and the problems tackled in this chapter. In Section 7.2, we discuss the proposed model along with its learning procedure. Section 7.3 presents our results with the proposed model on Active-3Yr dataset. Finally, in Section 7.4 we apply the proposed model to predict the popularity of a video before its publication on YouTube, commonly known as cold-start prediction setting.

7.1 Motivation

First, we observe from the Active-3Yr data that the popularity dynamics of videos in different categories have distinct temporal patterns. For example, Figure 7.1(a) and (b) depicts the evolution of the daily views for first 120 days of videos in the category *'News&Politics'* and *'Music'*, respectively . The y-axis in Figure 7.1(a) shows videos in category *'News&Politics'* achieve most of their views/popularity during early days. Whereas, as seen from Figure 7.1(b) for videos in category *'Music'*, the amount of views gathered is distributed much more evenly among the first 120 days. Furthermore, videos from the same channel tend to be similar, as seen from Figure 7.1(c) and (d). However, variations within a channel are much less than compared to variations among different channels from the same category.

One plausible approach to exploit the specific evolution patterns within the groups is to create customized models for each group. Correspondingly, in this work we address the question:



(a)

(b)

(c)

(d)

Figure 7.1: The distributions of fraction of total views gathered by a video in first 120 days (a) videos in category *'News&Politics'*, (b) videos in category *'Music'*, (c) videos from channel *'The Young Turks'*, and (d) videos in category *'Mnet Official'* . We can see that for *'News&Politics'* videos, slightly more than 35% of total views is gathered in first day whereas for videos in *'Music'* category it is only around 8%. Similarly, for channel *'The Young Turks'* videos, slightly more than 65% of total views is gathered in first day whereas for videos from channel *'Mnet Official'* gathers only around 35% of total views in the first day.

How does a specialized model for a group compare against the performance of a personalized model for each video in the group? A next step to explore for specialized models is to train them solely on the historical data in the group, i.e., create a pre-trained specialized model based on data from past videos in a specific group. Specialized pre-trained models allow us to answer the question: **What is the longitudinal effect of historical data on the accuracy of our forecasts?**

Lastly, predicting the popularity of content before being published (cold-start prediction) is a challenging task [Bandari et al., 2012; Martin et al., 2016]. Feature-driven methods have shown limited success in predicting the popularity of online content in cold-start setting [Bandari et al., 2012; Bakshy et al., 2011; Martin et al., 2016; Tsagkias et al., 2009; Abbar et al., 2018]. Whereas, generative models have shown state-of-the-art predictions for modeling popularity [Zhao et al., 2015; Crane and Sornette, 2008; Shen et al., 2014; Mishra et al., 2016], however they can not handle the cold-start predictions as they need some evolution data to fit individual models. Hence, in this work we seek to tackle the question: **Can we apply pre-trained models for cold-start predictions?**

7.2 Specialized Models for a Group of Videos

We start our discussion of the popularity prediction models for a group of videos by presenting our specialized model in Section 7.2.1. We then present the learning procedure for the proposed model in Section 7.2.2.

7.2.1 Model Formulation

The RNN-MAS proposed in Chapter 6 learns a different set of parameters for each video. Here, instead of learning different parameters for each video, we train a specialized RNN-MAS model that shares parameters across a group of videos. We argue for the shared formulation based on the common evolutionary patterns shared by videos in the same group, as seen in Section 7.1. Hence in the above setting, for a dataset, the total number of models learned for predicting popularity is equivalent to the number of different groups in the dataset.

A possible approach to identify groups in the YouTube dataset is to group videos based on the associated meta-information (see Section 3.2.1), i.e., category of a video (e.x., *Music*, *Education*, and others.) or the channel that uploaded the video. Hence, for creating various groups of different videos, we use the above-stated categorizations.

7.2.2 Model Learning

More formally, similar to the representation of data in Chapter 6, for a group of videos $\mathcal{V}$, let the daily views and the daily promotion volume for the i^{th} video on the d^{th} day be represented as $v_i[d]$ and $s_i[d]$, respectively. The corresponding event promotion series for the i^{th} video is denoted as S_i. Hence, the collection of event series is given by $\mathcal{S}_{1\ldots|\mathcal{V}|} = \{S_1, S_2, S_3, \ldots, S_{|\mathcal{V}|}\}$. A natural way to model $\mathcal{S}_{1\ldots|\mathcal{V}|}$ with RPP, is to maximize the joint log-likelihood of all parameters in RPP, e.g., $\Theta^{RPP} = \{W_o^P, \mathbf{h_t^P}, \alpha\}$, for N cascades, under the assumption that each of the $\mid \mathcal{V} \mid$ cascades are independent of each other. Hence, following Equation (5.10) we can write the joint likelihood as:

$$LL\left(\Theta^{RPP}|\mathcal{S}_{1\ldots|\mathcal{V}|}\right) = \sum_{n=1}^{|\mathcal{V}|}\sum_{i=1}^{|S_n|}\left[W_o^P\mathbf{h_{t_i}^P} + \alpha\tau_i + \frac{1}{\alpha}\exp\left(W_o^P\mathbf{h_{t_i}^P}\right) - \frac{1}{\alpha}\exp\left(W_o^P\mathbf{h_{t_i}^P} + \alpha\tau_i\right)\right] \quad (7.1)$$

Similarly the loss for attention volume series in the "per-group" RNN-MAS model for $\mathcal{V}$ can be obtained by combining the loss for all videos, given as:

$$MSE(\hat{\mathcal{V}}) = \frac{1}{D}\sum_{n=1}^{|\mathcal{V}|}\sum_{d=1}^{D}\left(W_o^{MAS}\left[\mathbf{h_d}, \mathbf{h_d^P}\right] - v_n[d]\right)^2 \quad (7.2)$$

where h_d and h_P are the hidden state for the volume RNN and for the RPP on day d for the "per-group" specialized model. The hidden states of volume RNN h_d, and of RPP h_P can be calculated by running the network forward by using inputs $s_i[d]$ and S_i, respectively. Analogous to per video model learning of RNN-MAS in Section 6.4.3, we can minimize the loss function in the Equation (7.1) and the Equation (7.2) to get parameters for the "per-group" model, W_o^P and W_o^{MAS}, respectively.

We note there are alternate ways of formulating a specialized model for a group of videos. For example, apart from sharing parameters, we can also share the input series between videos in the group for making predictions. We did try the setting mentioned above but noticed no performance gain in comparison to the setting of just sharing parameters.

7.3 Experiments

In this section, we will discuss the results for our proposed model for predicting the popularity of videos in the Active-3Yr dataset. The popularity is measured as the total views gathered by a video from days 91 to 120 (30 days) after it is published. The criterion used for evaluating our models is the Absolute Percentile Error (APE), used in Chapter 6 following the work of [Rizoiu et al., 2017].

We present our analysis of results in two different parts. In Section 7.3.1 we compare our results against the per video RNN-MAS (Chapter 6) model and the HIP [Rizoiu et al., 2017] model. In Section 7.3.2 we present a new setup for our prediction setting, which helps us to study the longitudinal effects of data on predicting the popularity of videos.

7.3.1 Volume Prediction

Setup. The methodology used for evaluating models is similar to the one used in Section 6.6.1. The temporal holdout setup of [Rizoiu et al., 2017] is used, where the first 90 days of data for a video is used for training and days 91-120 are used for performance evaluation. We use the meta-data available for each video to group videos based on their YouTube category or based on the channel (user) who uploaded the video. The meta-information is only used by the proposed specialized "per-group" model.

Overview of Methods. We run our experiments for predicting the total views during days 91-120, for each video, by using the following methods:

- **RNN-MAS:** Current state-of-the-art model; proposed in Chapter 6.
- **RNN-MAS-Cat:** A variant of RNN-MAS that estimates a single set of parameters for a group of videos based on their category and utilizes the learning mechanism described in Section 7.2.2.
- **RNN-MAS-Ch:** Similar to RNN-MAS-Cat, with groupings based on the channel (user) that uploaded a video.
- **HIP [Rizoiu et al., 2017]:** Previous state-of-the-art model for predicting views of video by using daily shares as promotions, outperforms linear regression baselines [Pinto et al., 2013] and [Szabo and Huberman, 2010].
- **HIP-Cat:** A variant of the HIP that estimates a single set of parameters for all videos in the same category; it is a specialized model similar in intent to the RNN-MAS-Cat.
- **HIP-Ch:** Similar to HIP-Cat but group of videos are based on their channel rather than category.

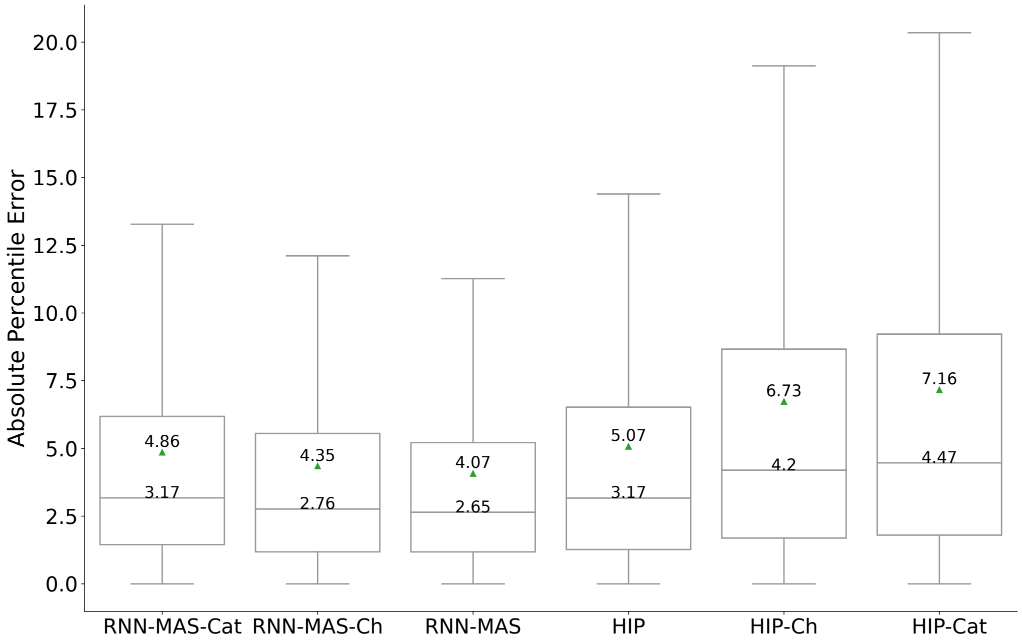


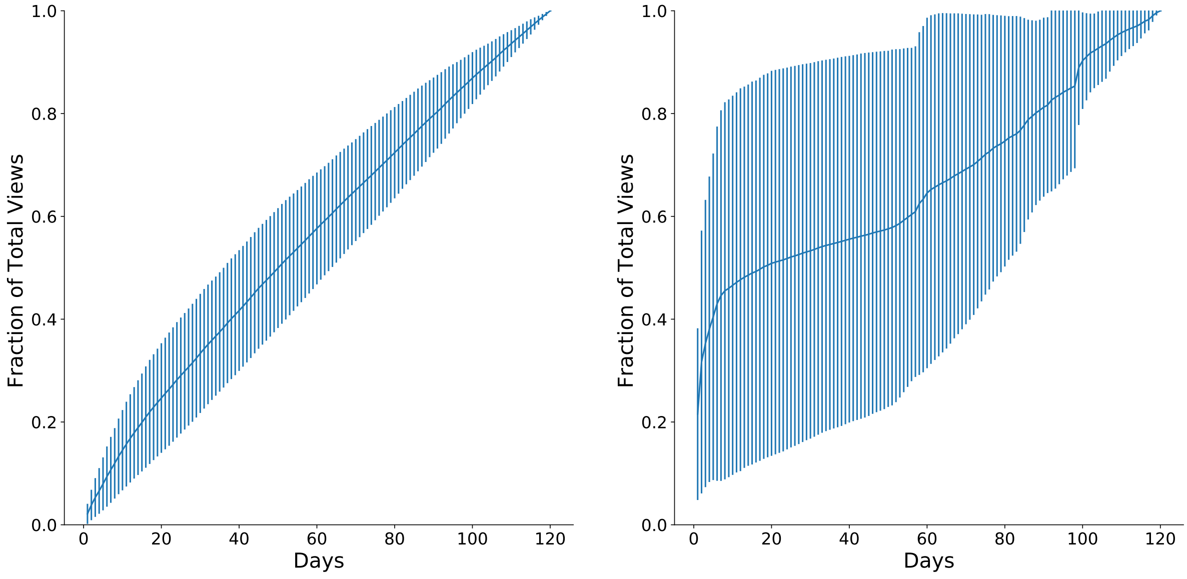
Figure 7.2: Absolute Percentile Error(APE) for different approaches on Active-3Yr dataset. We note, RNN-MAS based shared models although perform worse than their per video counterparts but are still better than HIP, especially the channel based RNN-MAS-Ch model, which outperforms HIP by 14%.

Results. Figure 7.2 presents results for all the above discussed methods on the Active-3Yr dataset. Our results are consistent with the observations made in the Chapter 6, that RNN-MAS outperforms HIP. A clear decrease in the performance of the shared models is visible when compared to its corresponding per video model, at least a decrease of 6% for RNN-MAS variants and 33% for HIP variants, for the mean performance. However, the performance of the channel based group models is better than the category based models. One plausible reason for the better performance for channel based models when compared to category based models is the significant variance in data within a category when compared to data within a channel, as seen in Figure 7.1. These results are in alignment with earlier work on popularity prediction, where [Martin et al., 2016] showed that author features are most indicative of future popularity after discarding temporal attributes.

Table 7.1 lists five best and worst channels for the performance of the RNN-MAS-Ch model when compared with RNN-MAS per video model. Interestingly as seen from Figure 7.3, the videos within the best-performing channel, *′AbrahamMateoVEVO′*, have very similar popularity evolution dynamics among each other, whereas, for the worst performing channel, *′ Universal*

Table 7.1: List of five best and worst performing channels for RNN-MAS-Ch model, alongwith the relative gain or decrease in the performance when compared to RNN-MAS model.

Name	Increase(%)	Name	Decrease(%)
AbrahamMateoVEVO	14.17	*Universal Orlando Resort*	24.08
VEGETTA777	13.04	*Stone Music Entertainment*	21.03
minutephysics	8.05	*SMTOWN*	17.92
MuzikPlay	7.56	*Visitmex*	17.59
The Young Turks	7.52	*SMTOWN*	17.42



(a)

(b)

Figure 7.3: The distributions of fraction of total views gathered by a video in first 120 days (c) videos from channel *AbrahamMateoVEVO*, and (d) videos in category *Universal Orlando Resort* . The videos from *AbrahamMateoVEVO* are more similar to each other with respect to the evolution oof popularity when compared to videos posted by *Universal Orlando Resort*.

Orlando Resort', videos mostly tend to differ from each other. Hence, we conjecture, the efficacy of shared models depends upon how related and similar are different videos within a group; more the similarity better is the performance.

One surprising result is the superior performance of shared models when compared to per video model of HIP. We observe an improvement of 14% and 4% for RNN-MAS-Ch and RNN-MAS-Cat over HIP for the mean performance. Results show us that even though sharing parameters across videos for RNN-MAS decreases performance, these models are still better than HIP, hence the advantage of maintaining far less number of models is worth considering. We believe the much larger number of parameters in RNN-MAS-Ch and RNN-MAS-Cat gives them the flexibility to be more expressive than HIP. Correspondingly, it may be the reason for better performance. We also show in Section 7.4 further extension of the shared model in other settings, like cold-start prediction, which were not possible either by HIP or RNN-MAS by themselves.

7.3.2 Volume Prediction with Historical Data

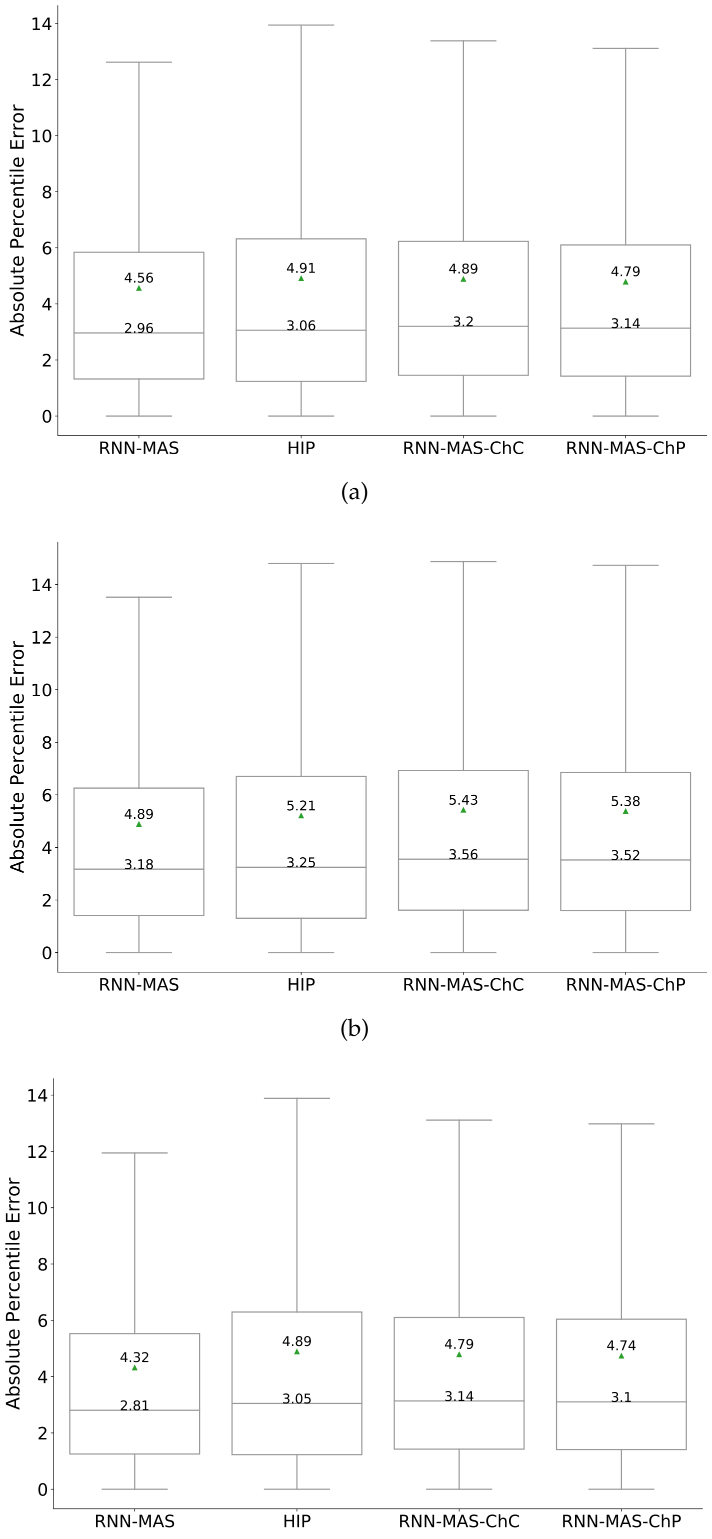
In this section, we analyze the longitudinal effect of data on the accuracy of forecasts. We want to know, how well we can predict the future popularity of a video by only utilizing historical information about a set of videos belonging to the same group that the original video belongs to. We remark, that in their original formulations, both RNN-MAS and HIP are incapable of analyzing this specific line of research objective as both of them need some data from the target video to estimate parameters. Whereas, the rationale of shared models can be naturally extended to train on historical videos belonging to the same group to learn a model for predicting future popularity.

Setup. The Active-3Yr dataset contains videos uploaded across three years, from January 2015 to December 2017. For every given year (2015, 2016, 2017) in the dataset, all videos uploaded between January to April form the training set for the year, and the videos uploaded from September to December are set aside as the test set for the given year. For every video in either the training or the testing set, we record its popularity and promotion data for 120 days after its publishing time. We note, we have purposely not included any video uploaded from May to August in either of the training or testing set. It helps us to mitigate the effect of accidental information leakage across training and testing set by making them non-overlapping across time ranges. The above approach essentially gives us three different training and test sets for evaluating models. Similar to the previous setup in Section 7.3.1, we predict the popularity of a video from day 91 to 120 (30 days). APE is used as the evaluation metric. The prediction setup in this section is different from the setup in Section 6.6 or Section 7.3.1. In the current setup, specialized "per-group" models do not have access to the daily views of the test video in its first 90 days for training. Whereas in earlier setups, it was available for training.

Overview of Methods. The models we evaluate for prediction setting in the section are as follows[1]:-

- **RNN-MAS:** Our model developed in Chapter 6, which learns a model per video basis. It does not train on historical data. Instead, it has access to all the individual data about a

[1] Note we do not present results HIP's variants and RNN-MAS-C, as we already showed the superior performance of RNN-MAS-Ch in Section 7.3.1.



(a)

(b)

(c)

Figure 7.4: Absolute Percentile Error(APE) for shared models based on historical data on Active-3Yr dataset. Figure (a), (b) and (c) show performances year 2015, 2016 and 2017 respectively.

video and trains a model per video basis. Instead of being a baseline, it acts as an indicator for upper bound on the performance.

- **HIP:** Similar to RNN-MAS, it is trained on full data for each video rather than the historical setup.

- **RNN-MAS-ChC:** A variant of RNN-MAS that estimates parameters for a set of videos based on their channel. The training data for this model is limited to the year test video is uploaded in. For example, if we are predicting popularity for a test video from the year 2017, then the training data also comes the year 2017.

- **RNN-MAS-ChP:** Similar to RNN-MAS-ChC, but it additionally has access to the training data from the current and previous year. For example, for a video in the test set for the year 2017 it will use training data from the year 2016 and 2017. We note, if the test video is from the year 2015 we use the videos from the ACTIVE'14 dataset to get training set for the year 2014 as we do not have any video in ACTIVE-3YR for the year 2014.

Results. Figure 7.4 presents our results on ACTIVE-3YR dataset for each of the test set in year 2015, 2016 and 2017. Expectedly the best results are from the per video RNN-MAS. However, the performance of RNN-MAS-ChC is equivalent to the performance of the HIP. The consistent performance of RNN-MAS-ChC when compared to models trained on full information shows how learning on historical data has enabled RNN-MAS to unearth the hidden structures in popularity evolution for a specific group of videos, and consequently can be utilized successfully for predicting popularity in the future for videos that belong to the same group. Hence, we can conclude that using historical data to make predictions based on RNN-MAS is viable and still competitive.

A slightly elusive result is the negligible improvement in performance for RNN-MAS-ChP when compared with RNN-MAS-ChC for all of the three years. We expected an increase in the performance of the model RNN-MAS-ChP owing to its access to additional data. The marginal gain in performance makes the presence of extra data from previous years almost unnecessary. We hypothesize, a possible reason for the marginal gain can be attributed to change in the underlying dynamics of the network across years, as shown in Section 3.3.1, and hence the data from distant past does not necessarily add any more generalizing capability.

7.4 Cold Start Popularity Prediction

In this section, we address the problem of predicting the popularity of an item before its publication, referred to as cold-start popularity prediction. Commonly, feature-driven methods have been used for prediction in the cold-start setting [Bandari et al., 2012; Goel et al., 2012; Bakshy et al., 2011; Tsagkias et al., 2009; Bandari et al., 2012]. The main idea behind feature-driven cold-start prediction models is to use meta-information about an item to compute its similarity with an item or group of items in historical data, and finally, on based identified items or items predict the popularity. However, these approaches have not been very successful in practice [Martin et al., 2016]. A reason for their timid performance can be attributed to the lack of the presence of temporal features deemed to be the most successful features for predicting online popularity [Cheng et al., 2014; Martin et al., 2016].

On the other hand, generative models [Zhao et al., 2015; Crane and Sornette, 2008; Shen et al., 2014; Mishra et al., 2016] are deemed unfit for cold-start predictions as they require some amount of popularity evolution data to estimate the parameters of the model. In this section, we will fill the gap of using generative models to predict popularity in cold start setting. In particular, we will predict the popularity of videos in the Active-3Yr dataset before their publication on YouTube by applying the shared model learned on historical data.

Next in Section 7.4.1 we will present our methodology for applying the shared model for cold-start prediction and finally Section 7.4.2 will present our results on the proposed approach.

7.4.1 Methodology

The performances of our shared models trained on historical data showed how a shared model could learn the internal representations of popularity dynamics of an item without observing the item. Hence, we propose to apply the shared model for predicting the popularity in the cold-start setting. However, in our current status, these models still need a promotion series to make predictions, and in a cold-start setting, these promotions are also not available.

We mitigate the aforementioned problem by generating representative samples of promotion for each video with the help of the shared model learned from historical data. Looking back at our RNN-MAS model, we identify every trained model also has a corresponding trained RPP model attached to it (see Figure 6.2(c)). Moreover, a trained RPP model can be efficiently utilized to simulate a series of events using the thinning algorithm [Ogata, 1999] (note that we followed the same procedure for estimating user influence in Section 6.7.3).

For simulating a series of events, we need an initial seed user to sample from the RPP model. In this work, we set the initial seed user to have a magnitude equal to the mean of the magnitudes of users who have started a tweet cascade of a video uploaded by the same channel. As the simulation from RPP is stochastic, we generate 200 different representative samples of tweet series for each of the channels. The generated samples act as the promotion we need for predicting popularity with a shared model.

Each of these 200 tweet series is used to generate corresponding views (popularity) series for the test video. The final views (popularity) for each day is calculated as the mean of these 200 different views.

7.4.2 Experiments

In this section, we start by introducing the setup followed in experiments. We then present a brief overview of different methods we compare for cold-start prediction, and finally, we present our results.

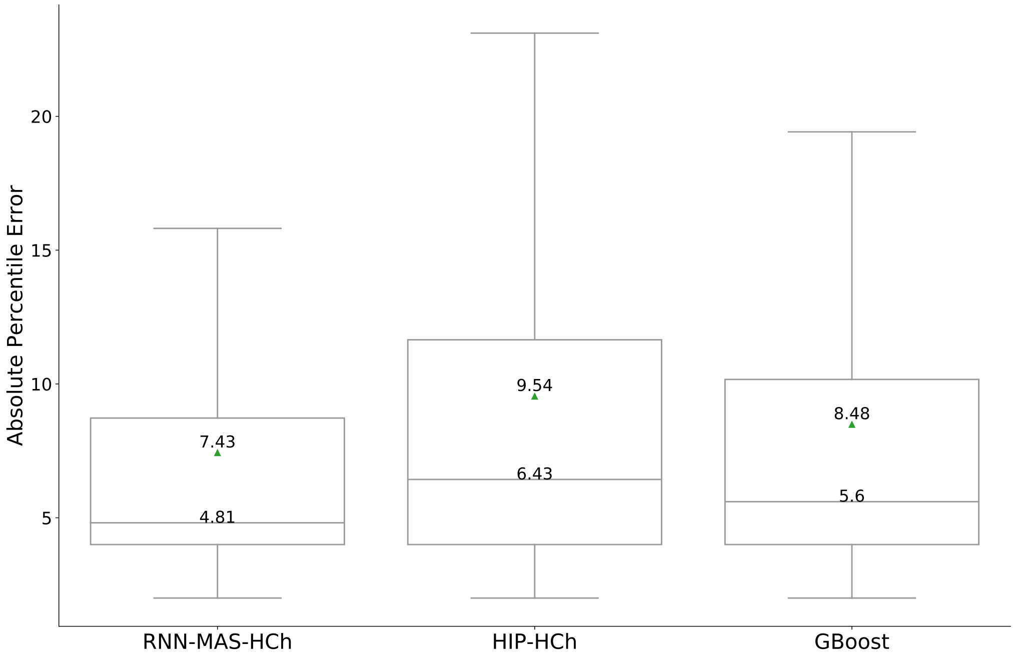
Setup. We divide the ACTIVE-3YR dataset into two parts as the training set and the testing set. The videos in the training set are uploaded between the period starting 1^{st} January 2015 to 30^{th} August 2016. For the testing set, we have all the videos uploaded in the year 2017. We aim to predict the total popularity gathered by a video in its first 30 days after being uploaded on YouTube. For calculating the performance of models, we report Absolute Percentile Error (APE) as our metric.

Overview of Methods. For the cold start prediction we use the following methods for predicting the total attention:

- **RNN-MAS-HCh:** A variant of the RNN-MAS model, proposed in this section, which learns the parameters from historical data by combining all the videos from a single channel. The only difference from models presented in earlier sections is that this model is trained on full historical data from January 2015 to August 2016.

- **HIP-HCh:** A variant of the HIP that is trained on the same data as the RNN-MAS-HCh.

- **GBoost:** A gradient boost algorithm that is trained on the same historical data as RNN-MAS-HCh. Additionally, it uses following features: category of video, the channel of video, 5-point summary of the number of followers of user who have tweeted about the video, 5-point summary of the distribution of time between consecutive tweets about the



H

Figure 7.5: Absolute Percentile Error(APE) for various methods used for cold-start prediction in Active-3Yr dataset. The performance of RNN-MAS-HCh model is better than all its alternatives.

video, day of upload of the video and time in hour when video was uploaded. We note, GBoost has access to more data than the above mentioned two models.

Results. Figure 7.5 presents our results on Active-3Yr dataset. We can observe that the best results are produced by the RNN-MAS-HCh model where it beats other methods by more than 15% and 14% for median and mean error respectively. The historical variant of the HIP model performs worse and has a mean APE of 9.54, indicating on average it predicts the popularity of videos off by 10% for their relative percentile bins. We also note that the variance in predictions for the cold start in Figure 7.5 are relatively large when compared against other prediction setups in Section 7.3.1 (Figure 7.2) and Section 7.3.2 (Figure 7.4), owing to the unknown (simulated approximation) of the promotion signal (tweets).

With all the extra information available to the GBoost model we expected it to have the best performance. However, to our surprise RNN-MAS-HCh model outperforms the GBoost model by at least 14% in both mean and median error. Results show that timing contains information useful for predicting views, hence necessitating a method to model event and volume data jointly.

7.5 Summary

In this chapter, we have presented extensions and applications of our RNN-MAS model developed in Chapter 6. We systematically show the efficacy of using shared models for prediction in different settings, and further, observe how we can use RNN-MAS as a pre-trained model for making predictions. Finally, we develop a strategy to make state-of-the-art cold-start predictions with RNN-MAS model, by utilizing historical data and generating representative promotion series. The various applications of the RNN-MAS model show its versatility and efficacy in modeling online social data with multiple asynchronous streams.

Chapter 8

Conclusion

In this chapter, we summarize the contributions therein this thesis for modeling and predicting popularity in social media. We also present a few potential research directions for future work.

8.1 Summary

This thesis aimed to make connections and links across methods for predicting and modeling popularity in social media. In particular, the thesis bridged gaps between methods based on different paradigms (feature-driven vs. generative) and methods for different types of data (event data vs. volume data). Moreover, we proposed two generalized formulations of the Hawkes Process accounting for essential diffusion properties in social media. Additionally, we derived new metrics for better explainability and interpretability of our models. In particular, we have made the following contributions:

- **Linking feature-driven and generative models.** Earlier work in modeling and predicting popularity in social media can be divided into two broad classes: feature-driven and generative methods. However, when describing a new method, researchers compared and evaluated the method against only one of the classes mentioned above. Even though the end goal of the two paradigms is the same, predicting the popularity of an online item, most of the prior work treated the two sets paradigms as incomparable and unusable in conjunction with each other due to different datasets and different evaluation settings. Our work in Chapter 4 [Mishra et al., 2016] established the first common benchmark for comparing feature-driven methods alongside generative methods in predicting the popularity of content in online social media, particularly on Twitter. We bridged the gap by first proposing a generative model based on Hawkes processes that utilized a prediction layer on top for predicting retweets popularity. Secondly, we identified a list of features that can be collected for any publicly available online data, for predicting popularity. Both, our proposed generative Hawkes method and feature-driven method outperform the previous state-of-the-art (SOTA) predictor in predicting the popularity of tweets as retweets. Furthermore, we combined identified features and Hawkes model to form a hybrid model that further increases the performance. We established the SOTA as of 2016; that outperformed the previous SOTA by 25% on average for predicting final size of retweet cascades. Our work has been further developed by [Wu et al., 2018a; Liao et al., 2019; Li et al., 2017b] for linking feature-driven and generative models. The links we formed between models helps us to gain performance as well as extract interpretable parameters to understand and characterize the evolution of popularity.

- **Generalized Hawkes Process.** Chapter 5 continued the study of utilizing the Hawkes process for modeling cascades in social networks. We extend the Hawkes process by proposing two novel formulations: i) a Hawkes model accounting for a finite population size, HawkesN, and ii) a non-parametric recurrent neural network based Hawkes model, Recurrent Point Process (RPP). In particular, our HawkesN model is the first Hawkes model that accounts for a finite population size in social media. The HawkesN model provides both a better explanation and prediction for cascades in social media. It is achieved by concurrently accounting for the self-excitation and the reduction in the size of the available population for cascade to grow. We further introduce a recurrent network based formulation of the Hawkes Process, RPP. RPP models the event intensity in a point process as a non-linear function of the history, and a recurrent neural network is used to learn this representation. Experiments on real and simulated data show the effectiveness of the RPP model. In the case of simulated data, we show how RPP can learn the true intensity of the model without the specification of the actual parametric form; as for the real datasets, the superiority of the model is shown by better predictive performance when compared with parametric alternatives suggested in Chapter 4.

- **Linking asynchronous event data models with volume based models.** We study the problem of predicting popularity in the presence of multiple information sources, asynchronous to each other in Chapter 6. Most of the earlier work, either considered popularity to evolve in a single specific social network, or aggregated data across multiple sources to make them synchronous. Chapter 6 introduced the first model, to the best of our knowledge, capable of handling both event data streams and volume data streams that are of different granularity and unfolds asynchronously in time to each other. Specifically, we introduced RNN-MAS, a recurrent neural network based model that can predict the popularity of YouTube videos under the external influence of tweets on Twitter. The main novelty of our work over previous work in literature lies in the joint modeling of daily views of video (periodic volume data) with individual tweets (aperiodic event data) without resorting to any aggregation techniques. On a large scale YouTube dataset, we show our model outperforms the previous SOTA by more than 17% on average. Our work highlights the efficacy of the joint modeling of multiple asynchronous streams of data over different networks to predict the popularity of online content. We further define new measures for explaining the virality and strength of latent sources (seasonality) for videos on YouTube. We also develop a new metric, loudness, that helps us to estimate the influence of users in Twitter to disseminate content on YouTube. To the best of our knowledge, it is one of the first work in user influence estimation across network boundaries. In Chapter 7, we show some exciting applications of RNN-MAS in learning shared parameters among videos of different categories or groups. By leveraging the idea of shared parameters, we learn a model for a video from the historical videos that belong to the same group as the video under consideration. With the learned parameters, we generate a representative promotion series for the video using our simulation setup. This simulated promotion is now utilized to predict the popularity of the video in the cold-start setting. Our proposed model performs consistently better than the other alternatives by at least 14% on average.

- **Dataset.** In chapter 3, we present three new datasets, i) News, ii) Active'14 and iii) Active-3Yr, for benchmarking popularity prediction in social media. News is the first dataset that allows benchmarking of both feature-driven and generative methods for popularity prediction simultaneously on a single dataset. Common benchmarking was plausible as News exposes a vast array of features related to information cascades that can be utilized by generative models as well as feature-driven methods. We hope the release of the News dataset, provides the research community a valuable resource to take further our work on linking methods across different modeling paradigms, feature-driven, and generative models. Further, we provide two YouTube videos datasets, Active'14 and Active-3Yr. Both datasets have aggregated daily views and shares from YouTube, and individual tweets from Twitter. The uniqueness of datasets lies in the presence of multiple streams of data available at different granularity levels related to online content (video). Both datasets are first of their kind, as they enable a systematic study of the popularity of an online item, video, on multiple platforms, YouTube and Twitter. Additionally, Active-3Yr dataset opens up the opportunity to study the longitudinal evolution of popularity on YouTube as it spans over three years from Jan 2015 to Dec 2017.

8.2 Discussion

The work done in the thesis spans across various paradigms on predicting and explaining popularity in social media. Although the work in the thesis has shown great potential in achieving state of the art results in predicting popularity, they still do not capture any causal or counterfactual behavior. The measures introduced in the thesis: i) response to unseen influence, ii) response to a unit promotion, and iii) loudness level, have provided a way to measure various desired quantities like the response to a promotion scheme or promotion strategy. Still, these measures only capture correlations in the collected data by simulating some 'what-if' scenarios based on past data. Hence, any algorithmic change in the underlying social media network that breaks the distribution of attention or the distribution of data in itself will also affect the results of the models presented in Chapter 6 and Chapter 7. The Hawkes process models presented in the Chapter 4 and Chapter 5 are more robust to the changes in the underlying social media network as they model the dynamics of attention for each cascade individually. Still, assumptions around how people react towards new content make Hawkes process models vulnerable. Hence, changes to how content is recommended or served in social media systems, like, facebook not showing the number of likes in a post or Twitter relaxing its constraint on the length of Tweet, may alter the shape of social-kernel used in Chapter 4 and Chapter 5. In summary, the results presented in the thesis will hold in general for models that capture individual dynamics. However, all the results at the aggregate level are dependent on the underlying distribution in the observed data.

8.3 Future Work

While this thesis addresses a wide range of models across popularity prediction by linking them together, at the same time it opens exciting avenues for future research. We envision the following future work:

- **Competing and collaborative cascades.** Modeling popularity with the Hawkes and related models have the implicit assumption of cascades evolving in isolation in the network. In reality, multiple cascades grow simultaneously in the network, competing for the same attention among each other [Myers and Leskovec, 2012; Srivastava et al., 2015], or even collaborating to garner attention [Huang et al., 2013]. It would be an exciting area of future research if we can incorporate the competition or collaboration explicitly into the framework of Hawkes Processes. Recent work by [Li et al., 2017b] tries to address the issue by utilizing a multivariate point process for modeling users activity; still, question remains on how to use them for predicting popularity. Hence, relaxing the constraint of independent evolution of content is a future work worth considering.

- **Beyond user followers and event times.** The point process models developed in the thesis and other related work has mainly concentrated on capturing user influence and temporal features as event times. However, a wealth of additional information associated with a cascade, text features, user meta-data, information of user interest and authority, to name a few, are mostly unexploited in the Hawkes models. Recent work like [Cao et al., 2017; Li et al., 2017a] tries to address the problem by employing end-to-end deep neural networks model. However, this loses the explainability power of generative models for not being able to untangle the effects of different features in an end-to-end model. We envision a more accurate estimation of cascades once the models systematically use these properties.

- **Exploration of other deep learning architectures.** In Chapter 5 and Chapter 6, we proposed novel formulations of the Hawkes process by utilizing a recurrent neural network. Though our proposed models produced state-of-the-art results, we acknowledge a few limitations to our work. Firstly, the proposed neural architecture does not allow us to model domain knowledge into the recurrent networks explicitly. One way of addressing it could be the utilization of a generative adversarial network (GAN) [Goodfellow et al., 2014], where we can use a feature-driven method as the discriminator and recurrent network as a generator [Wu et al., 2018a] function. Another significant drawback of recurrent neural networks based Hawkes process is the inability of the model to capture long-range effects for events. A straight forward solution is to use attention networks [Yang et al., 2016], to flexibly learn weights for combining history, based on their outputs and temporal contexts [Xiao et al., 2017; Liao et al., 2019]. Techniques that could efficiently incorporate domain knowledge and learn long term dependencies might result in much more useful models for estimating popularity dynamics.

- **Exploration of links between Hawkes and SIR models.** The finite population based HawkesN model inspired an unexplored connection between the SIR and Hawkes model. As shown in our work [Rizoiu et al., 2018], in the SIR model expected rate of new infections when marginalized over recovery events is equivalent to the intensity of events in the HawkesN model. The equivalence holds under a strict assumption on the exponential distribution of recovery events in the SIR model. We envision follow-up work in finding more generalized connections that are true for a broader class of recovery events distribution [Kong, 2019]. The proposed equivalence has opened many different gateways of utilizing techniques developed in epidemiology to estimate information diffusion models and vice-versa.

Bibliography

Abbar, S.; Castillo, C.; and Sanfilippo, A., 2018. To post or not to post: Using online trends to predict popularity of offline content. In *Proceedings of the 29th on Hypertext and Social Media*, 215–219. ACM. (cited on page 105)

Abisheva, A.; Garimella, V. R. K.; Garcia, D.; and Weber, I., 2014. Who watches (and shares) what on youtube? and when?: using twitter to understand youtube viewership. In *Proceedings of the 7th ACM international conference on Web search and data mining*, 593–602. ACM. (cited on page 20)

Agarwal, D.; Chen, B.-C.; and Elango, P., 2009. Spatio-temporal models for estimating click-through rate. In *Proceedings of the 18th international conference on World wide web*, 21–30. ACM. (cited on page 51)

Asur, S. and Huberman, B. A., 2010. Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, 492–499. IEEE Computer Society. (cited on page 37)

Bakshy, E.; Hofman, J. M.; Mason, W. A.; and Watts, D. J., 2011. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 65–74. ACM. (cited on pages 2, 3, 11, 12, 19, 27, 28, 83, 97, 105, and 113)

Bandari, R.; Asur, S.; and Huberman, B. A., 2012. The pulse of news in social media: Forecasting popularity. In *Sixth International AAAI Conference on Weblogs and Social Media*. (cited on pages 11, 105, and 113)

Bao, P.; Shen, H.-W.; Jin, X.; and Cheng, X.-Q., 2015. Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes. In *Proceedings of the 24th International Conference on World Wide Web*, 9–10. ACM. (cited on pages 64 and 67)

Bass, F. M., 1969. A new product growth for model consumer durables. *Management science*, 15, 5 (1969), 215–227. (cited on page 13)

Berger, J. and Milkman, K. L., 2012. What makes online content viral? *Journal of marketing research*, 49, 2 (2012), 192–205. (cited on page 2)

Bhagat, S.; Burke, M.; Diuk, C.; Filiz, I. O.; and Edunov, S., 2016. Three and a half degrees of separation. *Facebook Research*, 4 (2016). (cited on page 1)

Budak, C.; Agrawal, D.; and El Abbadi, A., 2011. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, 665–674. ACM. (cited on pages 3 and 83)

Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; and Cheng, X., 2017. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1149–1158. ACM. (cited on pages 20, 60, and 121)

Castillo, C.; El-Haddad, M.; Pfeffer, J.; and Stempeck, M., 2014. Characterizing the life cycle of online news stories using social media reactions. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, 211–223. ACM. (cited on page 20)

Cha, M.; Haddadi, H.; Benevenuto, F.; and Gummadi, K. P., 2010. Measuring user influence in twitter: The million follower fallacy. In *Fourth International AAAI Conference on Weblogs and Social Media*. (cited on pages 19 and 49)

Cha, M.; Kwak, H.; Rodriguez, P.; Ahn, Y.-Y.; and Moon, S., 2007. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 1–14. ACM. (cited on page 9)

Cha, M.; Mislove, A.; Adams, B.; and Gummadi, K. P., 2008. Characterizing social cascades in flickr. In *ACM SIGCOMM Workshop on Online Social Networks*. ACM. (cited on page 9)

Cha, M.; Mislove, A.; and Gummadi, K. P., 2009. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, 721–730. ACM. (cited on page 10)

Chandra, R. and Zhang, M., 2012. Cooperative coevolution of elman recurrent neural networks for chaotic time series prediction. *Neurocomputing*, 86 (2012), 116–123. (cited on page 19)

Chen, G.; Kong, Q.; and Mao, W., 2017. An attention-based neural popularity prediction model for social media events. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 161–163. IEEE. (cited on page 60)

Chen, G.; Kong, Q.; Xu, N.; and Mao, W., 2019. Npp: A neural popularity prediction model for social media content. *Neurocomputing*, 333 (2019), 221–230. (cited on page 60)

Cheng, J.; Adamic, L.; Dow, P. A.; Kleinberg, J. M.; and Leskovec, J., 2014. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, 925–936. ACM. (cited on pages 2, 11, 12, 24, 25, 27, 37, 38, 56, 57, 81, and 113)

Christakis, N. A. and Fowler, J. H., 2007. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357, 4 (2007), 370–379. (cited on page 2)

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, (2014). (cited on pages 20, 73, and 86)

Clauset, A.; Shalizi, C. R.; and Newman, M. E., 2009. Power-law distributions in empirical data. *SIAM review*, 51, 4 (2009), 661–703. (cited on pages 1 and 43)

Crane, R. and Sornette, D., 2008. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105, 41 (2008), 15649–15653. (cited on pages 2, 13, 16, 17, 42, 51, 64, 105, and 113)

Crane, R.; Sornette, D.; et al., 2008. Viral, quality, and junk videos on youtube: Separating content from noise in an information-rich environment. In *AAAI Spring Symposium: Social Information Processing*, 18–20. (cited on page 17)

Delay, D. and Vere-Jones, D., 2003. *An introduction to the theory of point processes. Vol. I*. Springer-Verlag. (cited on pages 16, 44, 46, and 75)

Ding, W.; Shang, Y.; Guo, L.; Hu, X.; Yan, R.; and He, T., 2015. Video popularity prediction by sentiment propagation via implicit network. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1621–1630. ACM. (cited on pages 3, 17, 24, 38, 64, and 67)

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L., 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the*

22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1555–1564. ACM. (cited on pages 3, 20, 74, 76, 78, 81, and 83)

Du, N.; Farajtabar, M.; Ahmed, A.; Smola, A. J.; and Song, L., 2015. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 219–228. ACM. (cited on page 71)

Du, N.; Song, L.; Rodriguez, M. G.; and Zha, H., 2013. Scalable influence estimation in continuous-time diffusion networks. In *Advances in neural information processing systems*, 3147–3155. (cited on page 81)

Elman, J. L., 1990. Finding structure in time. *Cognitive science*, 14, 2 (1990), 179–211. (cited on pages 7, 19, 73, 83, 84, and 86)

Farajtabar, M.; Wang, Y.; Rodriguez, M. G.; Li, S.; Zha, H.; and Song, L., 2015. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*, 1954–1962. (cited on page 71)

Filimonov, V. and Sornette, D., 2015. Apparent criticality and calibration issues in the hawkes self-excited point process model: application to high-frequency financial data. *Quantitative Finance*, 15, 8 (2015), 1293–1314. (cited on pages 16 and 42)

Fourer, R.; Gay, D. M.; and Kernighan, B. W., 1987. *AMPL: A mathematical programming language*. AT & T Bell Laboratories Murray Hill, NJ 07974. (cited on pages 65 and 66)

Gao, S.; Ma, J.; and Chen, Z., 2015. Modeling and predicting retweeting dynamics on microblogging platforms. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 107–116. ACM. (cited on pages 18, 64, and 67)

Gill, P.; Arlitt, M.; Li, Z.; and Mahanti, A., 2007. Youtube traffic characterization: a view from the edge. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 15–28. ACM. (cited on page 9)

Goel, S.; Watts, D. J.; and Goldstein, D. G., 2012. The structure of online diffusion networks. In *Proceedings of the 13th ACM conference on electronic commerce*, 623–638. ACM. (cited on pages 1 and 113)

Goldenberg, J.; Libai, B.; and Muller, E., 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12, 3 (2001), 211–223. (cited on page 13)

Gomez-Rodriguez, M.; Song, L.; Du, N.; Zha, H.; and Schölkopf, B., 2016. Influence estimation and maximization in continuous-time diffusion networks. *ACM Transactions on Information Systems (TOIS)*, 34, 2 (2016), 9. (cited on page 19)

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y., 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680. (cited on page 121)

Granovetter, M., 1978. Threshold models of collective behavior. *American journal of sociology*, 83, 6 (1978), 1420–1443. (cited on page 13)

Graves, A., 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, (2013). (cited on pages 7, 20, 73, 83, 84, and 86)

Hawkes, A. G., 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58, 1 (1971), 83–90. (cited on pages 6, 16, 38, 41, 62, and 84)

Hawkes, A. G. and Oakes, D., 1974. A Cluster Process Representation of a Self-Exciting Process. *J. Appl. Probab. '74*, 11, 3 (sep 1974), 493. (cited on pages 16, 61, and 76)

Helmstetter, A. and Sornette, D., 2002. Subcritical and supercritical regimes in epidemic models of earthquake aftershocks. *Journal of Geophysical Research: Solid Earth*, 107, B10 (2002), ESE–10. (cited on pages 16, 42, and 64)

Hirshleifer, D. and Hong Teoh, S., 2003. Herd behaviour and cascading in capital markets: A review and synthesis. *European Financial Management*, 9, 1 (2003), 25–66. (cited on page 2)

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9, 8 (1997), 1735–1780. (cited on pages 20, 73, 84, and 86)

Huang, T.-K.; Rahman, M. S.; Madhyastha, H. V.; Faloutsos, M.; and Ribeiro, B., 2013. An analysis of socware cascades in online social networks. In *Proceedings of the 22nd international conference on World Wide Web*, 619–630. ACM. (cited on page 121)

JAIN, A.; SINGH, A.; KOPPULA, H. S.; SOH, S.; AND SAXENA, A., 2016. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 3118–3125. IEEE. (cited on pages 19 and 73)

JONES, J. P., 1990. The double jeopardy of sales promotions. *Harvard business review*, 68, 5 (1990), 145–152. (cited on pages 90 and 96)

KALTENBRUNNER, A.; GOMEZ, V.; AND LOPEZ, V., 2007. Description and prediction of slashdot activity. In *2007 Latin American Web Conference (LA-WEB 2007)*, 57–66. IEEE. (cited on page 2)

KATZ, E. AND LAZARSFELD, P. F., 1955. Personal influence: the part played by people in the flow of mass communications. (1955). (cited on page 18)

KEELING, M. J. AND EAMES, K. T., 2005. Networks and epidemic models. *Journal of the Royal Society Interface*, 2, 4 (2005), 295–307. (cited on page 2)

KEMPE, D.; KLEINBERG, J.; AND TARDOS, É., 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146. ACM. (cited on page 19)

KINGMA, D. P. AND BA, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014). (cited on page 88)

KOBAYASHI, R. AND LAMBIOTTE, R., 2016. Tideh: Time-dependent hawkes process for predicting retweet dynamics. In *Tenth International AAAI Conference on Web and Social Media*. (cited on pages 18 and 64)

KONG, Q., 2019. Linking epidemic models and hawkes point processes for modeling information diffusion. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 818–819. ACM. (cited on page 122)

KWAK, H.; LEE, C.; PARK, H.; AND MOON, S., 2010. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, 591–600. AcM. (cited on pages 9, 10, and 43)

LESKOVEC, J.; MCGLOHON, M.; FALOUTSOS, C.; GLANCE, N.; AND HURST, M., 2007. Patterns of cascading behavior in large blog graphs. In *Proceedings of the 2007 SIAM international conference on data mining*, 551–556. SIAM. (cited on page 13)

LEWIS, E. AND MOHLER, G., 2011. A nonparametric em algorithm for multiscale hawkes processes. *Journal of Nonparametric Statistics*, 1, 1 (2011), 1–20. (cited on page 71)

LI, C.; MA, J.; GUO, X.; AND MEI, Q., 2017a. Deepcas: An end-to-end predictor of information cascades. In *Proceedings of the 26th international conference on World Wide Web*, 577–586. International World Wide Web Conferences Steering Committee. (cited on pages 20, 60, and 121)

LI, S.; GAO, X.; BAO, W.; AND CHEN, G., 2017b. Fm-hawkes: A hawkes process based approach for modeling online activity correlations. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1119–1128. ACM. (cited on pages 60, 118, and 121)

LIAO, D.; XU, J.; LI, G.; HUANG, W.; LIU, W.; AND LI, J., 2019. Popularity prediction on online articles with deep fusion of temporal process and content features. In *Thirty-Three AAAI Conference on Artificial Intelligence (AAAIâĂŹ19)*. (cited on pages 118 and 121)

LIN, T.; HORNE, B. G.; TINO, P.; AND GILES, C. L., 1996. Learning long-term dependencies in narx recurrent neural networks. *IEEE Transactions on Neural Networks*, 7, 6 (1996), 1329–1338. (cited on page 19)

LU, Y.; YU, L.; ZHANG, T.; ZANG, C.; CUI, P.; SONG, C.; AND ZHU, W., 2018. Collective human behavior in cascading system: Discovery, modeling and applications. In *2018 IEEE International Conference on Data Mining (ICDM)*, 297–306. IEEE. (cited on page 60)

MARTIN, T.; HOFMAN, J. M.; SHARMA, A.; ANDERSON, A.; AND WATTS, D. J., 2016. Exploring limits to prediction in complex social systems. In *Proceedings of the 25th International Conference on World Wide Web*, 683–694. International World Wide Web Conferences Steering Committee. (cited on pages 1, 2, 3, 11, 12, 25, 27, 28, 35, 37, 53, 81, 83, 105, 108, and 113)

MATSUBARA, Y.; SAKURAI, Y.; PRAKASH, B. A.; LI, L.; AND FALOUTSOS, C., 2012. Rise and fall patterns of information diffusion: model and implications. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 6–14. ACM. (cited on page 17)

MEI, H. AND EISNER, J. M., 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, 6754–6764. (cited on page 20)

Miritello, G.; Lara, R.; Cebrian, M.; and Moro, E., 2013. Limited communication capacity unveils strategies for human interaction. *Scientific reports*, 3 (2013), 1950. (cited on page 50)

Mishra, S.; Rizoiu, M.-A.; and Xie, L., 2016. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 1069–1078. ACM. (cited on pages 3, 23, 64, 65, 67, 71, 75, 76, 81, 83, 105, 113, and 117)

Mishra, S.; Rizoiu, M.-A.; and Xie, L., 2018. Modeling popularity in asynchronous social media streams with recurrent neural networks. In *Twelfth International AAAI Conference on Web and Social Media*. (cited on page 23)

Mohler, G. O.; Short, M. B.; Brantingham, P. J.; Schoenberg, F. P.; and Tita, G. E., 2011. Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106, 493 (2011), 100–108. (cited on page 46)

Myers, S. A. and Leskovec, J., 2012. Clash of the contagions: Cooperation and competition in information diffusion. In *2012 IEEE 12th international conference on data mining*, 539–548. IEEE. (cited on page 121)

Nazir, A.; Raza, S.; and Chuah, C.-N., 2008. Unveiling facebook: a measurement study of social network based applications. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, 43–56. ACM. (cited on pages 9 and 10)

Ogata, Y., 1999. Seismicity analysis through point-process modeling: A review. In *Seismicity patterns, their statistical significance and physical meaning*, 471–507. Springer. (cited on pages xvi, 46, 48, 62, 98, and 113)

Petrovic, S.; Osborne, M.; and Lavrenko, V., 2011. Rt to win! predicting message propagation in twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*. (cited on page 12)

Pintér, J. D., 2013. Lgo–a program system for continuous and lipschitz global optimization. *Developments in Global Optimization*, 18 (2013), 183. (cited on page 66)

Pinto, H.; Almeida, J. M.; and Gonçalves, M. A., 2013. Using early view patterns to predict the popularity of youtube videos. In *Proceedings of the sixth ACM international conference on Web search and data mining*, 365–374. ACM. (cited on pages 2, 3, 11, 12, 27, 28, 37, 82, 91, and 107)

Rizoiu, M.-A.; Mishra, S.; Kong, Q.; Carman, M.; and Xie, L., 2018. Sir-hawkes: Linking epidemic models and hawkes processes to model diffusions in finite populations. In *Proceedings of the 2018 World Wide Web Conference*, 419–428. International World Wide Web Conferences Steering Committee. (cited on pages 23, 64, 78, and 122)

Rizoiu, M.-A.; Xie, L.; Sanner, S.; Cebrian, M.; Yu, H.; and Van Hentenryck, P., 2017. Expecting to be hip: Hawkes intensity processes for social media popularity. In *Proceedings of the 26th International Conference on World Wide Web*, 735–744. International World Wide Web Conferences Steering Committee. (cited on pages xi, xxi, 2, 3, 18, 23, 28, 29, 67, 75, 82, 84, 85, 89, 91, 92, and 107)

Rizoiu, M.-A. and Xie, L. X., 2017. Online popularity under promotion: Viral potential, forecasting, and the economics of time. In *Eleventh International AAAI Conference on Web and Social Media*. (cited on pages 20, 89, 90, and 96)

Rodriguez, M. G.; Balduzzi, D.; and Schölkopf, B., 2011. Uncovering the temporal dynamics of diffusion networks. (cited on page 16)

Romero, D. M.; Tan, C.; and Ugander, J., 2013. On the interplay between social and topical structure. In *Seventh International AAAI Conference on Weblogs and Social Media*. (cited on page 12)

Roy, S. D.; Mei, T.; Zeng, W.; and Li, S., 2013. Towards cross-domain learning for social video popularity prediction. *IEEE Transactions on multimedia*, 15, 6 (2013), 1255–1267. (cited on page 20)

Salganik, M. J.; Dodds, P. S.; and Watts, D. J., 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311, 5762 (2006), 854–856. (cited on page 1)

Shamma, D. A.; Yew, J.; Kennedy, L.; and Churchill, E. F., 2011. Viral actions: Predicting video view counts using synchronous sharing behaviors. In *Fifth International AAAI Conference on Weblogs and Social Media*. (cited on pages 11 and 38)

Shen, H.; Wang, D.; Song, C.; and Barabási, A.-L., 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Twenty-eighth AAAI conference on artificial intelligence*. (cited on pages 2, 3, 17, 18, 38, 51, 64, 67, 71, 81, 105, and 113)

SORNETTE, D. AND HELMSTETTER, A., 2003. Endogenous versus exogenous shocks in systems with memory. *Physica A: Statistical Mechanics and its Applications*, 318, 3-4 (2003), 577–591. (cited on page 71)

SRIVASTAVA, A.; CHELMIS, C.; AND PRASANNA, V. K., 2015. Social influence computation and maximization in signed networks with competing cascades. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 41–48. IEEE. (cited on page 121)

SUH, B.; HONG, L.; PIROLLI, P.; AND CHI, E. H., 2010. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *2010 IEEE Second International Conference on Social Computing*, 177–184. IEEE. (cited on page 12)

SUTSKEVER, I.; VINYALS, O.; AND LE, Q. V., 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112. (cited on pages 7, 19, and 83)

SZABO, G. AND HUBERMAN, B. A., 2010. Predicting the popularity of online content. *Communications of the ACM*, 53, 8 (2010), 80–88. (cited on pages 2, 3, 11, 12, 27, 28, 51, 82, 91, and 107)

TSAGKIAS, M.; WEERKAMP, W.; AND DE RIJKE, M., 2009. Predicting the volume of comments on online news stories. In *Proceedings of the 18th ACM conference on Information and knowledge management*, 1765–1768. ACM. (cited on pages 105 and 113)

TSUR, O. AND RAPPOPORT, A., 2012. What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the fifth ACM international conference on Web search and data mining*, 643–652. ACM. (cited on page 12)

TURNER, V.; GANTZ, J. F.; REINSEL, D.; AND MINTON, S., 2014. The digital universe of opportunities: Rich data and the increasing value of the internet of things. *IDC Analyze the Future*, 16 (2014). (cited on page 1)

VINYALS, O.; TOSHEV, A.; BENGIO, S.; AND ERHAN, D., 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164. (cited on page 19)

WÄCHTER, A. AND BIEGLER, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106, 1 (2006), 25–57. (cited on pages 45, 65, and 66)

Wallinga, J. and Teunis, P., 2004. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of epidemiology*, 160, 6 (2004), 509–516. (cited on pages 16, 43, and 64)

Wang, Y.; Liu, S.; Shen, H.; Gao, J.; and Cheng, X., 2017a. Marked temporal dynamics modeling based on recurrent neural network. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 786–798. Springer. (cited on page 20)

Wang, Y.; Ye, X.; Zhou, H.; Zha, H.; and Song, L., 2017b. Linking micro event history to macro prediction in point process models. In *Artificial Intelligence and Statistics*, 1375–1384. (cited on page 20)

Watts, D. J. and Dodds, P. S., 2007. Influentials, networks, and public opinion formation. *Journal of consumer research*, 34, 4 (2007), 441–458. (cited on page 18)

Wu, Q.; Yang, C.; Zhang, H.; Gao, X.; Weng, P.; and Chen, G., 2018a. Adversarial training model unifying feature driven and point process perspectives for event popularity prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 517–526. ACM. (cited on pages 60, 118, and 121)

Wu, S.; Hofman, J. M.; Mason, W. A.; and Watts, D. J., 2011. Who says what to whom on twitter. In *Proceedings of the 20th international conference on World wide web*, 705–714. ACM. (cited on page 10)

Wu, S.; Rizoiu, M.-A.; and Xie, L., 2018b. Beyond views: Measuring and predicting engagement in online videos. In *Twelfth International AAAI Conference on Web and Social Media*. (cited on pages 12 and 31)

Xiao, S.; Yan, J.; Li, C.; Jin, B.; Wang, X.; Yang, X.; Chu, S. M.; and Zhu, H., 2016. On modeling and predicting individual paper citation count over time. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2676–2682. AAAI Press. (cited on page 71)

Xiao, S.; Yan, J.; Yang, X.; Zha, H.; and Chu, S. M., 2017. Modeling the intensity function of point process via recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*. (cited on pages 20 and 121)

Xu, H.; Wu, W.; Nemati, S.; and Zha, H., 2016. Patient flow prediction via discriminative learning of mutually-correcting processes. *IEEE transactions on Knowledge and Data Engineering*, 29, 1 (2016), 157–171. (cited on page 71)

Yang, J. and Leskovec, J., 2011. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, 177–186. ACM. (cited on page 17)

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E., 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 1480–1489. (cited on page 121)

Yu, H., 2015. Understanding the popularity evolution of online media: A case study on youtube videos. (2015). (cited on page 9)

Yu, H.; Xie, L.; and Sanner, S., 2014. Twitter-driven youtube views: Beyond individual influencers. In *Proceedings of the 22nd ACM international conference on Multimedia*, 869–872. ACM. (cited on pages 11 and 38)

Yu, H.; Xie, L.; and Sanner, S., 2015. The lifecyle of a youtube video: Phases, content and popularity. In *Ninth International AAAI Conference on Web and Social Media*. (cited on page 82)

Yu, L.; Cui, P.; Wang, F.; Song, C.; and Yang, S., 2017. Uncovering and predicting the dynamic process of information cascades with survival model. *Knowledge and information systems*, 50, 2 (2017), 633–659. (cited on pages 3, 17, 24, 38, and 81)

Zarezade, A.; Upadhyay, U.; Rabiee, H. R.; and Gomez-Rodriguez, M., 2017. Redqueen: An online algorithm for smart broadcasting in social networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 51–60. ACM. (cited on page 64)

Zhao, Q.; Erdogdu, M. A.; He, H. Y.; Rajaraman, A.; and Leskovec, J., 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1513–1522. ACM.

(cited on pages xvii, 2, 3, 6, 18, 24, 25, 38, 49, 51, 52, 53, 57, 58, 59, 64, 67, 71, 76, 81, 83, 105, and 113)

Zhou, K.; Zha, H.; and Song, L., 2013. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*, 1301–1309. (cited on page 71)