UNDERSTANDING, MODELING AND PREDICTING THE POPULARITY OF ONLINE CONTENT ON SOCIAL MEDIA APPLICATIONS

FLAVIO VINICIUS DINIZ DE FIGUEIREDO

UNDERSTANDING, MODELING AND PREDICTING THE POPULARITY OF ONLINE CONTENT ON SOCIAL MEDIA APPLICATIONS

Tese apresentada ao Programa de Pós-Graduação em Computer Science do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Computer Science.

Orientadora: Jussara Marques Almeida

Belo Horizonte
Junho de 2015

FLAVIO VINICIUS DINIZ DE FIGUEIREDO

UNDERSTANDING, MODELING AND PREDICTING THE POPULARITY OF ONLINE CONTENT ON SOCIAL MEDIA APPLICATIONS

Thesis presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Jussara Marques Almeida

Belo Horizonte
June 2015

Diniz de Figueiredo, Flavio Vinicius

F475u Understanding, Modeling and Predicting the Popularity of Online Content on Social Media Applications / Flavio Vinicius Diniz de Figueiredo. — Belo Horizonte, 2015

xxiv, 178 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais

Orientador: Jussara Marques Almeida

1. popularity. 2. trends. 3. time series. 4. user generated content. 5. ugc. 6. youtube. 7. lastfm. 8. twitter. 9. facebook. 10. social media. I. Título.

CDU 519.6*04(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Understanding, modeling and predicting the popularity of online content on social media applications

FLAVIO VINICIUS DINIZ DE FIGUEIREDO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Jussara Marques de Almeida Gonçalves
PROFA. JUSSARA MARQUES DE ALMEIDA GONÇALVES - Orientadora
Departamento de Ciência da Computação - UFMG

Ana Paula Couto da Silva
PROFA. ANA PAULA COUTO DA SILVA
Departamento de Ciência da Computação - UFMG

Anirban Mahanti
PROF. ANIRBAN MAHANTI
NICTA - Austrália

PROF. CAETANO TRAINA JÚNIOR
Instituto de Ciências Matemáticas e de Computação - USP

Fabrício Benevenuto de Souza
PROF. FABRÍCIO BENEVENUTO DE SOUZA
Departamento de Ciência da Computação - UFMG

Marcos André Gonçalves
PROF. MARCOS ANDRÉ GONÇALVES
Departamento de Ciência da Computação - UFMG

PROF. RENATO MARTINS ASSUNÇÃO
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 01 de junho de 2015.

Agradecimentos

Antes de tudo, agradeço a minha família: Mother, Father, Fabricio, Vinicius (o mais novo membro), Dudu e Aline, por todo o apoio ao longo dos anos. Sou grato também por ter tido Vanessa como companheira durante todo este tempo.

Agradeço aos amigos/amigas por todos os momentos de: felicidade, bebidas, bares, festivais, shows, caronas, viagens e conversas (aleatórias ou filosóficas). Tais experiências me guiaram para eu ser a pessoa que sou hoje. Para evitar esquecer de alguém, prefiro mencionar que sou grato os membros dos grupos: Sem Sintomas, LSD, NetsysLab-UBC, Lê Cabral, VoD e Vodinho. Como também a segunda família da casa do Seu João, por facilitar minha vinda até BH. Além disso, agradeço aquelas pessoas que me abrigaram quando estava fora do país, tomando um papel mais amplo do que apenas "room mates" e sim de amizade.

Por fim, agradeço também a Jussara Almeida, pela orientação nos últimos 7 anos, como também os vários outros orientadores que me ajudaram durante o doutorado: Fabrício Benevenuto, Christos Faloutsos, Marcos Gonçalves, Krishna Gummadi e Bruno Ribeiro. Como não pode faltar, agradeço também ao Cnpq, Capes, Google Brasil e IBM Research pelo financiamento durante os últimos 5 anos.

Resumo

Hoje em dia, o fenômeno denominado de mídia social emergiu como a forma predominante de publicação de conteúdo na Internet. Devido a esse grande sucesso, um entendimento de como os usuários criam, compartilham e disseminam conteúdo online pode trazer informações cruciais para criadores de conteúdo, provedores de Internet, marqueteiros online, dentre outros. Neste contexto, essa tese discute três principais objetivos sobre como a popularidade de mídia social evolui online. Inicialmente, apresentamos um estudo sobre como diferentes atributos textuais, sociais e do próprio conteúdo se relacionam com a popularidade do conteúdo de mídia social. Este estudo é feito com base em uma caracterização em larga escala do YouTube, a principal aplicação de compartilhamento de vídeos hoje em dia, como também com base em um estudo com usuários usando a ferramenta de crowdsourcing Amazon Mechanical Turk. No nosso segundo objetivo, propomos diferentes métodos de previsão de popularidade com objetivos de prever tanto a evolução de popularidade (ou tendências), como também valores futuros de popularidade do conteúdo de mídia social. Diferentemente de outros trabalhos, levamos em conta o equilíbrio entre o interesse restante no conteúdo após a predição e corretude das previsões, um fator negligenciado por abordagens anteriores de previsão. Por fim, apresentamos um estudo de como atividades dos usuários (e.g., assistir, compartilhar, curtir etc.) se relacionam com a popularidade do conteúdo de mídia social. Este terceiro trabalho é feito com bases de dados do YouTube, Twitter e do Last.FM. Na nossa análise, focamos em duas características complementares do comportamento de usuários: a revisita ao um mesmo conteúdo ao longo do tempo, como também as mudanças de interesse em conteúdos distintos ao longo do tempo. Os resultados dessa tese são discutidos com uma enfâse de aplicações como marketing online, provisionamento de conteúdo e plataformas de dados analíticos.

Abstract

Social media has emerged as the de-facto form of publishing on the Internet nowadays. Given the success of social media applications, understanding how users create, share and disseminate social media content online can provide valuable insights for content generators, online advertisers and Internet service providers (ISPs), among others. Motivated by this great success of social media applications, the objectives of this dissertation are threefold. Firstly, we aim at understanding how different textual, content and social features relate to the evolution of popularity of social media content. We achieve this based on a large scale characterization of the YouTube application, currently the largest video sharing platform, as well as a small scale crowdsourced user study. Secondly, we propose novel popularity prediction methods to predict not only future popularity *values*, but also the popularity evolution *trends* that social media content will achieve at future dates. Our proposed methods differ from previous work in two key aspects: (1) popularity trends are exploited to build specialized models of popularity values, and (2) our methods take into account not only the prediction accuracy, but also the remaining interest in the content after prediction, aiming at finding a good tradeoff between both. Lastly, we present two novel data mining techniques to understand how user activities (e.g., viewing, liking, sharing etc.) relate to the evolution of popularity of social media content. In this last step, we tackle two complementary effects of user activities: the revisit behavior of users to the same content, as well as the attention flows of users between different pieces of content. Our case studies on this last step are three different social media applications: YouTube, Twitter and LastFM. The three complementary studies presented in this dissertation are discussed in light of real world applications (e.g., advertising, provisioning and analytics platforms) that may benefit from our results.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

As we (humans) shift our content consumption and production practices to an online space (the Internet) [83, 105, 108], massive amounts of data on human behavior and cultural production is now readily available to aid researchers in understanding our information consumption habits. For example, social media and user generated content (UGC)[1], that is, media content which can (at least in theory) be created and/or curated by any user, is one of the driving forces of today's Internet. In other words, social media has become the de-facto form of publishing on some of the most popular Internet applications [31]. Focusing on video content, websites such as YouTube[2] receive over 1 billion unique users and attracts over 1 million different advertisers each month [151]. Even niche applications, such as Vimeo[3], whose target producers are independent filmmakers, manage to attract over 70 million unique users monthly [134]. Changing our perspective to other kinds of media, Flickr[4], a popular photo sharing application, announced in 2011 that over 6 billion photos were available in the application [48].

Focusing on how users consume online content nowadays, popular news websites rely not only on traditional advertising and subscriptions to drive traffic. There is also a heavy use of social media and viral marketing [18] campaigns that lead users to this professional content online. That is, user traffic, even to professional content nowadays is currently highly dependent on how information is propagated through users on social media applications as well on online and offline social networks. Another interesting

[1] We consider social media as a more general concept than UGC. It encompasses the technology that allows users to create, share, discuss, and curate online information. User generated content (UGC) the actual media content (e.g., videos) created by users and published/propagated through social media tools (see Table 1.1).

[2] `http://youtube.com`

[3] `http://vimeo.com`

[4] `http://flickr.com`

example is online music streaming services (OMSS) such as Spotify[5] and Last.FM[6], whose revenues depends on the online consumption habits of users, or simply, online user activities. Such websites usually rely on social networking and group geared features to attract and maintain their target audiences.

Given the success of some of the aforementioned applications – as well as the current large volume of online user activities in creating, curating, and consuming different types of content on a daily basis – understanding how users find such content and how content popularity evolves over time provides valuable insights for content generators, online advertisers and Internet service providers (ISPs), amongst others. For instance, from a system perspective, understanding these properties may drive the design of better analytic tools, a major market segment nowadays [83]. Online advertisers may also benefit from this information to better place contextual advertisements, while ISPs could exploit it to develop more cost-effective content delivery platforms and caching systems. From a social perspective, understanding the properties of content popularity could be used to better comprehend the human dynamics of consumption processes [34]. Also, content producers could benefit from insights on how user collaboration and collaborative social activities on Web 2.0 applications may impact content popularity, providing information on aspects related to their own fame on Internet applications.

The motivating theme of this dissertation is on understanding and modeling the popularity of online content on social media applications (e.g., UGC, as well as news websites or mainstream songs from musical artists), while at the same time providing valuable information to real world applications.

1.1 Motivation

In Figure 1.1 we show an example of a YouTube video, as well as the different features – such as the textual title and content's number of likes – as well as different user actions (e.g., web links through which users perform some action such as sharing the video) that surrounds the video. We define the content provided by the social media application, a YouTube video in this case, as a social media object. There exists a wide range of means to access an object such as the one shown in Figure 1.1. For example, search engines typically index such objects taking into account the different features available in the object (mostly the textual features [12]). Also, social sharing options provide users with means to propagate objects to their online friends or followers (e.g.,

[5] http://spotify.com

[6] http://last.fm

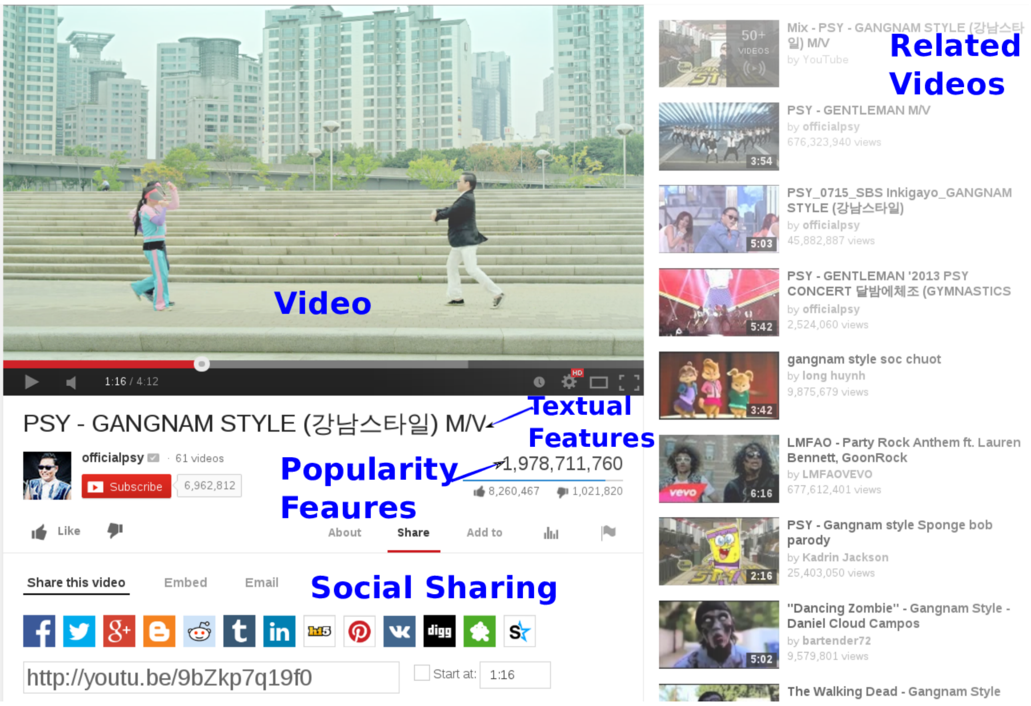

Related Videos
Video
PSY - GANGNAM STYLE (강남스타일) M/V
Textual Features
Popularity Feaures
1,978,711,760
Social Sharing
http://youtu.be/9bZkp7q19f0
Mix - PSY - GANGNAM STYLE (강남스타일) M/V
PSY - GENTLEMAN M/V
PSY_0715_SBS Inkigayo_GANGNAM STYLE (강남스타일)
PSY - GENTLEMAN '2013 PSY CONCERT 달밤에체조 (GYMNASTICS
gangnam style soc chuot
LMFAO - Party Rock Anthem ft. Lauren Bennett, GoonRock
PSY - Gangnam style Sponge bob parody
"Dancing Zombie" - Gangnam Style - Daniel Cloud Campos
The Walking Dead - Gangnam Style

Figure 1.1: Example of a Video's Page on YouTube.

online social network (OSN) sharing or e-mail sharing). [7]. Moreover, different websites such as blogs or news websites can also embed online objects on their pages[8]. Finally, internal browsing inside the application will also direct users to different content (as shown in the related videos links in the figure).

The aforementioned means of accessing objects will reflect on the different temporal patterns of evolution of popularity that social media content typically exhibit. Figures 1.2(a-c) shows some example time series of the daily popularity (number of hits/views) received by different YouTube videos. Also, Figures (d-f) shows some examples of popularity of Twitter hashtags. From the figures we can see different patterns of temporal evolution of popularity of objects. Take for instance, Figure 1.2(a). This example shows a YouTube music video that appears to have reached a steady state in the number of daily visits after an initial burst in growth (from zero to roughly 10,000 views daily). In contrast, Figure 1.2(b) shows a video that undergoes a popularity evolution consistent with a viral growth like pattern (exponential growth before the peak and exponential decay after). The last video example, Figure 1.2(c), appears to be composed of various viral patterns that relate to different real world events or incoming sources/links (referrers). In this specific case, the video is a song about New

[7] Offline sharing is also possible, but very difficult to account for.

[8] With embedding external content (the object) from a different provider is shown a webpage using specific HTML code.

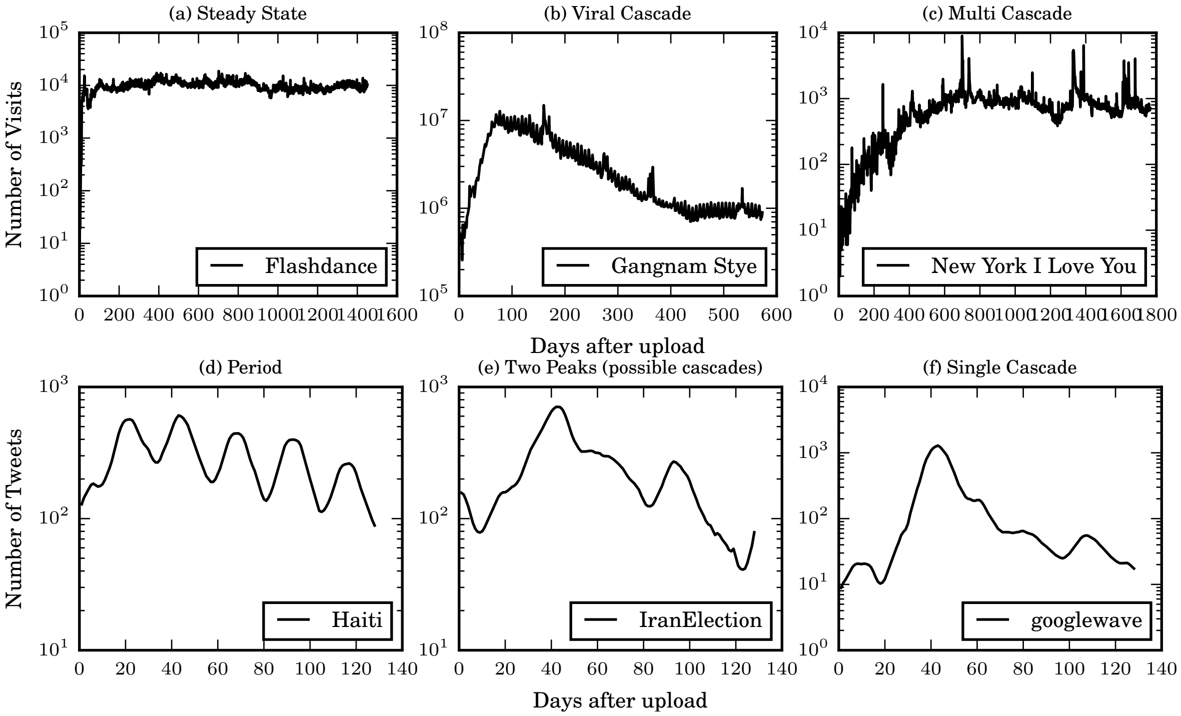


Figure 1.2: Example of Popularity Time Series for YouTube (top row) and Twitter (bottom).

York city. Looking into the different peaks in popularity, or spikes, they occur close to dates such as holidays and to the real world events related to the song (e.g., album release). Figures 1.2(d-f) shows examples of the popularity evolution of Twitter hashtags focused on the 128 hours around the peak hour (most popular hour). The first example, Figure 1.2(d), shows a hashtag whose popularity curve follows a periodic behavior. The other two Twitter examples show that before the peak, a exponential like growth is achieved by the hashtags. After the peak, either exponential or long-tailed decays can be seen depending on the example. In some cases, such as in Figure 1.2(e), a smaller cascade seems to follow up on the larger one. As shown by previous work, such cascade patterns (exponential growth followed by an exponential or long tail decay) are usually caused by viral like propagation [64, 91].

These different patterns, shown in the examples above, are interesting because they reveal important information on the evolution of popularity of social media objects. In this dissertation, we are specially interested in the events that cause such trends. More importantly, we want to find evidence of correlations between the trends above and different features of the object. In particular, we are interested in the impact of referrers (i.e., incoming links), since they reflect the propagation of the object online, as well as the impact of real world events, such as album releases from musical artists, on the popularity evolution.

> **Empirical Hypothesis** *In sum, the guiding hypothesis we evaluate throughout this dissertation is: Given the different features which may surround social media objects online – particularly the referrers (e.g., incoming links to objects) and features related to real world events (e.g., album releases) – will such features have an impact on the popularity evolution of the social media objects?*

Nevertheless, such a hypothesis is still very broad, and we now narrow it down as a problem statement and specific research goals in the next section.

1.2 Problem Statement

In general terms, the problem we intend to address is defined as follows. Given a collection O of social media objects, say YouTube videos, we define for each object $o \in O$ a set of popularity, content, social, referrer and real world features related to o. Popularity features account for measures of popularity, say views or comments over time. Content features can be textual information associated with the object, such as tags and object descriptions, or even how users perceive the quality of the content with ratings and likes. Social features are related to the user who posted the content and her online social network. For example, a followers or friends network in the application. Referrer features contain information about the incoming links exploited by users to reach object o from other websites. Finally, some features are related to external real world events such as the dates of album releases by music artists. We call these features real world features.

Throughout this dissertation, we shall mine evidence of how the importance of each of these features affect the evolution of popularity of objects in different social media applications. We are particularly interested in referrer features and external events (e.g., album releases by artists) since they provide valuable information on how users reach online information objects. These are two pieces of information largely neglected by previous work. As stated, we want to make use of these features on different data mining applications such as: (1) time series (or object) clustering and classification; (2) popularity trend and value prediction; and (3) modeling and mining the user activities that cause the different popularity trends. In more details, we consider that the information related to each of these features is available only up to the reference time t. When clustering time series, we want to make use of the information

Table 1.1: Definitions of Social Media, User Generated Content, Online Information and User Activities

	Definition	**Examples**
Social Media	Technology that enables online social behavior	Blogs, Video Blogs, Music Sharing, Ratings, Crowdsourcing
User Generated Content	Content generated by users	YouTube Videos, Blog posts, Microposts. Specific case of social media objects.
Social Media Objects	Memes and pieces of information propagated through social media	UGC, online news, online music, as well as more specific information such as text quotations, video frames
User Activities	User behavior (e.g., posting, liking, viewing, sharing etc.) that cause the online popularity of objects	Posting, sharing, commenting etc.

of the features up to a reference time t_r. In the case of popularity prediction, we aim at exploiting features to determine the popularity of objects at a target time $t_t = t_r + \delta$.

This general objective is narrowed down into three specific research goals we aim to achieve in this dissertation. Before introducing them in the next section, we present our definitions of *social media*, *user generated content*, *user activities*, and *social media objects*, which will be used throughout the remainder of this dissertation. These definitions are presented in Table 1.1. Social media is a broader concept which defines the tools, technologies and applications that allows users to create and share social media objects. User generated content (UGC) is the actual content, or media, generated by users. Examples of UGC are YouTube video's or blog posts. Online objects defines a piece of content, for instance a piece of UGC of even a snippet of content (e.g., a quotation), that is propagated through users online. Finally, user activities defines the user actions which cause online popularity (e.g., viewing and sharing content).

1.3 Research Goals

RG1 - Understanding Feature Importance: Our first research goal focuses on: (1) characterizing how object popularity evolves over time; (2) characterizing how object popularity evolution correlates with the referrers that most often lead users to objects (as well as with other content, social and popularity features); and, (3) modeling the evolution of popularity of objects. We note that, unlike previous work, that correlated different features of objects with final popularity [13, 139], here we are concerned with measuring the impact of referrers on how the popularity of each object evolves over time [91, 139]. At the same time we exploit this information in order to model the popularity evolution of individual objects. As a basis for comparison, we also study: (1) popularity features, that are related to temporal data about the evolution of popularity of individual objects; and (2) content features, such as the category of a video. In particular, we study how users perceive the content of social media objects and how such perception correlates with popularity.

RG2 - Predicting Object Popularity: After data characterization and modeling, we intend to exploit the available data to answer the following question: Is it possible to predict how the popularity of individual objects evolves over time? In other words, we want to know if it is possible to predict the popularity curve (or trend) of each object. We also investigate whether more effective methods to predict the popularity measures (e.g., views) of an object at a target date can be devised. This is done by exploiting the developed popularity trend prediction models (e.g., by building specialized models to pre-defined popularity trends). Our results showed that we can indeed improve popularity prediction models using trend prediction models. More importantly, unlike previous studies [4, 81, 110, 129, 150], we shall focus not only on predicting the popularity of a video at time $t_t = t_r + \delta$, but also on the evolution its popularity it may follow after prediction.

RG3 - Modeling and Mining Popularity Through User Activities: In our final research goal, we turn our focus on mining user activities on social media websites. User activities, such as tweeting and listening to songs, are the actions that eventually account for popularity. Thus, instead of looking at popularity through the use of time series only (as was mostly done in RG1 and RG2), in this third goal we change our emphasis to the user. Our first goal here is on understanding how the repeated-consumption behavior of users, or revisits,

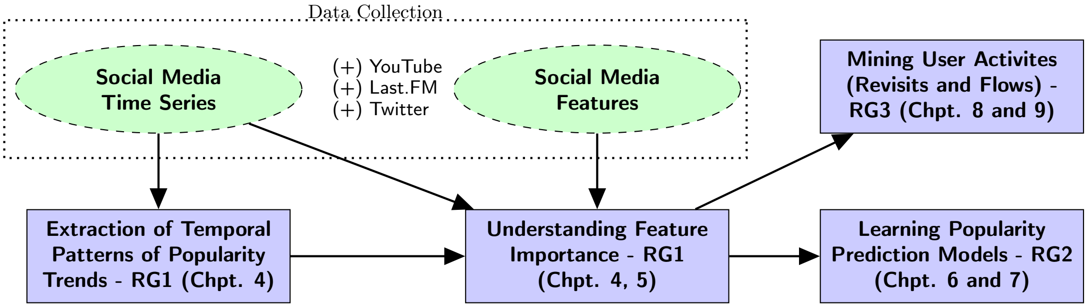

Data Collection
Social Media Time Series
(+) YouTube
(+) Last.FM
(+) Twitter
Social Media Features
Mining User Activites (Revisits and Flows) - RG3 (Chpt. 8 and 9)
Extraction of Temporal Patterns of Popularity Trends - RG1 (Chpt. 4)
Understanding Feature Importance - RG1 (Chpt. 4, 5)
Learning Popularity Prediction Models - RG2 (Chpt. 6 and 7)

Figure 1.3: Pictorial Representation of our Research Goals

impacts popularity. The revisit behavior has important implications on the social media application as it allows us to break popularity down into audience (unique users) and revisits (returning users). For instance, marketing services should care most about the audience of a particular content, as opposed to its total popularity, as each access does not necessarily represent a new exposed individual. After our study on revisits specifically, we broaden our view to the attention flows of users. In contrast to the "stickyness", which is captured by revisits, attention flows also capture how users change attention from one object to another. One interesting research question we tackle in here is whether objects compete or collaborate for user attention. Understanding object popularity from a competition and collaboration perspective is a novel task [89, 118, 119, 132], and brings important insights on how popularity evolves online.

1.4 Contributions and Outline of this Dissertation

In Figure 1.3 we show how the research goals of this dissertation relate to each other. We also emphasize the chapters in which each research goal is addressed. The rest of this dissertation is organized as follows. Chapter 2 discusses our related work and the background required for understanding the rest of the dissertation. Our main contributions are organized in the following chapters:

Chapter 3 Chapters 3 and 4 will focus on understanding feature importance to content popularity in the UGC application YouTube. Specifically on Chapter 3, we present a characterization of different aspects of popularity growth of YouTube videos from a service point of view. In order to achieve this, we collected three distinct datasets from the application and characterized different aspects of popularity growth, including: the time to achieve most of views; the fraction of views in the popularity peaks; and finally, which incoming links (i.e, referrers) most often lead users to such videos. Also, using clustering techniques, we extract the most common popularity evolution trends followed by videos. We then correlated different features from the videos with the trends and popularity values of videos. Our results are crucial to understand feature importance. These results provides the base knowledge needed to understand our findings in the next chapters.

Chapter 4 A study on the users' individual perceptions of YouTube videos and how these perceptions are connected with popularity. That is, we employ crowdsourcing tools and user surveys to understanding the relationships between explicit users' feedback on content and content popularity. In this chapter we tackle two simple but fundamental questions: (1) Can users reach consensus on a video they prefer from a pair of videos? (2) If consensus is reached, is the preferred video by the users the most popular one? Our goal on tackling these two questions is to to shed a light on the importance of users' perception on the popularity of objects. The first question aims at answering if a group of users will prefer a single piece of content (object). Whereas the second, aims at finding the relationship between the content (preferred by users) and the popularity.

Chapter 5 On Chapter 5 we begin our study on RG2. Specifically on this chapter, we build novel methods of popularity prediction. The case study of this chapter on News content that is shared on social media applications. In the news setting, there is a clear motivation to determine the future popularity (e.g., in two days) of an object, using only the information available shortly after the upload (e.g., a few hours) [18]. To predict popularity, we make use of a combined learning approach to: (1) extract popularity trends from popularity time series; and, (2) predict the popularity values of newly uploaded objects using these trends. The first step makes use of clustering techniques and represents each trend by a time series centroid. Our combined approach is quite effective, as it achieves results better than state-of-the-art baseline approaches. This prediction technique was also the winner on two out of three tasks of the ECML/PKDD 2014 Predictive Analytics Challenge.

Chapter 6 While news pages have a clear popularity prediction target time due to the timely nature of the content [18] (e.g., two days after the first hour), this target time definition is less clear in some social media settings (e.g., UGC). One example of the complexity behind social media popularity is the YouTube video of *Henri, le Chat-Noir*[9]. The first video of Henri was uploaded in 2007 and

[9] http://www.youtube.com/user/HenriLeChatNoir

remained in obscurity for years. However, in 2012 a user of the Tumblr social network found the video and posted it online[10]. Currently, the video has millions of visits from a wide range of different sources (e.g., OSNs, search engines, word-of-mouth and so forth). Motivated by such examples, and our understanding of UGC popularity from Chapters 3 and 4, we here create the TRENDLEARNER method. The goal of TRENDLEARNER is to predict the popularity trends of UGC. Moreover, TRENDLEARNER aims at capturing the tradeoff between the remaining interest in objects after prediction and the accuracy of predictions. Remaining interest captures how many views an object will receive after prediction was performed. In the UGC setting, capturing the remaining interest is important since it is unclear when an object will begin to become popular and/or interesting. This is a key contribution of our TRENDLEARNER method.

Chapter 7 Starting from this chapter we begin our study on RG3. In this chapter specifically, we focus on understanding the revisit behavior of users to social media objects. Repeated consumption, or revisits, account for a large fraction of the total popularity online [5]. Understanding how and why users evisit a single object is crucial for accurate popularity evolution models. Based on this characterization, we derive the PHOENIX-R model. We show how this model accurately captures the evolution of popularity of objects. An important facet of PHOENIX-R is that it also accurately models multiple bursts, or cascades, of visits to a single objects. As we have discussed, such bursts are related to referrers and real world events. Thus, by modelling them we take a step forwad in understanding their impact in popularity. Finally, our PHOENIX-R model is more accurate than state-of-the art competitors and can also be used for popularity prediction.

Chapter 8 Finally, we propose the A-FLUX approach, a user attention mining method designed to cope with the complex challenges of mining user visits to social media objects. One of the main technical contributions of A-FLUX is a probabilistic graphical model that captures the latent object-to-object transitions of user attention. We employ A-FLUX on large music streaming datasets, revealing interesting and meaningful user attention flow maps and patterns.

Finally, Chapter 9 concludes this dissertation and presents a discussion on directions for future work.

[10] `http://knowyourmeme.com/memes/henri-le-chat-noir`

Chapter 2

Background and Related Work

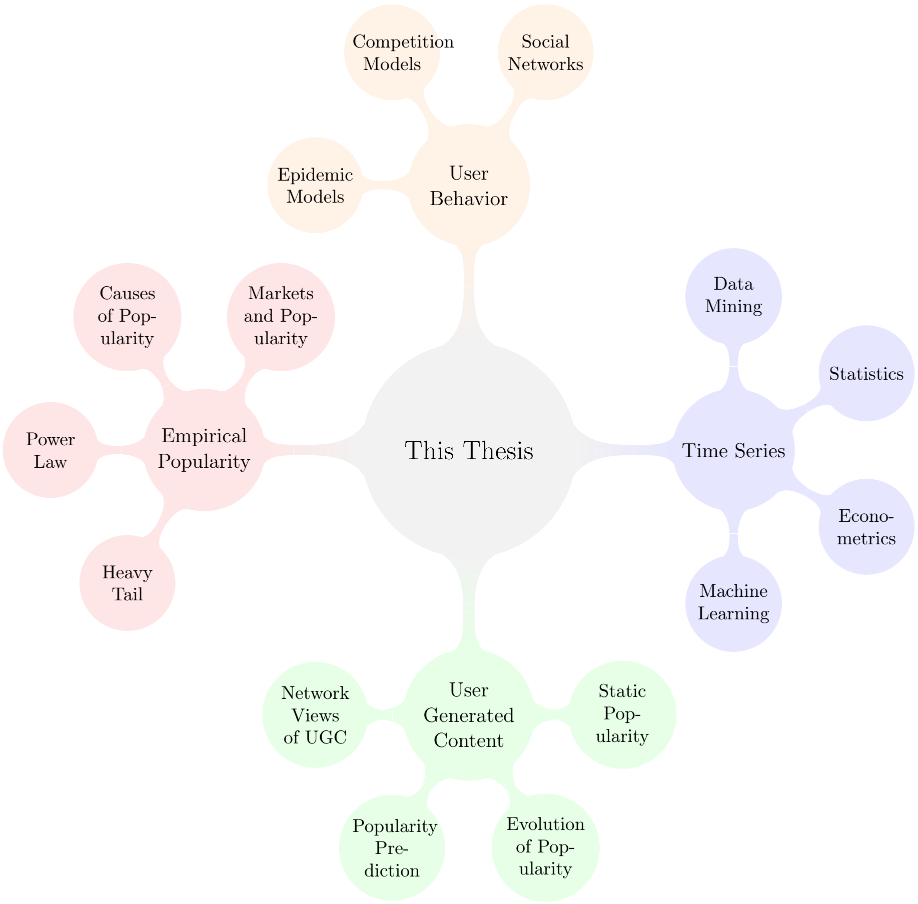


Figure 2.1: Mind Map of the Themes Related to this Dissertation

In this chapter, we present a summary of background knowledge and related work that is fundamental to the understanding of this dissertation. We discuss previous efforts related to each of the three research goals presented in Chapter 1, which can be grouped in to several topics as shown in Figure 2.1. In Section 2.1, we discuss empirical studies on popularity in general. Afterwards, in Section 2.2, we shift our focus to studies on user generated content, the type of content that is mostly explored in this thesis. This section is narrowed down as follows:

1. in Section 2.2.1 we discuss previous characterizations of social media objects popularity as a single static measure;
2. previous analyzes of the temporal evolution of social media objects popularity over time are discussed in Section 2.2.2;
3. Section 2.2.3 discusses popularity prediction models for social media;

We discuss some of the more recent approaches in understanding popularity through user activities and social network datasets through the use of competition and collaboration models, as well as epidemics based models, in Section 2.3. Finally, we provide a more in-depth summary of time series statistics and data mining techniques that are used by our research, or can be exploited in future developments, in Section 2.4. We summarize this chapter in Section 2.5.

2.1 Empirical Studies on Popularity

We begin our discussion with a brief summary of previous studies of general characteristics of popularity of different "quantities" of human knowledge (e.g., film revenues, book sales, votes received by political candidates, among others). This brief introduction, presented in Section 2.1.1, is useful to understand the nature of heavy-tailed distributions, which are used to characterize different properties of popularity of online information, such as the distribution of final popularity and decay in popularity over time. This is presented in Section 2.1.1. Afterwards, in Section 2.1.2 we review some efforts that analyzed the effects of different market and social factors that impact popularity.

2.1.1 Probability Distributions of Popularity

In the context of popularity, heavy-tailed distributions have been commonly used to explain different popularity phenomena. More specifically, a heavy-tailed distribution has the tail lower bounded by an exponential distribution, that is, their probability density function will decay slower than a exponential distribution [103]. Such distributions are suitable for popularity analysis, since, among other things, they accurately capture the behavior of most quantities having very low popularity, while very few contents will have very high popularity (hundreds of millions or even billions of views). Two heavy-tailed probability distributions are often used to quantify the popularity of different quantities are Power-law distributions [103] or the Log-normal distribution [28, 126]. In general, a quantity drawn from a Power-law distribution will have the probability density function:

$$p(x; C, \alpha) = Cx^{-\alpha}, \tag{2.1}$$

with any positive α, but typically falling the range $2 < \alpha < 5$. C is a normalization constant. Clauset *et al.* and Newman provide substantial studies on the nature of such distributions [28, 103]. In almost all of the cases the distribution above will only be observed for values of $x > x_{min}$ [28]. A Log-normal distribution has the form:

$$p(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}}\epsilon^{-(logx-\mu)^2/2\sigma^2}, \tag{2.2}$$

where μ and σ are respectfully the mean and standard deviation of the distribution.

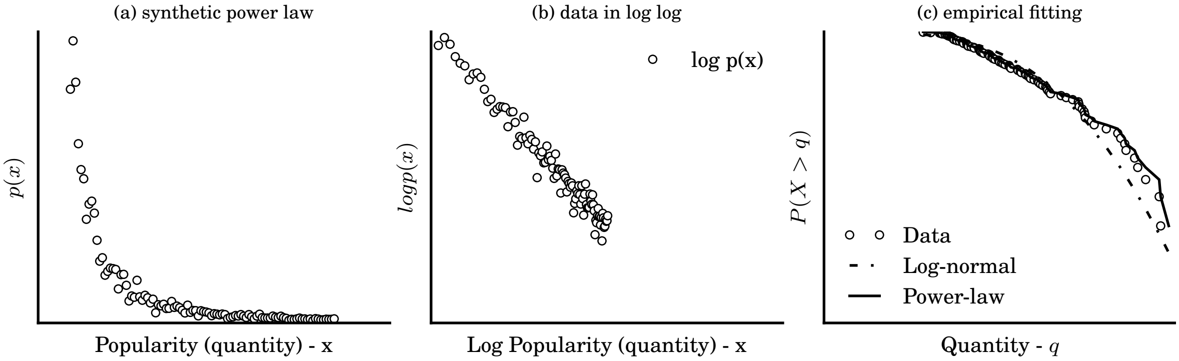


Figure 2.2: Synthetic Power-law Datasets.

An example of a Power-law distribution of a generic quantity is presented Figure 2.2. On the left (a) we show the density distribution, while on the center (b) we show the same distribution but using log scales on both axes. Notice the concentration of the distributions on lower values of the quantity, which are more probable to occur. Moreover, the linear behavior of the function on log scales is a simple means to find evidence of a heavy-tailed dataset. However, notice that on the figure to the right (c), which shows the complementary cumulative distribution function (CCDF), that both Log-normal and Power-law distributions provide reasonable fits to the data, due to both being heavy-tailed distributions. This confusion between different heavy-tailed distributions has been discussed in [28], and we here prefer to take the more cautious approach of stating that previous research found heavy-tailed distributions of popularity instead of Power-law or Log-normal specifically. We now discuss some of these studies.

The work of Sinha and Pan [126] provided an extensive analysis of the distributions of popularity for different quantities related to human knowledge (e.g., books or films revenues) or even ideologies (e.g., votes for political candidates). The study found the heavy-tailed behavior in most of such quantities. Also, the same study found that not only do final values of popularity tend to follow heavy-tailed distributions, but also, in some cases such as film revenues, the decay over time can also be modeled as such. However, in some cases, an exponential decay over time was also observed. This effect heavy-tailed popularity measures, and also the heavy-tailed/exponential popularity decay over time, was also verified in user generated content [22, 34, 78, 84, 133], as further discussed in Section 2.2.1.

There are many explanations for the existence of heavy-tailed distributions of popularity in most quantities of human knowledge. One of the most used explanations is the rich-gets-richer phenomenon. Rich-gets-richer, also known as cumulative advantage or preferential attachment, states that quantities with higher popularity will tend attract more attention over time [41]. Other authors, have looked into the stochastic processes based on exponential mixtures of exponential growth models [3, 67, 97, 115]. One example of such a process is the growth of cities, which has been shown is exponential in nature for individual cities. At the same time cities grow exponentially, new cities are also born with an exponential rate. These processes also cause heavy-tailed distributions. In any case, one important take-away here is that, although the aforementioned studies provide some empirical evidence of how to model popularity, little discussion is provided on what exactly causes such distributions of popularity. For instance, although the aforementioned models may explain the heavy-tailed behavior, they do not take the social and market structure surrounding online information into account. Such factors may explain the cause why some relatively unknown artists, like Psy,[1], may to rise to fame.

2.1.2 The Effects of Markets on Popularity

Motivated by the question of which factor has the most impact on popularity, quality or social influence, Salganik *et al.* [122] created an artificial musical market where users were asked to rate music based on their tastes. One set of users had knowledge of the popularity (based on previous ratings) of songs, while the other set of users had no such knowledge. The authors concluded that social influence leads to more

[1] Currently, the most viewed video on YouTube `http://www.youtube.com/watch?v=9bZkp7q19f0`

skewed popularity distributions and, surprisingly, less predictability. Even though this study neglects many factors of real world musical markets, it showed that social influence makes market prediction less effective. Thus, it provides evidence that the dynamics of popularity, even in restricted settings are difficult to account for. In a sense, these results related to another work by the same authors that suggested that viral like epidemics of online propagation of information (which are related to popularity [91,126]) are caused by the aggregation of small influences of a large number of unrelated individuals [140], which is a hard to predict phenomenon. Similarly, the hardness of predictability of popularity due to a wide range of social phenomena and different object features that exists in society has also been argued by the work of Lee *et al.* [79]. One simple example of these factors is also detailed in the work of Lakkaraju *et al.* [78]. Here, the authors showed that just the time of day and the title of a Reddit[2] post have significant impact on the popularity of the post.

Another important concept is the effect of exposure on the popularity of products or brands. A number of previous studies [37, 42, 52, 87] showed that purchase intent and awareness of a brand are related to the exposure (e.g., number of views) to an advertisement. This feature is usually known as promotion and is one of the causes of popularity growth (see Chapter 3). Those studies also provide valuable insights on how online markets behave and further motivate our work on music streaming services, discussed in Chapter 8.

2.2 Popularity of Social Media Objects

The focus of our dissertation is on the popularity of social media objects. As defined by Kaplan and Haelin [74], user generated content (UGC), online social networks (OSNs [96]) and even massive online games (such as Second Life and World of Warcraft [93]) all belong to the broader phenomenon of *social media*. Figure 2.3 presents an illustrative example of a typical social application. The picture depicts users connecting to one another via an internal OSN, while other users may access public content without being registered in the application. Note that both OSN related factors and external users may impact content popularity, as we shall further discuss in this section. The remainder of the section is divided as follows. We start by discussing earlier studies of social media popularity, which focused mainly on static views of popularity of UGC (Section 2.2.1). We then discuss some previous efforts to analyze the temporal evolution of popularity (Section 2.2.2) and which develop popularity prediction models

[2] http://reddit.com

(Section 2.2.3).

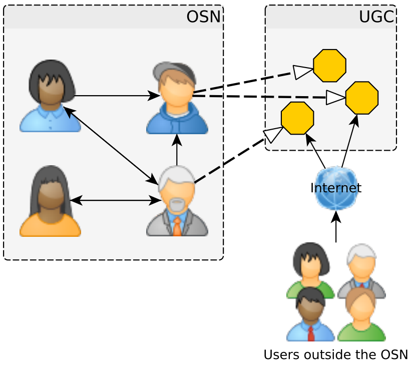


Figure 2.3: Example of Connections in a Social Media Application

2.2.1 Static Views of Popularity of Social Media Objects

Understanding the popularity of a piece of content (or object) is a subject that has gained attention from researchers since the earliest studies of social media. Most of these studies analyzed the total number of views measured at the time the data was crawled, or the “final” popularity, paying little or no attention at how content popularity evolves over time. Starting with video content, Cha *et al.* [22] presented an in-depth study of two video sharing systems. The authors analyzed the popularity distribution of objects, some few aspects on the popularity evolution, as well as content characteristics of YouTube and of a popular Korean video sharing service. Moreover, the authors investigated mechanisms to improve video distribution, such as caching and Peer-to-Peer (P2P) content distribution networks (CDNs). In a similar fashion, Chatzopoulou *et al.* [26] characterized the largest dataset of YouTube videos at the time, finding that, for older videos, moderate to strong correlations exist between popularity measured in number of views and other metrics such as number of comments and favorites. They also found that in the case of younger videos, such correlations are weaker, indicating that their long term popularity dynamics are not yet stable.

Wattenhofer *et al.* [139] analyzed the correlations between the popularity of YouTube videos and properties of various online social networks (OSN) created among users of the system. In particular, they found that characteristics of YouTube's comment-to-comment OSN (i.e., an OSN of links between users who comment each others' videos) are more strongly correlated with the popularity of a user's video than the characteristics of the subscription graph (although such correlation is still reasonably strong). This result indicates that active community collaboration can have a higher impact on the views a user receives through her videos. This study is in agreement with the one by Susarla *et al.* [128] which showed that subscriber links play an important role in the early popularity of videos. Another study was done by Borghol *et al.* [13] which also analyzed the correlations between popularity of YouTube videos and content factors, determined by groups of duplicate videos (or clones, as the authors call them), finding positive correlations the clone features and popularity. In details, the authors correlated final popularity with current popularity and clone groups. In order to achieve this, the authors make use of a linear regression model between the logarithm of past popularity plus the clone groups, the regressors or explanatory variables, and the logarithm of current popularity, the regressand or response, that is:

$$log(y_{current}) = \beta_0 * log(y_{past}) + \sum_{c \in C} \beta_c * \mathbb{1}_{v \in c} + \alpha \tag{2.3}$$

where $y_{current}$ is the current popularity, y_{past} is the past popularity, c is a clone group and C is the set of clone groups. Moreover, v is a video and $\mathbb{1}_{v \in c}$ is an indicator function that takes value of 1 if $v \in c$, that is, if v is part of the clone group c and 0 otherwise. β_i are the regression coefficients. With the model above, the authors showed that the addition of new binary explanatory variables, captured by $\sum_{c \in C} \beta_c * \delta_{v \in c}$, to the linear model, improved the regression quality captured by the coefficient of determination R^2 [69]. The authors argued that this result implies that popularity is *related* to content, since clone groups have the same content.

Flickr images were also the focus of attention of many early studies of UGC popularity. Zwol [133] measured the popularity of images, finding that heavy-tailed distributions explain both the total popularity and decay in popularity over time. It is interesting to note that many of the early studies on Flickr focused on folksonomies and tags [57, 82, 88], which are also examples of UGC. For instance, Marlow *et al.* [88] found heavy-tailed popularity distributions for tag popularity, where the popularity of a tag was measured either by number of images it annotates or number of user's that used the tag in their libraries. This result has also been observed when tags are used to annotate other kinds of media, such as videos and text data [45]. On the subject of diverse kinds of media, the work of Recuero [36] and of Leskovec *et al.* [84] also found heavy-tailed patterns in the distribution of the number of blogs a given sentence appears in.

In common, these studies provide important insights into content popularity in video sharing services and other UGC applications. However, most of them only focus on either a single snapshot of the popularity of objects or on a few snapshots only [22, 54, 133]. In this sense, these studies did not analyze the long-term temporal evolution of popularity as we do.

2.2.2 Popularity Evolution Over Time on Social Media Applications

The popularity evolution of online content has been the target of more recent studies. Focusing on YouTube videos, Borghol *et al.* [14] showed how weekly based views can be used to model video popularity. Also, the authors developed a model to determine the number of videos that may exceed a given popularity threshold, although such model does not indicate which *specific* videos these will be [14]. More recently the work of Islam *et al.* [68] showed that the weekly based modeling of video popularity in videos is still valid even years after upload. However, the synthetic model for predicting and generating the distribution of popularity *of a group of videos*, as proposed by the authors [14], does not. Zhou *et al.* [155], showed the importance of related videos links to popularity. Still on the subject, the importance of incoming links and content features to the final popularity of YouTube videos has been further analyzed recently [13]. It is important to note that some of these studies were done in parallel to ours on RG1 (which will be discussed on Chapter 3). Moreover, these studies are not focused on the prediction of popularity of *individual* videos (as is one our goals).

Broxton *et al.* [16] analyzed the popularity patterns of viral videos. According to the authors, a viral video is one that receives a large fraction of views from OSN applications. The authors developed a method to rank different sources of traffic to videos according to their potential in attracting more views. Brodersen *et al.* [15] made use of the same model to determine, amongst other things, if viral videos receive most views from the same geographic region. The authors showed that, surprisingly, most viral videos will have an initial burst in propagation in a diverse set of geographical regions, later falling back to the region of upload. Diversity was measured by entropy [32], which is defined as:

$$H(x) = -1 * \sum_{x \in X} p(x) log(p(x)) \tag{2.4}$$

where x is a geographical region (city or country) and $p(x)$ is the fraction of views such region. However, there is a possible bias in this results since estimation of $H(x)$ will be biased with few observations of x (the authors did not take into account the variability of the entropy). For instance, notice that with few observations, the values of $p(x)$ will be close to a uniform distribution because of the low initial popularity. Uniform distributions are the ones with maximum entropy [32]. Previous work also focused on geographical propagation of Twitter data [43, 71]. Among other things, authors find that some cities are trend-setters (sources of popular memes), while others are trend-consumers (sinks).

Focusing on image content, Cha *et al.* [24] analyzed the propagation of pictures through Flickr's internal OSN. The authors found that popularity (measured in number of favorite markings) of the most popular Flickr pictures exhibit close to linear growth. The authors discussed the importance of social links in the increase in popularity of images, showing that about 50% of favorite markings come from social cascades. That is, user A marks a picture as favorite after her friend user B marked the same picture. Another interesting work was done by Ratkiewicz *et al.* [114]. The authors investigated how external events, captured by search volume on Google Trends[3] and local browsing (i.e., university/community traffic), affect the popularity of Wikipedia articles. More recently, Khosla *et al.* [76] compared the use of image and social features for predicting the final popularity values of images. Their results are complementary to ours, as they focus on understanding a different social media application than the ones we study (Flickr). More importantly, the authors do not focus the long-term popularity trends, as we do.

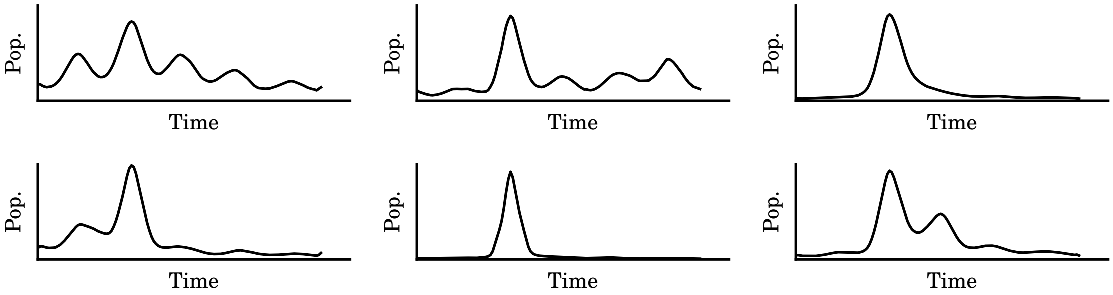


Figure 2.4: Examples of Popularity Evolution Trends on Twitter [146]

There have also been some efforts towards clustering social media objects in terms of their popularity temporal patterns. Crane and Sornette proposed Hawkes based models to explain how a burst in video popularity in terms of endogenous user interactions and external events [34]. Yang and Leskovec [146] proposed a time series clustering algorithm to identify trends on temporal patterns of popularity evolution. The model proposed by Matsubara *et al.* [91] provides a unifying analytical framework of the temporal patterns extracted by Crane and Sornette [34] and Yang and

[3] http://trends.google.com

Leskovec [146]. An example of such trends are depicted in Figure 2.4 for a toy dataset of 1000 Twitter hashtags. In the y-axis each figure shows the popularity *shape* of the trend, while on the x-axis is the time. The algorithm which extracts such trends [146] will be discussed in more details in Section 2.4, here we simply note that it is focused on the overall shape of popularity evolution and not popularity values, the reason why we omit numbers from both axes. We employ this algorithm as a component of our approach in dealing with popularity prediction (see Chapters 5 and 6).

Although the aforementioned efforts provide some insights into the evolution of content popularity, there is still little knowledge about which object features (e.g., video, link and popularity features) and system mechanisms (e.g., search) contribute the most to popularity growth. Thus, our analyses and findings greatly build on previous efforts, shedding more light into the complex task of social media objects popularity prediction. Moreover, unlike previous prediction efforts that focused on estimating future popularity measures, one of our focus in this dissertation. More specifically Chapter 6, is to tackle the challenge of predicting popularity trends while at the same time maximizing remaining interest after prediction, a task which, to the best of our knowledge, has not been studied yet.

2.2.3 Prediction of Popularity of Social Media Objects

We now focus on previous research that aimed at developing models to predict the popularity of a piece of content at a given future date. Our goal for the moment is to summarize the main previous efforts of the popularity prediction task. For simplicity, we leave the mathematical treatment of the data mining and time series techniques exploited by these efforts to be discussed in Section 2.4.

As stated by Lee *et al.* [79], popularity is related, in a complex way, to the social and psychological perspective of users regarding online content. Thus, deriving effective prediction models is not only difficult but also depends on characteristics of the target application. The same authors made use of a survival analysis approach [33] to predict the lifespan of online comments, that is, the probability that comments will still arrive at a comment thread after a given time t. They also developed models to predict the popularity of the thread at time t.

Focusing on Digg content, Lerman and Hogg [81] developed stochastic user behavior models to predict the popularity of Digg's stories based on early user reactions to new content and on aspects of the website design. The proposed model is very specific to Digg features, such as story up votes, and is not general enough for different kinds of social media content. Szabo and Huberman proposed a linear regression method to predict the popularity of YouTube and Digg content from early measures of user accesses [129]. This method has been later extended and improved with the use of multiple features [110].

In a different direction, Saez-Trumper *et al.* focused on the problem of identifying trendsetters - a twitter user who adopts, spreads and influences others with new trends before they become popular [121]. Regarding the rankings of an item (based on likes and dislikes), Yin *et al.* [150] proposed a model that took into account user personalities when casting votes, and developed a Bayesian model for ranking prediction. They tested their model in a popular IPhone application, JokeBox. Moreover, popularity prediction has also gained the attention in other contexts. In the particular context of search engines, the work by Radinsky *et al.* [112] proposed a model to predict future popularity, seasonality and the bursty behavior of queries.

We note that none of these prior efforts focused on the problem of *predicting popularity trends*. In particular, those focused on social media popularity prediction assume a fixed monitoring period, and do not explore the trade-off between prediction accuracy and remaining views after prediction. Even though some authors show the effectiveness of their methods for different monitoring periods [79, 110, 129], they did not discuss on methods how to determine such monitoring periods for each individual object, as we discuss in Chapter 6.

The previous efforts that are most related to ours are those reported in [107] and [4]. The former presents a model to predict whether a tweet will become a trending topic by applying a binary classification model (trending versus non-trending), learned from a set of objects from each class [107]. The objects are previously labeled by Twitter's internal mechanisms. Our work builds upon [107] by proposing a more general approach to detect *multiple* trends (classes), where trends are first automatically extracted and learned from a training set. Our approach also exploits the concept of shapelets [148] to reduce the classification time complexity, as we discuss in Chapter 6.

Ahmed *et al.* [4] proposed a clustering-based model for popularity prediction. The popularity curve is broken into multiple phases. For each phase, objects are clustered into representative trends, and such trends are used to build a transition graph with the probabilities of changes between trends along the popularity curve. Predictions are made by walking on such graphs. However, once again, the authors did not tackle the trade-off between remaining interest and prediction accuracy. In particular, as others [91, 110, 129], they do not investigate how long an object should be monitored before prediction, as we do here, and assume this information is given. Moreover, the paper is not clear on how to build the transition graph in practice (e.g., there is no separation between training and test sets). Adapting this method to tackle the early trend prediction problem is not straightforward, and is left for future work.

We also mention some other efforts to detect trending topics in various domains. Vakali *et al.* proposed a cloud-based framework for detecting trending topics on Twitter and blogging systems [131], focusing particularly on the implementation of the framework on the cloud, which is complementary to our goal. Golbandi *et al.* [56] focused on detecting trending topics for search engines. Despite the similar general goal, their solution applies to a very different domain, and thus focuses on different elements (query terms) and exploits different techniques (language models) for prediction. Finally the work of Jiang *et al.* [70], exploits content and social features to predict the day a video is going to peak. However, the authors do not provide a detailed analysis of the importance of each feature to popularity, as we do.

In sum, to our knowledge, we are the first to tackle the inherent challenges of producing predictions of popularity (trends and measures) as early and accurately as possible, on a per-object basis, recognizing that different objects may require different monitoring periods for accurate predictions. We build upon existing methods, extending them and combining them, to design novel solution to this problem.

As a summary of this whole section, we note that all of the studies that we have discussed up to here provided us with valuable understanding of popularity of social media objects. Nevertheless, important aspects, such as understanding the importance of various factors on popularity, specially the referrer features, as well as how to provide *useful* predictions of popularity while maintaining a reasonable amount of remaining interest, have been neglected or have not been thoroughly investigated in depth by these previous efforts.

2.3 Popularity Through the Lens of User Activities

In this section, we shift our focus to previous efforts on understanding popularity through user activities or online social networks. More specifically, we discuss efforts that emphasized studies on popularity using epidemics based models, repeated consumption, as well as studies that focused on competition and collaboration models. These papers are crucial for understanding our Phoenix-R and our A-Flux models, that are described in Chapters 7 and 8 respectively.

Epidemics Based Models - Previous work on information propagation on (OSNs has exploited epidemics based models [64] to explain the dynamics of the propagation process. An epidemic model describes the transmission of a “disease" through individuals. The simplest epidemic model is the Susceptible-Infected (SI) model. The SI model considers a fixed population divided into S susceptible individuals and I infected individuals. Starting with $S(0)$ susceptible individuals and $I(0)$ infected individuals, at each time step $\beta S(t-1)I(t-1)$ individuals get infected, and transition from the S state to the I state. The product $S(t-1)I(t-1)$ accounts for all the possible connections between individuals. The parameter β is the strength of the infectivity, or virus. The equations of the SI model are as follows:

$$\frac{dS}{dT} = -\beta SI \tag{2.5}$$

$$\frac{dI}{dT} = \beta SI \tag{2.6}$$

Cha *et al.* used an SI model to study how information (i.e., the "disease") disseminates through social links on Flickr [23], whereas Matsubara *et al.* [91] proposed an alternative model called SpikeM. SpikeM builds on an SI model by adding, among other things, a decaying power law infectivity per newly infected individual, which produces a behavior that is similar to the model proposed in [34]. The SpikeM model was used to capture the time series popularity for a single cascade. One of the reasons why the SI model is useful to represent online cascades of information propagation is that individuals usually do not delete their posts, tweets or favorite markings [23, 91]. Thus, once an individual is infected he/she remains infected forever (as captured by the SI model).

Repeated Consumption Models - Weng *et al.* [141] proposed an Yule-Simon [19] agent-based model to investigate the role of user activities (in special the limited attention of users) in the dissemination of information on Twitter. Similarly, Anderson *et al.* [5] investigated the repeated consumption of users by proposing a model (which is also a Yule-Simon based approach) that combines recency and content quality effects to predict the chance of a user re-consuming a given object.

Our work on the Phoenix-R model builds upon these past efforts – (epidemics based and repeated consumption models – to create a parsimonious model of evolution of online information. In Chapter 7 we shall describe the model and show how it compares to state-of-the art alternatives. Finally we point out to the very recent work of Hu *et al.*. Here, the authors focused on the defining longevity of social impulses, or multiple cascades [65]. Multiple cascades are also a form a of revisits. Our models on repeated consumption can also account for various cascades and, more importantly, they focus on capturing the number of visits an object will receive.

Competition and Colaboration Models - Latent Dirichlet Allocation (LDA) [10] is currently one of the most powerful and used tools for latent analysis in large datasets. In the context of user attention, Limited Attention LDA (LA-LDA) and Limited Attention Collaborative Topic Regression (LA-CTR) have extended the original LDA approach to incorporate the limited attention of users, being effective to support the recommendation of new content [72,73]. Myers and Leskovec [100] proposed the Clash model to mine the competition and collaboration among online memes. However, the Clash model has two main constraints, namely: (1) it captures competitive and collaborative behavior of memes by exploiting the follower links, and thus are more suited to domains where such links are a primary means of information dissemination such as online social networks (OSNs); and (2) the approach is based on maximum likelihood estimates, which have limitations due to long tail effects [116].

Our work on the A-FLUX model also focused on providing a latent factors approach to mining competition and collaboration. Different from the aforementioned approaches, we do not rely on OSN link data. This is specially interesting in the music streaming industry, our case study with A-FLUX. Here, it is very difficult to pin-point a who-exposed-whom to a piece of information, since music propagates online and offline [108]. We also point out that our A-FLUX model is also based on LDA. Also, the model is similar, in terms of the probabilistic graphical structure, as other models proposed in different settings such as text mining [61,90,137,144,149].

On a complementary effort, Ribeiro *et al.* modeled user activity as a commodity on membership based websites [118,119], showing that it can be used to predict if such sites will remain attractive over time. Finally, Matsubara *et al.* [89] made use of the Voltera-Lotka equations to model co-evolving, and possibly competing, time series of user attention to web products and services. Both of these efforts are time series based, whereas our work is focused on user activities. It thus provides a more fine grained view of the competition and collaboration issue.

2.4 Time Series Statistics and Data Mining

We now shift the focus, and describe some of the statistical and data mining techniques that are used by previous work and this dissertation. Time series can be summarized as a class of data that can be used to model a diverse set of phenomena such as stock markets, climate changes, earthquakes and, in our case, popularity evolution of social media objects. We note that this section *is not* an in-depth survey of time series or data mining in general. We only briefly discuss some of the techniques more related to the context of social media and popularity prediction. For a more detailed and comprehensive presentation, we refer to the following books [25, 125], Chapter 11 of [86] and the survey [50].

The rest of this section is divided as follows. We first discuss common representation of time series in Section 2.4.1, since this is the basis for any data mining task with this kind of data. Afterwards, we discuss the regression models commonly used to predict popularity of social media objects in Sections 2.4.2 and 2.4.3. We discuss time series distance measures and machine learning approaches in Section 2.4.4.

2.4.1 Time Series Representation

In research areas such as Statistics [25, 125] and Econometrics [142][4] it is common to represent time series using definitions from the *stochastic processes* literature. Since this is a more general representation, we begin by briefly describing stochastic processes. We then narrow this definition down to the vector representation of time series commonly used in data mining (as well as this dissertation).

A *stochastic process* is denoted as:

$$\{x_{t_i}\}_{i=1}^{\infty} = x_{t_1}, x_{t_2}, x_{t_3}, \cdots , \tag{2.7}$$

where x_{t_i} are values in $\mathbb{R}$. Each such observation defines the quantity which the time series captures. The values t_i represent the points in time (or indexes) for each quantity x_{t_i}. A necessary condition is that $t_1 < t_2 < t_3 \cdots t_n$, that captures the nature of a series. It is common for quantities to be observed at uniform lengths from one another, thus making the use of the index variable t_i unnecessary in most applications. Thus, a simpler notation is $\{x_t\}_{t=1}^{\infty}$. Such processes can also be written as a cumulative probability density function of each t_i:

$$F(c_1, c_2, \cdots) = P(x_i \leq c_1, x_2 \leq c_2, \cdots), \tag{2.8}$$

with $c_i \in \mathbb{R}$. We note that this definition is more commonly used in descriptive statistics. Such a notation is useful for understanding the statistical properties of time series. Notice that in both notations the time series are represented up to infinity, thus

[4] Econometrics is informally defined as the study of economics using statistics and computer science, or simply quantitative economics.

the notion of a never ending process. For example, a simple stochastic process is the one bellow:

$$MA(q) \to x_t = \sum_{i=0}^{q} \beta_i \epsilon_{t-i}, \tag{2.9}$$

which is an example of a *moving average* (MA) model of level q. Here, β_i are the parameters and ϵ_{t-i} is white noise (e.g, Gaussian error) [25].

While the stochastic process based definition is more general, in practice we observe a subsequence of the time series. That is, a vector $\mathbf{x}$ of observations is observed. In this sense, a time series can be summarized simply as a sequence of data points measured at different times steps [50]. Thus, we define a time series vector as:

$$\mathbf{x} =< x_{t_1}, x_{t_2}, x_{t_3}, \cdots, x_{t_n} >, \tag{2.10}$$

where $\mathbf{x}$ is an observation vector, again composed of values $x_{t_i} \in \mathbb{R}$. The same comment for uniform indexes apply in this case, thus turning the definition above in the one below:

$$\mathbf{x} =< x_1, x_2, x_3, \cdots, x_n > . \tag{2.11}$$

When appropriate, we take the liberty to define a process as a time series stream, which is:

$$\hat{\mathbf{x}} =< x_{t_1}, x_{t_2}, x_{t_3} \cdots \tag{2.12}$$

This notation captures the intuition of a never ending process, but using the vector like notation that is exploited throughout the dissertation.

We take some time to discuss how different studies on social media instantiate $\mathbf{x}$. On some applications, such as YouTube or Flickr, popularity is usually measured at coarse granularity such as days [24, 34] or even weeks [14]. On applications with finer time granularities of interest, such as Twitter [146], successive indexes t_i can be aggregated in order to capture popularity in seconds, minutes or days, depending on how the series will be used. Due to the fact that we only observe popularity for a finite period, we never really observe the defined stream, but in reality a sample of the time series which we can further sub-sample. Moreover, instead of representing the time series as absolute quantities, one can model each object as a change in quantity (the derivative), that is $x_{t_i} - x_{t_{i-1}}$ or even relative changes in regard to other objects. Such an approach has been used by Ahmed *et al.* [4] to cluster similar time series of social media objects according to their change rates. Other authors [79] simply deal with non-uniform indexes and raw quantities.

It is also important to note that most *classical* [25, 125] stochastic process based time series models are unsuited for the study and prediction of popularity in time series in social media [91]. This occurs because models based on moving averages or *auto regression* (AR) assume stationarity (their multivariate analogs, such as *vector auto regressive* - VAR and *vector auto regressive moving average* - VARMA, also have the same assumption). Two necessary, but not sufficient, conditions for stationarity is that the mean and the variance of the probability distribution in Eq. 2.8 have to be independent of time t. Popularity time series will fail to meet these conditions [91][5]. Take as example the series plotted in Figure 2.4; the nature of having a significant peak breaks both conditions. Moreover, even models which can deal with non stationarity, such as ARIMA based models [25, 125], fail to account for the heavy-tailed decay in popularity present in popularity time series. While stationary based models are unsuitable for this task, *linear regression* [110, 129] and *state space* [112] models have been applied to popularity time series with some success. We now discuss such models.

2.4.2 Linear Regression Models

Ordinary least squares (OLS) linear regression models have been adopted by previous research [110, 129] as a means build models of popularity prediction. A OLS regression is defined as follows:

$$\mathbf{y}_{t+h} = \mathbf{X}_t^T \mathbf{\Theta} + \boldsymbol{\epsilon}, \tag{2.13}$$

where $\mathbf{X}_t$ is a matrix of multiple time series column vectors (also called the covariate matrix), each with observations up to reference time t, $\mathbf{y}_{t+h}$, is the response, and $\boldsymbol{\epsilon}$ is the error of the model. The notation, $\mathbf{X}^T$ represents a matrix transpose. Solving the OLS equation for $\mathbf{\Theta}$ will define the prediction model, that is, the parameters $\mathbf{\Theta}$, that minimizes the mean squared error[6] (MSE):

[5] There is a nice quote which is: “Experience with real-world data, however, soon convinces one that both stationarity and Gaussianity are fairy tales invented for the amusement of undergraduates” [130]

[6] For clarity, we shall drop the index $t + h$ in the vector $\mathbf{y}_{t+h}$

$$mse(\mathbf{y}) = n^{-1}\sum_{i=1}^{n}(y_i - \bar{y})^2. \tag{2.14}$$

However, if applied to heavy-tailed data this model may fail to produce accurate predictions. One of the premises for linear regression is that of independence between the errors, $\boldsymbol{\epsilon}$, and the response $\mathbf{y}_{t+h}$. Due to the heavy-tailed nature of popularity, this premise is violated. For example, if we take the *mse* equation above and apply it to a heavy-tailed distribution, it is better to reduce the error on the most popular objects since any arithmetic mean is biased towards higher values. Thus, a correlation between errors and parameters will exist in the model. To understand this, picture the notion that *mse* will try to get the most popular content correct, since they have a large impact in the mean error. Given that only a handful of these objects exist, the model may be wrong for the majority of content.

In order to mitigate this behavior, the authors [110, 129] suggest minimizing the mean relative squared error (MRSE) instead, that is:

$$mrse(\mathbf{y}) = n^{-1}\sum_{i=1}^{n}(\frac{y_i - \bar{y}}{y_i})^2 \tag{2.15}$$

In order to create such model, we can slightly change the OLS equations. That is, the new equation will have the form:

$$\mathbf{1} = \mathbf{T}_t^T\boldsymbol{\Theta} + \boldsymbol{\epsilon}, \tag{2.16}$$

where $\mathbf{1}$ is a vector of ones, and $\mathbf{T}_t$ is a matrix composed of column vectors with the following normalization:

$$\mathbf{t}^j =< x_1/y, x_2/y, \cdots, x_n/y >, \tag{2.17}$$

which is the original time series vector divided by the response variable. The proof for that this model minimizes MRSE can be found in [110].

In the social media setting, both the studies of Szabo and Huberman [129] and the work of Pinto *et al.* [110] made use of the above models for prediction. The former, uses only the sum of popularity up to a certain time t as the covariate. The latter, uses the multiple observations of popularity. Moreover, this latter work also discusses how knowing trends *before hand* can help reduce prediction errors, by creating a specialized model for each trend. Another important detail is that the model can also be improved by adding new features to the covariate matrix $\mathbf{T}_t$. Such features capture the similarity between time series using a radial basis kernel.

2.4.3 State Space Regression Models

State space models are also usually explored for popularity predictions. A state space model can be defined as follows [25]:

$$y_t = \mathbf{h}_t^T \mathbf{\Theta}_t + e \tag{2.18}$$

$$\mathbf{\Theta}_t = \mathbf{G}\mathbf{\Theta}_{t-1} + \mathbf{F}\boldsymbol{\epsilon}_t. \tag{2.19}$$

It is important to note that the model is defined for a *single* time series y. The popularity at time t, y_t, is captured by the component vector $\mathbf{h}_t$ (we shall detail this shortly) and the parameter matrix $\mathbf{\Theta}_t$, with e being a Gaussian random noise. Notice that the parameters are time dependent, and are updated according to the pre-defined matrix $\mathbf{G}$. Moreover, $\boldsymbol{\epsilon}_t$ is a vector of errors also updated according to a pre-defined matrix $\mathbf{F}$.

Stationary and linear regression models can be instantiated as state space models based on the definition of $\mathbf{h}_t$, $\mathbf{G}$ and $\mathbf{F}$. For example, by setting $\mathbf{h}_t$ to be the previous popularity observations, $\mathbf{G}$ and $\mathbf{F}$ to be $\mathbf{0}$, we get an OLS model. Note that in this case, $\mathbf{\Theta}_t$ is no longer updated over time. For predictions considering multiple steps ahead $t + h$, the values of y_t and $\mathbf{\Theta}_t$ are updated until time $t + h$.

The study in [112] created different state space models by setting $\mathbf{h}_t$ to capture, trend, seasonal and surprise components of the time series. The models used by the authors are instantiations of the Holt-Winters Linear models [25]. Combinations of these three components can be used to create distinct models that capture different aspects of popularity. In order to define which components to use, the authors make use of a classification model based on query features. Moreover, parameter matrices $\mathbf{G}$ and $\mathbf{F}$ are estimated for each individual time series. Estimation by the authors was done using gradient descent techniques, such as the Levenberg-Marquardt algorithm. We also employ these techniques when learning our Phoenix-R model in Chapter 7. However, due to fact that the authors use Holt-Winters Linear models, their approach to social media time series is limited since these time series will exhibit non-linear trends (as we shall discuss in Chapter 7).

2.4.4 Distance Measures and Machine Learning Tasks

To finish up this section, we discuss the machine learning tasks of classification and clustering [99] in the time series context. One of the key elements for employing such tasks in time series is the effective definition of a distance measure [6] since a wide range of clustering or classification techniques exploit distances (or inversely similarity). To name a few, we can cite K-Means and Affinity Propagation for clustering tasks. We can also cite K-nearest Neighbors and Support Vector Machines for classification tasks. It is out of the scope of this dissertation to detail all of these techniques, and we refer the reader to a machine learning book for a description of them [99]. We shall however at the end of this section detail the KSC algorithm, proposed by Yang *et al.* [146], which is a variation of K-Means and used in our prediction models.

As stated, one of the key tasks in mining time series is the definition of a "good" distance measure [6]. One of the simplest definition of distances is the Euclidean one, which is define for two vectors ($\mathbf{x}$ and $\mathbf{y}$) as:

$$d_{euc.}(\mathbf{x},\mathbf{y}) = ||\mathbf{x}-\mathbf{y}||_2 \tag{2.20}$$

$$= \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \tag{2.21}$$

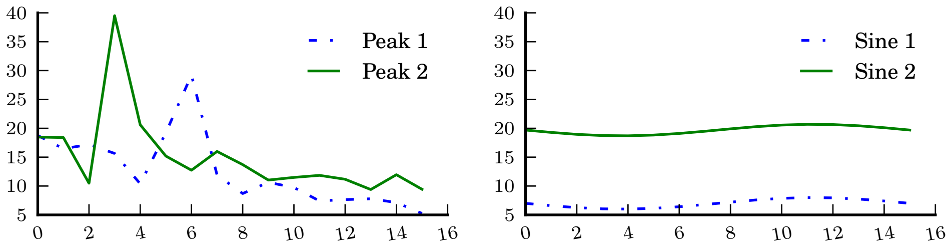


Figure 2.5: Two peak based time series and two sine based ones

where $||\cdot||_2$ is the l2-norm operator. Despite simple, the euclidean distance has major drawbacks, as pointed out by previous studies [6, 146]. For example, such this measure fails to account for the shifted behavior of time series. Take as an example the popularity time series shown in Figure 2.5. On the left plot, we show two time series which grow in popularity up to a maximum and then tend to decay over time. We call these *peak* time series. On the right, we show two time series generated by a sine process, being thus referred to as *sine*. If we were to use a simple nearest neighbors clustering algorithm with these series using euclidean distance, it would fail to distinguish the peak time series from the two sine curves. This occurs because of two effects: (1) the peak time series have their global maximum at different times, even though they come from similar processes; (2) the difference in volume between the two sine time series is very large, even though they are both sine processes. Effect 1 is related to shifts in the index (x-axis), while effect 2 is related to differences in scale (y-axis).

In order to exemplify this in a learning context, we state that if we were to compute the pairwise euclidean distances between these series, the distance between $Sine1$ and $Sine2$ would be larger than the distances between $Sine1$ and $Peak1$, or $Sine2$ and $Peak2$. Because of this, a clustering algorithm would group $Sine1$ with $Peak1$ and $Sine2$ with $Peak2$. That is, a distance based learner to be unable to distinguish the two processes. Different distance measures can be used to mitigate either or both of the aforementioned effects. For example, normalizing the time series could mitigate the scale effect, but not the shift one. A measure with is invariant to both shifts and scale is said to be complexity invariant [6].

Yang *et al.* proposed a distance measure which is invariant to both shifts and scales [146]. This measure, d_{ksc}, is defined as follows:

$$d_{ksc}(\mathbf{x}, \mathbf{y}) = \min \alpha, q \quad \frac{||\mathbf{x} - \alpha \mathbf{y}_{(q)}||_2}{||\mathbf{x}||_2}. \tag{2.22}$$

α is scaling parameter, and q a shift on $\mathbf{y}$. For a fixed q, there exists an exact solution for α by computing the minimum on d_{ksc} (i.e., setting the gradient to zero), which is: $\alpha = \frac{\mathbf{x}^{'}\mathbf{y}}{||\mathbf{y}||}$. In contrast, there is no simple way to compute shifting parameter q. Thus, in our implementation of KSC, whenever we measure the distance between two series, we search for the optimal value of q considering all integers in the range $(-n, n)$[7]. If we apply this metric to the time series in Figure 2.5, it would correctly find that the sine series are closer to each other than the peak ones (and vice-versa). Note that the best values of both parameters are found to minimize the metric.

We choose to use d_{ksc} in our dissertation since it has some desirable properties. Not only is it complexity invariant, but it was shown to be effective on the task extracting trends from social media [146]. Moreover, it can also be correctly used in a K-Means like clustering algorithm, which is called the KSC algorithm, which is defined as follows:

1. The time series are uniformly distributed to k random clusters C_i, where i =

[7] Shifts are performed in a rolling manner, where elements at the end of the vector return to the beginning. This maintains the symmetric nature of $d_{ksc}(\mathbf{x}, \mathbf{y})$.

$1 \cdots k$;

2. Cluster centroids are computed based on the members of each cluster. In K-Means based algorithms, the objective is to find centroid **c** that minimizes:

$$\mathbf{c} = \arg\min_{\boldsymbol{c}} \sum_{\mathbf{x} \in C_i} d_{ksc}(\boldsymbol{x}, \boldsymbol{c}) \tag{2.23}$$

.

We refer the reader to the original KSC paper for more details on how to find **c** [146];

3. Each time series vector **x** is assigned to the nearest centroid based on distance metric d_{ksc};

4. Return to Step 2 until convergence, i.e., until all objects remain within the same cluster in Step 3.

Many other distance measures for comparing time series exist in the literature. For example, Batista *et al.* [6] also also focused on the task of defining a scale and shift invariant distance measure. However, the efficacy of such a measure in extracting popularity trends of social media objects has not yet been measured. Other previous work also make use and extended on the notion of Dynamic Time Warp (DTW) [113, 135]. DTW is not a distance measure. It is an algorithm which finds the optimal alignment between consecutive time series points. In essence it deals with problems in shifts to align time series, and then computes distances using any given distance measure. We note however that in this work we want to extract trends from popularity time series. Not only was the KSC measure and algorithm shown to be useful in this context, the other measures we mentioned cannot be directly employed in a K-Means framework, thus extracting the representative centroid (or trend) would require more complex clustering algorithms, such as Affinity Propagation. The reason such measures cannot be directly used is the optimization objective in Eq. 2.23, since not all distance measures have a provable centroid.

2.5 Summary and Roadmap

In this chapter we reviewed previous studies and the background knowledge required to understand this dissertation. The structure of the rest of the dissertation is shown in

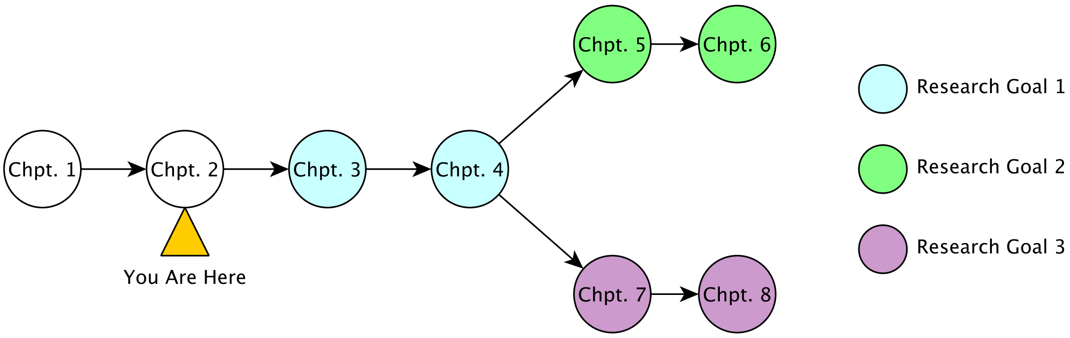


Figure 2.6: Thesis Roadmap (Related Work)

Figure 2.6, which we now discuss. Starting from Chapter 3, we begin our studies on our first research goal (RG1). Recall that, RG1 is focused on understanding the importance of different in relation to social media popularity. The first chapter on RG1, Chapter 3, focused on a characterization of YouTube videos popularity evolution. This is followed by Chapter 4, that presents our study on user perceptions of content and how these perceptions relate to popularity. Different from previous work (or at least previous work before some of our studies were performed), no other provided a characterization of video popularity focusing on novel features such as incoming links. Also, no previous studies had been done in looking into the human perceptions of content as we do.

Regarding our second Research Goal (RG2), Chapters 5 and 6, presents novel methods for popularity prediction on social media. Not only does the work of these chapters improve on previous methods, but more importantly, our study explores the trade-off between accurate and remaining interest in the content after prediction, a problem not yet tackled by previous research. Chapter 5 presents our prediction methods for news content. Whereas, Chapter 6 presents our study on predicting social media popularity considering the tradeoff between remaining interest and accuracy.

Finally, on Chapters 7 and 8 presents our studies on RG3. Recall that, RG3 is focused on understanding popularity through user activities dataset. Chapter 7 present our PHOENIX-R model for understanding revisit behavior in social media. Whereas, Chapter 8 presents our A-FLUX model developed to mine user attention flows. Both tasks complement each other and present novel insights into the popularity evolution issue.

Chapter 3

On the Dynamics of Social Media Popularity

Recall that our first research goal is on understanding feature importance on the popularity evolution of social media objects. In this chapter, we present our initial and fundamental results on this topic. It is important to point out that understanding the factors that impact the popularity dynamics of social media can drive the design of effective information services, besides providing valuable insights to content generators and online advertisers. Taking YouTube as case study, we analyze how video popularity evolves since upload, extracting popularity trends that characterize groups of videos. More importantly, we also analyze the referrers that lead users to videos, correlating them, features of the video and early popularity measures with the popularity trend and final observed popularity the video will experience. On the next chapter (Chapter 4) we shall continue this study from a users' perception of content context. This will be followed by our discussion on Research Goal 2 (RG2) on Chapters 5 and 6. Whereas RG3 will be discussed Chapters 7 and 8.

3.1 Introduction

User generated content (UGC) has emerged as the predominant form of online information sharing nowadays. The unprecedented amount of information being produced is one of the driving forces behind the success of the social media phenomenon [31,74]. This phenomenon is a shift from the traditional media where, instead of content being produced mostly by a few selected individuals, anyone, in theory, can produce and share content online. However, the "information overload" that accompanies the huge amount of social media being produced has its drawbacks. For example, it is ever-so-difficult to find and filter relevant content to oneself. Nevertheless, some pieces of content (or *objects*) succeed in attracting the attention of millions of users, while most remain obscure. This leads to the heavy tailed characteristic of content popularity [28, 126], where a few objects become very popular while most of them attract only a handful of views. *What makes one particular object become hugely popular while the majority receive very little attention? Which factors affect how the popularity of an object will evolve over time?* These are major questions in the social media context that drive our present work.

A plethora of different factors may impact social media popularity, including the object's content itself, the social context in which it is inserted (e.g., characteristics of the object's creator and her social neighborhood or influence zone), mechanisms used to access the content (e.g., searching, recommendation), and specific characteristics of the application that may promote the visibility of some objects over the others. Some of these factors contribute to the rich-gets-richer phenomenon [41], which can partially explain the heavy-tailed nature of content popularity. Others, such as links to the object from a popular blog and events in the real world, are external to the application and still may impact the object's future popularity.

Given the importance of social media on society nowadays, understanding the extent to which these factors impact the popularity of social media and how popularity evolves over time provides valuable insights for content generators, online advertisers and Internet service providers (ISPs), amongst others [30, 94, 98, 111, 131]. In this first study of our dissertation, we aim at investigating how different factors impact popularity dynamics of social media, focusing on YouTube as case study. YouTube is currently the most popular video sharing application, with over 100 hours of video shared per minute[1], and a total estimated number of shared videos that had surpassed 4 billion in early 2012[2]. It is a rich application that embeds several mechanisms, such as search, list of related videos, and top lists, that may affect how a video is disseminated, thus impacting its popularity.

Thus, in this chapter aim at performing a deep study of the evolution of popularity of user generated videos on YouTube. Towards our goal, we collected a public set of statistics available in the system that provides for each video: (a) its popularity as a function of time, and (b) a set of referrers, i.e., links used by users to access the video, along with the number of views for which each referrer is responsible. Given the great diversity of content on YouTube, our characterization is done on three different datasets, namely, popular videos that appear on the world-wide top

[1] http://www.youtube.com/yt/press/statistics.html

[2] http://www.reuters.com/article/2012/01/23/us-google-youtube-idUSTRE80M0TS20120123

lists maintained by YouTube; videos that were removed from the system due to copyright violation; and, a dataset of videos sampled according to a random procedure (i.e., random queries). Focusing on number of views as popularity metric, our study addresses five questions:

Q1 - How early do videos reach the majority of observed views? we intend to assess how fast a video achieves most of its observed popularity. This is key to determine the time period during which different information services can benefit more from a video. For example, ad placement services will be more effective if ads are posted on videos *before* most of their views are consumed. Moreover, search engines may misleadingly use observed popularity to favor some videos in their rankings, even when videos are no longer attractive. Our results show that some videos, such as top and copyright protected videos, achieve most of their views very early on, whereas videos selected from random queries tend to take longer to attract most of its observed views.

Q2 - Is popularity concentrated in bursts? We want to know whether video popularity is concentrated on a few days or weeks. This question complements Q1, offering valuable insights into how quickly the interest in the video raises and vanishes. Moreover, knowing the peak potential of a video (based on the most popular day/week) is valuable for services like advertisement campaigns. We find that top and copyright protected videos tend to experience popularity bursts, with a large fraction of their final observed views concentrated on single week or a even single day, whereas the popularity of videos selected from random queries tends to be less concentrated.

Q3 - Are there governing trends that characterize common groups of video popularity evolution? We here aim at bridging Q1 and Q2 by extracting the popularity trends of common groups of videos. To that end, we make use of a time series clustering algorithm [146] to infer the popularity trends. Focusing on videos from top lists and selected from random queries, we find that the same four types of popularity trends are observed in both datasets. One trend consists of videos that tend to remain attractive over time with an always increasing popularity. The other trends account for videos that tend to peak in popularity for a short while, with three different popularity decay characteristics after the peak.

Q4 - Which incoming links (or referrers) are more important for video popularity, and how early do they occur? The previous questions focus on understanding popularity based only on the popularity time series. Here, we want to know how users reach these videos. There are multiple forms through which users can reach a particular piece of content and, thus, there are multiple driving forces that may impact the popularity of a video. Identifying such forces is crucial for designing more cost-effective content dissemination strategies. For instance, should a content creator invest time on perfecting the keywords describing their videos (for better search rankings) or focus on campaigning videos in online social networks? Our results show that internal YouTube mechanisms, such as search engines and related videos, are the most important mechanisms that drive users to content, implying that YouTube itself handles a great power to drive video popularity through its internal mechanisms.

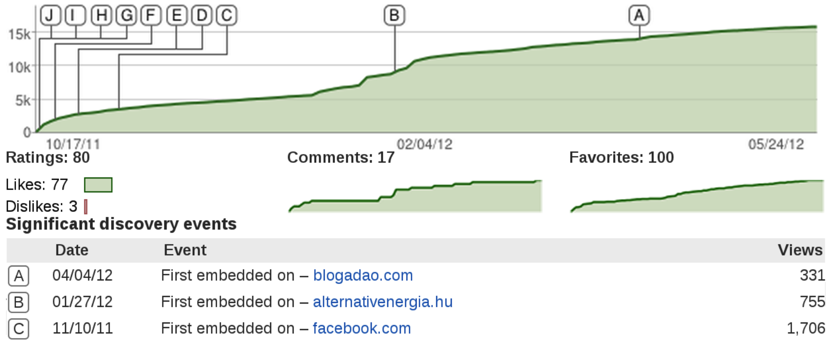
Q5 - What are the associations between features related to the video, to its early popularity measures and referrers with the popularity trend (or final observed popularity) of the video? We aim at measuring the associations between features related to the video (e.g., category, upload date, age), early points in the popularity time series and referrers with the identified popularity trends (Q3) and popularity measures. We show that videos that follow the same trend tend to also have similar content (based on video category) and referrers. For example, music videos tend to remain popular over time and are generally found through search engines, while videos related to news tend to have a small but significant attention period and are found through more diverse sources (e.g., external websites and viral propagation). Moreover, different features are more correlated with popularity trends and measures at different moments of the video's lifespan, motivating the use of some of them to build popularity prediction models.

The rest of this chapter is organized as follows. Section 3.2 discusses our data collection methodology. Section 3.3 presents our characterization of YouTube popularity curves (Q1 and Q2), while Section 3.4 shows the different popularity trends of YouTube videos (Q3). Next, Section 3.5 characterizes the relative importance of different referrers (Q4), whereas Section 3.6 discusses the correlations between various features and popularity (Q5). Finally, Section 3.7 concludes the chapter discussing the implications of our results.

3.2 Datasets

As our case study, we analyze the following datasets, which are publicly available[3]:

[3] http://vod.dcc.ufmg.br/traces/youtime/



Ratings: 80

Likes: 77 🟩

Dislikes: 3 🟥

Comments: 17

Favorites: 100

Significant discovery events

	Date	Event	Views
A	04/04/12	First embedded on – blogadao.com	331
B	01/27/12	First embedded on – alternativenergia.hu	755
C	11/10/11	First embedded on – facebook.com	1,706

Figure 3.1: YouTube's Insight Data Example (Some Referrers Were Trimmed)

Top: 27,212 videos from top lists maintained by YouTube (e.g., most viewed videos, most commented videos).

YouTomb: 120,862 videos with copyright protected content, identified by the the MIT YouTomb project[4]. This is the first effort to characterize copyright protected videos.

Random topics: 24,482 videos collected as result of random queries submitted to YouTube's search API. To build such queries, we first randomly selected, according to a uniform distribution, an entity from the Yago semantic database [75]. Yago entities cover topics such as popular movies (e.g., Blade Runner) to common items (e.g., chair). The (textual) name of the entity was then submitted as a query to the YouTube's search API, and we selected the most relevant video in the result list. We queried for 30,000 entities and discarded queries with empty results[5].

For each video, we collected YouTube's insight data associated with it, which is publicly available on the video's home page. This insight data consists of various features of the video, including three time series of how the numbers of views, comments and favorite markings of the video evolved over time, since the video was uploaded. It also includes a set of referrers that led users to the video. The time series are daily for videos with less than 100 days of age, while 100 evenly distributed points are provided for videos with more than 100 days of age. Other features, such as the video category and upload date, were also scrapped from the HTML page of each video. In Figure 3.1 we show an example of YouTube's insight data that was available up to 2013 (before a change in YouTube's user interface). Currently, the referrer (discovery events) information, comments and favorites time series are no longer provided.

We processed our collected datasets to remove: (1) videos with missing or inconsistent information; and (2) videos uploaded on the same day of our crawling. Table 3.1 provides a summary of each cleaned dataset, presenting the total number of videos,

[4] `http://youtomb.mit.edu/`

[5] We do *not* claim our dataset is a random sample of YouTube videos. Nevertheless, for the sake of simplicity, we use the term Random videos to refer to videos from this dataset.

Table 3.1: Crawled Datasets (after cleanup)

Video Datasets	Top	YouTomb	Random
# of Videos	18,422	102,888	21,935
Average # of of views	1,064,264	273,696	131,473
Average video age (days)	170	750	526

Table 3.2: Distribution of Video Age

	Top	YouTomb	Random
age (days) ≤ 7	4,303	0	109
$7 <$ age (days) ≤ 30	6,543	0	563
$30 <$ age (days) ≤ 365	4,627	13,379	8,159
age (days) > 365	2,949	89,509	13,104

average number of views per video, and average video age. Video age, measured in number of days, is defined as the difference between the crawling date (or the removal date, for videos in the YouTomb dataset) and the upload date. We note that YouTomb videos are on average older than videos in the Top and Random datasets. Moreover, Top videos are, as expected, more popular, on average, than YouTomb videos, which, in turn, tend to attract more views than videos in the Random dataset (on average).

We also note that video ages vary significantly, as shown in Table 3.2. Most videos in the YouTomb and Random datasets are over 1 year old, or have ages between 1 month and 1 year. In contrast, videos in the Top dataset tend to have a bi-modal age range, with most being either a few days old or over 1 year. Given such variability, we analyze popularity evolution separately for videos in each age range. However, to avoid hurting presentation with too many graphs, we focus on results computed over all videos in each dataset, pointing out significant differences across age ranges when appropriate.

The features we collected, shown in Table 3.3, are grouped into three classes, namely video, referrer, and popularity features. Video features include category, upload date, age, and the duration of the time window w that represents a single observation in the video's popularity time series (see below). The video category is defined based on the YouTube's list of categories, which includes *Autos/Vehicles, Comedy, Education, Entertainment, Gaming, Film/Animation, Howto/Style, Music, News/Politics, Shows, Nonprofit/Activism, People/Blogs, Pets/Animals, Travel/Events, Science/Technology, and Sports*. The referrer features include the first date and the number of views asso-

Table 3.3: Summary of Features

Class	Feature Name	Type
Video	Video category	Categorical
	Upload date	Numerical
	Video age	Numerical
	Time window size (w)	Numerical
Referrer	Referrer first date	Numerical
	Referrer # of views	Numerical
Popularity	# of views	Numerical
	# of comments	Numerical
	# of favorites	Numerical
	change rate of views	Numerical
	change rate of comments	Numerical
	change rate of favorites	Numerical
	Peak fraction	Numerical

ciated with each referrer category. Referrers are categorized into *External, Featured, Search, Internal, Mobile, Social and Viral.* The *External* category represents websites (often other OSNs and blogs) that have links to the video. The *Featured* category contains referrers that come from advertises about the video in other YouTube pages or featured videos on top lists and on the front page. The *Search* category includes referrers from search engines, which comprise only Google services. *Internal* referrers correspond to other YouTube mechanisms, such as the "Related Video" feature. *Mobile* includes all accesses that come from mobile devices. *Social* referrers consist of accesses from the page of the video owner or from users who subscribed to the owner or to some specific topic. Finally, some other referrers are grouped into *Viral.* The popularity features include the final numbers of views, comments and times the video was marked as favorite, the trend in these measures captured by the corresponding average change rates, and the largest fraction of all observed views that happened in a single time window (peak fraction). Jointly, these features capture properties of the popularity curve.

We note some limitations of the data provided by YouTube. Each popularity curve is registered with at most 100 points, regardless of the video age. Thus, the video's time window w is defined as the video age divided by 100. In order to be able to estimate video popularity on a daily basis, we performed linear interpolation between the 100 points provided. Moreover, YouTube does not provide information on every referrer that led users to the videos, but rather on ten *important* ones (according to YouTube). In total, the available referrers account for only 36%, 25% and 35% of all the views of videos in the Top, YouTomb, and Random datasets, respectively.

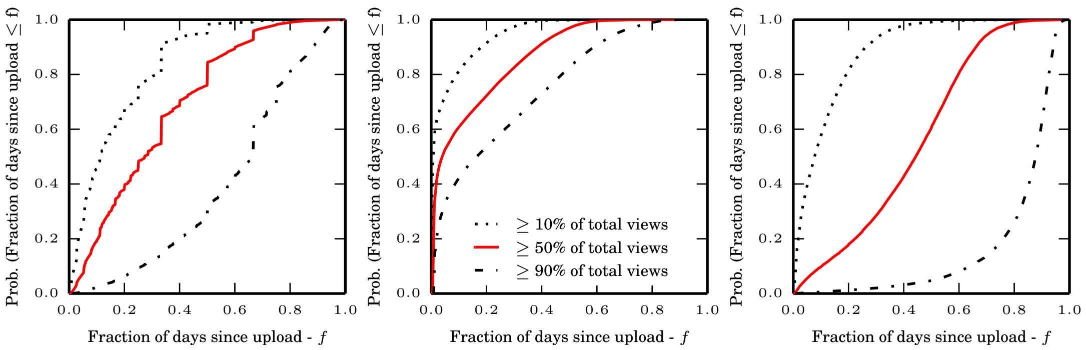
3.3 Understanding Video Popularity Growth

Recall that we established 5 questions that our study aims to address. We start by analyzing the popularity growth patterns of videos in our three datasets, focusing on two aspects: (1) the time interval until a video reaches most of its observed popularity, and (2) the bursts of popularity experienced by a video in short periods of times (e.g., days or weeks). We use the number of views as popularity metric because previous studies have found large correlations between final number of comments (or favorites) and final view count [26]. Moreover, we have also found positive correlations, ranging from 0.18 to 0.24, for both pairs of metrics, taken at each point in time (instead of only for the final snapshot, as previously done).

3.3.1 How Early do Videos Reach Most of their Observed Views (Q1)?

Figure 3.2 shows the cumulative distributions of the amount of time it takes for a video to receive *at least 10%*, *at least 50%* and *at least 90%* of their final (observed) views, measured at the time our data was collected. Time is shown normalized by the age of the video, which is here referred to as the video's *lifespan*. That is, the y-axis shows the fraction of videos that achieved at least 10%, 50%, and 90% of their final views (considering the final views at the time we crawled the data) in a period of time that does not exceed the value shown in the x-axis (which is normalized by the total time since the video was uploaded).

We note that, for half of the videos (y-axis) in the Top, YouTomb and Random datasets, it takes at most 67%, 17% and 87%, respectively, of their total lifespans (x-axis) until they receive at least 90% of their final views. If we consider at least 50% of their final views, the fractions are 27%, 4% and 44%, respectively, following a similar trend (as also found for the mark of 10% of the views). Conversely, around 34% of Top videos take at least 20% of their lifespans to reach at least 10% of their observed popularity. Similarly, 19% of videos in the Random dataset experience a similar dormant period before starting to receive most views. In contrast, only 8% of the YouTomb videos take 20% or more of their lifespans to reach at least 10% of their observed popularity.



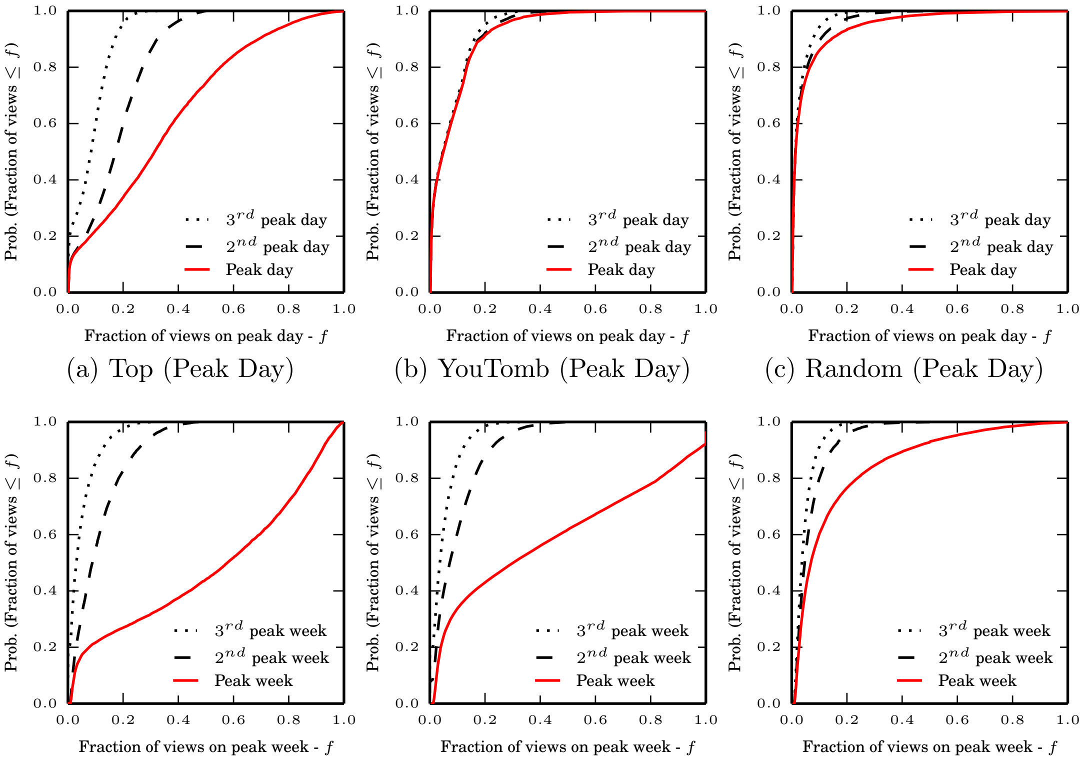
(a) Top (b) YouTomb (c) Random

Figure 3.2: Cumulative Distributions of Time Until Video Achieves at Least 10%, 50% and 90% of its Total Observed Popularity (time normalized by video's lifespan).

Thus, comparing the results across datasets, YouTomb videos tend to get most of their views earlier in their lifespans, followed by videos in Top and Random datasets. As videos in the top lists tend to be more popular, the difference between the results for Top and Random datasets are somewhat predictable. Possible reasons as to why YouTomb videos tend to receive most of their views even earlier are: (1) as many of these videos consist of popular TV shows and music trailers, a natural interest in this content closer to when it is uploaded is expected, and (2) being aware that such videos contain copyright protected content, users may seek them quicker after upload, before the violation is detected and they are removed from YouTube.

We note that since lifespan is a normalized metric, these results may be impacted by the distributions of video ages (Table 3.2). In particular, recall that such distribution is skewed towards older videos in the YouTomb dataset: around 86% of them have at least 1 year of age. This bias may influence the results. However, we also note that 59% of the videos in the Random dataset also fall into the same age range. Yet, in comparison with YouTomb, videos in the Random dataset get most of their views later.

Thus, to reduce any bias caused by age differences, we repeat our analyses separately for videos in each age range. Table 3.4 shows results for the time until a video achieves at least 90% of its views, presenting averages and standard deviations for each age range and dataset. Similar results occur for videos in most age ranges: YouTomb videos reach at least 90% of their views much earlier in their lifespans than Top videos, which are followed by videos in the Random dataset. The only exception occurs for the youngest videos, for which there is no much difference across the datasets.



(a) Top (Peak Day) (b) YouTomb (Peak Day) (c) Random (Peak Day)

(d) Top (Peak Week) (e) YouTomb (Peak Week) (f) Random (Peak Week)

Figure 3.3: Cumulative Distributions of the Fraction of Total Views on the First, Second and Third Peak Days/Weeks.

3.3.2 Is Popularity Concentrated in Bursts (Q2)?

We now investigate the popularity bursts experienced by the videos. We first analyze the distributions of the fraction of views a video receives on its most popular (i.e., peak) day, shown in Figure 3.3 and summarized in Table 3.5 for videos falling in different age ranges. Figure 3.3 also shows distributions for the second and third most popular days. Each curve in a graph of Figure 3.3 shows the fraction of videos (y-axis) that receive at most f% (shown in x-axis, as a fraction) of its final views on the given peak day.

Figure 3.3-a) shows that Top videos experience a very distinct peak day: 50% of them receive between 31% and 100% of their views on a single (peak) day. In comparison, the same fraction of videos receive between 17% and 50% of their views on the second peak day, and between 8% and 34% of their views on their third peak day. Thus, Top videos clearly experience a burst of popularity on a single day. This is in sharp contrast with videos in the YouTomb and Random datasets (Figures 3.3-b and 3.3c), where the three curves are very close to each other and skewed towards smaller fractions of views. While these results might reflect diverse popularity patterns,

Table 3.4: Normalized Time Until at Least 90% of Total Views, Grouped by Video Age (mean μ, and standard deviation σ).

	Top		YouTomb		Random	
	μ	σ	μ	σ	μ	σ
age (days) ≤ 7	.64	.10	-	-	.60	.16
$7 <$ age ≤ 30	.56	.19	-	-	.66	.21
$30 <$ age ≤ 365	.50	.27	.10	.13	.80	.17
age > 365	.77	.23	.26	.23	.85	.12

Table 3.5: Fraction of Views on Peak Day Grouped by Video Age (mean μ, and standard deviation σ).

	Top		YouTomb		Random	
	μ	δ	μ	δ	μ	δ
age (days) ≤ 7	64%	.16	-	-	63%	.20
$7 <$ age ≤ 30	35%	.15	-	-	33%	.19
$30 <$ age ≤ 365	23%	.16	21%	.13	8%	.11
age > 365	2%	.03	20%	.03	1%	.02

with more videos in the Random and YouTomb datasets having multiple (smaller) daily peaks, we note that the interpolation performed over the collected data might introduce distortions in this analysis, particularly given the large fraction of older videos in those two datasets.

To cope with these possible distortions, we also analyze the distributions of the fraction of views on the first, second and third peak weeks. Figures 3.3(d-f) show that videos in all datasets tend to exhibit some burst of popularity on a single week. However, the general trend remains the same as the one observed for daily peaks: the peak week tends to be more significant for Top videos, followed by videos in the YouTomb and Random datasets.

The same general conclusions, for both weekly and daily popularity peaks, also hold for videos falling in different age ranges, as illustrated in Table 3.5 for daily peaks.

3.3.3 Discussion

In this section we characterized content popularity growth, focusing on our first two research questions (Q1 and Q2). In general, we note that results vary according to the analyzed dataset. While Top and YouTomb videos tend to be more concentrated and receive most views earlier in their lifespans, videos in the Random dataset exhibit less clear bursts, particularly at the daily granularity, and tend to take longer to receive most views. These results contrast and complement previous analyses of YouTube videos, where the authors characterized a sample of videos uploaded on a single day, concluding that they exhibit concentrated popularity growth patterns [14]. By analyzing different datasets, composed of videos with different characteristics, our study is able to reveal different aspects of YouTube as a whole.

These results might be useful for a wide range of social media services. For example, they raise the question of whether (and when) it is beneficial to incorporate popularity estimates into search engine rankings. For videos that receive most of its views in short time periods (such as videos in the Top and YouTomb datasets), adding this information into the ranking after the period of interest has already passed might hide other (possibly more relevant) videos (e.g., newly uploaded videos). Another interesting argument is for advertisement services. The notion that popular content may have a higher ad-visibility has been discussed only recently [17]. However, focusing on the final observed popularity may be misleading, since posting ads on popular videos does not necessarily promote a higher amount of *future audience* as we shall discuss in Chapter 3.

3.4 Popularity Temporal Dynamics (Q3)

We now characterize the temporal dynamics of popularity of YouTube videos, aiming at identifying governing popularity trends that characterize groups of videos in our datasets. To that end, we employ the KSC algorithm [146], which is a K-Means like clustering algorithm focused on extracting similar trends (or shapes) from time series. KSC is based on a distance metric that captures the similarity between two time series with scale and time shifting invariants (see our discussion on Chapter 2).

KSC requires all time series to have equal length. Thus, we focus on videos with more than 100 days, whose popularity time series is defined by 100 evenly distributed observations, that is, the original crawled data with no interpolation[6]. Each such observation represents the popularity of the video at a time window w, whose duration depends on the video age. We also focus on the Top and Random datasets, since the non-interpolated data from the YouTomb dataset has all zeros after the removal date, which leads to time series with various lengths that cannot be handled by KSC.

[6] The popularity curves of those videos capture longer term popularity dynamics and trends.

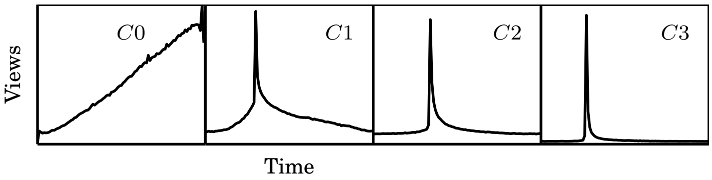


Figure 3.4: Popularity Trends (Cluster Centroids) in Both Random and Top Datasets.

After such filtering, we are left with 4,527 and 19,562 videos in the Top and Random datasets, respectively. These are the videos analyzed in this section (and in Section 3.6).

Like K-means, the KSC algorithm requires the target number of clusters k to be given as input. We use the β_{CV} heuristic [92] to define the best value of k. The β_{CV} is defined as the ratio of the coefficient of variation (CV) of the intracluster distances and the coefficient of variation of the intercluster distances. The smallest value of k after which the β_{CV} remains roughly stable should be selected, as a stable β_{CV} implies that new splits affect only marginally the variations of intra and intercluster distances. The values of β_{CV} seem to stabilize for k=4, for both analyzed datasets. We confirmed this choice by plotting the clustering cost, silhouette and Hartigan's index metrics [146], and by visually inspecting the members of each cluster. The best value of k was 4 according to all these techniques.

Figure 3.4 shows the discovered popularity trends (the centroids of the identified clusters), which govern popularity evolution in our datasets. Each graph shows the number of views as function of time. Note that the same four popularity trends are present in both analyzed datasets. Moreover, Table 3.6 presents, for each cluster, the number of videos that belong to it as well as the average number of views, the average change rate in the number of views, and the fraction of views at the peak time window of these videos. The average change rate is the average difference between two (non-cumulative) measures taken in successive time windows. Thus, it captures the trend in the number of views of the video: a positive (negative) change rate indicates an increase (decrease) with time, whereas a change rate equal to 0 indicates stability. Table 3.6 shows the average change rate computed over the total duration of the video's lifespan. The peak fraction, also shown in Table 3.6, is the ratio of the maximum number of views in a time window divided by the final number of views of the video.

As shown in Figure 3.4, cluster 0 consists of videos that remain popular over time, attracting an increasing number of views per time window as time passes, as indicated by the large positive change rates (Table 3.6). This behavior is specially strong in the Top dataset, with an average change rate of 1,112 views per window, which corresponds

Table 3.6: Summary of Popularity Trends

	Top				Random			
	$C0$	$C1$	$C2$	$C3$	$C0$	$C1$	$C2$	$C3$
Number of Videos	958	1,370	1,084	1,115	4,023	6,718	5,031	3,790
Avg. Number of Views	711,868	6,133,348	1,440,469	1,279,506	305,130	108,844	64,274	127,768
Avg. Change Rate	1112	395	51	67	47	7	4	4
Avg. Peak Fraction	0.03	0.04	0.19	0.74	0.03	0.03	0.08	0.28

to roughly a week in that dataset. The videos in cluster 0 have also no significant peaks, as the average fractions of views in the peak windows are very small (Table 3.6). The other three clusters are predominantly defined by a single peak in popularity followed by a steady decline. The main difference is the rate of decline, which is much slower in Cluster 1, somewhat faster in Cluster 2, and very sharp in Cluster 3. This difference is more clear if we analyze the peak fractions and the average change rates in Table 3.6.

Given the popularity (i.e., scale) invariant nature of the KSC algorithm, it is important to highlight the differences between the clusters in the Top and Random datasets. To that end, we make use of the numbers in Table 3.6. Although very similar clusters exist in both datasets (determined both by the shape of the centroids and the fraction of videos in each cluster), notice that the change rates in popularity for the videos in the Top dataset are much higher (for every cluster) than the corresponding rates in the Random dataset. For example, videos in Cluster 0 (which remain popular over time) in the Top dataset experience a change in number of views in consecutive time windows of 1,112 views, on average. In contrast, videos in the Random dataset experience a change of only 47 views, on average.

Also notice how the peak fractions in the Top dataset are higher than those in the Random dataset (in all clusters but Cluster 0). However, the average number of views in Cluster 0 in the Top dataset is the smallest one when compared to the other clusters in the same dataset. For the Random dataset, this is the opposite. This is very interesting, as it indicates that the most popular videos in the Top dataset are in Clusters 1-3, that is, they experience clear popularity peaks, being more popular in shorter time windows. However, given the very high change rates experienced by videos in Cluster 0 (in Top), we might speculate that videos in this cluster will become more popular over time, as they capture enough interest to remain receiving visitors over time. We might also speculate that, as time passes and the Top videos in Clusters 1-3 loose their appeal to the audience, the relative distribution of popularity across clusters in the Top dataset will be more similar to that in the Random dataset. This is a conjecture that requires further investigation in the future.

It is also important to note that Clusters 1, 2 and 3 were previously uncovered in other YouTube or Twitter datasets [34, 80, 91]. Crane and Sornette [34] explained their occurrences by a combination of endogenous user interactions and external factors. According to them, Cluster 1 consists of videos that experience word-of-mouth popularity growth resulting from epidemic-like propagation through the social network; Cluster 2 includes videos that experience a sudden popularity burst, due to some external event, but continue spreading through the social network afterwards; and Cluster 3 consists of videos that experience a popularity burst for some reason (e.g., spam) but do not spread through the social network. However, these previous studies relied mostly on peak popularity analyses [80] and fitting power-law decays after the peak [34, 91]. Instead, we here use an unsupervised learning algorithm that makes our task of discovering popularity trends more general and robust. For example, the thresholds in peak volume that define different trends in these previous studies are not clearly defined. In contrast, such peaks emerge clearly in our clusters (as shown in Table 3.6).

Notice however that no previous study that analyzed video popularity time series or other UGC time series has identified a trend similar to Cluster 0, possibly because of the models they adopted, which focus on power-law like behavior [34, 91] or due to inherent differences in media consumption trends for different media types [146]. The existence of Cluster 0 can be attributed to three possible reasons. Firstly, there are certain topics that users will continue to revisit over time [5, 136], and thus the content will not follow a rise-and-fall pattern (as proposed in [91]). Secondly, the propagation of these topics is much slower [136], being the pattern we see still part of the growth period in interest in that particular topic. Lastly, YouTube's own growth in popularity over time may cause the audience of interest in some videos to increase. Intuitively, a combination of these factors will likely be the case, and only recently researchers have started looking into the implications of each of them [5, 136].

Finally, we note that other time series clustering techniques could also be employed to extract popularity trends from our datasets. For example, one could consider first using *Symbolic Aggregate Approximation*, SAX [85] to represent the time series, and then applying traditional clustering methods (e.g., K-Means). However, SAX assumes that time series values are normally distributed, which is not true for our data (even after log and z-transformations). We argue that KSC is a suitable choice of clustering algorithm to our study because it: (1) requires only the number of clusters as input, (2) requires no data pre-processing, and (3) has well defined and interpretable centroids, which facilitates analyzing and drawing useful insights from the results.

This section bridges our study on Q1 and Q2, and thus have similar implications for social media services. So far we characterized video popularity focusing on popularity time series only. We have yet to discuss possible reasons behind content popularity and popularity trends. We explore these issues in the next two sections. Throughout the rest of the chapter, we refer to Clusters 0, 1, 2 and 3 as $C0$, $C1$, $C2$, and $C3$, respectively.

3.5 Referrer Analysis (Q4)

The dynamics of information propagation through friends in social networks has been studied before [21]. However, on YouTube, as on other social media applications, word-of-mouth is not the only mechanism through which information is disseminated. We here tackle this issue by investigating important referrers that lead users to videos (Section 3.5.1) and their first access since video upload (Section 3.5.2). These analyses are performed on our three original datasets (Table 3.1).

3.5.1 Which Referrers are More Important for Video Popularity (Q4a)?

Table 3.7: Referrer Categories and Statistics (t_{view}: number of views (x 10^9); f_{view}: the fraction of views; f_{time}: fraction of times a referrer from the given category was the first referrer of a video).

	Top			YouTomb			Random		
	t_{view}	f_{view}	f_{time}	t_{view}	f_{view}	f_{time}	t_{view}	f_{view}	f_{time}
EXTERNAL	0.57	0.11	0.35	0.81	0.16	0.41	0.07	0.08	0.22
FEATURED	0.72	0.14	0.03	0.10	0.02	0.00	0.11	0.14	0.00
INTERNAL	1.50	0.29	0.67	1.85	0.36	0.65	0.14	0.18	0.34
MOBILE	0.26	0.05	0.51	0.02	0.00	0.02	0.03	0.03	0.05
SEARCH	1.05	0.20	0.36	1.80	0.35	0.52	0.29	0.37	0.41
SOCIAL	0.36	0.07	0.35	0.01	0.00	0.01	0.01	0.00	0.12
VIRAL	0.81	0.16	0.79	0.59	0.12	0.62	0.16	0.20	0.55

Recall that the referrers in our datasets were grouped into seven categories: External, Featured, Search, Internal, Mobile, Social, and Viral. Table 3.7 shows the number (n_{view}) and fraction (f_{view}) of views for which each category is responsible. The table shows that search and internal YouTube mechanisms are key channels through which users reach content on the system, and we note that YouTube search is responsible for more than 99% of all Search referrers. Oliveira *et al.* [109] posed the hypothesis that search is the main method for reaching content on video sharing websites, verifying it through questionnaires with volunteers. Whereas our results confirm their hypothesis for videos in the Random dataset, we find that YouTube internal features (e.g., “Related Videos") play an even more important role to content dissemination for Top videos. For YouTomb videos, both categories are roughly equally important. In general, we find that search is more important to Random and YouTomb videos, as they are not systematically exposed to users as videos from top lists are. We also note the importance of the Viral category in all datasets, particularly Random.

We further analyze the importance of each referrer category by computing the distributions of the number of views for which each category is responsible, *taking only videos that received accesses from the given category*, and computing percentages based on the *total views from referrers only (accounted views)*. Figures 3.5(a-c) show box plots containing 1^{st}, 2^{nd} and 3^{rd} quartiles, 9^{th} and 91^{th} percentiles, and the mean, for each category and each dataset[7]. Unlike Table 3.7, which shows aggregated results (i.e., results for *all videos in each dataset*), these plots allow us to assess the importance of each referrer category for individual videos.

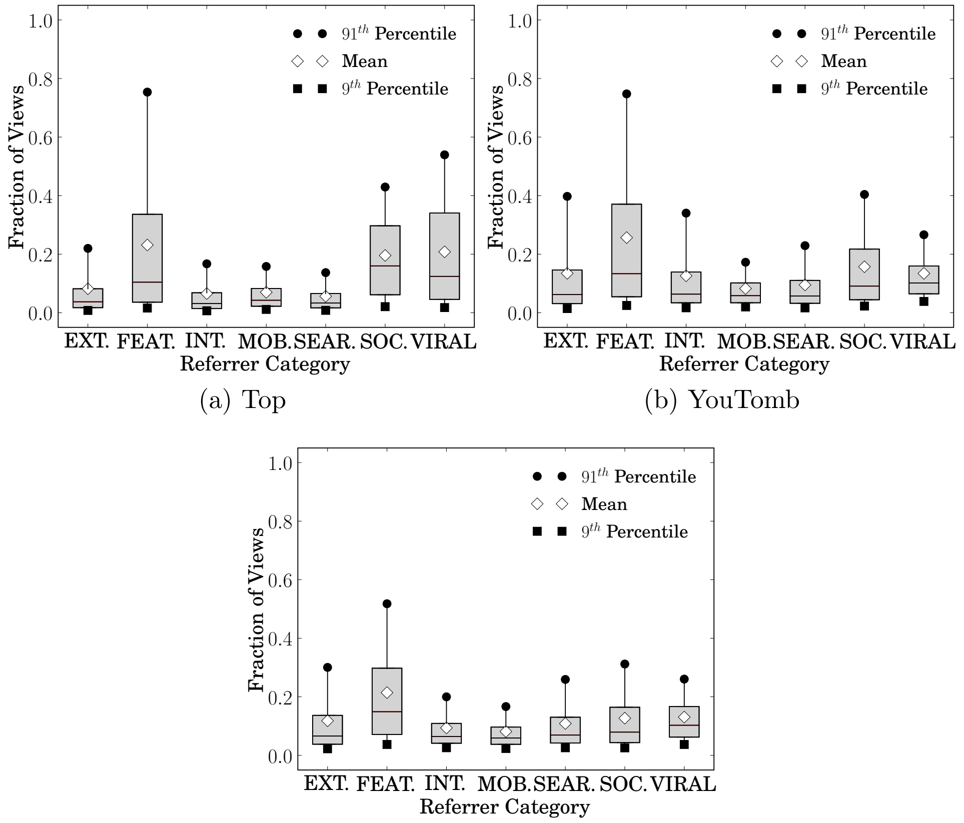
For example, Table 3.7 shows that Social referrers do not appear to be important for YouTomb dataset as a whole. However, taking only copyright protected videos with at least one Social referrer, Figure 3.5-b) shows that, for 25% of such videos (1^{st} quartile), more than 22% of the accounted views come from subscription links. Thus, users do subscribe to other users who post copyright protected content. The Featured category is a similar case. For Top videos, the Social, Featured and Viral categories are responsible for more than 30%, 33% and 34%, respectively, of the accounted views for 25% of the videos with referrers from each such category (Figure 3.5-a). Finally, Featured referrers play a key role to attract views to Random videos: 25% of the videos with Featured referrers received at least 30% of the accounted views from such referrers (Figure 3.5-c).

It is hard to tell whether one referrer influences the number of views from other referrers. For example, a Top video may experience a popularity growth from Social and Viral referrers *after* being featured in the top list. Next, we study this issue by analyzing how early in a video's lifespan each type of referrer is used.

3.5.2 How Early do Referrers Appear (Q4b)?

We now analyze the referrers that first lead users to a video. Table 3.7 also shows the fractions of videos that had the first referrer falling into each category (f_{time}). Since

[7] For any given referrer category, at least 1,000 videos received views for which it is responsible.



(a) Top

(b) YouTomb

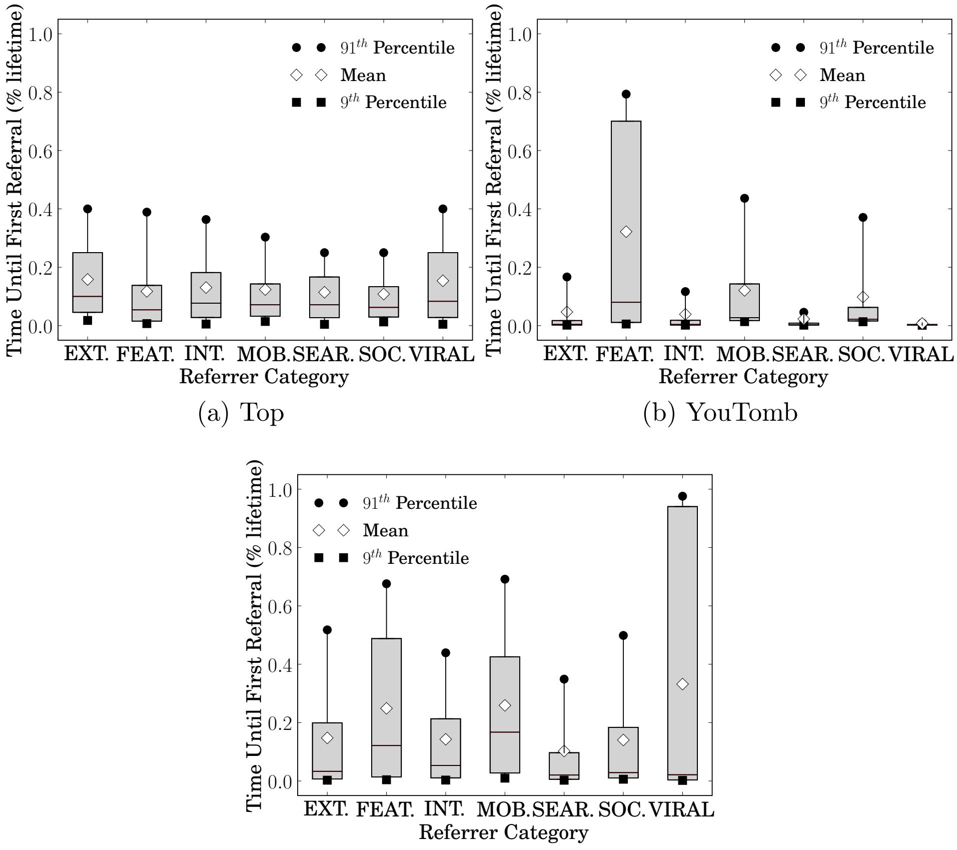
(c) Random

Figure 3.5: Fraction of Views From Each Referrer Category.

YouTube provides only the *day* each referrer was first used, there might be ties with multiple categories, and the sum of f_{time} may exceed 100% for a dataset.

In general, viral spreading and internal YouTube mechanisms appear as primary forms through which users reach the content for the first time, in all three datasets. For example, the first referrers for 79%, 67%, and 51% of the Top videos are from the Viral, Internal, and Mobile categories, respectively. For the YouTomb dataset, Internal, Viral, and Search contain the first referrers for 65%, 62% and 52% of the videos, respectively. For the Random dataset, the first referrers of 55%, 41%, and 34% of the videos are from the Viral, Search, and Internal categories, respectively. Interestingly, mobile devices are also a relevant front door to Top videos, whereas for YouTomb and Random videos, the YouTube search engine accounts for a large fraction of the first referrers.

Figures 3.6(a-c) show the distributions of the difference between the time of the first referrer access and the time the video was uploaded, measured as a fraction of the video's lifespan. For the Top and YouTomb datasets, referrers (of any category) tend to happen very early: for 75% of the Top and YouTomb videos, most referrer categories have their first appearances during the first quarter of the video's lifespan. Indeed, only



(a) Top

(b) YouTomb

(c) Random

Figure 3.6: Time Until the First Referrer Access (normalized by video's lifespan).

9% of the Top videos have their first referrer access (of any category) after 40% of their lifespans. The exception is the Featured category on YouTomb: those referrers tend to take more time to appear. This suggests that YouTube may try to avoid featuring videos that are suspicious or have potential to be copyright protected. For Random videos, in general, Search, Internal, External, and Social referrers tend to appear earlier than other types of referrers. Thus, users are more likely to initially find such videos through social links, search, other YouTube mechanisms or external websites, instead of receiving them via e-mail or viewing them on mobile devices.

3.5.3 Discussion

We here focused on identifying the most important referrers that lead users to a video (Q4). Our results are useful to help content creators to increase their viewership. For instance, search engines seem to attract most viewers to content, and they do so early on the video's lifespan (Table 3.7). However, focusing on particular videos, we find that this may not hold for every case (Figure 3.5). One suggestion to content creators would thus be to provide good textual descriptions of video content, which would likely help search engine users to find it. Afterwards, a careful monitoring of how the video propagates on external websites and internal OSNs may be used to further boost viewership.

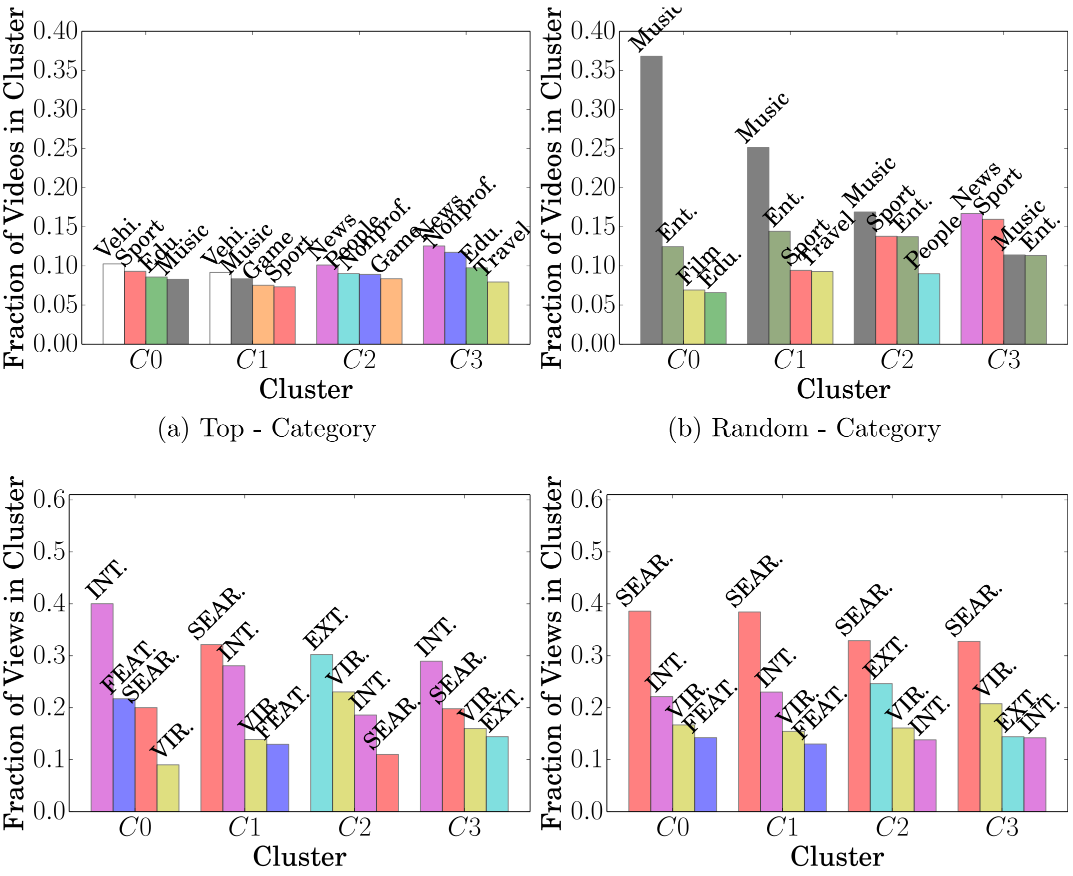
3.6 Associations Between Various Features and Popularity (Q5)

We now tie the analyses of the previous sections together by assessing how different features are associated with the identified popularity trends, and also with final observed popularity values. We first analyze whether videos that follow a similar popularity trend tend to have content in the same topic and be reached through similar referrers (Section 3.6.1). We then measure the correlations between various features (shown in Table 3.3) and the popularity trend and observed popularity value of the videos (Section 3.6.2). As in Section 3.4, we here focus on the Top and Random datasets.

3.6.1 What Kinds of Content and Referrers are Responsible for Each Popularity Trend? (Q5a)

We start by analyzing whether videos that follow a similar popularity trend (same cluster) tend to have content in the same categories. For both datasets, we found that the distributions of the number of videos across categories in each cluster are statistically different from the distribution computed over all videos in the dataset, according to a Chi-Square test with p-value < 0.01. Thus, videos in different clusters tend to be concentrated around different categories (or topics). In Figures 3.7(a-b) we show the fractions of videos in the top 4 categories in each cluster, for each dataset.

Starting with Top videos, Figure 3.7(a) shows a clear divergence in the topics of the videos in each cluster. Clusters $C0$ and $C1$, which consist of videos that tend to attract viewers for longer periods, are composed mostly by videos about music, sports, and automobiles, while journalistic videos (news), video blogs (people) and videos related to activism (non-profit) are the most common topics in clusters $C2$ and $C3$, which tend to have much shorter viewer retention periods. This mostly likely occurs because such videos tend to be interesting only during short time periods. For the Random dataset, Figure 3.7(b) shows that videos with music and entertainment content are very frequent in all four clusters. This may occur due to a natural bias of copyrighted content and of the queries used to build that dataset. Regardless, the



(a) Top - Category

(b) Random - Category

(c) Top - Referrer

(d) Random - Referrer

Figure 3.7: Fractions of Videos Per Category (a,b) and Fraction of Views Per Referrer (c,d) For Each Cluster.

frequencies of these categories tend to decrease, while news tends to become more frequent in clusters $C2$ and $C3$.

We now turn to the referrers used to reach videos in each cluster, and analyze the fractions of views each type of referrer is responsible for, on average[8]. Once again, for both datasets, the distributions of these fractions in each cluster are statistically different from the distribution computed for all videos in the dataset. Thus, the types of referrers that attract the largest fractions of views do vary depending on the popularity trend. Figures 3.7(c-d) show the results for each dataset, focusing again on the top 4 referrers per cluster. Note that the only dataset where Search is the most important type of referrer for all trends is Random, due to the nature of its crawling process. However, Search, Internal and Viral referrers tend to be among the top 4 referrers in all clusters of both datasets. Moreover, Featured referrers are among the most important ones for videos that remain attractive for some time ($C0$ and $C1$), while

[8] These fractions are computed based on the final number of views received through the referrers.

External referrers play an important role for videos that experience a sudden burst of popularity ($C2$ and $C3$).

3.6.2 What are the Correlations Between Features and Popularity Trends and Values? (Q5b)

Finally, we measure the correlations between the features shown in Table 3.3 and the popularity trends and the final popularity values of the videos (at the time of data crawling). To that end, we use the maximal information coefficient, or MIC [117]. MIC results range from 0, for no correlation, to $+1$, for strong (positive or negative) correlation. This novel metric captures the normalized mutual information measure between two features. It measures correlations between different types of features (e.g., categorical and numeric) and is able to detect non-linear and even periodic types of relationships, a limitation of other coefficients (e.g., Pearson and Spearman). We also used the Information Gain and the Gini coefficient [32] to measure the correlations, obtaining qualitatively similar results.

Since the values of some referrer and popularity features vary with time, we compute MIC results for various *monitoring periods*. That is, we express the monitoring period as a fraction of the video's lifespan, and compute feature values only for that period. For example, the correlation between number of views and popularity trends for a monitoring period of 10% is computed taking the number of views received during the first ten time windows, since each time series has 100 windows. By doing so, we can identify the most important features in different phases of the video's lifespan.

Table 3.8 shows, for each dataset, MIC results between the features and *popularity trends*, for monitoring periods equal to 1%, 5%, 50% and 100%. As the number of features is large, we aggregate MIC results for each feature class - video, referrer, and popularity, and present mean (μ) and maximum MIC for the features in each class. Similarly, Table 3.9 shows the MIC results between features and final (observed) *popularity values*. Since YouTube provides only the total number of views associated with each referrer, we only consider these features for a monitoring period equal to 100%, taking only the other referrer features (e.g., date of each referrer) for shorter periods.

We start by noting that, as the monitoring period increases, popularity features tend to greatly surpass the others in importance for correlations with both trends and popularity values. For trends, the popularity feature with maximum MIC is peak fraction, whereas for popularity values, it is number of views. For both of them, the correlations are above 0.5 for monitoring periods beyond 50%, in both datasets.

Table 3.8: Average (μ) and Maximum (max) Maximal Information Coefficient (MIC) Values per Feature Type for Popularity Trend.

Dataset	Feature	Monitoring Phase 1% of Lifespan		5% of Lifespan		50% of Lifespan		100% of Lifespan	
		μ	max	μ	max	μ	max	μ	max
Top	Popularity Features	.15	.19	.15	.27	.18	.63	.26	.84
	Referrer Features	.11	.17	.11	.18	.12	.18	.11	.19
	Video Features	.02	.19	.02	.19	.02	.19	.02	.19
Random	Popularity Features	.04	.07	.05	.17	.13	.52	.20	.75
	Referrer Features	.04	.08	.04	.08	.05	.08	.08	.15
	Video Features	.01	.07	.01	.07	.01	.07	.01	.07

Table 3.9: Average (μ) and Maximum (max) Maximal Information Coefficient (MIC) Values per Feature Type for Total (Observed) Popularity Value.

Dataset	Feature	Monitoring Phase 1% of Lifespan		5% of Lifespan		50% of Lifespan		100% of Lifespan	
		μ	max	μ	max	μ	max	μ	max
Top	Popularity Features	.31	.48	.32	.56	.41	.88	.57	1
	Referrer Features	.16	.27	.17	.28	.20	.31	.32	.74
	Video Features	.11	.31	.11	.31	.11	.31	.11	.31
Random	Popularity Features	.16	.32	.22	.48	.36	.89	.51	1
	Referrer Features	.09	.11	.10	.13	.12	.18	.26	.68
	Video Features	.08	.18	.08	.18	.08	.18	.08	.18

However, for shorter periods, the other (referrer and video) features are also very important. This is interesting for popularity prediction tasks [4, 110, 112, 129], since popularity features computed over short monitoring periods might be very unstable, and prediction must rely mostly on other pieces of information about the video, such as its category and referrers. Indeed we exploit both referrer and category features when predicting popularity in Chapter 6.

We further note that the relative average importance of each feature group is the same for both datasets: popularity features are more important than referrer features, which are more important than video features. The major differences between both datasets lie in the individual features within each feature class, as we discuss below. We now focus on the correlations computed for popularity trends (Table 3.8). Albeit not shown in the table, for short monitoring periods (e.g., 1%), the most important video feature is the video age (MIC of 0.19 for Top and 0.07 for Random), while the number of views is the most important popularity feature (MIC of 0.19 for Top and 0.17 for Random). The most important referrer feature is the date of the first Internal referrer (MIC=0.17) for Top videos and the date of the first External referrer (MIC=0.08) for videos in the Random dataset. As time passes, the fraction of views on the peak window becomes the most important feature overall. This is expected, since popularity trends are either concentrated on peaks or exhibit linear growth (Section 3.4).

In contrast, when correlating with the final observed popularity values (Table 3.9), the most important feature is the number of views, for all monitoring periods. At very early points in time (1% of lifespan), the most important video and referrer features are video age (tied with upload date with MIC = 0.31) and the date of the first *Viral* referrer (MIC=0.26) for Top videos. For the Random dataset, they are video age (again, tied with upload date with MIC=0.18) and the date of the first *Search* referrer (MIC=0.1).

Thus, from the perspective of popularity prediction, having fixed the monitoring period, the most important features to be explored depend on whether one aims at predicting a trend or a value. For example, previous work showed that, by knowing the trend of a video *before hand*, the accuracy of the prediction of popularity values can be improved [110,145]. However, we are not aware of any previous effort to predict the popularity *trend* of a video (or UGC in general) as we do in Chapters 5 and 6.

We also note that, when correlating with both trends and observed popularity values with a monitoring period of 1%, the Music category is in the top 10 most correlated features for the Random dataset[9]. The News category is also in the top 10 features when correlating with trends. This result is in agreement with Figures 3.7(a-b), which show a more skewed concentration of categories across trends in the Random dataset. Moreover, when correlating with both trends and observed popularity values with a monitoring period of 100%, some referrer features, mainly the number of views from the referrers, are in the top 10 most important features, in both datasets. We believe that such features would also be important at shorter monitoring periods. However, their values are not available in our dataset. Thus we cannot test this hypothesis.

3.6.3 Discussion

The correlations unveiled in this section motivate the need to explore a diverse set of features for popularity prediction tasks. Most previous efforts explored only early points in the popularity time series [4,110,112,129]. Our results show that they could benefit from considering also other features. In particular, we found that while some referrer and video features may be useful to predict, at very early stages in the video's lifespan, how its popularity will evolve over time (the trend), early popularity measures are the most useful features to predict future popularity values. However, as discussed

[9] We represented each category by a binary feature, and computed correlations for each category separately.

by previous work [110, 145], one task may complement the other. Also, the differences in relative importance of individual features across datasets, particularly when considering early periods after video upload, raises a question of whether different prediction methods (i.e., methods that exploit different sets of features) should be designed for different groups of videos.

Other applications that may benefit from our results are recommender systems. By exploring important features that correlate with popularity, useful recommendations may be produced even before a video becomes popular. However, in this case a chicken-and-egg problem arises. Will a video become popular because it is interesting or due to the recommendation engine? Investigating causality between factors that impact content popularity is an important open question, which we leave for future work.

3.7 Summary

In this chapter have characterized the dynamics of video popularity on the currently most popular video sharing system, YouTube. Driven by 5 research questions, we analyzed how the popularity of individual videos evolve since upload (Q1 and Q2), extracted common trends of popularity evolution (Q3), characterized the types of referrers that lead users to videos (Q4), and correlated popularity trends and final observed popularity values with various features (Q5). Our analyses were performed on three YouTube datasets, providing a broad view of the popularity evolution for a diverse set of videos.

We found that copyright protected (YouTomb) videos tend to get most of their views much earlier in their lifespans, followed by Top videos, and then videos in the Random dataset. We also found that Top videos tend to experience significant popularity bursts, receiving a large fraction of their views on a single day (or week). YouTomb videos also follow this pattern, and this is less of a case for Random videos. However, using a time series clustering algorithm, we found that the same 4 popularity trends seem to explain how video popularity evolves in both Top and Random datasets.

We also characterized the main referrers that led users to videos in each dataset. Particularly, we showed that search and internal YouTube mechanisms, such as lists of related videos, are key mechanisms that attract users to the videos. Whereas Search referrers account for the largest fraction of views to videos in the Random dataset, internal mechanisms play an even more important role to content dissemination for Top and YouTomb datasets. Also, our correlation results show that various video and referrer features can be explored for popularity prediction, and not only features extracted from early points of the popularity time series, as done by most previous efforts.

Our main findings can be applied in several contexts, as discussed next.

Content Distribution: we found that, even after short monitoring periods, there exists some correlations between popularity trends and the analyzed features, motivating their use for predicting *popularity trends*. Content distribution networks could use such predictions, together with observed popularity estimates, for load balancing, by provisioning videos predicted to remain popular for longer (i.e., videos in $C0$ or $C1$) to more capable servers. For videos predicted to be in $C2$ or $C3$, as their popularity growth rates decrease there is a high chance that the attention for them will drop. Such videos should then be provisioned by less capable servers or sent to secondary storage. Similarly, this knowledge could be used by ISPs for local caching.

Online Advertising: our results also suggest that different video categories tend to more often follow different popularity trends (e.g., $C0$ and $C1$ are dominated by music and sports videos, while $C2$ and $C3$ by news and non-profit ones). This knowledge could be used by advertisers to drive the selection of the video categories for ad placement, and by online advertising platforms to provide category-based price differentiation for advertisers (e.g., higher prices for categories that tend to remain popular for longer). Our results are also potentially useful for content publishers, who may profit from ads placed on their videos. The finding that Search (and Featured) referrers attract more views for videos that remain attractive over time (i.e., videos in $C0$ and $C1$) suggests that content publishers could periodically refine the keywords assigned to their videos (e.g., tags, title) to target different queries over time. For example, after a cycle of popularity growth and decay, publishers could adjust the video keywords and descriptions to possibly target other searchers that exploit related terms to find the video.

Monitoring Fame and Popularity: From a social perspective, understanding content popularity could be used for monitoring fame and popularity of content producers, and analyzing how users seek-out and consume information on real world events (e.g.,natural disasters, gossip news).

The results provided in this chapter represent a major cornerstone in the development of the rest of this dissertation. For instance, in the next chapter we shall focus on a complementary analysis analyzing how user preferences relate to information popularity in social media applications. Also, our findings on correlations between features and popularity trends/values motivated our popularity prediction methods presented in Chapters 5 and 6. Finally, our results on the importance of referrers for popularity were also used as motivations for our PHOENIX-R and A-FLUX models, discussed in Chapters 7 and 8.

Chapter 4

Users Perception of Content and Popularity

In the previous chapter we presented a characterization of how different features relate with the popularity evolution of YouTube videos. In this chapter, we seek to understand the extent to which the content by itself determines the popularity of a YouTube video. Using mechanical turk as experimental platform, we asked users to evaluate pairs of videos, and compared users' relative perception of the videos' content against the videos' relative popularity as reported by YouTube. This chapter will end our discussions on our first research goal, understanding feature importance. In this sense, users' perception of content is defined as a content feature of the social media objects. After this chapter, our studies on RG2 (predicting popularity) will be discussed in Chapters 5 and 6. Moreover, on RG3 (Chapters 7 and 8), we shall present another user centric view on popularity, our studies on user activities.

4.1 Introduction

What drives content popularity in a social media application? Recently, this question has attracted a lot of research attention as social media sites become increasingly popular platforms for exchanging information. An unresolved part of this question is about the relative roles of two primary forces that drive the popularity of a piece of information: (i) its content, i.e., the interestingness, topicality, or quality of the information *as perceived by users*, and (ii) its dissemination mechanisms, such as its propagation by word-of-mouth, blogs or mass media channels. It stands to reason that both factors matter, but the extent to which they impact the overall popularity of a piece of information remains an open question.

Our studies so far, as well as many previous studies on how information becomes popular in social media sites, focused on dissemination related factors (e.g, social influence, mechanisms that expose content to users, time of upload) [13, 78, 122, 123], ignoring the role of content itself. Other previous efforts, instead, analyzed social media content focusing on exploring content features for data mining tasks such as popularity prediction [147] and video classification [46], analyzing popularity differences in groups of content duplicates [13], and capturing content importance as a parameter in popularity evolution models [91]. In this chapter we take a different and complementary approach, focusing on understanding the extent to which content matters for popularity of videos on the YouTube social media site.

Our methodology attempts to assess *users' relative perception* of the contents of pairs of videos through user surveys conducted over Amazon mechanical turk. Users in our experiments are exposed only to the video content, and we took care to not subject them to other factors (inherent to the YouTube environment) that may impact user perceptions of content (e.g., user comments, social links, appearance of content in external sites). Specifically, we present to users pairs of videos from the same major topic and uploaded around the same date, and ask them to choose which one (none or both): (1) *they enjoyed more*, (2) *they would be more willing to share with friends*, and (3) *they predicted would become more popular on YouTube*. The first question targets the user's individual perception of content interestingness, the second captures the user's perception of the interests of her social circle (and thus the chance of the content spreading through it), and the third captures the expectations of the user on a global scale. Our goals are to assess, for each of these questions, whether users reach consensus, and, when there is consensus, whether user perceptions match the relative popularity achieved by the videos reported by YouTube.

We find that there is no consensus among participants in many evaluations, even when the popularity (reported by YouTube) of the evaluated videos differs by orders of magnitude. The lack of consensus is more striking when it comes to sharing and, to a lesser extent, liking choices, and also depends on the topic of the video content. This suggests that users' perceptions about content are quite subjective and that content is not the most important factor that drives popularity in many cases. Interestingly, our results also show that, whenever participants reach consensus, their choices, particularly for question (3), almost always match the video with largest popularity reported by YouTube, suggesting that, in these cases, content has a significant impact and predictive power on the popularity of YouTube videos.

Note that the goals of our study are complementary to those of previous work. In particular, Salganik *et al.* [122] also relied on a user study to understand popularity dynamics. However, they focused on the impact of social influence on popularity, whereas we focus on the role of content and rely on users to evaluate the content in a setup that is isolated (to the extent possible) from dissemination mechanisms that might influence popularity. To our knowledge, the human perceptions of content and how they correlate to popularity in a social media site (YouTube) have not been analyzed in any previous work. Accordingly, our experimental setup and findings are very different from those in prior studies. Thus, this work is a first step towards addressing the broad and fundamental question about the role of content in determining popularity of a piece of information, and our proposed experimental methodology, discussed next, a key contribution towards that goal.

In the next section we discuss our methodology. This is followed by our results in Section 4.3. Section 4.4 concludes this chapter.

4.2 Methodology

Aiming at taking a further step towards understanding to which extent the content itself impacts the popularity of social media, we have designed a crowd sourced based study of YouTube videos. Our study is guided by two questions:

Q1 Given a pair of videos with similar topic, can users reach consensus on the relative popularity of the videos?

Q2 When users do reach consensus, does the preferred video match the most popular video on YouTube?

Question Q1 is focused on the collective notion of popularity reported by the users in our experiment, who are subject only to the content itself. This notion relates to whether a user likes and/or would be willing to share a video more than the other, and also whether a user, despite personal tastes, believes one video would become more popular than the other. Question Q2 aims at comparing this notion with the popularity achieved by the videos on YouTube, measured by the total number of views at the time we collected the videos, which can be affected by various factors, other than content alone.

4.2.1 Datasets

Given our two research questions, the datasets employed on our evaluations need to eliminate as much bias as possible. For instance, a video may be more popular than another simply because it is older. Thus, the datasets employed in the Chapter 3 were shown to unsuitable for our user study. Thus, in order to identify pairs of videos with similar topic, we used Freebase[1], a collaborative semantic knowledge database that covers over 30 million topics, ranging from sports (e.g., baseball) to individuals (e.g., Muhammad Ali). Specifically, we crawled YouTube for videos that are indexed under the same Freebase topic on its API. We focused on two topics - *major league baseball*[2] and *music videos*[3], as they span different user interests and are neither too specific nor too general[4].

For each topic, we downloaded videos that were uploaded from the US on April 2012, considered safe by YouTube's safe search, and could be embedded in external sites. Videos were downloaded on August 2013. By studying videos of similar topic we factored out the notion of popularity due to latent social, cultural and psychological issues. For example, soccer is less popular than baseball in the US. For the same reason we focused on videos uploaded from the same country. Moreover, by focusing on videos uploaded around the same time, we factored out popularity variations due to first mover advantage [13] and upload date [78]. We also limited the potential of users disliking a video because they found it offensive by using the YouTube's safe search. Finally, we considered only videos that could be embedded so that user evaluations could be done outside YouTube, and thus be unaffected by the other pieces of information (e.g., number of views, user comments) provided by YouTube on a video's page.

To select videos with various YouTube popularity values, we defined three ranges: low popularity, defined by a number of views ranging from 10 to 100, medium popularity, with number of views between 1,000 and 10,000, and high popularity, with number of views between 100,000 and 1,000,000. For each topic, we selected 3 videos of each popularity range, with each video having between 4 and 6 minutes of duration.

4.2.2 Human Intelligence Tasks

We ran our user experiments on Amazon mechanical turk (MT). To recruit participants, we posted the pre-requisite that only master workers (i.e., the best workers as ranked by MT) based on the US could perform our task.

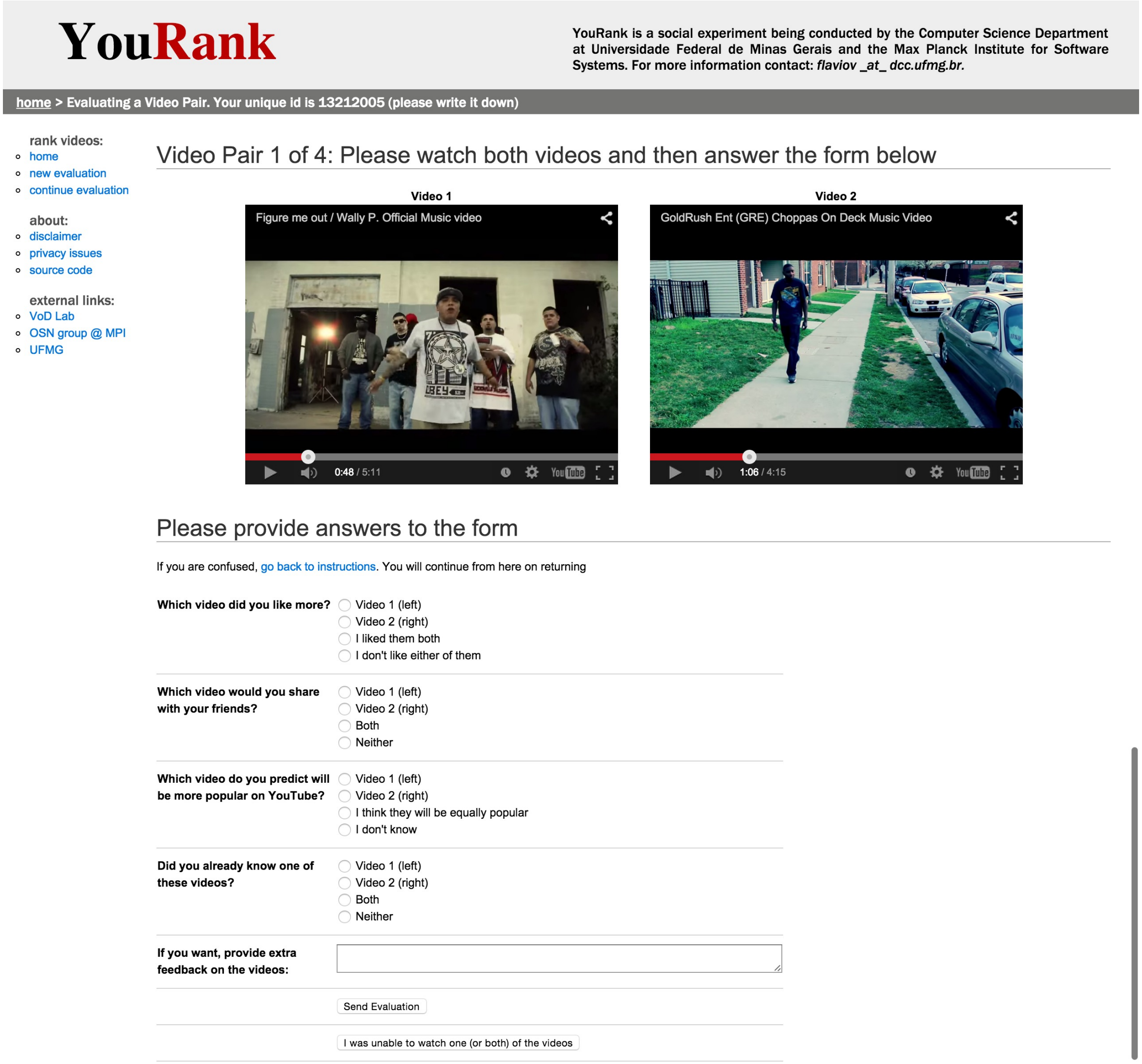
The first step was the build, for each topic, all 36 pairings for the 9 selected videos. These pairs were assigned to 9 *folds*, so as to have only unique videos in each fold (4

[1] `http://www.freebase.com`

[2] `http://www.freebase.com/m/09p14`

[3] `http://www.freebase.com/m/0mdxd`

[4] Music in general would cover a very broad set of user interests, such as music lessons and dance videos, whereas topics such as Muhammad Ali might be too specific even for sport fans.



YouRank

YouRank is a social experiment being conducted by the Computer Science Department at Universidade Federal de Minas Gerais and the Max Planck Institute for Software Systems. For more information contact: *flaviov _at_ dcc.ufmg.br.*

home > Evaluating a Video Pair. Your unique id is 13212005 (please write it down)

rank videos:
- home
- new evaluation
- continue evaluation

about:
- disclaimer
- privacy issues
- source code

external links:
- VoD Lab
- OSN group @ MPI
- UFMG

Video Pair 1 of 4: Please watch both videos and then answer the form below

Video 1

Video 2

Please provide answers to the form

If you are confused, go back to instructions. You will continue from here on returning

Which video did you like more?
○ Video 1 (left)
○ Video 2 (right)
○ I liked them both
○ I don't like either of them

Which video would you share with your friends?
○ Video 1 (left)
○ Video 2 (right)
○ Both
○ Neither

Which video do you predict will be more popular on YouTube?
○ Video 1 (left)
○ Video 2 (right)
○ I think they will be equally popular
○ I don't know

Did you already know one of these videos?
○ Video 1 (left)
○ Video 2 (right)
○ Both
○ Neither

If you want, provide extra feedback on the videos:

Send Evaluation

I was unable to watch one (or both) of the videos

Figure 4.1: Example of a Video Pair Evaluation on YouRank

pairs per fold). We deployed these video pairs on a web application we built, called YouRank. On YouRank, each user watches one fold of videos. Users are assigned to folds following a round-robin schedule. YouRank shows users only the embedded video stream, hiding any other video information kept by YouTube. Log in was based on random ids to preserve privacy. A snapshot of the application is shown in Figure 4.1.

After logging in, each user was asked to answer the demographic survey shown in Figure 4.2 (top). For questions 3 to 5, the possible answers were: 1) never; 2) rarely (few times a year); 3) occasionally (few times a month); 4) often (few times a week); and 5) very often (once or more daily).

Next, the user was asked to watch 4 pairs of videos, and, for each pair, answer the form shown in Figure 4.2 (bottom). For each question the user had to pick *one*

S1. How old are you?
S2. Are you a male or a female?
S3. How often do you watch a video on YouTube?
S4. How often do you share YouTube videos with friends or colleagues?
S5. How often do you share any kind of online content with friends or colleagues?

E1. Which video did you enjoy watching more?
E2. Which video you would be most willing to share with a friend or group of friends?
E3. Which video do you predict will be more popular on YouTube?

Figure 4.2: YouRank Forms - Demographic Survey (top); Video Evaluation Form (bottom)

out of four options: a) Video 1 (left); b) Video 2 (right); c) Both; d) Neither. Thus, two neutral options (c-d) were available in case the user could not decide on a single video. An optional task of providing feedback in free text form was also available for each pair. We asked users to refrain from visiting the video page on YouTube, and to indicate whether they had watched any of the videos in the past. To avoid bias due to user fatigue, the pairs of a fold were randomized whenever a new user was assigned to the fold.

Upon task completion, we payed 4.50 US dollars to each user. Since each fold consists of 4 pairs of videos, each one from 4 to 6 minute long, a user was expected to work for roughly 45 minutes. Thus, our payment covers MT suggested hourly rate of 6 US dollars. In practice, the users took on average 44.8 minutes to complete the task, although some user evaluations were disregarded, as discussed in the next section.

4.2.3 Evaluation Metrics

To tackle question Q1, we measured consensus for each pair of videos using the Fleiss' Kappa (κ) score of agreement [47]. This score varies from -1 to 1, while values above .4 are often interpreted as fair to good agreements, and above 0.75 as very good agreements [47]. We determine that consensus was reached if the null hypothesis of negative or no agreement ($\kappa \leq 0$) can be rejected. The same score is achieved regardless of whether the neutral responses, i.e. options c-d, are included. Thus, we compute it over all responses. When summarizing multiple tests, we apply Bonferroni correction to rule out significance due to random chance [1].

To answer Q2, we focused only on pairs of videos for which consensus was reached and computed the fraction $\hat{p}$ of those pairs for which the preferred video matches the one with larger popularity on YouTube. We then used an exact binominal sign test based on Clopper-Pearson confidence interval [47] to test whether $\hat{p}$ is above random chance (i.e., $\hat{p} > 0.5$)[5].

4.2.4 Representativeness and Reproducibility

Representativeness is an important but very challenging question that arises in any empirical study that tackles a broad question, as we do here. We tried to design an experimental methodology that is as thorough as possible, given our practical constraints. We chose one of the most popular social media sites: YouTube. We recruited only master MT workers, who are more expensive but are known to perform their tasks better. This limited us in terms of number of video pairs and number of user evaluations per pair, given our budget. Thus, we chose our videos carefully: we compared videos across 3 vastly different (10 times difference) popularity levels (with multiple videos per level). To avoid the impact of extraneous factors, we only compared videos that belonged to the same topic (repeating it for 2 different topics), selected videos of similar age, and only considered evaluations if the user had never seen the video before. To ensure that our sample sizes are not too small to draw our conclusions, we applied conservative and exact statistical tests, adequate for them, presenting results for different significance levels. Thus, we designed our experiments to yield the most accurate and representative results, within our constraints. Moreover, in favor of reproducibility, we make the YouRank source code and our gathered data publicly available[6].

Nevertheless, we acknowledge that it is impossible to generalize the findings to all social media sites without future studies. Instead we aim at providing insights that are valuable to undrstand popularity differences in one particular application – YouTube. Moreover, we hope that this work will encourage future efforts to apply our proposed methodology across different applications and over more content instances.

4.3 Results

We now discuss the results of two rounds of MT experiments, one for each selected topic (*major league baseball* and *music videos*). We ran each round until 72 users had finished their tasks. In both rounds, some users refused the task after logging in,

[5] This test is known for being suitable to small samples, as our case.

[6] `http://github.com/flaviovdf/yourank`

Table 4.1: Answers to S3, S4 and S5 in the Demographic Survey (Fig. 4.2).

	Major League Baseball			Music Videos		
	S3	S4	S5	S3	S4	S5
Never	0%	4%	1%	0%	1%	0%
Rarely	0%	18%	12.5%	0%	22%	13%
Occasionally	8%	39%	28%	21%	45%	32%
Often	48%	29%	37.5%	39.5%	28%	37%
Very Often	44%	10%	21%	39.5%	4%	18%

which caused some imbalance in the number of evaluators per fold. We also disregarded evaluations in which the users reported they (1) were unable to watch one of the videos (2 evaluations), and (2) had watched at least one of the videos before (5% and 8% of the cases for the major league baseball and music video experiments, respectively). Thus, we consider only evaluations in which users were exposed to new content so as to minimize a possible bias due to previous knowledge. After these filters, we were left with 6 to 10 evaluations per video pair (8 evaluations, on average).

We summarize the answers to the demographic survey next.

4.3.1 Demographic Survey

In both rounds, all users were from the US, as required by our task. They were roughly balanced across genders: 53% and 42% of the users (of 72 per round) were males in the baseball and music experiments, respectively. Also, in both rounds, the majority (57%) had from 20 to 45 years of age, 5% were under 20, and the others were over 45 years old.

The answers of users regarding their viewing and sharing habits (S3-S5 in Figure 4.2) are summarized in Table 4.1. The evaluations of pairs of videos, discussed below, should be interpreted in light of these answers. Note that participants of both rounds of experiments are avid YouTube viewers: they watch YouTube videos at least occasionally, and most of them do it often (39.5-48%) or very often (39.5-44%). Moreover, most users share YouTube content occasionally (39-45%), or often (28-29%), whereas only 22% of the users in both rounds share YouTube videos only rarely or never. Finally, in both rounds, users tend to have more active sharing patterns when it comes to online content in general, as expected.

We now turn to the evaluation of the pairs of videos, discussing the results in light of our two key driving questions.

Table 4.2: Fraction of Video Pairs that Rejected the Fleiss' Kappa Null Hypothesis of $\kappa \leq 0$. The columns correspond to the questions in Figure 4.2.

	Major League Baseball			Music Videos		
	E1	E2	E3	E1	E2	E3
p-value .05	25%	13%	52%	11%	2.7%	13%
.01	19%	8%	41%	8%	2.7%	11%
.001	16%	5%	36%	5%	2.7%	8%

Table 4.3: Average Values of κ for Pairs that Rejected the Null Hypothesis of $\kappa \leq 0$. The columns correspond to the questions in Figure 4.2.

	Major League Baseball			Music Videos		
	E1	E2	E3	E1	E2	E3
p-value .05	.68	.64	.74	.63	.53	.65
.01	.75	.76	.78	.63	.53	.69
.001	.79	.76	.83	.65	.62	.86

4.3.2 Can Users Reach Consensus?

Table 4.2 shows, for both rounds of experiments, the fractions of pairs in which users reached consensus, that is, pairs for which the null hypothesis of $\kappa \leq 0$ can be rejected, while Table 4.3 shows the average scores for pairs that passed the null hypothesis. Results are shown separately for each question in Figure 4.2, and for different significance levels (p-values). Smaller p-values imply more confidence on results.

In general, for any considered significance level, and for both topics, the fraction of pairs that passed the test tends to be very small (with few exceptions). The fraction is larger when users were asked which video they predicted would be more popular (E3). Thus, user agreement is easier when it comes to the collective knowledge of popularity. However, this happened in at most 52% of the pairs (p-value = 0.05). Users agreed much less often when asked which video they enjoyed the most (E1), reflecting a natural heterogeneity of user interests. The consensus was even rarer when users were asked which video they would share (E2), possibly reflecting also the user heterogeneity in terms of social activities and their perceptions of the interests of their social networks. Table 4.3 shows that, when consensus was reached, the agreements were on average good (above 0.4) or very good (above 0.75).

These findings can be illustrated by the following feedbacks on the same music video:

U1: "i didn't care for either one but the girl in the second video was stunningly beautiful so i would share that one"

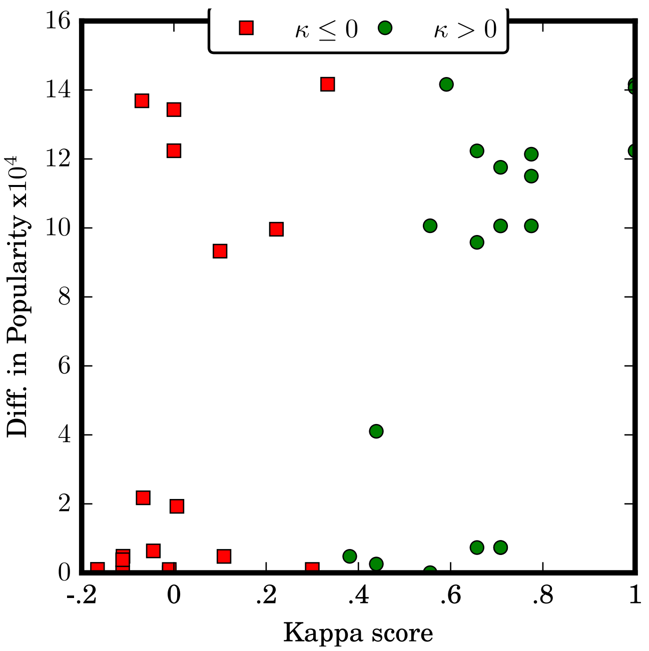


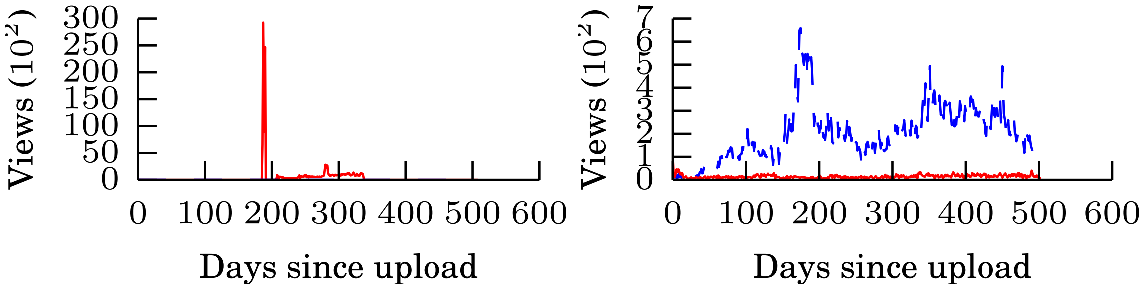
Figure 4.3: Level of Agreement κ vs Popularity Gap in Pairs of Videos

U2: “Video 2 was sad and dark and I didn’t like the girl’s voice.”
U3: “I secretly like Evanescence but I would never let my friends know.”
U4: “The video on the left was much better music for my tastes”

The divergence in opinions reveals the more egocentric notions of liking and sharing content. U2 dislikes the video because of the tone of the song, while U4 likes it because of personal taste. U1 would share the video because of the girl in it, whereas U3 would not share it because she does not want her friends to know she likes the band. Also, we notice that our current results of the diverging perceptions across users for the same video agrees with previous findings on the individuality of motivations for online participation [59, 138, 154].

Recall that users evaluated pairs of videos that covered a wide range of popularity values on YouTube. Thus, one may ask whether users could reach consensus more often for pairs of videos with a larger gap in their relative popularity. Surprisingly, we found no strong trend towards that, as illustrated in Figure 4.3 for E3 in the major league baseball experiments (p-value = .05). Very low κ values were obtained even for videos with popularity gap of hundreds of thousands views.

Table 4.2 also shows that the agreements are more common for major league baseball videos than for music videos. While this may be related to a more diverse range of personal interests for music videos (e.g., U4’s feedback), it may also relate to promotional campaigns for this kind of content. Such campaigns may cause videos to be popular for a short while, regardless of user tastes. One example is a music video



(a) One music video

(b) Two baseball videos

Figure 4.4: Popularity Curves Provided by YouTube for Example Videos.

Table 4.4: Fractions of Cases of Consensus that Match YouTube's Popularity. Values with * (**) are above random chance with p-value = 0.05 (0.01)

Major League Baseball			Music Videos		
E1	E2	E3	E1	E2	E3
100%**	100%*	84%**	75%	100%	100%*

(see Figure 4.4-a) that experienced a burst in popularity, possibly caused by promotion (professional or amateur such as in a blog) but was unable to remain popular over time.

Nevertheless, there are many cases of lack of consensus even for baseball videos. In Figure 4.4-b we show one example of a pair of videos whose popularity curves, reported by YouTube, differ considerably. One of the videos has over 100 times more views (in total) than the other, and remains more popular throughout the monitored period. Yet, the users of our experiment could not reach consensus on which video they preferred in none of the questions. Further investigating these videos, we noted that the most popular video has a watermark that affected user opinions in our experiment, as indicated by the feedbacks below. This suggests that other latent factors, other than simply content, may play a role on driving the popularity of social media.

U5: "The watermark on video 2 ruins it"
U6: "The first video had an annoying watermark on the front, and pop up tabs common on YouTube, it was very distracting", the gap towards the end was also very annoying."

4.3.3 When There Is Consensus, Does It Match the Relative Popularity of Videos on YouTube?

Focusing on the pairs for which consensus was reached (p-value = 0.05), we computed the fraction of pairs in which the video preferred by MT users matches the video with higher popularity on YouTube. The results are shown in Table 4.4. Note that, whenever consensus is reached, user preferences match YouTube's popularity in almost all cases. Note also that this result is above random chance (p-value = 0.05) in most cases. The cases where random chance cannot be ruled out are due to small number of pairs with consensus. Thus, if users can reach consensus on their opinions, the video they prefer are likely to become more popular on YouTube.

4.4 Summary

In the traditional media (e.g., newspapers, TV, radio, movie), dissemination mechanisms are closely tied to the content generators, who have a vested interest in promoting the content they generate. Content is traditionally generated or selected by professionals (e.g., journalists, singers or actors) on behalf of organizations (e.g., newspapers, movie studios or television networks) that have widely varying ability to promote their content (via advertisement campaigns to their audience). Differently, social media is dominated by content generated by ordinary users. The dissemination mechanisms are democratized and are only loosely coupled with the content generators. As discussed in Chapter 3, two important dissemination mechanisms in social media are (i) crowd-endorsements: information that is "liked" by crowds is promoted in search results and recommended to others (on home page and as personalized recommendations), and (ii) viral propagation over a social network of users: anyone who finds the information content interesting can "share" it with their friends and propagate it virally by the way of the word-of-mouth. The democratization of "dissemination mechanisms" in social media offers the hope that information popularity would be driven to a larger extent by its content (more precisely, how users perceive or like the content) than it is in traditional media. Thus, the work presented on this chapter gives the first step towards understanding the extent to which this is true.

To that end, we relied on user evaluations of pairs of YouTube videos of similar topic, factoring out the dissemination related factors. We found that users' perception of content is very subjective, since in many cases users did not reach consensus at which video they liked or would share more, or predicted would become more popular. This result indicates the difficulty in determining the role of content in driving popularity, and complements previous observations that users cannot estimate the extent of visibility of their content [9]. However, whenever there was consensus, the preferred video almost always matched the one with higher popularity on YouTube, highlighting the key role played by content in those cases.

Our results have important implications in various contexts. For social media researchers, they highlight the role of content in determining the popularity of a piece of information and the need to account for it in future studies. From a media site operator's or viral marketer's perspective, our findings have implications for popularity prediction. Our observation that when there is user consensus the video with preferred content is always more popular can be leveraged by marketers or advertisers to compare new videos against old ones with known popularities to quickly define which of the new videos have more chance of attracting viewers. It also motivates future research on how a site operator (e.g., YouTube) can design a scalable way for gathering users' feedback comparing newly uploaded videos with older ones to predict which of the new ones will more likely become popular.

Up to this point in the dissertation we have looked into how different features and user preferences correlate with content popularity in social media. As of now, our results have presented the foundation to understand our following discussions on RG2 (popularity prediction) and RG3 (mining user activities). From this point, the reader can follow up on RG2 or RG3 in any order. RG2 starts with the next chapter, and ends in Chapter 6. In these two chapters, we shall build upon our characterizations so far to develop new models to predict the popularity of social media. In Chapter 7, we shall begin our analysis on user activities, as task that also builds upon our results on how users preferences relate to popularity. Our work on mining user activities ends in Chapter 8.

Chapter 5

News Content Popularity Prediction Using Time Series Trends

In the previous chapters we analyzed the relationships between various dissemination (e.g., referrals) and content features with content popularity, aiming at uncovering fundamental knowledge about popularity dynamics. In this chapter, as well as in the next one, we shift our attention to our second research goal, and apply this knowledge in the design of popularity prediction methods.

One key aspect of the methods we propose is the use of common popularity trends extracted from the data to build more accurate prediction models. This idea is inspired by the results in [110] showing that the knowledge of such trends can improve prediction by building specialized models for each trend. In that direction we propose two sets of methods.

The first one, described in this chapter, is based on two steps. We first use time series clustering techniques, notably the KSC algorithm [146], to extract common temporal trends of content popularity. Next, we use linear regression models using as input predictors both content features (e.g., numbers of visits and mentions on online social networks) and metrics that capture the distance between the early popularity time series observed up to prediction time and the trends extracted in the first step.

The methods proposed here assume a fixed monitoring period for all objects during which the predictor variables are measured. Focusing on news content, the goal is to predict news popularity, estimated by number of visits and social network engagement, 48 hours after its upload using only information available in the first hour of upload. We note that our proposed solution was the winner of two of the three tasks of the European Conference on Machine Learning and Principles and Practice of

Knowledge Discovery in Databases 2014 Predictive Analytics Challenge [1]

In Chapter 6 we build upon the results discussed in this chapter by proposing methods to predict the popularity trend of a piece of content. We focus then on user generated content, notably YouTube videos. Unlike news, which typically have clear deadlines on when the predictions should be performed [18], UGC experiences very diverse popularity trends, with viewer ship not necessarily concentrated in sharp peaks (as discussed in Chapter 3). In that case, different objects may require different monitoring periods for accurate popularity prediction. Thus determining the monitoring period for each individual object is a key aspect of our solutions.

5.1 Introduction

With the ever-growing production of online content, characterizing and predicting user engagement (e.g., number of visits or social engagement such as Facebook likes) on content may have multiple beneficial values such as: (1) understanding the human dynamics of information consumption; (2) supporting the decisions of content producers and providers on different tasks (e.g., marketing and content filtering); and, (3) understanding the physical processes that govern the growth of viewership on the Web. Several previous studies [18, 24, 129] have characterized some of the factors that cause the popularity growth of different kinds of social media content. Complementarity, various others [4, 107, 110, 129] have focused on the task of popularity prediction. We focus here on the latter task, aiming at predicting the popularity of a piece of content.

Popularity prediction is a difficult and important task since it mostly translates into income and profits for content providers, creators and consumers alike. For example, more visitors to a news web page may lead to more ad-clicks and sales. Moreover, content provisioning to a large amount of users may require decisions such as geographical sharding of content to servers (due to the increased traffic). Thus, if planning is not performed correctly, longer latencies and loading times, and thus, fewer users may be expected. Finally, accurate and early predictions can lead to better services to the end consumer, such as search engine rankings [112].

Our model exploits the temporal features related to news web pages (e.g., past visits and social engagement), as well as typical popularity (i.e., number of visits) time series trends that exist in the dataset. Such trends are extracted via unsupervised learning methods. Specifically, it combines the temporal features with features that capture the distances between the popularity time series for each news web page and

[1] `http://sites.google.com/site/predictivechallenge2014`

the extracted trends. We present a data characterization that motivates the design of our solution, and show the gains in prediction accuracy (ranging from 15% to 27%) when it is compared to state of the art alternatives.

The rest of this chapter is organized as follows. We formally describe the prediction problem in Section 5.2. In Section 5.3 we present the datasets used in the remainder of this chapter. Moreover, Section 5.4 presents our baselines as well as our proposed solution. Our experiments and results are presented in Section 5.5. Finally, Section 5.6 concludes the chapter.

5.2 Problem Definition

Recall that our goal is to predict the popularity of news web pages (or simply news pages) that are disseminated in social media applications. In this setting, we can formalize the popularity prediction problem we tackle as follows. Let $\mathcal{H}$ be a set of web hosts (e.g., `nytimes.com`), where a single host $h \in \mathcal{H}$ is comprised of a set of pages, $\mathcal{P}$ be the set of all pages, where $p \in \mathcal{P}$ is a single page, and $\mathcal{P}_h$ be the set of all pages from host h. 44 Moreover, let $\mathcal{F}$ be a set of features associated with each page $p \in \mathcal{P}$, where each feature value is computed up to a certain *reference time* t_r (e.g., $t_r = 1$ hour). Thus, using the set of features ($\mathcal{F}$) and the set of pages ($\mathcal{P}$), a matrix $\boldsymbol{X}_{tr}$ with $|\mathcal{P}|$ rows and $|\mathcal{F}|$ columns is defined for the values of features measured up to the reference time. Moreover, a row $\boldsymbol{x}_{p,tr}$ of the matrix $\boldsymbol{X}_{tr}$ defines the measurements for the given page[2]. Using the measurements $\boldsymbol{X}_{tr}$, our goal is to predict the user engagement on each page up to a *target time* t_t (where $t_t > t_r$).

We here focus on the following metrics of user engagement, referred to as the response variables: number of visits $v_{p,tt}$, number of Facebook likes $f_{p,tt}$, and number of Twitter mentions $m_{p,tt}$. All of them are cumulative measures, computed from page's upload up until time t_t. We can then define vectors of $|\mathcal{P}|$ rows for each response variable (e.g., $\boldsymbol{v}_{tt}$), or in more general terms, we can define a matrix $\boldsymbol{Y}_{tt}$ with three columns, one for each response variable:$\boldsymbol{Y}_{tt} = [\boldsymbol{v}_{tt}, \boldsymbol{f}_{tt}, \boldsymbol{m}_{tt}]$.

With these definitions, the prediction task can be stated as a supervised machine learning task. Given a set of news pages for which both $\boldsymbol{X}_{tr}$ and $\boldsymbol{Y}_{tt}$ are available (the training set $\mathcal{P}^{train}$), our goal is to learn a function that maps $f(\boldsymbol{X}_{tr}) \to \boldsymbol{Y}_{tt}$. Ideally, such a function will generalize well for new pages not used in the training set, also known as the test set or $\mathcal{P}^{test}$. This function is usually defined as the *model*. The baseline methods, presented next, as well as our approach, introduced in Section 5.4,

[2] For simplicity, we shall identify rows using p.

Table 5.1: ECML/PKDD Challenge Dataset

# of Hosts	100
# of News Pages	30,000
Total # of Tweets	432,381
Total # of Facebook Likes	6,847,457
Total # of Visits	42,986,599

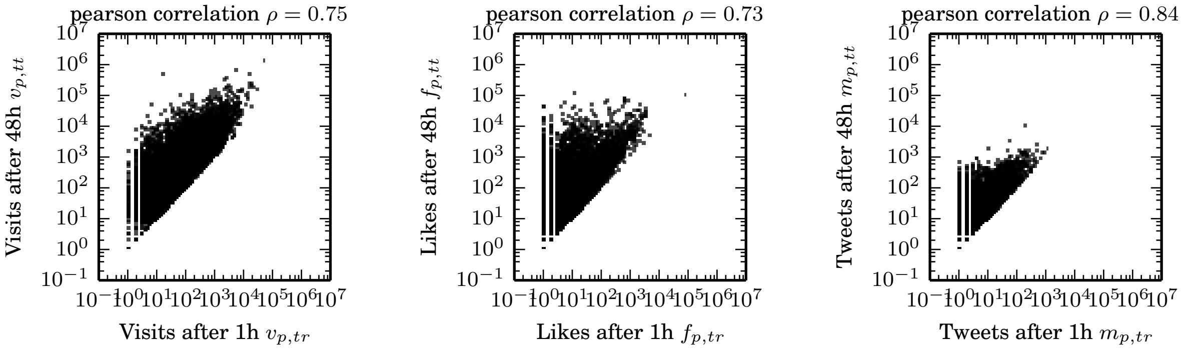
explores linear regression method to learn the model. Moreover, unless otherwise noted, we use a fixed $t_r = 1h$ as well as a $t_t = 48hs$ from now on, since these are the reference and target times defined in the Predictive Analytics Challenge.

5.3 Datasets

Before describing our method, we present a brief characterization of the datasets we use. Throughout this chapter, our case study is on predicting the popularity of news pages. Thus, we employ the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2014 Predictive Analytics Challenge dataset for this task. The details of this dataset are shown in Table 5.1, which we now discuss.

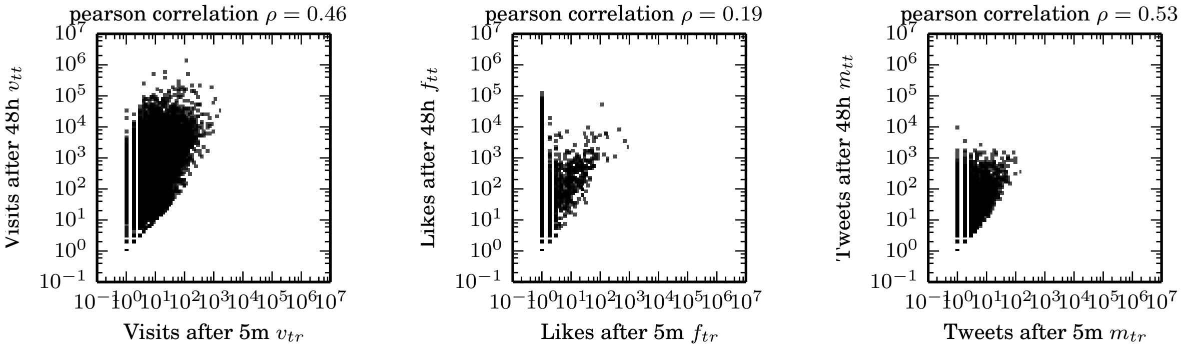
The data is composed of 30,000 news pages from 100 different hosts. Each host has exactly 300 pages. The total number of Tweets was of 432,381, whereas the number of Facebook likes was 6,847,457. A total of 42,986,599 visits was accounted for in the dataset. Before detailing our method, as well as our baselines, we now present a motivating characterization of some features in the dataset. We initially show the correlations between the user engagement metrics measured up to the reference time t_r and their respective values at the target time t_t. Figure 5.1 shows these correlations for the number of visits $v_{p,tr}$ (Figure 5.1-a), Facebook likes $f_{p,tr}$ (Figure 5.1-b) and Twitter mentions $m_{p,tr}$ (Figure 5.1-c), using $t_r = 1$ hour. Note that both axes of the graphs are in log-scales. Also, a value of 1 was added to each measure on each page (e.g., the axis for visits shows $log(1 + v_{p,tr})$).

The figure shows that a strong linear correlation in log scales (captured by the Pearson correlation coefficient ρ) exists for each engagement metric, as observed in [129]. Values of ρ exceed 0.73 for Facebook likes, reaching 0.84 for Twitter mentions. Such strong positive correlations motivate the use of linear regression methods to predict log-scaled engagement measures. By itself, these results serve as motivation that the number of visits at t_r can be used to predict future values. However, the whole



(a) Visits (b) Facebook Likes (c) Twitter Mentions

Figure 5.1: Correlations Between the Predictors Number of Visits v_{tr}, Facebook Likes f_{tr} and Twitter Mentions (Tweets) m_{tr} in one hour and their respective values after 48 hours. Each variable has been incremented by one due to log transformed x and y axes.

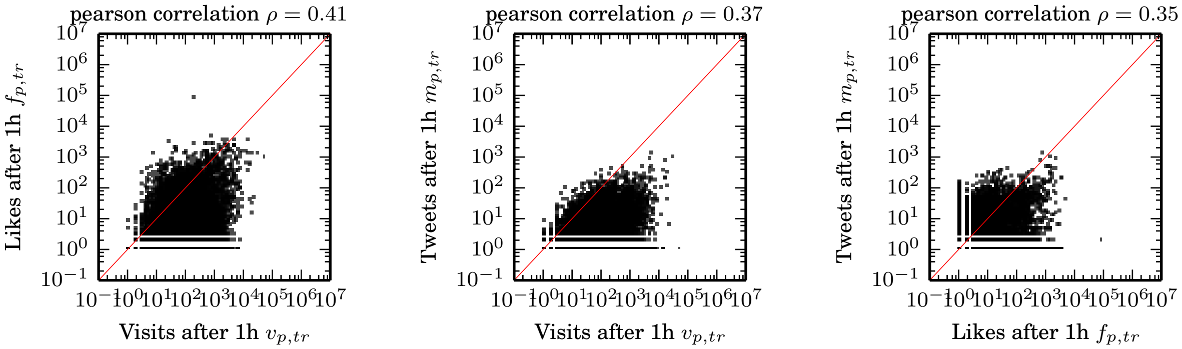


(a) Visits (b) Facebook Likes (c) Twitter Mentions

Figure 5.2: Correlations Between the Predictors Number of Visits $v_{p,tr}$, Facebook Likes $f_{p,tr}$ and Twitter Mentions (Tweets) $m_{p,tr}$ in 5 minutes and their respective values after 48h. Each variable has been incremented by one due to log transformed x and y axis.

history of measures for each metric can also be useful to predict popularity values at $t_t = 48$ hours. This is exemplified in Figure 5.2, which shows scatter plots similar to those in Figure 5.1, but now assuming that $t_r = 5$ minutes. The figure shows that in some cases (such as visits and Twitter mentions), moderate correlations (e.g. $\rho = 0.46$ for visits and $\rho = 0.53$ for Twitter mentions) already exist even very soon after the page was created.

We also looked at the correlations between engagement metrics. Figure 5.3 shows that moderate correlations exist between every pair of metrics (e.g., ρ of at least 0.35), which motivates our approach of multiplying different metrics to mitigate multi-collinearity issues. More surprisingly, we find that there exists pages that have more Facebook likes (and Twitter mentions) than actual visits (points above the 45



(a) Visits vs Facebook Likes

(b) Visits vs Twitter Mentions

(c) Facebook Likes vs Twitter Mentions

Figure 5.3: Correlations Between the Predictors Number of Visits $v_{p,tr}$, Facebook Likes $f_{p,tr}$ and Twitter Mentions (Tweets) $m_{p,tr}$. Each variable has been incremented by one due to log transformed x and y axis.

degree line in each plot). This result indicates that not every like or tweet implies in a visit, and suggests that measuring popularity on a single online social network service may be misleading, since people are not necessarily visiting the news pages. Finally, this result also suggests that we may not be able to completely rely on a single metric (e.g., Facebook likes) to predict the other (e.g., number of visits), since only moderate correlations exist between them.

We now look into the motivation for also exploiting the host, day of the week and time of the day as predictors. Figure 5.4 shows the correlations between number of visits at t_r=1 hour and at t_t=48 hours for two hosts in our dataset. We note that host 68 (shown in black) has, very similar values of $v_{p,tr}$ and $v_{p,tt}$ for most pages (i.e., most pages are on the 45 degree line). Such finding implies that most pages of this host will not grow in views. In fact, if we train the SH model for this host only, it will find that the parameter θ has a value of 1.10, that is, $log(1 + v_{p,tt}) = 1.10log(1 + v_{p,tr})$. In contrast, host 3 shows a clear increase in popularity values for almost every page. In fact, the SH model, trained specifically for host 3, will capture the relationship between $v_{p,tr}$ and $v_{p,tt}$ as being $log(1 + v_{p,tt}) = 2.04log(1 + v_{p,tr})$. This difference between hosts motivates the candidate utility of exploiting these features for prediction. As we shall discuss, we indeed make use of them in the form of indicator variables that to boost (positively or negatively) the general relationship that exists in the whole dataset (see Figure 5.1) to relationships specific to the behavior of each host. Similarly, we can correct for the behavior for different upload days and hours.

So far we have provided evidence that past popularity and future popularity of news pages are correlated. Similar results were observed in UGC content on Chapter 3.

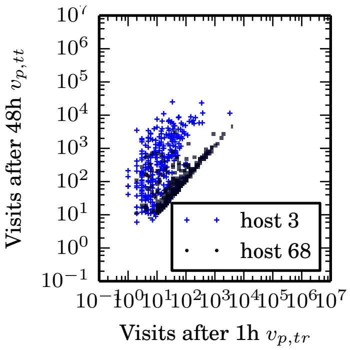


Figure 5.4: Correlation Between $v_{p,tr}$ and $v_{p,tt}$ for Selected Hosts.

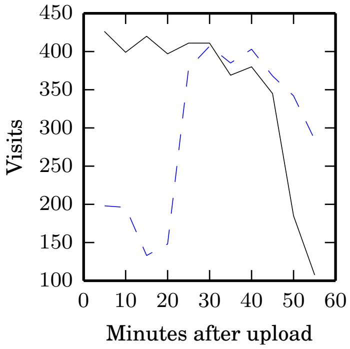


Figure 5.5: Popularity Evolution of Two Selected Pages.

Also, we have motivated that using content features, such as the host id, and temporal features, such as day of week, can also be exploited for predictions. Together with our previous findings in Chapters 3 and 4, the knowledge produced so far will serve as basis for our prediction methods detailed this chapter and the next chapter. However, before continuing, we also aim at providing evidence that the different trends followed by objects (again, see Chapter 3) can also be used to predict popularity.

In Figure 5.5 the evolution in the number of visits for two news pages, selected from our dataset, that have similar popularity in terms of total number of visits. The figure shows that the numbers of visits of the two pages evolve over time according to very different processes. The news page shown in the black/solid line is steadily decreasing in popularity over time, whereas the news page in the blue/dashed line experiences a sharp increase in popularity 25 minutes after its upload. Such an example motivates the need for the Mixed-Trend model. Indeed, in [110] the authors argued that prediction accuracy could be improved by building specialized models for each popularity trend, although no attempt to learn popularity trends and tackle such specialization was done. By incorporating the similarity of news pages to previously identified trends, as proposed here, we can effectively capture such differences in popularity curves, and thus improve prediction accuracy, as we shall now discuss.

5.4 Baseline Methods and Our Approach

Based on the characterization of the data, we now discuss how to exploit our findings for popularity prediction. We use three previously proposed prediction methods as baselines for comparison. These baselines were presented in Chapter 2. For the sake of clarity, the baselines are here re-discussed with more details in order to provide a better understanding of our results. We start this section by discussing existing state of the art solutions used as baselines in our experimental study. We then discuss our proposed model (Section 5.4.2). Finally, we discuss how cross-validation and parameter tuning is performed (Section 5.4.3).

5.4.1 Baseline Methods

One of the simplest prediction models, the Szabo-Huberman (SH) model [129], defines one single feature for each page[3], which is the number of visits measured up to the reference time t_r. Using $t_r = 1$ hour, the SH model represents a single page as $\boldsymbol{x}_{p,1h} =< v_{p,1h} >$. The SH model thus makes use of the following linear relation to provide predictions:

$$log(1 + \boldsymbol{v}_{tt}) = log(1 + \boldsymbol{X}_{tr})\theta.$$

Using linear regression, the parameter vector $\boldsymbol{\theta}$ (with only one cell in this case - θ), is solved by minimizing:

$$\min_{\boldsymbol{\theta}} ||log(1 + \boldsymbol{X}_{tr})\boldsymbol{\theta} - log(1 + \boldsymbol{v}_{tt})||_2^2,$$

where $|| \cdot ||_2^2$ is the squared $l2 - norm$. The log transform is required given the linear correlations between $log(1 + \boldsymbol{v}_{tr})$ and $log(1 + \boldsymbol{v}_{tt})$ unveiled by the authors. The goal of this objective function minimizes the sum of squared errors on the log transformed data. We shall make use of the same objective since it is the one defined in the Predictive Analytics Challenge. However, we do note that in order to provide prediction in non-log transformed values, the authors suggest changing the linear regression objective by one based on the relative error, that is:

$$\min_{\boldsymbol{\theta}} ||(\boldsymbol{X}_{tr}\boldsymbol{\theta} - \boldsymbol{v}_{tt}) \circ \boldsymbol{v}_{tt}^{-1}||_2^2.$$

[3] The model was originally proposed for YouTube videos and Digg news.

where the inverse of a vector is defined as the cell-wise inverse, while $\circ$ is the cell-wise product (e.g., $\boldsymbol{x} \circ \boldsymbol{y} =< x_1 y_1, ..., x_n y_n >$).

Pinto *et al.* [110] extended the SH model by incorporating the whole history of the number of visits to the vector $\boldsymbol{x}_{p,tr}$. Using 5-minute time windows, the vector is defined as:

$$\boldsymbol{x}_{p,tr} =< v_{p,5min}, v_{p,10min}, v_{p,15min}, \cdots, v_{p,55min}, v_{p,1h} > .$$

Defining $\boldsymbol{v}_{p,tr}$ as the vector of visits measured in fixed length time windows (e.g., 5 minutes)[4], the model above can be re-written as: $\boldsymbol{x}_{p,tr} = \boldsymbol{v}_{p,tr}$.

The same authors proposed a second model, called the MRBF model, which extends the set of features of each page by adding *distance* features. Such distance features, measured using Radial Basis Functions[5], are computed between the vector $\boldsymbol{v}_{p,tr}$ and a fixed number C of vectors for other pages, randomly selected from the training set. To avoid over-fitting, the authors suggest using ridge regression on the MRBF model. Both the ML and MRBF models were originally evaluated in terms of the relative errors, and not in terms of the log based regression as we do here.

Our last baseline is the model proposed by Castillo *et al.* [18]. In a very similar approach to the SH model, the authors also made use of a linear regression on log scales. However, instead of using one visit feature, the authors also explored social engagement features. Thus, a possible representation for a news page is:

$$\boldsymbol{x}_{p,1h} =< v_{p,1h}, f_{p,1h}, m_{p,1h} > .$$

In addition to these features, the authors also added other features, such as the entropy of tweets related to the news page. Since such features are unavailable in our dataset, we leave them out of the definition of the model. Finally, to mitigate issues of *multi-collinearity*, that is, correlation between predictors in the model, the authors suggest representing each page as:

$$\boldsymbol{x}_{p,1h} =< v^2_{p,1h}, f^2_{p,1h}, m^2_{p,1h}, v_{p,1h}f_{p,1h}, v_{p,1h}m_{p,1h}, f_{p,1h}m_{p,1h} > .$$

Since this model was initially proposed for news websites, we shall simply refer to it as the News model.

[4] The model presented by Pinto *et al.* [110] defines the amount of visits on each time window (v_i) not as cumulative (total views up to the window) as we do here, but actually as the amount of views gained in that window ($v_i - v_{i-1}$ in our notation). We found that using cumulative values lead to better results in terms of root mean squared error, thus we maintain our definition.

[5] $RBF(\boldsymbol{x}, \boldsymbol{y}) = e^{||\frac{\boldsymbol{x}-\boldsymbol{y}}{\gamma}||^2_2}$, where γ is an input parameter.

5.4.2 Our Approach

Our approach combines the ideas described in the previous section with new features not explored by previous work. Moreover, as a novelty aspect, we make use of trend features extracted via clustering of visit time series. We first describe the features we explore without considering these popularity trends (Section 5.4.2.1). Later, we discuss how we extract popularity trends and extend our model to include the distances between the popularity curve already observed of the page that is target of prediction and the previously identified trends (Section 5.4.2.2).

5.4.2.1 Mixed Model

We borrow some of the ideas of the baselines by exploring the following temporal features for each page: (1) the time series of the number of visits to a page (each observation is recorded at each 5-minute time windows) - $\boldsymbol{v}_{p,tr}$; (2) two time series of user engagement which measure the number of Facebook likes - $\boldsymbol{f}_{p,tr}$, and the number of Twitter mentions - $\boldsymbol{m}_{p,tr}$; (3) a time series of the average time each user spends on the page - $\boldsymbol{a}_{p,tr}$; (4) the weekday (e.g., Monday to Sunday) and hour (e.g., 0 to 23) the page was created - d_p and c_t. Moreover, we explore a single non-temporal feature which is the host to which each page belongs - h_p.

We encode the weekday and hour the page was created, as well as its host in a binarized manner. That is, each value is represented by a sparse vector, where one cell, representing the given weekday (hour or host) has a value of one, and all other cells are zeroes. For example, a page uploaded on a Tuesday is represented as $< 0, 1, 0, 0, 0, 0, 0 >$. Thus, we represent the weekday in which a page was uploaded as a vector $\boldsymbol{d}_p$, the hour as $\boldsymbol{c}_p$, and the host as $\boldsymbol{h}_p$. In this sense, each host, day of the week, and hour of the day become an *indicator variable.*

With these features, one possible manner of representing each page is:

$$\boldsymbol{x}_{p,1h} =< \boldsymbol{v}_{p,tr}, \boldsymbol{f}_{p,tr}, \boldsymbol{v}_{p,tr}, \boldsymbol{a}_{p,tr}, \boldsymbol{d}_{p,tr}, \boldsymbol{c}_{p,tr}, \boldsymbol{h}_{p,tr} > .$$

However, to mitigate multi-collinearity issues and to capture the behavior of hosts with non-linear popularity growth (discussed in the next section), we represent each page as:

$$\begin{aligned}\boldsymbol{x}_{p,1h} =<&\boldsymbol{v}_{p,tr}, \boldsymbol{f}_{p,tr}, \boldsymbol{v}_{p,tr}, \boldsymbol{a}_{p,tr}, \boldsymbol{v}_{p,tr} \circ \boldsymbol{v}_{p,tr}, \boldsymbol{f}_{p,tr} \circ \boldsymbol{f}_{p,tr}, \boldsymbol{m}_{p,tr} \circ \boldsymbol{m}_{p,tr}, \boldsymbol{a}_{p,tr} \circ \boldsymbol{a}_{p,tr},\\ &\boldsymbol{v}_{p,tr} \circ \boldsymbol{f}_{p,tr}, \boldsymbol{v}_{p,tr} \circ \boldsymbol{m}_{p,tr}, \boldsymbol{v}_{p,tr} \circ \boldsymbol{a}_{p,tr}, \boldsymbol{f}_{p,tr} \circ \boldsymbol{m}_{p,tr},\\ &\boldsymbol{f}_{p,tr} \circ \boldsymbol{a}_{p,tr}, \boldsymbol{m}_{p,tr} \circ \boldsymbol{a}_{p,tr}, \boldsymbol{v}_{p,tr} \circ \boldsymbol{v}_{p,tr} \circ \boldsymbol{v}_{p,tr},\\ &\boldsymbol{f}_{p,tr} \circ \boldsymbol{f}_{p,tr} \circ \boldsymbol{f}_{p,tr}, \boldsymbol{m}_{p,tr} \circ \boldsymbol{m}_{p,tr} \circ \boldsymbol{m}_{p,tr}, \boldsymbol{a}_{p,tr} \circ \boldsymbol{a}_{p,tr} \circ \boldsymbol{a}_{p,tr}, \boldsymbol{d}_{p,tr}, \boldsymbol{c}_{p,tr}, \boldsymbol{h}_{p,tr} > .\end{aligned}$$

We refer to this model as the Mixed model. The $\circ$ multiplications capture the same intuition as that of squaring the sum $(v_i + f_i + m_i)^2$ for each time window. Moreover, we also add the cubic terms (e.g., v_i^3) for the number of visits, Facebook likes, Twitter mentions and active time. To learn the model parameters we solve an linear regression task for each response variable.

5.4.2.2 Mixed-Trend Model

In order to capture the trend of each time series, we incorporate to the Mixed model features that capture the distance of the popularity curve of the target page measured during the reference time t_r to given trends, which were previously identified using an unsupervised learning method. Specifically, we experiment with K-Means clustering [62] and KSC clustering [146] to extract such trends from the training set. For each response variable, we define a matrix $\boldsymbol{T}_{tr}$, where each row is the time series of the response for a given page:

$$\boldsymbol{t}_{p,tr} =< \delta_{5min}, \delta_{10min}, \cdots, \delta_{55min}, \delta_{1h} > .$$

With the reference time of 1 hour, and a window length equal to 5 minutes, this matrix will have $|\mathcal{P}|$ rows and 12 columns. Each entry of the matrix, δ_i, represents the *number visits* gained in that time window, i.e., $\delta_i = v_i - v_{i-1}$. We note that, using this matrix to extract trends is a common approach in the literature [107, 146].

The time series trends can be considered as the most common *shapes* of the different vectors $\boldsymbol{t}_{p,tr}$. Different techniques will extract shapes in different manners from a given training set. For example, the K-Means algorithm will group time series into k clusters according to the squared Euclidean distance:

$$dist_{km}(\boldsymbol{t}, \boldsymbol{o})_{km} = ||\boldsymbol{t} - \boldsymbol{o}||_2^2.$$

As we have discussed in Chapter 2 and re-iterate here, the KSC algorithm groups times series based on a distance metric that is invariant of scale in the popularity axis and shifts in the time axis [146]. That is, two pages that have their popularities evolving according to similar processes (e.g., linear growth) will be assigned to the same cluster by KSC, regardless of the popularity values. Also, two pages that have stable popularity over time except for a peak in a single window will also be clustered together, regardless of the time when the peak occurred and the peak value. KSC is mostly a direct translation of the K-Means algorithm, except for the distance metric used, which is defined as:

$$dist_{ksc}(\boldsymbol{t}, \boldsymbol{o}) = \min_{\alpha,q} \quad \frac{||\boldsymbol{t} - \alpha \boldsymbol{o}(q)||_2}{||\boldsymbol{o}||_2}.$$

where $\boldsymbol{o}(q)$ is the operation of shifting vector $\boldsymbol{o}$ by q units. For a fixed q, the exact solution for α, obtained by computing the minimum of $dist_{ksc}$, is: $\alpha = \frac{\boldsymbol{t}'\boldsymbol{o}(q)}{||\boldsymbol{t}'||_2}$. The optimal value of q is found by considering all integers in the range of the size of the time series vectors (e.g., (-12,12)).

It is important to note that, unlike KSC, K-Means is not scale invariant. Thus, in order to make the method invariant in terms of popularity we apply the following transforms. Initially, we apply a $log(1 + \boldsymbol{T}_{tr})$ to the time series matrices. Secondly, we z-normalize (zero mean normalization) each log transformed time series vector. While this approach will keep the popularity invariance, since time series will have values in the same range, it does not tackle the time shifting invariant, as KSC does. We also note that both K-Means and KSC receive k, the target number of clusters, as input.

Given a new page p for which a prediction is to be made, we can compute the distances between its popularity time series during the reference time t_r and each previously identified trend by simply computing the distances from $\boldsymbol{t}_{p,tr}$ to each cluster center (considering a fixed time window equal to t_r), after clustering in the training set is done using either K-Means or KSC. Thus, for each clustering method we can define a vector $\boldsymbol{s}_{p,tr}$ which includes the distances to the extracted trends. The Mixed-Trend model is thus the incorporation of these distance vectors into the Mixed model.

5.4.3 Evaluation Methodology

The results discussed in this section are computed on the training set of the Predictive Analytics Challenge dataset, which consists of 30,000 news pages from 100 different hosts, each host with exactly 300 pages. We did not made use of the test set since the response variables $\mathbf{Y}_{tt}$ are not publicly available on the test set. Instead, we evaluate our models by employing Generalized Cross Validation, as described below.

For the SH, ML, News and Mixed models, model parameters were learnt by the regression method, i.e., by minimizing the sum of squared errors on the log transformed data. However, for the MRBF model, the parameter γ (used by the MRBF function), the regularization parameter of the ridge regression as well as the number C of pages selected to build Radial Basis Functions must be determined. Similarly, the number of clusters k must be given as input for the Mixed-Trend model.

Ideally, a temporal split of training and test sets would be performed to determine these parameters. However, given that the upload date of each page is not provided in the Predictive Analytics dataset, we decided to employ Generalized Cross Validation (GCV) [62] to define the best parameter values. GCV is equivalent to leave-one-out cross validation (LOOCV). In LOOCV, one page per time is used to evaluate a model which is trained on the rest of the pages. Thus, for each page, we computed the squared error between the predicted and real values. GCV computes the same squared error for each page without the need of manually splitting the dataset into train and test sets. Specifically, only one model is trained for the whole dataset, and the GCV computes the LOOCV error for every page [6]. When comparing different model parameters, we measure the root mean squared error (RMSE) between the predicted and actual value for each page. The parameters with lowest RMSE were chosen.

For the Mixed-Trend model, we searched for the best value of k (i.e., number of clusters) in the $[1, 100]$ range, finding it to be k=50 (for both K-Means an KSC algorithms) in all cases. For the MRBF model, we search for values of γ and of the ridge regularization parameter considering the following options: $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. We also searched for the best value of C out of the options: $\{10, 50, 100\}$. The best parameter values were adopted in each case. When performing clustering, we make use of the entire dataset since we found that isolating a single page using the traditional LOOCV has little to no effect on our results.

We finally note that the SH, ML and MRBF models are defined for a single engagement measure (e.g., number of visits). In order to evaluate these models for different engagement measures, we make the appropriate changes to the input features (e.g., changing from $v_{p,tr}$ to $f_{p,tr}$ or $m_{p,tr}$ in SH model).

5.5 Results

We now discuss the prediction results in terms of the root mean squared error (RMSE) when measured using generalized cross validation (GCV). The results produced by all models, when using the best parameter values as discussed previously, are shown in

[6] The following website provides a good summary of GCV `http://robjhyndman.com/researchtips/crossvalidation/`

Table 5.2: Number of Features $|\mathcal{F}|$ and Prediction Results (Root mean squared error - RMSE)

	SH	ML	MRBF	News	Mixed	Mixed-Trend KSC	Mixed-Trend K-Means	Mixed-Trend K-Means on $\boldsymbol{Y}_{tt}$
$\lvert\mathcal{F}\rvert$	1	12	22 up to 112	60	347	397	397	397
Visits	1.355	1.299	1.088	1.267	1.005	**0.991**	**0.983**	0.989
Facebook Likes	1.835	1.793	1.534	1.525	1.390	**1.383**	**1.380**	1.378
Twitter Mentions	0.863	0.852	0.779	0.786	0.669	**0.667**	**0.667**	0.666

Table 5.2. On the table we also show the number of features of each model. Moreover, in last column of the table we also show the RMSE values obtained on the challenge server, that is, when measuring RMSE based on $\boldsymbol{Y}_{tt}$ and not using GCV.

Considering only the baselines, we find that the SH model performs worse than all other methods, whereas the MRBF model is the best baseline, except for predicting Facebook likes, for which the News model is the best baseline. More importantly, our proposed Mixed and Mixed-Trend models greatly outperform all baselines, for all three response variables. Moreover, by exploiting the distances to previously identified trends, the Mixed-Trend models, using either KSC or K-Means to extract the trends from the training set, also provides improvements over the simpler Mixed model, particularly for predicting number of visits. Compared to the baselines, the improvements of the Mixed-Trend models vary from 15% (for Twitter mentions against the MRBF model) to 27% (for the number of visits against the SH model). Finally, we note only marginal differences in RMSE (if any) between extracting trends using K-Means or KSC.

Before concluding, it is important to discuss whether over-fitting is occurring in our models. We argue that this is not the case based on three results. Initially, from the last column of Table 5.2 we can see that the results for the Mixed-Trend K-Means model on the evaluation server test set is very close (and sometimes even smaller) than the one measured by GCV. Secondly, we also trained models using Ridge and Lasso regression [62], finding no improvements over the ordinary least squares linear regression we employ. Finally, we point out the result by Stone [127], which shows that minimizing cross validated errors is asymptotically equivalent to minimizing Alkaike's Information Criterion (AIC). A similar result exists for linear models when using the Bayesian Information Criterion [124] (BIC). In order to avoid over-fitting, both AIC and BIC penalize more complicated models. Thus, we also compared AIC and BIC values finding that the Mixed-Trend models always performs better than baseline approaches. These results indicate that on the Predictive Analytics Challenge dataset no over-fitting is occurring. However, it is impossible to generalize such a finding to any dataset. Thus, we point out that the use of regularized regression may be necessary on different datasets.

5.6 Summary

We have presented a novel model that exploits popularity time series (trends) and linear regression to predict user engagement on content. Three variations of the model were presented — Mixed, Mixed-Trend KSC and Mixed-Trend K-Means — together with a data characterization that motivates their design. Our results show that our best model, the Mixed-Trend K-Means, provides gains in prediction accuracy ranging from 15% to 27% when compared with state of the art approaches[7].

In the next chapter we further explore the popularity prediction task, returning our focus on user generated content. Unlike news, UGC has no clear prediction deadlines. More importantly, UGC content can exhibit more complex growth patterns. Such properties asks for a special treatment of the reference and target times. Thus, our discussion on the next chapter will address the tradeoff between how early vs how accurate can we predict the popularity of a piece of content. Our solution for this tradeoff, called TRENDLEARNER, builds upon our results on this chapter. That is, we continue to explore popularity trends in order to develop accurate prediction models.

[7] We note that all of our source code is available at: `http://github.com/flaviovdf/ecmlpkdd-analytics-challenge-2014`.

Chapter 6

Early Prediction of Popularity Trends of User Generated Content

In this chapter we present our study on predicting the popularity of user generated content (UGC). As we have motivated in Chapter 1, popularity prediction is a valuable task to content providers, advertisers, as well as social media researchers. However, specially in the UGC setting, it is also a challenging task due to the various factors factors that affect content popularity in social systems. Thus, in this chapter we focus on the problem of predicting the popularity *trend* of a piece of UGC (object) *as early as possible*. Unlike our work on Chapter 5, we explicitly address the inherent tradeoff between prediction accuracy and remaining interest in the object after prediction, since, to be useful, accurate predictions should be made *before* interest has exhausted. This tradeoff is inherent in the UGC setting, where, different from the news pages, a clear definition of reference and target times are not determined by stakeholders of the content. Moreover, given the heterogeneity in popularity dynamics across objects, this tradeoff has to be solved on a per-object basis, making the prediction task harder. We tackle this problem with a novel two-step learning approach in which we: (1) extract popularity trends from previously uploaded objects, and then (2) predict trends for newly uploaded content.

This chapter will be the last one discussing RG2, popularity prediction. Starting from the next Chapter 7, we shall shift our focus to mining user activities and how they relate to popularity. Nevertheless, the models we propose to model and mine user activities can also be used for popularity prediction if need be. However, they do not dead with the tradeoff between prediction accuracy and remaining interest in the object as is done in this chapter.

6.1 Introduction

The success of Internet applications based on user generated content (UGC)[1] has motivated questions such as: How does content popularity evolve over time? What is the potential popularity a piece of content will achieve after a given time period? How can we predict popularity evolution of a particular piece of UGC? For example, from a system perspective, accurate popularity predictions can be exploited to build more cost-effective content organization and delivery platforms (e.g., caching systems, CDNs). They can also drive the design of better analytic tools, a major segment nowadays [83, 153], while online advertisers may benefit from them to more effectively place contextual advertisements. From a social perspective, understanding issues related to popularity prediction can be used to better understand the human dynamics of consumption. Moreover, being able to predict popularity on an automated way is crucial for marketing campaigns (e.g. created by activists or politicians), which increasingly often use the Web to influence public opinion.

Challenges: However, predicting the popularity of a piece of content, an *object*, in a social system is a very challenging task. This is mostly due to the various phenomena affecting the popularity prediction of social media – which were observed on the datasets we use (as well as others) [91, 152] – as well as the diminishing interesting in objects over time, which implies that popularity predictions must be timely to capture user interest and be useful in real work settings. Both challenges can be summarized as follows:

1. Due to the easiness with which UGC can be created, many factors can affect an object's popularity. Such factors include, for instance, the object's content, the social context in which it is inserted (e.g., social neighborhood or influence zone of the object's creator), the mechanisms used to access the content (e.g., searching, recommendation, top-lists), or even an external factor, such as a hyperlink to the content in a popular blog or website. These factors can cause spikes in the surge of interest in objects, as well as information propagation cascades which affect the popularity trends of objects.
2. To be useful in a real scenario, a popularity prediction approach must identify popularity trends *before the user interest in the object has severely diminished.* To illustrate this point, Figure 6.1 shows the popularity evolution of two YouTube videos: the video on the left receives more than 80% (shaded region) of all views received during its lifespan in the first 300 days since upload, whereas the other video receives only about half of its total views in the same time frame. If we were

[1] YouTube, Flickr, Twitter, and so forth

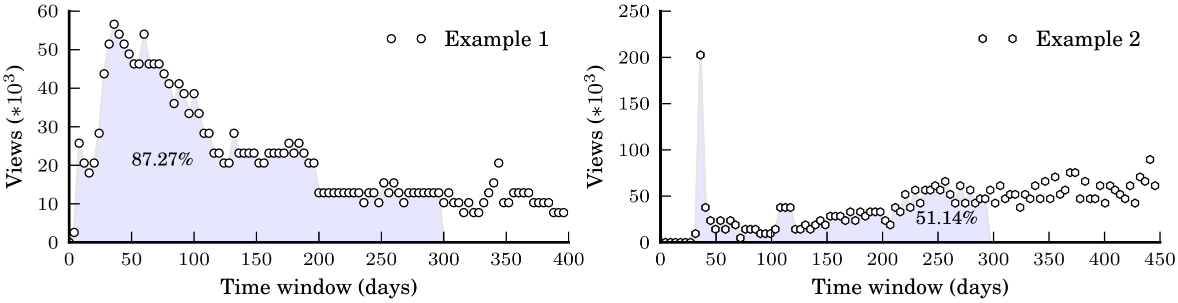


Figure 6.1: Popularity Evolution of Two YouTube Videos.

to monitor each video for 300 days, most potential views of the first video would be lost. In other words, not all objects require the same monitoring period, as assumed by previous work, to produce accurate predictions: for some objects, the prediction can be made earlier. Thus, the tradeoff should be solved on a *per-object* basis, which implies that determining the duration of the monitoring period that leads to a good solution of the tradeoff for *each object* is part of the problem.

These challenges set UGC objects apart from more traditional web content. For instance, news media [18] tends to have clear definitions of monitoring periods, say predicting the popularity of news after one day using information from the first hour after upload. This is mostly due to the timely nature of the content, which is reflected in the popularity trends usually followed by news media (see Chapter 3) – interest is usually concentrated in a peak window (e.g., day) and dies out rather quickly. Thus, mindful of the challenges above, we here tackle the problem of UGC popularity *trend* prediction. That is, we focus on the (hard) task of predicting popularity *trends*. Trend prediction can help determining, for example, if an object will follow a viral pattern (e.g., Internet memes) or will continue to gain attention over time (e.g., music videos for popular artists). Moreover, we shall also show that, by knowing popularity trends *beforehand*, we can improve the accuracy of models for predicting popularity measures (e.g., hits). Thus, by focusing on predicting trends, we fill a gap in current research since no previous efforts has effectively predicted the popularity *trend* of UGC taking into account challenges (1) and (2).

We should stress that one key aspect distinguishes our work from previous efforts to predict popularity [4, 18, 81, 110, 129, 150] – we explicitly address the inherent tradeoff between prediction accuracy and how early the prediction is made, assessed in terms of the remaining interest in the content after prediction. All previous popularity prediction efforts considered fixed monitoring periods for all objects, which is given as input. We refer to this problem as *early prediction*.

In terms of applications, knowing that an object will be popular early on can help advertisers to plan out specific revenue models [55]. Such knowledge can also help out on geographic content sharding [39] for better content delivery. On the other hand, being aware that an object will not be popular at all, as early as possible, allow low access content to be tiered down to lower latency servers/geographic regions, whereas advertisers can use this knowledge to avoid bidding for ads in such content. Finally, early prediction is of utmost importance to content producers – knowing whether a piece of content will be follow a certain trend can help in their promotion strategies and in the creation of new content.

TrendLearner: We tackle this problem with a novel two-step combined learning approach. First, we identified popularity trends, expressed by popularity timeseries, from previously uploaded objects. Then, we combine novel time series classification algorithms with object features for predicting the trends of new objects. This approach is motivated by the intuition that it might be easier to identify the popularity trend of an object if one has a set of possible trends as basis for comparison. More important, we propose a new trend classification approach, namely TrendLearner, that tackles the aforementioned tradeoff between prediction accuracy and remaining interest after prediction on a per-object basis. The idea here is to monitor newly uploaded content on an online basis to determine, for each monitored object, the earliest point in time when prediction confidence is deemed to be good enough (defined by input parameters), producing, as output, the probabilities of each object belonging to each class (trend). Moreover, unlike previous work, TrendLearner also combines the results from this classifier (i.e., the probabilities) with a set of object related features [44], such as category and incoming links, building an ensemble learner.

In sum, our main contributions include:

1. The proposal of TrendLearner, a new effective and efficient popularity trend classification method that considers multiple classes, represented by cluster centroids, and combines class probabilities with features commonly associated with UGC objects to build a more effective trend predictor;
2. The definition of novel metrics related to prediction accuracy and how early and biased such predictions are. TrendLearner optimizes both metrics, achieving better results than the baselines;
3. The use of TrendLearner to improve the prediction of popularity metrics (e.g., number of views), with improvements over the baselines of around 33%, at least.

The rest of this chapter is organized as follows. Next section discusses related work. We state our target problem in Section 6.2, and present our approach to solve it in

Section 6.3. We introduce the metrics used to evaluate our approach in Section 6.4. Our main experimental results are discussed in Section 6.5. Section 6.6 offers conclusions and directions for future work.

6.2 Problem Statement

The early popularity trend prediction problem can be defined as follows. Given a training set of previously monitored user generated objects (e.g., YouTube videos or tweets), $\mathcal{D}^{train}$, and a test set of newly uploaded objects $\mathcal{D}^{test}$, do: (1) extract popularity trends from $\mathcal{D}^{train}$; and (2) predict a trend for each object in $\mathcal{D}^{test}$ as early and accurately as possible, particularly before user interest in such content has significantly decayed. User interest can be expressed as the fraction of all potential views a new content will receive until a given point in time (e.g., the day when the object was collected). Thus, by predicting as early as possible the popularity trend of an object, we aim at maximizing the fraction of views that still remain to be received *after prediction*.

Note that there is a tradeoff between prediction accuracy and the remaining fraction of views: it is expected that the longer we monitor an object, the more accurately we can predict its popularity trend; but often this would imply a reduction of the remaining interest in the content. Determining the earliest point in time when prediction can be made with reasonable accuracy is an inherent challenge of the early popularity prediction problem, given that it must be addressed on a per-object basis. In particular, we here treat it as a multi-class classification task, where the popularity trends automatically extracted from $\mathcal{D}^{train}$ (step 1) represent the classes into which objects in $\mathcal{D}^{test}$ should be grouped.

Table 6.1 summarizes the notation used throughout the chapter. Each object $d \in \mathcal{D}^{train}$ is represented by an n-dimensional time series vector $\mathbf{s}_d =< p_{d,1}, p_{d,2}, \cdots, p_{d,n} >$, where $p_{d,i}$ is the popularity (i.e., number of views) acquired by d *during* the i^{th} time window after its upload. Intuitively, the duration of a time window w could be a few hours, days, weeks, or even months. Thus, vector $\mathbf{s}_d$ represents a time series of the popularity of a piece of content measured at time intervals of duration w (fixed for each vector). New objects in $\mathcal{D}^{test}$ are represented by streams, $\hat{\mathbf{s}}_d$, of potentially infinite length ($\hat{\mathbf{s}}_d =< p_{d,1}, p_{d,2}, \cdots$). This captures the fact that our trend prediction/classification method is based on monitoring each test object on an online basis, also determining when a prediction with acceptable confidence can be made (see Section 6.3.2). Note that a vector can be seen as a contiguous subsequence of a stream. Note also that the complete dataset is referred to as $\mathcal{D} = \mathcal{D}^{train} \bigcup \mathcal{D}^{test}$.

Table 6.1: Notation. Vectors ($\mathbf{x}$) and matrices ($\mathbf{X}$), in bold, are differentiated by lower and upper cases. Streams ($\hat{\mathbf{x}}$) are differentiated by the hat accent ($\hat{}$). Sets ($\mathcal{D}$) are shown in fancy letters and variables (d) are shown in regular lower case letters, respectively.

Symbol	Meaning	Example
$\mathcal{D}$	dataset of UGC content	YouTube videos
$\mathcal{D}^{train}$	training set	-
$\mathcal{D}^{test}$	testing set	-
d	a piece of content or object	video
$\mathcal{D}_i$	cluster/class/trend i	-
$\mathbf{c}_{\mathcal{D}_i}$	centroid of cluster/class i	-
$\mathbf{s}_d$	time series vector for object d	$\mathbf{s}_d =< p_{d,1}, \cdots, p_{d,n} >$
$\hat{\mathbf{s}}_d$	time series stream for object d	$\hat{\mathbf{s}}_d =< p_{d,1}, , \cdots$
$p_{d,i}$	popularity of d at i-th window	number of views
$\mathbf{s}_d(i)$	index operator	$< 7, 8, 9 > (2) = 8$
$\mathbf{s}_d(i : j)$	slicing operator	$< 7, 8, 9 > (2 : 3) =< 8, 9 >$
$\mathbf{S}$	matrix with set of time series	all time series

6.3 Our Approach

We here present our solution to the early popularity trend prediction problem. We introduce our trend extraction approach (Section 6.3.1), present our novel trend classification method, TrendLearner (Section 6.3.2), and discuss practical issues related to the joint use of both techniques (Section 6.3.3).

6.3.1 Trend Extraction

To extract temporal patterns of popularity evolution (or trends) from objects in D^{train}, we employed the time series clustering algorithm called K-Spectral Clustering (KSC) as in Chapter 3. We note that the authors of the KSC algorithm [146] are focused mainly on the time series *clustering task*, aiming at studying temporal patterns of online content, and not on *predicting* popularity trends based on information collected during a monitoring period. Thus, the joint use of the KSC algorithm with TrendLearner (Section 6.3.2) to predict as early and accurately as possible popularity trends is a novel contribution of this work. Recall from Chapter 2 that KSC is mostly a direct translation of K-Means and that each cluster's centroid defines the trend that objects in the cluster (mostly) follow. Also, each cluster defines a class in our task of predicting trends for new objects (Section 6.3.2). Thus, we refer to the discovered trends (clusters) as *classes*.

Before introducing our trend classification method, we make the following observation that is key to support the design of the proposed approach: each trend, as defined by a centroid, is conceptually equivalent to the notion of *time series shapelets* [148]. A shapelet is informally defined as a time series subsequence that is in a sense maximally representative of a class. As argued in [148], the distance to the shapelet can be used to classify objects with more accuracy and much faster than state-of-the-art classifiers. Thus, by showing that a centroid is a shapelet, we choose to classify a new object based only on the distances between the object's popularity time series up to a monitored time and each cluster's centroid.

This is one of the points where our approach differs from the method proposed in [27], which uses the complete $\mathcal{D}^{train}$ as reference series, classifying an object based on the distances between its time series and *all* elements of each cluster. Given $|\mathcal{D}^{train}|$ objects in the training set and k clusters (with $k << |\mathcal{D}^{train}|$), our approach is faster by a factor of $\frac{|\mathcal{D}^{train}|}{k}$.

Definition: *For a given class $\mathcal{D}_i$, a shapelet $\mathbf{c}_{\mathcal{D}_i}$ is a time series subsequence such that:* (1) $dist(\mathbf{c}_{\mathcal{D}_i}, \mathbf{s}_d) \leq \beta, \forall \mathbf{s}_d \in \mathcal{D}_i$; and (2) $dist(\mathbf{c}_{\mathcal{D}_i}, \mathbf{s}_{d'}) > \beta, \forall \mathbf{s}_{d'} \notin \mathcal{D}_i$, where β is defined as an optimal distance for a given class. With this definition, a shapelet can be shown to maximize the information gain of a given class [148], being thus the most representative time series of that class.

We argue that, by construction, a centroid produced by KSC is a shapelet with β being the distance from the centroid to the time series within the cluster that is furthest away from its centroid. Otherwise, the time series that is furthest away would belong to a different cluster, which contradicts the KSC algorithm. This is an intuitive observation. Note that a centroid is a shapelet *only* when using K-Means based approaches, such as KSC, to define class labels. In the case of learning from already labeled data a shapelet finding algorithms [148] should be employed.

6.3.2 Trend Prediction

Let $\mathcal{D}_i$ represent class i, previously learned from $\mathcal{D}^{train}$. Our task now is to create a classifier that correctly determines the class of a new object as early as possible. We do so by monitoring the popularity acquired by each object d ($d \in \mathcal{D}^{test}$) since its upload on successive time windows. As soon as we can state that d belongs to a class with *acceptable confidence*, we stop monitoring it and report the prediction. The heart of this approach is in detecting *when* such statement can be made.

6.3.2.1 Probability of an Object Belonging to a Class

Given a monitoring period defined by t_r time windows, our trend prediction is fundamentally based on the distances between the subsequence of the stream $\hat{\mathbf{s}}_d$ representing d's popularity curve from its upload until t_r, $\hat{\mathbf{s}}_d(1 : t_r)$, and the centroid of each class. To respect shifting invariants, we consider all possible starting windows t_s in each centroid time series when computing distances. That is, given a centroid $\mathbf{c}_{\mathcal{D}_i}$, we consider all values from 1 to $|\mathbf{c}_{\mathcal{D}_i}| - t_r$, where $|\mathbf{c}_{\mathcal{D}_i}|$ is the number of time windows in $\mathbf{c}_{\mathcal{D}_i}$. Specifically, the probability that a new object d belongs to class $\mathcal{D}_i$, given $\mathcal{D}_i$'s centroid, the monitoring period t_r and a starting window t_s, is:

$$p(\hat{\mathbf{s}}_d \in \mathcal{D}_i \mid \mathbf{c}_{\mathcal{D}_i}; t_r, t_s) \propto exp(-dist(\hat{\mathbf{s}}_d(1 : t_r), \mathbf{c}_{\mathcal{D}_i}(t_s : t_s + t_r - 1))) \qquad (6.1)$$

where $(x{:}y)$ $(x \leq y)$ is a moving window slicing operator (see Table 6.1). As in [27, 29, 110], we assume that probabilities are inversely proportional to the exponential function of the distance between both series, given by function $dist$ (The KSC distance function, see Chapter 2), normalizing them afterwards to fall in the 0 to 1 range (here omitted for simplicity). Figure 6.2 shows an illustrative example of how both time series would be aligned for probability computation[2].

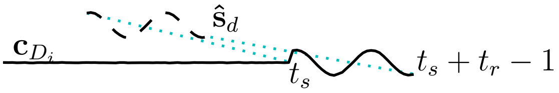


Figure 6.2: Example of Alignment of Time Series (dashed lines) for Probability Computation.

With Equation 6.1, we could build a classifier that simply picks the class with highest probability. But this would require t_s and t_r to be fixed. As shown in Figure 6.1, different time series may need different monitoring periods (different values of t_s and t_r), depending on the required confidence.

Instead, our approach is to monitor an object for successive time windows (increasing t_r), computing the probability of it belonging to each class at the end of each window. We stop when the class with maximum probability exceeds a class-specific threshold, representing the required minimum confidence on predictions for that class. We detail our approach next, focusing first on a single class (Algorithm 1), and then generalizing it to multiple classes (Algorithm 2).

[2] In case $|\mathbf{c}_{\mathcal{D}_i}| < |\hat{\mathbf{s}}_d(1 : t_r)|$, we try all possible alignments of $\mathbf{c}_{\mathcal{D}_i}$ with $\hat{\mathbf{s}}_d(1 : t_r)$.

Algorithm 1 Define when to stop computing probability of object $\hat{\mathbf{s}}_d$ belonging to class $\mathcal{D}_i$, based on minimum confidence θ_i, and minimum and maximum monitoring periods γ_i and γ^{max}.

```
 1: function PerClassProb($\hat{\mathbf{s}}_d$, $\mathbf{c}_{\mathcal{D}_i}$, $\theta_i$, $\gamma_i$, $\gamma^{max}$)
 2:     $p \leftarrow 0$
 3:     $t_r \leftarrow \gamma_i - 1$                                  ▷ Start at previous window
 4:     while $p < \theta_i$ do                                  ▷ Extend monitoring period
 5:         $t_r \leftarrow t_r + 1$                              ▷ Move to next current window
 6:         if $t_r > \gamma^{max}$ then                          ▷ Monitoring period ended
 7:             return $0, \gamma^{max}$
 8:         end if
 9:         $p \leftarrow AlignComputeProb(\hat{\mathbf{s}}_d, \mathbf{c}_{\mathcal{D}_i}, \theta_i, t_r)$
10:     end while
11:     return $t_r, p$                          ▷ Return monitoring period and probability
12: end function
13: function AlignComputeProb($\hat{\mathbf{s}}_d$, $\mathbf{c}_{\mathcal{D}_i}$, $\theta_i$,$t_r$)
14:     $t_s \leftarrow 1$; $p \leftarrow 0$
15:     while $(t_s \leq |\mathbf{c}_{\mathcal{D}_i}| - t_r)$ and $(p < \theta_i)$ do
                               ▷ Iterate over possible values of $t_s$, aligning both series
16:         $p' \propto exp(-dist(\hat{\mathbf{s}}_d(1:t_r), \mathbf{c}_{\mathcal{D}_i}(t_s : t_s + t_r - 1)))$
17:         $p \leftarrow max(p, p')$
18:         $t_s \leftarrow t_s + 1$
19:     end while
20:     return $p$
21: end function
```

Algorithm 1 shows how we define when to stop computing the probability for a given class $\mathcal{D}_i$. The algorithm takes as input the object stream $\hat{\mathbf{s}}_d$, the class centroid $\mathbf{c}_{\mathcal{D}_i}$, the minimum confidence θ_i required to state that a new object belongs to $\mathcal{D}_i$, as well as γ_i and γ^{max}, the minimum and maximum thresholds for the monitoring period. The former is used to avoid computing distances with too few windows, which may lead to very high (but unrealistic) probabilities. The latter is used to guarantee that the algorithm ends. We allow different values of γ_i and θ_i for each class as different popularity trends have overall different dynamics, requiring different thresholds[3]. The algorithm outputs the number of monitored windows t_r and the estimated probability p. The loop in line 4 updates the stream with new observations (increases t_r), and function *AlignComputeProb* computes the probability for a given t_r by trying all possible alignments (i.e., all possible values of t_s). For a fixed alignment (i.e., fixed t_r and t_s), *AlignComputeProb* computes the distance between both time series (line 15) and the probability of $\hat{\mathbf{s}}_d$ belonging to $\mathcal{D}_i$ (line 16). It returns the largest probability

[3] Indeed, initial experiments showed that using the same values of γ_i (and θ_i) for all classes produces worse results.

Algorithm 2 Define when to stop computing probabilities for each object in $\mathcal{D}^{test}$, considering the centroids of all classes ($\mathbf{C}_D$), per-class minimum confidence ($\boldsymbol{\theta}$) and monitoring period ($\boldsymbol{\gamma}$), and maximum monitoring period (γ^{max}).

```
 1: function MultiClassProbs( $\mathcal{D}^{test}$, $\mathbf{C}_D$, $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$,$\gamma^{max}$)
 2:     $\mathbf{t} = [0]$                                   ▷ Per-object monitoring period vector
 3:     $\mathbf{P} = [[0]]$                                 ▷ Per-object, per-class probability matrix
 4:     $n_{objs} \leftarrow |\mathcal{D}^{test}|$           ▷ Number of objects to be monitored
 5:     $t_r \leftarrow min(\boldsymbol{\gamma})$            ▷ Init $t_r$ with minimum $\gamma_i$
 6:     while $(t_r \leq \gamma^{max})$ and $(n_{objs} > 0)$ do
 7:         for all $\hat{\mathbf{s}}_d \in \mathcal{D}^{test}$ do          ▷ Predict class for each object
 8:             for all $\mathbf{c}_{\mathcal{D}_i} \in \mathbf{C}_D$ do    ▷ Get centroid of each class
 9:                 $p(i) \leftarrow AlignComputeProb(\hat{\mathbf{s}}_d, \mathbf{c}_{\mathcal{D}_i}, \theta_i, t_r)$
10:             end for
11:             $maxp \leftarrow max(\mathbf{p})$        ▷ Get max. probability and corresponding class for $t_r$
12:             $maxc \leftarrow argmax(\mathbf{p})$
13:             if $(maxp > \boldsymbol{\theta}(maxc))$ and $(t_r \geq \boldsymbol{\gamma}(maxc))$ then      ▷ Stop if maxp and $t_r$ exceeds per-class thresholds
14:                 $\mathbf{t}(d) \leftarrow t_r$                 ▷ Save current $t_r$
15:                 $\mathbf{P}(d) \leftarrow \mathbf{p}$          ▷ Save current $\mathbf{p}$ in row $d$
16:                 $n_{objs} \leftarrow n_{objs} - 1$
17:                 $\mathcal{D}^{test} \leftarrow \mathcal{D}^{test} - \{\hat{\mathbf{s}}_d\}$
18:             end if
19:         end for
20:         $t_r \leftarrow t_r + 1$
21:     end while
22:     return $\mathbf{t}, \mathbf{P}$                  ▷ Return monitoring periods and probabilities
23: end function
```

representing the best alignment between $\hat{\mathbf{s}}_d$ and $\mathbf{c}_{\mathcal{D}_i}$, for the given t_r (lines 17 and 20). Both loops that iterate over t_r (line 4) and t_s (line 15) stop when the probability exceeds the minimum confidence θ_i. The algorithm also stops when the monitoring period t_r exceeds γ^{max} (line 7), returning a probability equal to 0 to indicate that it was not possible to state the $\hat{\mathbf{s}}_d$ belongs to $\mathcal{D}_i$ within the maximum monitoring period allowed (γ^{max}).

We now extend Algorithm 1 to compute probabilities and monitoring periods for all object streams in $\mathcal{D}^{test}$, considering all classes extracted from $\mathcal{D}^{train}$. Algorithm 2 takes as input the test set $\mathcal{D}^{test}$, a matrix $\mathbf{C}_D$ with the class centroids, vectors $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ with per-class parameters, and γ^{max}. It outputs a vector $\mathbf{t}$ with the required monitoring period for each object, and a matrix $\mathbf{P}$ with the probability estimates for each object (row) and class (column), both initialized with 0 in all elements. Given a valid monitoring period t_r (line 6), the algorithm monitors each object d in $\mathcal{D}^{test}$ (line 7) by first computing the probability of d belonging to each class (line 9). It then takes, for each object d, the largest of the computed probabilities (line 11) and the associated class (line 12), and tests whether it is possible to state that d belongs to that class with enough confidence at t_r, i.e., whether: (1) the probability exceeds the minimum confidence for the class, *and* (2) t_r exceeds the per-class minimum threshold (line 13). If the test succeeds, the algorithm stops monitoring the object (line 16), saving the current t_r and the per-class probabilities computed at this window in $\mathbf{t}$ and $\mathbf{P}$ (lines 14-15). After exhausting all possible monitoring periods ($t_r > \gamma^{max}$) or whenever the number of objects being monitored n_{objs} reaches 0, the algorithm returns. At this point, entries with 0 in $\mathbf{P}$ indicate objects for which no prediction was possible within the maximum monitoring period allowed (γ^{max}).

Having $\mathbf{P}$, a simple classifier can be built by choosing for each object (row) the class (column) with maximum probability. The value in $\mathbf{t}$ determines how early this classification can be done. However, we here employ a different strategy, using matrix $\mathbf{P}$ as input features to another classifier, as discussed below. We compare our proposed approach against the aforementioned simpler strategy in Section 6.5.

6.3.2.2 Probabilities as Input Features to a Classifier

Instead of directly extracting classes from $\mathbf{P}$, we choose to use this matrix as input features to another classification algorithm, motivated by previous results on the effectiveness of using distances as features to learning methods [29]. Specifically, we employ an extremely randomized trees classifier [51], as it has been shown to be effective on different datasets [51], requiring little or no pre-processing, besides producing models that can be more easily interpreted, compared to other techniques like Support Vector Machines[4]. Extremely randomized trees tackle the over fitting problem of more common decision tree algorithms by training a large ensemble of trees. They work as follows: 1) for each node in a tree, the algorithm selects the best features for splitting based on a random subset of all features; 2) split values are chosen at random. The decision of these trees are then averaged out to perform the final classification. Although feature search and split values are based on randomization, tree nodes are still chosen based on the maximization of some measure of discriminative power such as Information Gain, with the goal of improving classification effectiveness.

We extend the set of probability features taken from $\mathbf{P}$ with other features associated with the objects. The set of object features used depends on the type of UGC under study and characteristics of the datasets ($\mathcal{D}$). We here use the features shown

[4] We also used SVM learners, achieving similar results.

in Table 3.3, combining them with the probabilities in $\mathbf{P}$. We refer to this approach as TrendLearner.

We note that there are alternative strategies to combine a learner based on Algorithm 2 and one based on the object features. We tried Co-Training [106], which combines learners based on different input features. However, it failed to achieve better results than just combining the features, most likely because it depends on feature independence, which may not hold in our case. We also experimented with Stacking [40], which yielded similar results as the proposed approach. Nevertheless, either strategy might be more effective on different datasets or types of UGC, an analysis that we leave for future work.

6.3.3 Putting It All Together

A key point that remains to be discussed is how to define the input parameters of the trend extraction approach, that is, the number of clusters k, as well as the parameters of TrendLearner, namely vectors $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$, γ^{max}, and the parameters of the adopted classifier.

We choose the number of clusters k based primarily on the β_{CV} as was done in Chapter 3. Since our case study is on the same dataset, the choice was of $k = 4$. Regarding the TrendLearner parameters, we here choose to constrain γ^{max} with the maximum number of points in our time series (100 in our case, as discussed in Chapter 3). As for vector parameters $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$, a traditional cross-validation in the training set to determine their optimal values would imply in a search over an exponential space of values. Moreover, note that it is fairly simple to achieve best classification results by setting $\boldsymbol{\theta}$ to all zeros and $\boldsymbol{\gamma}$ to large values, but this would lead to very late predictions (and possibly low remaining interest in the content after prediction). Instead, we suggest an alternative approach. Considering each class i separately, we run a one-against-all classification for objects of i in $\mathcal{D}^{train}$ for values of γ_i varying from 1 till γ^{max}. We select the smallest value of γ_i for which the performance exceeds a minimum target (e.g., classification above random choice, meaning Micro-F1 greater than 0.5), and set θ_i to the average probability computed for all class i objects for the selected γ_i. We repeat the same process for all classes. Depending on the required tradeoff between prediction accuracy and remaining fraction of views, different performance targets could be used. Finally, we use cross-validation in the training set to choose the parameter values for the extremely randomized trees classifier, as further discussed in Section 6.5.

We summarize our solution to the early trend prediction problem in Algorithm 3.

Algorithm 3 Our Solution: Trend Extraction and Prediction

```
function TrendExtraction($\mathcal{D}^{train}$)
    $k \leftarrow 1$
    while $\beta_{CV}$ is not stable do
        $k \leftarrow k + 1$
        $\mathbf{C}_D \leftarrow KSC(\mathcal{D}^{train}, k)$
    end while
    Store centroids in $\mathbf{C}_D$
end function
function TrendLearner($\mathbf{C}_D$, $\mathcal{D}^{train}$, $\mathcal{D}^{test}$)
    $\boldsymbol{\theta}, \gamma, \mathbf{P}^{train} \leftarrow LearnParams(\mathcal{D}^{train}, \mathbf{C}_D)$
    $TrainERTree(\mathcal{D}^{train}, \mathbf{P}^{train} \bigcup$ obj. feats$)$
    $\mathbf{t}, \mathbf{P} \leftarrow MultiClassProbs(\mathcal{D}^{test}, \mathbf{C}_D, \boldsymbol{\theta}, \gamma)$
    return $\mathbf{t}, PredictERTree(\mathcal{D}^{test}, \mathbf{P} \bigcup$ obj. feats$)$
end function
```

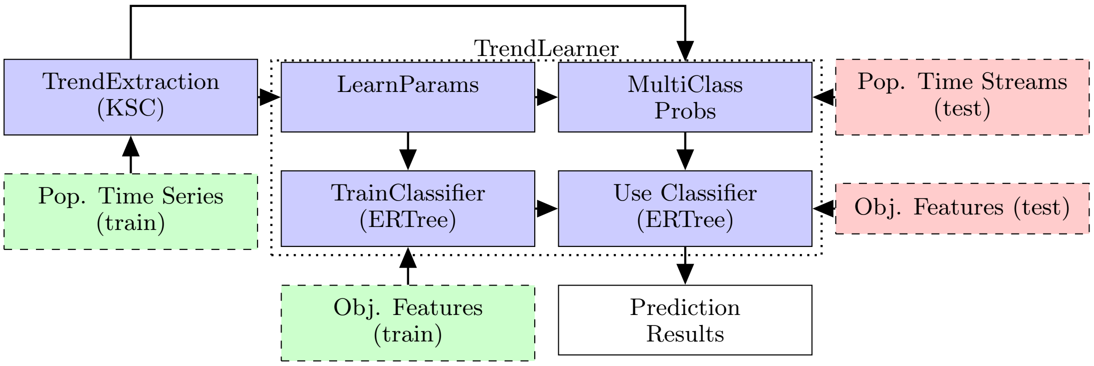


Figure 6.3: Pictorial Representation of Our Solution

In particular, TrendLearner works by first learning the best parameter values and the classification model from the training set ($LearnParams$ and $TrainERTrees$), and then applying the learned model to classify test objects ($PredictERTrees$), taking the class membership probabilities ($MultiClassProb$) and other object features as inputs. A pictorial representation is shown in Figure 6.3. Compared to previous efforts [27], our method incorporates multiple classes, uses only cluster centroids to compute class membership probabilities (which reduces time complexity), and combines these probabilities with other object features as inputs to a classifier, which, as shown in Section 6.5, leads to better results.

6.4 Evaluation Methodology

As discussed in Section 6.2, an inherent challenge of the early popularity trend prediction problem is to properly address the tradeoff between prediction accuracy and how early the prediction is made. Thus, we evaluate our method with respect to these two aspects.

We estimate prediction accuracy using the traditional Micro and Macro $F1$ metrics, which are computed from precision and recall. The precision of class c, $P(c)$, is the fraction of correctly classified videos out of those assigned to c by the classifier, whereas the recall of class c, $R(c)$, is the fraction of correctly classified objects out of those that actually belong to that class. The $F1$ of class c is given by: $F1(c) = \frac{2 \cdot P(c) \cdot R(c)}{P(c)+R(c)}$. Macro F1 is the average $F1$ across all classes, whereas Micro F1 is computed from global precision and recall, calculated for all classes.

We evaluate how early our correct predictions are made computing the remaining interest (RI) in the content *after* prediction. The RI for an object $\mathbf{s}_d$ is defined as the fraction of all views up to a certain point in time (e.g., the day when the object was collected) that are received *after* the prediction. That is, $RI(\mathbf{s}_d, \mathbf{t}) = \frac{sum(\mathbf{s}_d(\mathbf{t}(d)+1:n))}{sum(\mathbf{s}_d(1:n))}$ where n is the number of points in d's time series, $t(d)$ is the prediction time (i.e., monitoring period) produced by our method for d, and function sum adds up the elements of the input vector. In essence, this metric captures the future potential audience of $\mathbf{s}_d$ after prediction.

We also assess whether there is any bias in our *correct* predictions towards more (less) popular objects by computing the correlation between the total popularity and the remaining interest after prediction for each object. A low correlation implies no bias, while a strong positive (negative) correlation implies a bias towards earlier predictions for more (less) popular objects. We argue that, if any bias exists, a bias towards more popular objects is preferred, as it implies larger remaining interests for those objects. We use both the Pearson linear correlation coefficient (ρ_p) and the Spearman's rank correlation coefficient (ρ_s) [69], as the latter does not assume linear relationships, taking the logarithm of the total popularity first due to the great skew in their distribution [22, 34].

6.5 Experimental Results

In this section, we present representative results of our trend prediction approach. We also show the applicability of our approach to improve the accuracy of state-of-the-art popularity prediction models (Section 6.5.3). Given our focus on user generate content, we evaluate our TrendLearner approach with the same datasets as the ones used in Chapter 3, that is, time series of YouTube videos. Also, we use the same four clusters extracted in that chapter (which were also extracted using the KSC algorithm) as the time series trends. See Section 3.4 for details. Our results were computed using 5-fold cross validation, i.e., splitting the dataset D into 5 folds, where 4 are used as training set $\mathcal{D}^{train}$ and one as test set $\mathcal{D}^{test}$, and rotating the folds such that each fold is used for testing once. As discussed in Section 6.3, trends are extracted from $\mathcal{D}^{train}$ and predicted for videos in $\mathcal{D}^{test}$.

Since we are dealing with time series, one might argue that a temporal split of the dataset into folds would be preferred to a random split, as we do here. However, we choose a random split because of the following. Regarding the object features used as input to the prediction models, no temporal precedence is violated, as the features are computed only during the monitoring period t_r, *before* prediction. All remaining features are based on the distances between the popularity curve of the object until t_r and the cluster centroids. As we argue below, the same clusters and centroids found in our experiments were consistently found in various subsets of each dataset, covering various periods of time. Thus, we expect the results to remain similar if a temporal split is done. However, a temporal split of our dataset would require interpolations in the time series, as all of them have exactly 100 points regardless of video age. Such interpolations, which are not required in a random split, could introduce serious inaccuracies and compromise our analyses.

We now discuss our trend prediction results, which are averages of 5 test sets along with corresponding 95% confidence intervals. We here refer to the clusters as *classes*. We start by showing results that support our approach of computing class membership probabilities using only centroids as opposed to all class members, as in [27] (Section 6.5.1). We then evaluate our TRENDLEARNER method, comparing it with three alternative approaches (Section 6.5.2).

6.5.1 Are shapelets better than a reference dataset?

We here discuss how the use of centroids to compute class membership probabilities (Equation 6.1) compare to using all class members [27]. For the latter, the probability of an object belonging to a class is proportional to a summation over the exponential of the (negative) distance between the object and every member of the given class.

An important benefit of our approach is a reduction in running time: for a given object, it requires computing the distances to only k time series, as opposed to the complete training set $|\mathcal{D}^{train}|$, leading to a reduction in running time by a factor of $\frac{|\mathcal{D}^{train}|}{k}$, as discussed in Section 6.3.1. We here focus on the classification effectiveness of the probability matrix $\mathbf{P}$ produced by both approaches. To that end, we consider a classifier that assigns the class with largest probability to each object, for both matrices.

Table 6.2: Classification Using Centroids Only vs. Using All Class Members: Averages and 95% Confidence Intervals.

Monitoring	Centroid		Whole Training Set	
period t_r	Micro F1	Macro F1	Micro F1	Macro F1
1 window	$.24 \pm .01$	$.09 \pm .00$	$.29 \pm .04$	$.11 \pm .01$
25 windows	$.56 \pm .02$	$.52 \pm .01$	$.53 \pm .04$	$.44 \pm .08$
50 windows	$.67 \pm .03$	$.65 \pm .03$	$.64 \pm .05$	$.57 \pm .09$
75 windows	$.70 \pm .02$	$.68 \pm .02$	$.69 \pm .08$	$.61 \pm .12$

Table 6.2 shows Micro and Macro F1 results for both approaches, computed for fixed monitoring periods t_r (in number of windows) to facilitate comparison. We show results only for the Top dataset, as they are similar for the Random dataset. Note that, unless the monitoring period is very short (t_r=1), both strategies produce statistically tied results, with 95% confidence. Thus, given the reduced time complexity, using centroids only is more cost-effective. When using a single window *both* approaches are worse than random guessing (Macro F1 = 0.25), and thus are not interesting.

6.5.2 TrendLearner Results

We now compare our **TrendLearner** method with three other trend prediction methods, namely: (1) **P only**: assigns the class with largest probability in **P** to an object; (2) **P + ERTree**: trains an extremely randomized trees learner using **P** only as features; (3) **ERTree**: trains an extremely randomized trees learner using only the object features in Table 3.3. Note that TrendLearner combines ERTree and **P** + ERTree. Thus, a comparison of these four methods allows us to assess the benefits of combining both sets of features.

For *all* methods, when classifying a video d, we only consider features of that video available up until $\mathbf{t})(d)$, the time window when TrendLearner stopped monitoring d. We also use the same best values for parameters shared by the methods, chosen as discussed in Section 6.3.3. Both Tables 6.3 (for the Top dataset) and 6.4 (for the Random dataset), show the best values of vector parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$, selected considering a Macro-F1 of at least 0.5 as performance target (see Section 6.3.3). These results are averages across all training sets, along with 95% confidence intervals. The variability is low in most cases, particular for $\boldsymbol{\theta}$. Recall that γ^{max} is set to 100. Regarding the extremely randomized trees classifier, we set the size of the ensemble to 20 trees, and the feature selection strength equal to the square root of the total number of features, common choices for this classifier [51]. We then apply cross-validation within

Table 6.3: Best Values for Vector Parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ (Averages and 95% Confidence Intervals) for the Top Dataset

	Top Dataset			
	D_0	D_1	D_2	D_3
$\boldsymbol{\theta}$	$.250 \pm .015$	$.257 \pm .001$	$.272 \pm .003$	$.303 \pm .006$
$\boldsymbol{\gamma}$	28 ± 16	89 ± 8	5 ± 0.9	3 ± 0.5

Table 6.4: Best Values for Vector Parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$ (Averages and 95% Confidence Intervals) for the Random Dataset

	Random Dataset			
	D_0	D_1	D_2	D_3
$\boldsymbol{\theta}$	$.250 \pm .001$	$.251 \pm .001$	$.269 \pm 0.001$	$.317 \pm 0.001$
$\boldsymbol{\gamma}$	33 ± 0.6	74 ± 2	45 ± 9	17 ± 3

the training set to choose the smoothing length parameter (n_{min}), considering values equal to $\{1, 2, 4, 8, 16, 32\}$. We refer to [51] for more details on the parametrization of extremely randomized trees.

Still analyzing Tables 6.3 and 6.4, we note that classes with smaller peaks (D_0 and D_1) need longer minimum monitoring periods γ_i, likely because even small fluctuations may be confused as peaks due to the scale invariance of the KSC distance metric used [5]. However, after this period, it is somewhat easier to determine whether the object belongs to one of those classes (smaller values of θ_i). In contrast, classes with higher peaks (D_2 and D_3) usually require shorter monitoring periods, particularly in the Top dataset, where videos have popularity peaks with larger fractions of views (Table 3.6). Indeed, by cross-checking results in Tables 3.6, 6.3 and 6.4, we find that classes with smaller fractions of videos in the peak window (D_0 and D_1 in Top, and D_0, D_1 and D_2 in Random) tend to require longer minimum monitoring periods so as to avoid confusing small fluctuations with peaks from the other classes.

We now discuss our classification results, focusing first on the Micro and Macro F1 results, shown in Table 6.6 and Table 6.5, for the Top and Random datasets respectively. From both tables we can see that TrendLearner consistently outperforms all other methods in both datasets and on both metrics, except for Macro F1 in the Random dataset, where it is statistically tied with the second best approach (**P** only). In contrast, there is no clear winner among the other three methods across both datasets.

[5] Indeed, most of these videos are wrongly classified into either D_2 or D_3 for shorter monitoring periods.

Table 6.5: Comparison of Trend Prediction Methods (Averages and 95% Confidence Intervals) for the Top Dataset

	Top Dataset			
	P only	**P**+ERTree	ERTree	TrendLearner
Micro F1	.48 ± .06	.48 ± .06	.58 ± .01	.62 ± .01
Macro F1	.44 ± .06	.44 ± .06	.57 ± .01	.61 ± .01

Thus, combining probabilities and object features brings clear benefits over using either set of features separately. For example, in the Top dataset, the gains over the alternatives in average Macro F1 vary from 7% to 38%, whereas the average improvements in Micro F1 vary from 7% to 29%. Similarly, in the Random dataset, gains in average Micro and Macro F1 reach up to 14% and 11%, respectively. Note that TrendLearner performs somewhat better in the Random dataset, mostly because videos in that dataset are monitored for longer, on average (larger values of γ_i). However, this superior results comes with a reduction in remaining interest after prediction, as we discuss below.

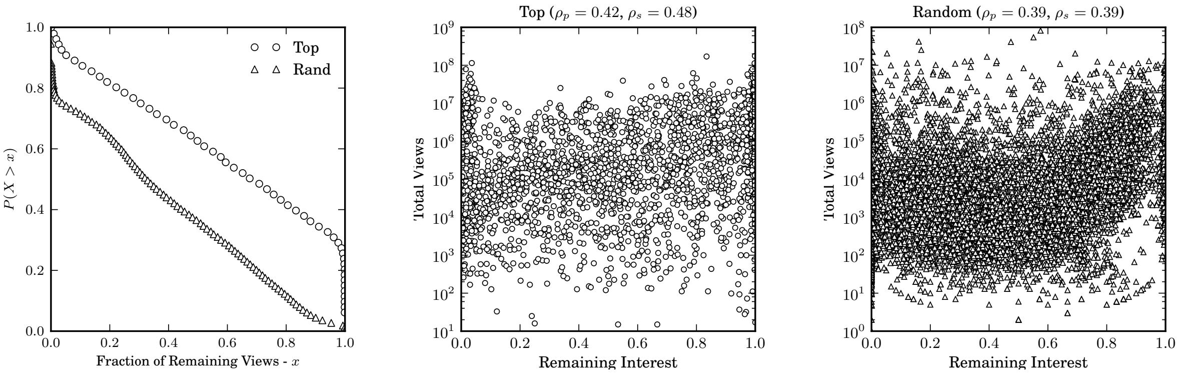
We note that the joint use of both probabilities and object features renders TrendLearner more robustness to some (hard-to-predict) videos. Recall that, as discussed in Section 6.3.2.1, Algorithm 2 may, in some cases, return a probability equal to 0 to indicate that a prediction was not possible within the maximum monitoring period allowed. Indeed, this happened for 1% and 10% of the videos in the Top and Random datasets, respectively, which have popularity curves that do not closely follow any of the extracted trends. The results for the **P** only and **P** + ERTree methods shown in Tables 6.6 and 6.5 do *not* include such videos, as these methods are not able to do predictions for them (since they rely only on the probabilities). However, both ERTree and TrendLearner are able to perform predictions for such videos by exploiting the object features, since at least the video category and upload date are readily available as soon as the video is posted. Thus, the results of these two methods in Tables 6.6 and 6.5 contemplate the predictions for *all* videos[6].

We now turn to the other side of the tradeoff and discuss how early the predictions are made. These results are the same for all four aforementioned methods as all of them use the prediction time returned by TrendLearner. For all *correctly* classified videos, we report the remaining interest RI after prediction, as well as the Pearson (ρ_p) and

[6] For the cases with probability equal to 0, the predictions of TrendLearner and ERTree were made with t_r=γ^{max}, when Algorithm 2 stops. Since we set γ^{max}=100, those predictions were made at the last time window, using all available information to compute object features. Nevertheless, note that, in those cases, the remaining interest (RI) after prediction is equal to 0.

Table 6.6: Comparison of Trend Prediction Methods (averages and 95% confidence intervals) for the Random Dataset

	Random Dataset			
	P only	**P**+ERTree	ERTree	TrendLearner
Micro F1	$.67 \pm .02$	$.62 \pm .01$	$.65 \pm .01$	$.71 \pm .01$
Macro F1	$.69 \pm .02$	$.63 \pm .01$	$.63 \pm .01$	$.70 \pm .01$



(a) Remaining Interest (RI)

(b) Total Views vs. RI (Top)

(c) Total View vs. RI (Random)

Figure 6.4: Remaining Interest (RI) and Correlations Between Popularity and RI for Correctly Classified Videos.

Spearman (ρ_s) correlation coefficients between remaining interest and (logarithm of) total popularity (i.e., total number of views), as informed in our datasets.

Figure 6.4(a) shows the complementary cumulative distribution of the fraction of *RI* after prediction for both datasets, while Figures 6.4(b) and 6.4(c) (log scale on the y-axis) show the total number of views and the *RI* for each video in the Top and Random datasets, respectively. All three graphs were produced for the union of the videos in all test sets. Note that, for 50% of the videos, our predictions are made before at least 68% and 32% of the views are received, for Top and Random videos, respectively. The same *RI* of at least 68% of views is achieved for 21% of videos in the Random dataset. In general, for a significant number of videos in both datasets, our correct predictions are made before a large fraction of their views are received, particularly in the Top dataset.

We also point out a great variability in the duration of the monitoring periods produced by our solution: while only a few windows are required for some videos, others have to be monitored for a longer period. Indeed, the coefficients of variation of these monitoring periods are 0.54 and 1.57 for the Random and Top datasets, respectively.

This result emphasizes the need for choosing a monitoring period on a per-object basis, a novel aspect of our approach, and not use the same fixed value.

Moreover, the scatter plots in Figures 6.4(b-c) show that some moderately positive correlations exist between the total number of views and RI. Indeed, ρ_p and ρ_s are equal to 0.42 and 0.48, respectively, in the Top dataset, while both metrics are equal to 0.39 in the Random dataset. Such results imply that our solution is somewhat biased towards more popular objects, although the bias is not very strong. In other words, for more popular videos, TrendLearner is able to produce accurate predictions by potentially observing a smaller fraction of their total views, in comparison with less popular videos. This is a nice property, given that such predictions can drive advertisement placement and content replication/organization decisions which are concerned mainly with the most popular objects.

6.5.3 Applicability to Regression Models

Motivated by results in [110,145], which showed that knowing popularity trends *beforehand* can improve the accuracy of regression-based popularity prediction models, we here assess whether our trend predictions are good enough for that purpose. To that end, we use the state-of-the-art ML and MRBF regression models proposed in [110]. The former is a multivariate linear regression model that uses the popularity acquired by the object d on each time window up to a reference date t_r (i.e., $p_{d,i}$, $i = 1...t_r$) to predict its popularity at a target date $t_t = t_r + \delta$. The latter extends the former by including features based on Radial Basis Functions (RBFs) to measure the similarity between d and specific examples, previously selected from the training set.

Our goal is to evaluate whether our trend prediction results can improve these models. Thus, as in [110], we use the mean Relative Squared Error (mRSE) to assess the prediction accuracy of the ML and MRBF models in two settings: (1) a general model, trained using the whole dataset (as in [110]); (2) a specialized model, trained for each *predicted* class. For the latter, we first use our solution to predict the trend of a video. We then train ML and MRBF models considering as reference date each value of $\mathbf{t}(d)$ produced by TrendLearner for each video d. Considering a prediction lag δ equal to 1, 7, and 15, we measure the mRSE of the predictions for target date $t_t = \mathbf{t}(d) + \delta$.

We also compare our specialized models against the state-space models (SSMs) proposed in [112]. These models are variations of a basic state-space model that represent query and click frequency in Web search, capturing various aspects of popularity dynamics (e.g., periodicity, bursty behavior, increasing trend). All of them take as in-

Table 6.7: Mean Relative Squared Error Various Prediction Models and Lags δ (averages and 95% confidence intervals)

Prediction Model	Top Dataset			Random Dataset		
	$\delta = 1$	$\delta = 7$	$\delta = 15$	$\delta = 1$	$\delta = 7$	$\delta = 15$
generalML	$.09 \pm .005$	$.42 \pm .02$	$.75 \pm .04$	$.01 \pm .001$	$.06 \pm .005$	$.11 \pm .01$
generalMRBF	$.08 \pm .005$	$.52 \pm .05$	$1.29 \pm .17$	$.01 \pm .001$	$.1 \pm .01$	$.26 \pm .03$
best SSM	$.76 \pm .01$	$.63 \pm .02$	$.64 \pm .03$	$.90 \pm .002$	$.69 \pm .005$	$.54 \pm .006$
specializedML	$.08 \pm .005$	$.27 \pm .01$	$.38 \pm .02$	$.009 \pm .001$	$.04 \pm .0003$	$.06 \pm .003$
specializedMRBF	$.08 \pm .005$	$.32 \pm .04$	$.47 \pm .08$	$.009 \pm .001$	$.04 \pm .0004$	$.06 \pm .008$

put the popularity time series during the monitoring period t_r. Thus, though originally proposed for the Web search domain, they can be directly applied to our context. Both regression and state-space models are parametrized as originally proposed[7].

Table 6.7 shows average mRSE for each model along with 95% confidence intervals, for all datasets and prediction lags. Comparing our specialized models and the original ones they build upon, we find that using our solution to build trend-specific models greatly improves prediction accuracy, particularly for larger values of δ. The reductions in mRSE vary from 10% to 77% (39%, on average) in the Random dataset, and from 11% to 64% (33%, on average) in the Top dataset[8]. The specialized models also greatly outperform the state-space models: the reductions in mRSE over the best state-space model are at least 89% and 27% in the Random and Top datasets (94% and 59%, on average). These results offer strong indications of the usefulness of our trend predictions for predicting popularity measures.

Finally, it is important to discuss why the state-space models did not work well in our context. The main reason we found was that Holt-Winters based models can only capture the linear trends in time series, that is, linear growth and decay. By using the KSC distance function, we can identify and group UGC time series with non-linear trends [91, 146], and create specific prediction models for these cases. Also, these models are trained independently for each target object, using early points of the time series. Another possible reason for the low performance in our context might be that, unlike in [112] where the models were trained with hundreds of points of each time series, we here use much less data (only points up to $\mathbf{t}^{[d]}$).

[7] The only exception is the number of examples used to compute similarities in the MRBF model: we used 50 examples, as opposed to the suggested 100 [110], as it led to better results in our datasets.

[8] The only exception is the MRBF model for δ=1 in the Top dataset, where general and specialized models produce tied results.

6.6 Summary

In this chapter, we have identified and formalized a new research problem. To the extent of our knowledge, we are the first work to tackle the problem of *early prediction* of popularity trends in UGC. We were motivated in studying this problem based on our previous knowledge on the complex patterns and causes of popularity in UGC [44]. Different from other kinds of content, e.g., news, which have clear definitions of monitoring periods, target and prediction dates for popularity, the complex nature of UGC calls for a popularity prediction solution which is able to determine these dates automatically. We here provided such a solution – TrendLearner.

We have also proposed a novel two-step learning approach for early prediction of popularity trends of UGC. Moreover, we defined new metrics for measuring the effectiveness of popularity of UGC content, the remaining interest, which is optimized by TrendLearner as to provide not only accurate, but also timely, predictions. Thus, unlike previous work, we addresses the tradeoff between prediction accuracy and remaining interest in the content after prediction on a per-object basis.

We performed an extensive experimental evaluation of our method, comparing it with state-of-the-art, representative solutions of the literature. Our experimental results on two YouTube datasets showed that our method not only outperforms other approaches for trend prediction (a gain of up to 38%) but also achieves such results before 50% or 21% of videos (depending on the dataset) accumulate more than 32% of their views, with a slight bias towards earlier predictions for more popular videos. Moreover, when applied jointly with recently proposed regression based models to predict the popularity of a video at a future date, our method outperforms state-of-the-art regression and state-space based models, with gains in accuracy of at least 33% and 59%, on average, respectively.

With this chapter we conclude our studies on predicting the popularity of social media content. Our approaches are novel in the sense that they predict both the trend and the number of hits a piece of content will receive, a joint-task not exploited by any previous work. The next chapter begins our work on mining user activities (RG3). Understanding user activities is a complementary task to the work done on popularity prediction (RG2). Finally, it is important to point out that the data mining techniques proposed for mining user activities (RG3) can also be used for popularity prediction. However, our discussion on those techniques are more focused on understanding on how user activities relate with popularity.

Chapter 7

Revisit Behavior in Social Media

How many listens will an artist receive on a online radio? How many of these visits are new or returning users? In this chapter, we begin our study on modeling and mining content popularity from the perception of user activities. Specifically on this chapter, we investigate the effect of revisits (successive visits from a single user) on content popularity. Using four datasets of user activity, with up to tens of millions of media objects (e.g., YouTube videos, Twitter hashtags or LastFM artists), we show the effect of revisits in the popularity evolution of such objects. Secondly, we propose the Phoenix-R model which captures the popularity dynamics of individual objects. Phoenix-R has the desired properties of being: (1) parsimonious, being based on the minimum description length principle, and achieving lower root mean squared error than state-of-the-art baselines; (2) applicable, the model is effective for predicting future popularity values of objects.

Recall that the work presented in Chapters 3, 5, and 6 mostly made use of time series and object features. Moreover, our work on Chapter 4 focused on *user perceptions* of content. As we have shown, different features (e.g., referrer features), as well as the user perceptions of content, are related to the evolution of popularity of social media objects in varying degrees. However, those previous chapter did not look into user activities as is done in this chapter and the next one. In details, the rest of this dissertation will focus on presenting novel large scale data mining techniques that are used to unveil how user activities relate to popularity evolution of objects. These studies complement and extend our previous chapters as we shall now discuss.

7.1 Introduction

How do we quantify the popularity of a piece of content in social media applications?

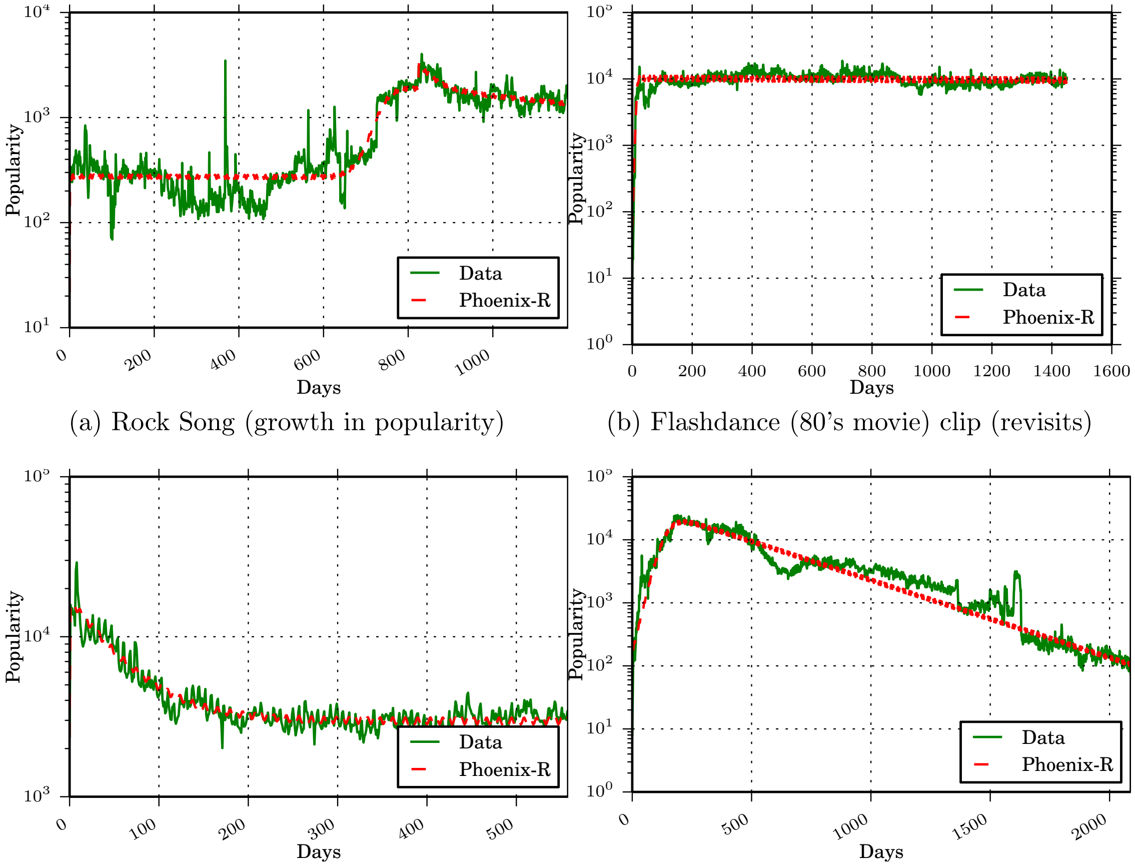
Should we consider only the audience (unique visitors) or include revisits as well? Can the revisit activity be explored to create more realistic popularity evolution models? These are important questions in the study of social media popularity. In this chapter, we take the first step towards answering them based on four large traces of user activity collected from different social media applications: Twitter, LastFM, and YouTube.

However, a key aspect that has not been explored by most previous work is the effect of revisits on content. The distinction between audience (unique users), revisits (returning users), and popularity (the sum of the previous two) can have large implications for different stakeholders of these applications - from content providers to content producers - as well as for internal and external services that rely on user activity data. For example, marketing services should care most about the audience of a particular content, as opposed to its total popularity, as each access does not necessarily represent a new exposed individual. Even system level services, such as geographical sharding [39, 131], can be affected by such distinction, as a smaller audience served by one data center does not necessarily imply that a smaller volume of activity (and thus lower load) should be expected. As prior studies of content popularity in social media do not clearly distinguish between unique and returning visits, the literature still lacks fundamental knowledge about content popularity dynamics in this environment.

We here aim at investigating and modeling the effect of revisits on popularity, thus complementing prior efforts on the field of social media popularity. Our goals are: (1) Characterizing the revisits phenomenon and show how it affects the evolution of popularity of different objects (videos, artists or hashtags) on social media applications; (2) Introducing the PHOENIX-R model that captures the evolution of popularity of individual objects, while explicitly accounting for revisits. Also, we develop the model so that it can capture multiple cascades, or outbreaks, of interest in a given object.

Our results shows that that when analyzing total popularity values, revisits account from 40% to 96% of the popularity of an object (on median), depending on the application. Moreover, when looking at small time windows (e.g., hourly) revisits can be up to 14x more common than new users accessing the object. Based on these, and other findings, we derive the PHOENIX-R model. PHOENIX-R explicitly addresses revisits in social media behavior and is able to automatically identify multiple cascades [65] using only popularity time series data. The PHOENIX-R model is also scalable. Fitting is done in linear time and no parameters are required.

Figure 7.1 shows the different behaviors which can be captured by the Phoenix-R model. Notice how the model captures a growth in the popularity of video (a), videos which have a plateau like popularity after the upload (b), and two different single cascade dynamics (c-d). Previous models, such as the SpikeM [91] and TemporalDy-



(a) Rock Song (growth in popularity)

(b) Flashdance (80's movie) clip (revisits)

(c) Korean Music Video (single cascade)

(d) User Dancing Video (single cascade)

Figure 7.1: Different YouTube Videos as Captured by the Phoenix-R Model.

namics [112] are unable to capture behaviors such as the ones shown in the figure. The SpikeM [91] approach models single cascades only, whereas the TemporalDynamics [112] models, are linear in nature.

The rest of this chapter is organized as follows. Section 7.2 presents an overview of definitions and background. This is followed by Section 7.3 which presents our characterization. Phoenix-R is described in Section 7.4. The validation of Phoenix-R and it's applicability are presented in Section 7.5. Finally, we conclude the chapter in Section 7.6.

7.2 Definitions and Background

In this section we present the definitions used throughout the chapter (Section 7.2.1). Next, we discuss existing models of popularity dynamics of individual objects (Section 7.2.2). Some of these models were already discussed in more details on Chapter 2. We revisit their discussion in this chapter for a clearer understanding of our Phoenix-R model.

7.2.1 Definitions

Recall that we defined an **object** as a piece of media content stored on a social media application. Specifically for this chapter, an object on YouTube is a video, whereas, on an online radio like LastFM, we consider (the webpage of) an artist as an object. We also define an object on Twitter as a hashtag or a *musictag*[1]. A **user activity** is the act of accessing - posting, re-posting, viewing or listening to - an object on a social media application. The **popularity** of an object is the aggregate behavior of user activities on that object. We here study popularity in terms of the most general activities in each application: number of views for YouTube videos, number of plays for LastFM artists, and number of tweets with a hashtag. The popularity of an object is the sum of **audience** (user's first visit) and, **revisits** (returning users). The evolution of the popularity of an object over time defines a **time series**.

7.2.2 Existing Models of Object Popularity Dynamics

The use of epidemic models [64] have been successfully used by previous work to understand how information disseminates in social media applications. The simplest of these models is the Susceptible-Infected (SI) model, which we have discussed in Section 2.3. Starting with an initial population of S susceptible individuals and I infected individuals, the evolution of the model is governed by two differential equations:

$$\frac{dS}{dT} = -\beta SI \tag{7.1}$$

$$\frac{dI}{dT} = \beta SI. \tag{7.2}$$

At ach time step, $\beta S(t-1)I(t-1)$ individuals get infected, transitioning from state S to state I. The product SI accounts for all the possible connections between individuals. The parameter β is the strength of the infectivity, or virus.

$\frac{dS}{dT} = -\beta SI$ and $\frac{dI}{dT} = \beta SI$. The evolution of the model is governed by the dynamics of at each time step, $\beta S(t-1)I(t-1)$ individuals get infected, transitioning from state S to state I. The product SI accounts for all the possible connections between individuals. The parameter β is the strength of the infectivity, or virus.

[1] Users informing their followers which artists they are listening to.

Table 7.1: Comparison of Phoenix-R With Other Approaches

	Revisits	Non-Linear	Forecasting	Multi Cascade
SI [64]		✓		
SpikeM [91]		✓	✓	
TemporalDynamics [112]			✓	
Phoenix-R	✓	✓	✓	✓

Cha *et al.* [23] modelled information propagation on the Flickr social media application using an SI model. Moreover, Matsubara *et al.* [91] extended this model to account for a power law infectivity per newly infected individual. This new model is called SpikeM. One of the reasons why the SI model is useful to represent online cascades of information propagation is that individuals usually do not delete their posts, tweets or favorite markings [23,91]. Thus, once an individual is infected he/she remains infected forever (as captured by the SI model). However, when considering popularity in general, the "forever infected" assumption may be false [5, 118].

Other models that can be explored in the study of content popularity dynamics are auto-regressive models and state space models, such as the Holt-Winters model and its extensions [112]. However, these models are linear in nature, and thus cannot account for more complex temporal dynamics observed in online content [91]. Although, these models have been successful in predicting *normalized* query behavior in search engines [112], the descriptive power of such models is less attractive. For example, Holt-Winters based models are very general, that is, they are used to predict time series behavior, but will not take into account cascades, revisits or information dissemination. From a descriptive point of view, these models are of little help to understand the actual process that drives popularity evolution. Recently, the work of Hu *et al.* focused on the defining longevity of social impulses, or multiple cascades [65]. However, unlike our approach, the authors are not focused on modeling the long term popularity of objects.

Table 7.1 summarizes the key properties of the aforementioned models as well as of our new Phoenix-R model. In comparison these approaches, Phoenix-R explicitly captures both revisits and multiple cascades, allows for non-linear solutions, and can be used for accurate forecasting. The next section presents the effect of revisits in both long and short term content popularity evolution for real world datasets. This is followed by the definition of the Phoenix-R model.

Table 7.2: Datasets of User Activities Mined With Phoenix-R

Application	# of User Activities	# of Users	# of Objects
MMTweet	1,086,808	215,376	25,060
LastFM	19,150,868	992	107,428
Twitter	476,553,560	17,069,982	49,293,684
YouTube	-	-	2,901,605

7.3 Content Revisit Behavior in Social Media

We now analyze the revisit behavior in various social media applications. We describe the datasets used in our analysis, and then discuss our main characterization findings.

7.3.1 Datasets

Our study is performed on four large user activity datasets, which are presented in Table 7.2 and we now summarize:

- The Million Musical Tweets Dataset (MMTweet): consists of 1,086,808 tweets of users about artists they are listening to at the time [63]. We focus on the artist of each tweet as an object. A total of 25,060 artists were mentioned in tweets.
- The 2010 LastFM listening habits dataset (LastFM): consists of the whole listening habits (until May 5th 2009) of nearly 1,000 users, with over 19 million activities on 107,428 objects (artists) [20].
- The 476 million Twitter tweets corpus (Twitter): accounts for roughly 20% to 30% of the tweets from June 1 2009 to December 31 2009 [146], and includes over 50 million objects (hashtags) tweeted by 17 million users.
- The YouTube dataset: Since 2013, YouTube began to provide the full daily time series (known as insight data) of visits for videos in the page of each video. We crawled the time series of roughly 3 million YouTube videos, as done in Chapter 3. However, different from the previous YouTube dataset, this one contains the full daily time series of each video. This information was unavailable at the time the study of Chapter 3 was done.

7.3.2 Main findings

Our goal is to analyze how the popularity acquired by different objects, in the long and short runs, is divided into audience and revisits. In particular, we aim at assessing

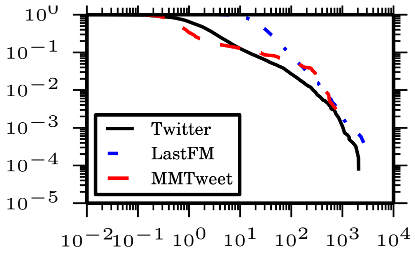


Figure 7.2: Distributions (CCDF) of $\frac{\#Revisits}{Audience}$.

to which extent the number of revisits may be *larger* than the size of the audience, in which case popularity is largely a sum of repeated user activities. Since this property may vary depending on the type of content, we perform our characterization on the LastFM, MMTweet, and Twitter datasets. We leave the YouTube dataset out of this analysis since, unlike the other datasets, it does not contain individual user activities, but only popularity time series. We will make use of the YouTube dataset to fit and evaluate our PHOENIX-R model, in the next section.

We first analyze the distribution of the final values[2] of popularity, audience, and revisits across objects in each dataset. For illustration purposes, Figure 7.2 shows the complementary cumulative distribution function of the ratio of the number of revisits to the audience size for all datasets, computed for objects with popularity greater than 500. We filtered out very unpopular objects, which attract very little attention during the periods of our datasets (over 6 months each). Note that the probability of an object having more revisits than audience (ratio greater than 1) is large. Indeed, though rare, the ratio of revisits to audience size reaches 10^2 and even 10^3.

In order to better understand these findings across all datasets, Table 7.3 shows, for each dataset: (1) the median of the ratio of number of revisits to audience size, (2) the median of the ratio of number of revisits to total popularity; and (3) the percentage of objects where the revisits dominate the popularity (i.e., ratio of number of revisits to the audience size greater than 1). Note that revisits dominate popularity in 66% of the Twitter objects. Moreover, on median, 62% of the total popularity of these objects is composed of revisits, which account for 1.7 times more activities than the visits by new users (audience size). Again, for LastFM artists, revisits are over 25 times more frequent than the visits by new users (on median), and the revisits dominate popularity in *all* objects. In contrast, the ratios of number of revisits to audience size and to total popularity are smaller for MMTweet objects, most likely because users do not tweet about artists they are listening to all the time, but rather only when they wish to share this activity with their followers. Yet, the revisits dominate popularity in 33%

[2] Values computed at the time the data was collected.

Table 7.3: Relationships between Revisits, Audience and Popularity.

Dataset	Median $\frac{\#Revisits}{Audience}$	Median $\frac{\#Revisits}{Popularity}$	% objects with $\frac{\#Revisits}{Audience} > 1$
Twitter	1.70	0.62	66%
MMTweet	0.68	0.40	33%
LastFM	25.39	0.96	100%

Table 7.4: Quartiles of the Ratio $\frac{\#Revisits}{Audience}$ for Various Time Windows w.

Dataset	Time window (w)	25^{th} percentile	Median	75^{th} percentile
Twitter	1 hour	1.08	3.93	12
	1 day	1	2.5	6.28
	1 week	0.66	1.69	4.28
	1 month	0.56	1.44	3.75
MMTweet	1 hour	0.25	0.66	12.5
	1 day	0.55	0.83	1.26
	1 week	0.41	0.73	1.41
	1 month	0.31	0.56	1.17
LastFM	1 hour	20	21	25
	1 day	21	28	41
	1 week	20	30.5	55.25
	1 month	14	25	48

of the objects. These results provide evidence that, at least in the long run, revisits are much more common than new users for many objects in different applications. For microblogs, though less intense, this behavior is still non-negligible.

We further analyze the effect of revisits on popularity, focusing now in the short term, by zooming into smaller time windows w. Specifically, we analyze the distributions of the ratios of number of revisits to audience size computed for window sizes w equal to one hour, one day, one week, and one month. Table 7.4 shows the three distribution quartiles for the various window sizes and datasets considered. These quartiles were computed considering only window sizes during which the popularity acquired by the object exceeds 20. We adopted this threshold to avoid biases in time windows with very low popularity, focusing on the periods where the objects had a minimal attention (note that 20 is still small considering that each trace has millions of activities).

Focusing first on the LastFM dataset, we note that, regardless of the time window size, the number of revisits is at least one order of magnitude (14x) larger than the audience size for at least 75% of the analyzed windows (25^{th} percentile). In fact, the ratio between the two measures exceeds 55 for 25% of the windows (75^{th} percentile) on the weekly case. In contrast, in the MMTweet dataset, once again, the ratios are much smaller. Nevertheless, at least 25% of the of the windows we observe a burst of revisits in very short time, with the ratio exceeding 12 for the hourly cases. Once again, we suspect that these lower ratios may simply reflect that users do not tweet about every artist they list to. Thus, in general, we have strong evidence that, for music-related content, popularity is mostly governed by revisits, as opposed to new users (audience).

The same is observed, though with less intensity, in the Twitter dataset. Revisits are more common than new users in 50% of the time windows, for all sizes considered. Indeed, considering hourly time windows, popularity is dominated by revisits for 75% of the cases. While large ratios, such as those observed for LastFM, do not occur, the number of revisits can still be 12 times larger than the audience size during a single hour in 25% of the Twitter hourly windows.

Our main conclusions so far are: (1) for most objects in the Twitter and LastFM datasets, popularity, measured both in the short (as short as 1 hour periods) and long runs, is mostly due to revisits than to audience size; and (2) revisits are less common on the MMTweet dataset, which we believe is due to data sparsity, but are still a significant component of the popularity acquired by a large fraction of the objects (in both long and short runs). These findings motivate the need for models that explicitly account for revisits in the popularity dynamics, which we discuss next.

7.4 The Phoenix-R Model

In this section we introduce the proposed Phoenix-R model (Section 7.4.1), and discuss how we fit the model to a given popularity time series (Section 7.4.2). In the next section we present results on the efficacy of the model on our datasets when compared to state-of-the-art alternatives, as well as the applicability of the Phoenix-R model on popularity prediction.

Similar to the previous chapters, we present vectors ($\mathbf{x}$) in bold. Sets are shown in non-bold calligraphy letters ($\mathcal{X}$), and variables are represented by lower case letters or Greek symbols (x, β). Moreover, $\mathbf{x}(i)$ means data index i (with indexes starting from from 1), and $\mathbf{x}(: i)$ means sub-vector up to i.

7.4.1 Deriving the Model

The Phoenix-R model is built based on the 'Susceptible-Infected-Recovery' (SIR) compartments, extending the basic model in order to capture revisits and multiple cascades. Specifically, Phoenix-R captures the following behavior *for each individual object*:

- We assume a fixed population of individuals, where each individual can be in one of three states: susceptible, infected and recovered.

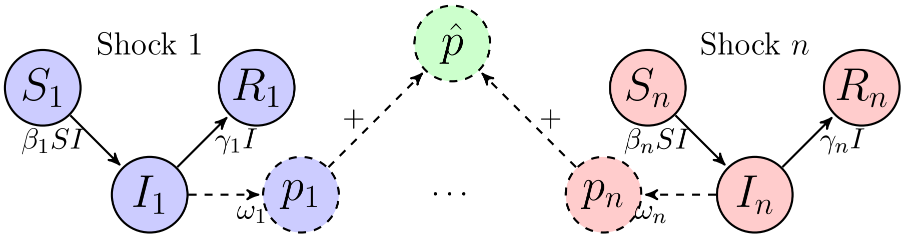


Figure 7.3: Individual Shocks that Account for the Phoenix-R Model.

- At any given time s_i, an external shock i causes initial interest in the object. The shock can be any event that draws attention to the object, such as a video being uploaded to YouTube, a news event about a certain subject, or even a search engine indexing a certain subject for the same time (thus making an object easier to be found). We assume that the initial shock s_1 is always caused by one individual.
- New individuals discover the object by being infected by the first one. Moreover, after discovery, these "newly infected" ndividuals can also infect other individuals, thus contributing to the propagation.
- Infected individuals may access (watch, play or tweet) the object. It is important to note that being infected does not necessarily imply in an access. For example, people may talk about a trending video before actually watching it. Each infected individual accesses the object following a Poisson process with rate ω ($\omega > 0$)[3].
- After some time, individuals lose interest in the object, which, in the model, is captured by a recovery rate γ.
- Multiple external shocks may occur for a single object.

Figure 7.3 presents the Phoenix-R model. In the figure, three compartments are shown for each shock i, namely S_i, I_i, and R_i. These compartments represent the number of susceptible, infected and recovered individuals for the shock i, respectively. Variable p_i, associated with shock i, measures the popularity acquired by the object due this shock. The total popularity of the object, i.e., the sum of the values of p_i for all shocks, is denoted by $\hat{p}$. We first present the model for a single shock, and then generalize the solution for multiple shocks. For the sake of simplicity, we drop the subscripts while discussing a single shock. We present the model assuming discrete time, referring to each time tick as a time window.

Each shock begins with a given susceptible population ($S(0)$) and one infected individual ($I(0) = 1$). The total population is fixed and given by ($N = S(0) + 1$). The R compartment captures the individuals that have already lost interest in the object. Similarly the SI model, βSI susceptible individuals become infected in each

[3] Both [5, 66] show the poissonian behavior of mutiple visits from the same user.

time window. Moreover, γI individuals loose interest in (i.e., recover from) the object in each window. Revisits to the object are captured by the rate ω. Thus ω is the expected number of accesses of an individual up to time t. The probability of the individual accessing the object k times during a time interval of τ windows is given by:

$$P(v(t+\tau) - v(t) = k) = \frac{(\omega\tau)^k e^{-\omega\tau}}{k!}. \tag{7.3}$$

We assume that the shock starts at time zero, thus focusing the dynamics *after* the shock. Under this assumption, the equations that govern a single shock are:

$$S(t) = S(t-1) - \beta S(t-1)I(t-1) \tag{7.4}$$

$$I(t) = I(t-1) + \beta S(t-1)I(t-1) - \gamma I(t-1) \tag{7.5}$$

$$R(t) = R(t-1) + \gamma I(t-1) \tag{7.6}$$

$$p(t) = \omega I(t). \tag{7.7}$$

The equation $p(t) = \omega I(t)$ accounts for the expected number of times infected individuals access the object, thus capturing the popularity of the object at time t due to the shock. We can also define the expected audience size of the object at time t due to the shock, $a(t)$, as:

$$a(t) = (1 - e^{-\frac{\omega}{\gamma}})\beta S(t-1)I(t-1). \tag{7.8}$$

Each newly infected individual ($\beta S(t-1)I(t-1)$) will stay infected for γ^{-1} windows (see [64]). The probability of generating at least one access while the individual is infected is:

$$1 - P(v(t+\gamma^{-1}) - v(t) = 0) = 1 - e^{-\frac{\omega}{\gamma}}. \tag{7.9}$$

Thus, we here capture the individuals that were infected at some time and generated at least one access.

The Phoenix-R model is thus defined as the sum of the popularity values due to multiple shocks. We discuss how to determine the number of shocks in the next section. Given a set of shocks $\mathcal{S}$, where shock i starts at given time s_i, the popularity

$\hat{p}$ is:

$$\hat{p}(t) = \sum_{i, s_i \in \mathcal{S}} p_i(t - s_i)\mathbb{1}[t > s_i] \tag{7.10}$$

where $\mathbb{1}[t > s_i]$ is an indicator function that takes value of 1 when $t > s_i$, and 0 otherwise. Audience size $\hat{a}(t)$ can be similarly defined. Also, both in the single shock and in the PHOENIX-R models, the number of revisits at time t, $\hat{r}(t)$, can be computed as $\hat{r}(t) = \hat{p}(t) - \hat{a}(t)$. The overall population that can be infected is defined by:

$$N = \sum_i N_i = \sum_i S(0)_i + 1. \tag{7.11}$$

Note that we assume that the population of different shocks do not interact, that is, an infected individual from shock s_i does not interact with a susceptible one from shock s_j, where $i \neq j$. While this may not hold for some objects (e.g., people may hear about the same content from two different populations), it may be a good approximation for objects that become popular in large scale (e.g., objects that are propagated world wide). It also allows us to have different values of β_i, γ_i, and ω_i for each population. Intuitively, the use of different parameters for each shock captures the notion that some objects may be more (or less) interesting for different populations. For example, samba songs may attract more interest from people in Brazil.

Adding a period: In some cases, the popularity of an object may be affected by periodical factors. For example, songs may get more plays on weekends. We add a period to the PHOENIX-R model by making ω fluctuate in a periodic manner. That is:

$$\omega_i(t) = \omega_i * (1 - \frac{m}{2} * (sin(\frac{2\pi(t+h)}{e}) + 1)), \tag{7.12}$$

where e is the period, and sin is a sine function. For example, for daily series we may set $e = 7$ if more interest is expected on weekends. Since an object may have been uploaded on a Wednesday, we use the shift h parameter to correct the sine wave to peak on weekends. The amplitude m captures oscillation in visits. The same period parameters are applied to every shock model. This approach is similar to the one adopted in [91].

The final PHOENIX-R model will have 5 parameters to be estimated from the data *for each* shock, namely, $S(0)_i$, β_i, γ_i, ω_i, s_i; plus the m and h period parameters. The last two do not change for individual shocks. We decided to fix e in our experiments to 7 days, when using daily time windows, and $e = 24$ hours when using hourly series.

7.4.2 Fitting the Model

We now discuss how to fit the PHOENIX-R parameters to real world data. Our goal is to produce a model that delivers a good trade-off between parsimony (i.e., small number of parameters) and accuracy. To that end, three issues must be addressed: (1) the identification of the start time of each individual shock; (2) an estimation of the cost of the model associated with multiple shocks; and, (3) the fitting algorithm itself. Note that one key component of the fitting algorithm is model selection: it is responsible for determining the number of shocks that will compose the PHOENIX-R model, choosing a value based on the cost estimate and model accuracy.

Finding the start times s_i of the shocks: Intuitively, we expect each shock to correspond to a peak in the time series. Indeed, previous work has looked at the dynamics of single shock cascades, finding a single prominent peak in each cascade [7, 91]. With this in mind, instead of searching for s_i directly, we initially attempt to find peaks. We can achieve both tasks using a continuous wavelet transform based peak finding algorithm [38]. We chose this algorithm since it has the following key desirable properties. Firstly, it can find peaks regardless of the "volume" (or popularity in the present context) in the time windows surrounding the peaks. It does so by only considering peaks with a high signal to noise ratio in the series. Secondly, the algorithm is fast, with complexity in the order of the length, n, of the time series ($O(n)$). Lastly and more importantly, using the algorithm we can estimate both the peaks and the start times of the shocks that caused each peak. We shall refer to the algorithm as $FindPeaks$.

As stated $FindPeaks$ makes use of a continuous wavelet transform to find the peaks of the time series. Specifically, we apply the Mexican Hat Wavelet[4] for this task. The Mexican Hat Wavelet is parametrized by a half-width l. We use half-widths (l) of values $\{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ to find the peaks. Thus, for the peak identified at position k_i, with wavelet determined by the parameter l_i, we define the start point of the shock s_i as: $s_i = k_i - l_i$. We found that using the algorithm with the default parameters presented in [38], combined with our MDL fitting approach (see below), proved accurate in modeling the popularity of objects[5].

Estimating the cost of the model with multiple shocks: we estimate the cost of a model with $|S|$ shocks based on the minimum description length (MDL) principle [60, 102], which is largely used for problems of model selection. To apply the MDL principle, we need a coding scheme that can be used to compress both the model

[4] `https://en.wikipedia.org/wiki/Mexican_hat_wavelet`

[5] We used the open source implementation available with SciPy (`http://scipy.org`)

parameters and the likelihood of the data given the model. We here provide a new intuitive coding scheme, based on the MDL principle, for describing the PHOENIX-R model with $|\mathcal{S}|$ shocks, assuming a popularity time series of n elements (time windows). As a general approach, we code natural numbers using the $\log^*$ function (universal code length for integers)[6] [60], and fix the cost of floating point numbers at $c_f = 64$ bits.

For each shock i, the complexity of the description of the set of parameters associated with i consists of the following terms: $\log^*(n)$ for the s_i parameter (since the start time of i can be at any point in the time series); $\log^*(S_i(0))$ for the initial susceptible population; and $3 * c_f$ for β_i, γ_i, and ω_i. We note that an additional cost of $\log^*(7) + 2 * c_f$ is incurred if a period is added to the model. However, we ignore this component here since it is fixed for all models. Therefore, it does not affect model selection. The cost associated with the set of parameters $\mathcal{P}$ of all $|\mathcal{S}|$ shocks is:

$$Cost(\mathcal{P}) = |\mathcal{S}| \times (\log^*(n) + \log^*(S_i(0)) + 3 * c_f) + \log^* |\mathcal{S}|. \tag{7.13}$$

Given the full parameter set $\mathcal{P}$, we can encode the data using Huffman coding, i.e., a number of bits is assigned to each value which is the logarithm of the inverse of the probability of the values (here, we use a Gaussian distribution as suggested in [102] for the cases when not using probabilistic models.).

Thus, the cost associated with coding of the time series given the parameters is:

$$Cost(\mathbf{t} \mid \mathcal{P}) = -\sum_{i=1}^{n} \log(p_{gaussian}(\mathbf{t}(i) - \mathbf{m}(i); \mu, \sigma)). \tag{7.14}$$

where $\mathbf{t}$ is the time series data and $\mathbf{m}$ is the time series produced by the model (i.e., $\mathbf{t}(i) - \mathbf{m}(i)$ is the error of the model at time window i.) Here, $p_{gaussian}$ is the probability density function of a Gaussian distribution with mean μ and standard deviation σ estimated from the model errors. We do not include the costs of encoding μ and σ because, once again, they are constant for all models. The total cost is:

$$Cost(\mathbf{t}; \mathcal{P}) = \log^* n + Cost(\mathcal{P}) + Cost(\mathbf{t} \mid \mathcal{P}). \tag{7.15}$$

This accounts for the parameters cost, the likelihood cost, and the cost of the data size.

Fitting algorithm: The model fitting approach is summarized in Algorithm 4. The algorithm receives as input a popularity time series $\mathbf{t}$. It first identifies candidate shocks using the *FindPeaks* method, which returns the peaks $\mathbf{p}$ and the start times

[6] $\log^*(x) = 1 + \log^*(\log x)$ if $x > 1$. $\log^*(x) = 1$ otherwise. We use base-2 logarithms.

Algorithm 4 Fitting the PHOENIX-R Model. Only the time series is required as input.

```
 1: function FITPHOENIXR($\mathbf{t}$)
 2:     $\epsilon = 0.05$
 3:     $\mathbf{s} \leftarrow \{\}$
 4:     $\mathbf{p}, \mathbf{s}' \leftarrow FindPeaks(\mathbf{t})$
 5:     $\mathbf{s}[1] = 0$
 6:     $\mathbf{s} \leftarrow append(\mathbf{s}')$
 7:     $\mathcal{P} \leftarrow \{\}$
 8:     $min\_cost \leftarrow \infty$
 9:     for $i \leftarrow 1$ to $|s|$ do
10:         $\mathcal{F} \leftarrow LM(\mathbf{t}, \mathbf{s}(:i))$
11:         $\mathbf{m} \leftarrow PhoenixR(\mathcal{F})$
12:         $mdl\_cost \leftarrow Cost(\mathbf{m}, \mathbf{t}, \mathcal{F})$
13:         if $mdl\_cost < min\_cost$ then
14:             $min\_cost \leftarrow mdl\_cost$
15:             $\mathcal{P} \leftarrow \mathcal{F}$
16:         end if
17:         if $mdl\_cost > min\_cost * (1 + \epsilon)$ then
18:             break
19:         end if
20:     end for
21:     return $\mathcal{P}$
22: end function
```

$\mathbf{s}'$ of the corresponding shocks in decreasing order of peak volume (line 3). To account for the upload of the object, we include one other shock starting at time $s_1 = 0$, in case a shock was not identified in this position. Each s_i is stored in vector $\mathbf{s}$, ordered by the volume of the each identified peak (with the exception of $s_1 = 0$ which is always in the first position) (lines 4 and 5). We then fit the PHOENIX-R model using the Levenberg-Marquardt (LM) algorithm adding one shock at a time, in the order they appear in $\mathbf{s}$ (loop in line 9), that is, in decreasing order of peak volume (after the initial shock). Intuitively, shocks that lead to larger peaks account for more variance in the data. For each new shock added, we evaluate the MDL cost (line 12). We keep adding new shocks as long as the MDL cost decreases (line 13) or provided that an increase of at most ϵ over the best model is observed[7] (line 17). We set the Levenberg-Marquardt algorithm to evaluate the mean squared errors of the model and adopt a threshold ϵ equal to 5%. We also note that we initialize each parameter randomly (uniform from 0 to 1), except for $S_i(0)$ values. For the first shock we do test multiple initial values: $S_1(0) = 10^3$, 10^4, 10^5, and 10^6. The other $S_i(0)$ values are initialized to the corresponding peak volume.

7.5 Experiments

In this section we discuss the experimental evaluation of the PHOENIX-R model. Initially, we present results on the efficacy of the model on our datasets when compared to

[7] MDL based costs will decrease with some variance and then increase again. The ϵ threshold is a guard against local minima due to small fluctuations.

state-of-the-art alternatives (Section 7.5.1) Next, we show results on the applicability of the model for popularity prediction (Section 7.5.2)[8].

7.5.1 Is Phoenix-R Better than Alternatives?

We compare Phoenix-R with two state-of-the-art alternatives: the TemporalDynamics [112], used to model query popularity; and the SpikeM model [91], which captures single cascades. We compare these models in terms of time complexity, accuracy, estimated by the root mean squared errors (RMSE), and cost-benefit. For the latter, we use the Bayesian Information Criterion (BIC) [112], which captures the tradeoff between cost (number of parameters) and accuracy of the model.

In terms of time complexity, we note that the Phoenix-R model scales linearly with the length of the time series n. This is shown in Figure 7.4, which presents the number of seconds (y-axis) required to fit a time series with a given number of time windows (x-axis). TemporalDynamics also has linear time complexity [112]. In contrast, the equations that govern the SpikeM model requires quadratic ($O(n^2)$) runtime on the time series length, making it much less scalable to large datasets.

In terms of accuracy, we make an effort to compare Phoenix-R with the alternatives in fair settings, with datasets with similar characteristics from those used in the original papers. In particular, when comparing with TemporalDynamics, we run the models proposed in [112] selecting the best one (i.e., the one with smallest root mean squared error) for each time series. Moreover, we use long term daily time series (over 30 days), with a total popularity of at least 1,000[9]. We compare Phoenix-R and TemporalDynamics under these settings in our four datasets, including YouTube.

When comparing with SpikeM, we use Twitter hourly time series trimmed to 128 time windows around the largest peak (most popular hour). We focus on the 500 most popular of these times series for comparison. We chose this approach since this is the same dataset explored by the authors. Moreover, we focus on a smaller time scale because the SpikeM model was proposed for single cascades only.

Table 7.5 shows the average RMSE (along with corresponding 95% confidence intervals) computed over the considered time series for all models. Best results of each comparison (including statistical ties with significance of 0.01) are shown in bold. Note that Phoenix-R has statistically lower RMSE than TemporalDynamics in all datasets. These improvements come particularly from the non-linear nature of Phoenix-R , which better fits the long term popularity dynamics of most objects. The difference

[8] All of our source code is provided at: `http://github.com/flaviovdf/phoenix`

[9] Similar results were achieved using other thresholds.

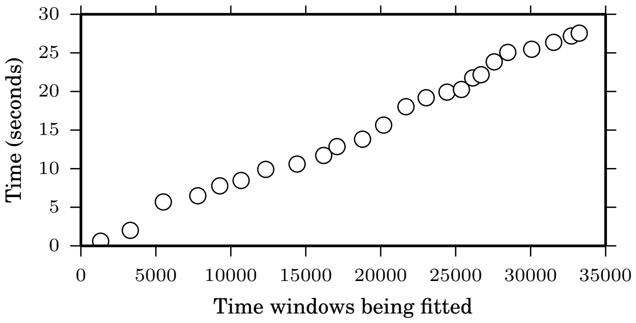


Figure 7.4: Scalability of Phoenix-R

Table 7.5: Comparison of Phoenix-R with TemporalDynamics [112] and SpikeM [91]: Average RMSE values (with 95% confidence intervals in parentheses). Statistically significant (p-value of 0.01) results (including ties) are shown in bold.

	Phoenix-R vs. TemporalDynamics (daily series)		Phoenix-R vs. SpikeM (hourly series)	
	RMSE Phoenix-R	RMSE TemporalDynamics	RMSE Phoenix-R	RMSE SpikeM
MMTweet	**2.93** (± 0.23)	4.18 (± 0.49)	-	-
LastFM	**7.09** (± 0.23)	8.31 (± 0.32)	-	-
Twitter	**72.05** (± 6.08)	194.79 (± 20.49)	**10.83** (± 1.61)	**9.77** (± 2.24)
YouTube	**280.58** (± 29.29)	3429.19 (± 577.76)	-	-

between the models is more striking for the YouTube dataset, where most time series cover long periods (over 4 years in some cases). The linear nature of TemporalDynamics largely affects its performance in those cases, as many objects do not experience a linear popularity evolution over such longer periods of time. As result, Phoenix-R produces reductions on average RMSE of over one order of magnitude. In contrast, the gap between both models is smaller in the LastFM dataset, where the fraction of objects (artists) for which a linear fit is reasonable is larger. Yet, Phoenix-R produces results that are still statistically better, with a reduction on average RMSE of 15%.

When comparing with SpikeM, the Phoenix-R model produces results that are statistically tied. We consider this result very positive, given that this comparison favors SpikeM: the time series cover only 128 hours, and thus there is no much room for improvements from capturing multiple cascades, one key feature of Phoenix-R . Yet, we note that our model is more general and suitable to modeling popularity dynamics in the longer run, besides being much more scalable, as discussed above.

As a final comparison, we evaluate the cost-benefit of the models using BIC, as suggested by [112]. We found that Phoenix-R out performs TemporalDynamics in terms of BIC on at least 80% of the objects in all datasets but LastFM. For LastFM objects, the reasonable linear evolution of popularity of many objects, makes the cost-benefit of TemporalDynamics superior. Yet, Phoenix-R is still the preferred option in

30% of the objects in this dataset. Compared to SpikeM we also find that once again, statistically equal BIC scores are achieved for both models.

7.5.2 Predicting Popularity with Phoenix-R

We here assess the efficacy of Phoenix-R for predicting the popularity of objects a few time windows into the future, comparing it against TemporalDynamics[10]. To that end, we train the Phoenix-R and TemporalDynamics models for each time series using 5%, 25%, and 50% of the initial daily time windows. We then use the δ time windows following the training period as validation set to learn model parameters. In each setting, we train 10 models for each time series, selecting the best one on the validation period. We then use the selected model to estimate the popularity of the object δ windows after the validation (test period). We experiment with δ equal to 1, 7 and 30 windows.

Table 7.6 shows the average RMSE of both models on the test period. Confidence intervals are omitted for the sake of clarity, but the best results (and statistical ties) in each setting are shown in bold. Phoenix-R produces more accurate predictions than TemporalDynamics in practically all scenarios and datasets. Again, the improvements are quite striking for the YouTube dataset, mainly because the time series cover long periods (over 4 years in some cases). While the linear TemporalDynamics model fits reasonably well the popularity dynamics of some objects, it performs very poorly on others, thus leading to high variability in the results. In contrast, Phoenix-R is much more robust, producing more accurate predictions for most objects, and thus being more suitable for modeling and predicting long periods of user activity.

7.6 Summary

In this chapter we presented the Phoenix-R model for describing social media popularity evolution time series. Before introducing the model, we showed the effect of revisits on the popularity of objects on large user activity datasets. Our main contributions are:

- **Discoveries:** We explicitly show the effect of revisits in social media popularity.
- **Explanatory model:** We introduce the Phoenix-R model, which explicitly accounts for revisits and multiple cascades. Factors that ere not captured by state-of-the art alternatives.

[10] We do not use SpikeM for this task, as it is suitable for tail forecasting only (i.e., predicting after the peak)

Table 7.6: Comparing Phoenix-R with TemporalDynamics [112] for Prediction. The values on the table are RMSE. Statistically significant results are in bold

		5%			25%			50%		
		1	7	30	1	7	30	1	7	30
MMTweet	PhoenixR	**11.61**	**12.78**	**15.15**	**8.67**	**6.74**	**8.82**	**4.08**	**6.87**	**13.58**
	TempDynamics	**17.07**	**17.41**	**16.52**	**9.63**	**10.78**	**14.46**	**25.19**	**23.08**	**30.39**
Twitter	PhoenixR	**53.68**	**60.78**	**215.76**	**132.21**	**135.15**	**210.30**	**75.58**	**229.59**	**254.93**
	TempDynamics	**104.45**	**129.36**	**255.69**	643.39	643.83	786.50	420.74	587.86	598.75
LastFM	PhoenixR	**2.37**	**3.97**	**5.71**	**8.60**	**12.06**	**14.66**	**11.34**	**15.03**	**15.43**
	TempDynamics	6.47	7.03	8.00	11.15	14.62	17.86	14.91	18.15	18.80
YouTube	PhoenixR	**91.62**	**106.38**	**138.88**	**83.76**	**113.14**	**147.04**	**127.53**	**97.97**	**115.97**
	TempDynamics	3560.65	3631.09	3661.81	5091.82	5107.82	5143.70	4136.14	4139.73	4169.26

- **Scalable and Parsimonious:** Our fitting approach make's use of the MDL principle to achieve a parsimonious description of the data. We also show that fitting the model is scalable (linear time).
- **Effective:** of model: We showed the effectiveness of the model not only when describing popularity time series, but also when predicting future popularity values for individual objects. Improvements over state-of-the art alternatives can be up to one order of magnitude, depending on the dataset.

In the next chapter we shall present our final study on social media popularity. More specifically, we shall now focus on the attention flows across objects (e.g., transitions between objects). More importantly, we also discuss how the competition and collaboration for attention between objects affect popularity in social media applications. After that, Chapter 9 concludes this dissertation.

Chapter 8

Mining User Attention Flows

User attention is arguably one of the most scarce and vied for commodities of today's Internet economics. Selling user attention is a multi-billion dollar business that sustains some of the most popular social media applications. Unfortunately, social media user attention flows present a variety of complex system behaviors unfit to an analysis with existing approaches, such as asynchronous users with mixed but similar behavior, repeated consumption effects, niche artist sparsity, rich-get-richer effects, and the effects of fluctuations due to external shocks. Mindful of these challenges in this chapter we propose A-FLUX, a user attention mining method designed to cope with the complex challenges of attention flow mining.

With A-FLUX, we present a complementary study on the popularity of social media objects as the one done with the PHOENIX-R model. Whereas in the previous chapter we focused only on revisit behavior, with A-FLUX we present a data mining approach to understand attention flows across objects. One of the main technical contributions of A-FLUX is a probabilistic graphical model that captures the latent object-to-object transitions of user attention. We also use modulated Markov models to capture user attention short and heavy tails observed in real-world datasets.

Our case study is on a popular social media Online Music Streaming Service (OMSS). OMSSs are the fastest growing revenue streams of the music industry. OMSS ad earnings are shared with record labels in proportion to their artists "air time". Thus, understanding this marketplace entails mapping how user attention flows between artists (objects). We employ A-FLUX on large datasets crawled from a popular OMSS, revealing interesting and meaningful user attention flow maps and patterns. Specifically, we observe that overall user attention seem to be elastic, as many artists seem to cooperate for user attention, although we also find evidence of competition, notably newcomers that steal user attention away from others.

8.1 Introduction

There is no free lunch on today's Internet. In some of the most popular social media applications, user attention is a valuable commodity that users trade for services, fueling advertisement sales of billions of dollars. Music streaming is a particularly interesting case study. Online Music Streaming Services (OMSSs) currently account for 40% of all digital revenue of artists and record labels, and may soon be the dominating form of revenue of the music industry [104]. OMSSs translate user attention in revenues through online ads and subscription services: for each song played part of the revenue is shared with record labels and their artists.

The first step to understand this marketplace of attention is to study how user attention flows across different artists. While there are many reasons behind user attention flows – user musical interests, recommendation systems, and even the alphabetical order of track, album, and artist names – the end result is a flow of user attention between artists. Given the relevance of the problem, there is a demand for methods that can answer the following questions:

> *Which artists cooperate and which compete for attention? Are OMSSs zero-sum attention markets?*

In a zero-sum attention market an artist gains attention at the expense of another artist. Unfortunately, user attention flows in OMSSs depend on complex phenomena that make it hard to analyze. Some of these phenomena, which were observed in our datasets of user plays gathered from a popular OMSS, are:

a) *Asynchronous users with mixed but similar behavior*: Users who like similar artists will not start their playlists at the same time (e.g., they may live in distinct time zones) or listen to songs in the same order. Indeed, regarding the latter we observe that 97% of the inter-artist transitions[1] $a \to b$, $a \neq b$ happen less than ten times in one of our datasets with over 200 million user plays. Still, attention can flow from artist a to artist b through artist c, $a \to c \to b$, even though there is no direct transition $a \to b$. Our method should be able to cope with this.

b) *Repeated consumption*: Users tend to listen to artists in bursts, more than what one would expect at random in a shuffled playlist[2]. For instance, over 60% of all played songs in our largest dataset are consecutive plays of the same artist, i.e., intra-artist attention flows are 1.5× larger than inter-artist flows. But at random

[1] The attention of user u flows from artist a to artist b if b is the next artist u listens to after a. This event is a *transition* $a \to b$.

[2] This effect may be intentional or because continuously listening to the same artist is the player's default.

the probability of re-consuming the same artist is only 0.2%. Our method must cope with intra-artist attention flows overpowering the analysis of rarer inter-artist attention flows.

c) *Biased observations and small subpopulations*: Data of online user behavior is biased towards subpopulations of users interested in the website services; and OMSSs are no exception. But we still want to be able to analyze underrepresented subpopulations: say, the behavior of heavy metal fans in a dataset dominated by fans of teenage pop artists. Two difficulties arise here: (c.1) the signal of the largest subpopulation can be much stronger than that of smaller subpopulations; and (c.2) small subpopulations show few inter-artist transitions. For instance, in our largest dataset up to 74% of the artists see fewer than ten "plays" leading to inter-artist transitions.

d) *External shock effects*: Attention paid to an artist may spike due to external shocks such as concerts, media exposure, and album releases.

We are aware of no previous model of attention flows that cope with the aforementioned effects (a-d). Most previous work focuses on modeling user attention to a single "object" (e.g., artist) [18, 112, 129], and thus does not capture the inter-artist attention flows that are essential to attention flows. As we show in our results, the same issue is present in other latent factor approaches, including those that capture limited attention [73]. Those approaches often model user-to-artist plays, rather than explicit transitions, as we do.

The models that do focus on defining competition and collaboration through user behavior [100] are designed for large datasets of well represented subpopulations as the maximum likelihood point estimates used in the method cannot cope with rare transitions [116]. More importantly, such methods are applied to Online Social Networks (OSNs), requiring social graphs to determine when users are exposed to pieces of information. Exact times of exposure to music is largely unaccounted for in OMSSs, mostly due to the various ways that users are exposed to new songs [108].

Our main contribution in this chapter is the new A-FLUX method (available for download[3]), which accurately captures user attention flows in OMSSs taking into account all four effects (a-d). A-FLUX includes a modulated Markov model that captures the long and short tails of repeated attention to single artists and a probabilistic graphical model that captures the latent inter-artist transitions of user attention. We show that A-FLUX can reveal interesting and semantically meaningful patterns of user attention flows by employing it to our datasets. In particular, the Bayesian latent fac-

[3] `http://github.com/flaviovdf/aflux`

tor approach of A-FLUX enables the finding of relevant patterns, avoiding problems associated with point estimates.

Using A-FLUX, we also take a step towards tackling our motivating questions by finding strong evidence that OMSS attention is elastic. Specifically, we find various cases of artists which seem to cooperate for attention, whereas some others seem to steal attention away. Finding evidence of attention elasticity in OMSSs is a novel application that has not been explored by previous work. Although it is not possible to generalize our findings to different forms of music consumption (e.g., radio plays), A-FLUX is a general method that can be readily applied to other OMSS datasets.

In the next section, we detail the datasets we gathered from a popular OMSS and mined with A-FLUX. Afterwards, in Section 8.3 we define the A-FLUX model. Afterwards, we present in Section 8.4 our results on applying A-FLUX to uncover attention flow patterns from our datasets. Then, in Section 8.5 we present a comparison of A-FLUX with other baseline approaches. Finally, Section 8.6 concludes this chapter.

8.2 OMSS Datasets

We apply A-FLUX on two datasets crawled from Last.FM[4], a popular social media application and a OMSS. Last.FM aggregates various forms of digital music consumption, ranging from desktop/mobile media players to streaming services (theirs and others)[5]. As stated, Last.FM also has social media features such as an OSN, allowing the creation of user groups, as well as providing demographical data about users.

Our datasets are:

Last.FM-1k Collected using a snowball sampling [20] approach. After the snowball sampling, 992 uniformly random users were selected. The dataset contains, for each user, the complete listening history (all plays) from February 2005 to May 2009, the self-declared nationality, age (at the time), and registration date [20]. This dataset accounts for 18.5 million (user, artist, time) triples, and 107,397 unique artists. This is the same dataset explored in Chapter 7.

Last.FM-Groups Crawled in 2014, using the user groups from Last.FM. We manually selected 17 groups: *Active Users*, *Music Statistics*, *Britney Spears*, *The Strokes*, *Arctic Monkeys*, *Miley Cyrus*, *LMFAO*, *Katy Perry*, *Jay-Z*, *Kanye West*, *Lana Del Rey*, *Snoop Dogg*, *Madonna*, *Rihanna*, *Taylor Swift*, *Adelle*, and *The Beatles*. For each group, we crawled the listening history (from February 2005 to August 2014)

[4] `http://last.fm`

[5] Aggregation is done using plugins available on other OMSSs and desktop/mobile media players.

Table 8.1: Summary of our Last.FM datasets

	Last.FM-1k	Last.FM-Groups
# Artists	107,397	836,625
# Users	992	15,329
# Plays	18,548,702	218,377,124
Lifespan	2005 to 2009	2005 to 2014

of a subset of the users (the first users listed in the group)[6]. The total number of crawled users in 15,329. The number of users per group crawled varies from 16 to 2,343 (median of 421). This dataset has 836,625 unique artists and roughly 218 million triples. While this dataset has biases towards more active users and pop artists (given our choice of groups), it has over 10 times more users and plays than the Last.FM-1k dataset. Morever, it allows us to analyze the behavior of fans (based on group membership) of different artists. This is a desirable property to mine user behavior when major music event happens (e.g., album releases). We also crawled the age and nationality of all the users.

Table 8.1 presents the total numbers of artists, users, and music plays on each dataset. We cross-referenced the datasets with other public music databases. That is, we also use the Million Songs data[7], containing song durations, to estimate the time dedicated to an artist, and the MusicBrainz dataset[8] to gather release dates of albums (and singles). We collected the dates of 285 releases by 11 artists[9], with at least 12 and at most 40 releases per artist.

The work of Nowak [108] discussed the social-material relations of music consumption, concluding that even the same user still relies on multiple forms of music consumption (e.g., legal and illegal downloading, streaming services, CDs, etc). Because of these various means of consumption, our music streaming case study, Last.FM, presents itself as a good platform for studying *online behavior*. The service aggregates user accesses from desktop media players (which incorporate legal and illegal downloads), free, and also paid streaming services. The issues of analyzing a OMSS that does not aggregate various forms of consumption is further emphasized by [8], where the authors showed the disagreement between web and social music services (in terms of artist popularity).

It is important to point out that attention flows are influenced by internal mechanisms employed by OMSSs, such as recommendation services and user interfaces.

[6] We focus on the first (more active) users due to rate limits.

[7] `http://labrosa.ee.columbia.edu/`

[8] `http://musicbrainz.org`

[9] A subset of the artists in the user groups of Last.FM-Groups.

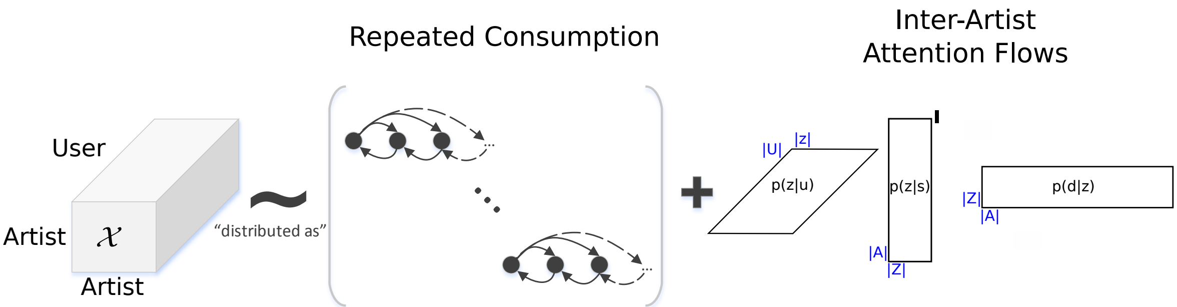


Figure 8.1: The A-FLUX Model: Data Representation by Tensor $\boldsymbol{\mathcal{X}}$ (left), the Repeated Consumption Modulated Markov Model (center) and the Inter-artist Model (right).

These are an integral part of the business model and, more importantly, our datasets reflect a multitude of such effects since Last.FM aggregates various forms of music consumption. Due to the role of such effects in OMSS this a desirable property when measuring attention flows. However, A-FLUX can be used in datasets which do not reflect these effects if necessary (e.g., when comparing recommendation engines or user interfaces).

8.3 The A-FLUX Model

We derive A-FLUX to capture two user behaviors, namely the repeated consumption of artists – intra-artist flows – and the inter-artist attention flows. We exploit stochastic complementation [95] to isolate these two behaviors, and propose two complementary markovian models, as illustrated in Figure 8.1. The fixation model consists of a modulated Markov model that accurately captures both long and short attention tails of repeated consumption. The inter-artist attention flow model exploits a probabilistic graphical model that captures the latent artist-to-artist transitions of user attention. Our goal is to capture piecewise stationary processes governing user attention flows. This way, the dataset can be analyzed on time windows where user behavior is roughly stationary.

A naïve way to capture user attention flows between artists s (source) and d (destination) is to build a transition probability matrix $\mathbf{P}_{s,d}$ with the probability that a user listens to an artist d after listening artist s. This approach has undesirable properties [116]. Matrix $\mathbf{P}$ is estimated through maximum likelihood point estimates obtained by dividing the number of transitions $s \to d$ by the total number of transitions out of s. Unfortunately, for 74% of the artists in our largest dataset the denominator has fewer than ten outgoing transitions. This creates two undesirable effects: (1)

there are not enough samples to accurately estimate the transition probabilities for most artists; and, (2) the transition probability matrix $\mathbf{P}$ is sparse, stating that it is impossible to flow from an artist s to an artist d when no user has done so in the past.

In contrast, A-FLUX divides user attention between intra-artist and inter-artist flows, and uses a latent space Bayesian approach. The intra/inter flow separation is possible by treating attention flows as a reducible system, where we model the strong memory of intra-artist transitions – some users continuously listen to the same artist for hours – as only interfering with the inter-artist dynamics through limited user attention. This creates an effective separation between the intra-artist model and the inter-artist model. User's limited attention is the glue that correlates intra and inter artist plays by considering that, given a user's limited attention, there will be only a limited number of listened songs. This *budget* of songs is first spent at listening to songs of the same artist s. One play takes us to an inter-artist transition from artist s to artist d, $s \neq d$, which then again transitions to the intra-artist model of artist d. The inter-artist attention flows are captured by a graphical model, using a Bayesian approach to estimate inter-artist transitions, thus avoiding problems associated with point estimates and being robust to infrequent transitions of small subpopulations of interest. A key point to ensure this separation is modeling the intra-artist song plays as a Makov chain that only allows entrance and exit from and to other artists at the same state. This way, the incoming and outgoing flows to and from other artists are independent of the intra-artist memory, achieving the desired system reducibility.

Before describing A-FLUX, we justify our design choice of not using Phoenix-R (described in Chapter 7) as part of A-FLUX. Recall that Phoenix-R is focused on revisits, one factor also captured by A-FLUX. However, with A-FLUX we aim at capturing user attention flows using a Markovian system. The epidemic approach of Phoenix-R is non-Markovian. That is, coupling Phoenix-R with the inter-artist attention flows of A-FLUX would not lead to a Markovian and reducible system. Nevertheless, since both models aim at analyzing complementary behaviors, the results on this chapter are complementary to our previous studies.

8.3.1 Data Representation and Notation

Let $\mathcal{D}$ be a dataset consisting of $(\text{user}, \text{artist}, \text{timestamp})$ tuples observed over a time window $[0, T]$. Let $\mathcal{U}$ be the set of users and $\mathcal{A}$ the set of artists in $\mathcal{D}$. Mining $\mathcal{D}$ through its original coordinate system $\mathcal{U} \times \mathcal{A} \times [0, T]$ is problematic because of effects such as "asynchronous users with mixed but similar behavior" – issue (a) in the introduction.

To circumvent this effect, we change the coordinate system, transforming $\mathcal{D}$ into a 3-mode tensor $\boldsymbol{\mathcal{X}}$ over $\mathcal{U} \times \mathcal{A} \times \mathcal{A}$. One mode has dimension $|\mathcal{U}|$ and represents the users. The other two modes both have dimension $|\mathcal{A}|$ and represent sources s and destination d artists.

More specifically, let all users (artists) to be numbered between one and $|\mathcal{U}|$ ($|\mathcal{A}|$). Given n_{usd}, the number of times user $u \in \mathcal{U}$ transitioned from $s \in \mathcal{A}$ to $d \in \mathcal{A}$, we define tensor $\boldsymbol{\mathcal{X}} = \left[\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{|\mathcal{U}|}\right]$, where $\mathbf{X}_u$ is:

$$\mathbf{X}_u = \begin{bmatrix} n_{u11} & \cdots & n_{u1|\mathcal{A}|} \\ \vdots & \ddots & \vdots \\ n_{u|\mathcal{A}|1} & \cdots & n_{u|\mathcal{A}||\mathcal{A}|} \end{bmatrix} \tag{8.1}$$

This data representation, illustrated in the left side of Figure 8.1 (left), is essentially distinct from other tensor decompositions that mine $\mathcal{D}$ in its original "user", "object" and "time" coordinates as the three tensor modes [90, 143]. These techniques are meant to capture synchronous user behavior. In our results, we show how our data representation allows us to capture the asynchronous but similar behavior patterns that emerge when we have a mixed population of users, spread across different time zones and with different temporal activity patterns.

A-FLUX defines two complementary models as shown in Figure 8.1: an inter-artist attention flow model that captures the transitions between different artists, represented by the values in matrix $\mathbf{X}_u$ for $s \neq d$, and a fixation model, which captures the intra-artist transitions ($s = d$). We describe each model next.

8.3.2 Inter-Artist Attention Flow Model

The goal of our inter-artist attention flow model is to capture the latent transitions between different artists based on the listening habits of users. Towards that goal, we experimented with various non-negative tensor factorization techniques, such as PARAFAC [116]. In the end we opted for a Bayesian approach to decompose the tensor, since, among other things, has been shown to be more scalable than tensor factorization [90] and has a probabilistic interpretation.

In our model, the latent space $\mathcal{Z}$ defines a set of *transitions* between pairs of artists s and d, and each latent factor z in this space defines a transition pattern shared by a group of users. We refer to each latent factor z as a *attention flow gene*, and the collection of genes as a *genome*. These terms are inspired by the "Music Genome Project", a proprietary approach that aims to capture the individual preferences of users

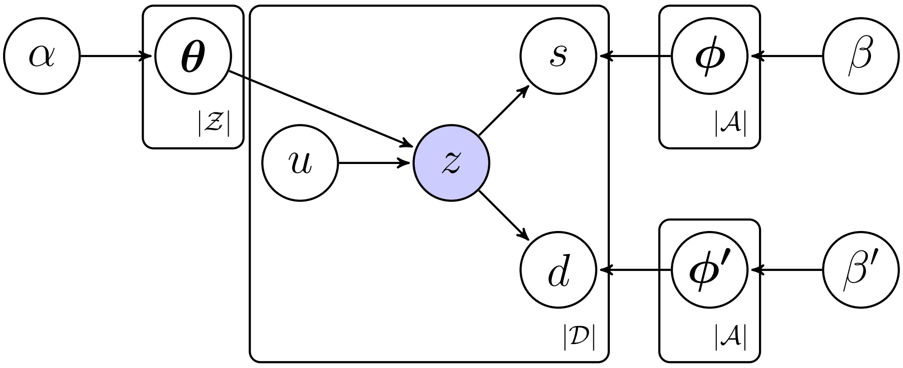


Figure 8.2: The Inter-Artist Attention Flow Model.

using musical information retrieval techniques[10]. Also, the notion of genes determining user behavior has been used before in the social media context [11].

In the following, we use the "·" notation to imply a sum over a given dimension (e.g., $n_{\cdot sd} = \sum_{u \in \mathcal{U}} n_{usd}$).

8.3.2.1 Model Derivation

To model inter-artist transitions, we define $\boldsymbol{\mathcal{X}}^- = \boldsymbol{\mathcal{X}} - diagonals(\boldsymbol{\mathcal{X}})$ by removing the cases where $s = d$ from $\boldsymbol{\mathcal{X}}$, since this behavior is captured by the Fixation model. We aim at producing a decomposed view of $\boldsymbol{\mathcal{X}}$ by estimating the tensor $\boldsymbol{\mathcal{P}}^- = \left[\mathbf{P}_1^-, \cdots, \mathbf{P}_{|\mathcal{U}|}^-\right]$, where $\mathbf{P}_u^-$ is the transition matrix for user u. We then estimate the inter-artist attention flow probabilities by aggregating the behavior of every user into the transition matrix $\mathbf{P}^-$. The graphical model used to define the intra-artists model is shown in Figure 8.2. We now describe this model.

To estimate $\mathbf{P}^-$ we decompose $\boldsymbol{\mathcal{X}}^-$ to a latent space (or genome) $\mathcal{Z}$. Parameter $k = |\mathcal{Z}|$ is an input variable determining the number of genes (or latent factors) to be estimated. The three other inputs are the hyper-parameters α, β_s, and β_d. The outputs of our model are three matrices, $\mathbf{\Theta}$, $\mathbf{\Phi}_s$, and $\mathbf{\Phi}_d$, as well as a vector $\mathbf{z}$. $\mathbf{\Theta}$ has $|\mathcal{U}|$ rows and $|\mathcal{Z}|$ columns, where each cell contains the probability of a user u generating a given gene z, $p(z|u)$, i.e:

$$p(z|u) = \mathbf{\Theta}(u, z) = \theta_{z|u}(z) = \frac{n_{zu} + \alpha}{n_{\cdot u} + |\mathcal{Z}|\alpha} \tag{8.2}$$

where n_{zu} is the number of times the user activated a transition (s, d) because of gene z, and is estimated from the data (see next section). Both matrices $\mathbf{\Phi}_s$, and $\mathbf{\Phi}_d$ have $|\mathcal{Z}|$ rows and $|\mathcal{A}|$ columns, and contain the probabilities of the source s and destination

[10] http://www.pandora.com/about/mgp

d, respectively, given the gene z. That is:

$$p(s|z) = \mathbf{\Phi}_s(z, s) = \phi_{s|z}(s) = \frac{n_{sz} + \beta_s}{n_{\cdot z} + |\mathcal{A}|\beta_s} \tag{8.3}$$

$$p(d|z) = \mathbf{\Phi}_d(z, d) = \phi_{d|z}(s) = \frac{n_{dz} + \beta_d}{n_{\cdot z} + |\mathcal{A}|\beta_d}. \tag{8.4}$$

where, once again, n_{sz} and n_{dz} are estimated from the data. Finally, vector $\mathbf{z}$ contains the probabilities of each gene $z \in \mathcal{Z}$, referred to as $p(z)$, which are also estimated from the data. When learning the model, we can define $p(z) \propto n_z$.

Given Equations 8.2-8.4 and the graphical model in Figure 8.1, we can describe the generative process for the tensor $\boldsymbol{\mathcal{X}}^-$ as:

1. For a given user u:
   a) Sample $\theta_{z|u} \sim Dirichlet(\cdot \mid \alpha)$
2. For each (s, d):
   a) Draw a latent topic from $z \sim Multinomial(\theta_{z|u})$
   b) Draw a source s from $s \sim Multinomial(\phi_{s|z})$
   c) Draw a destination d from $d \sim Multinomial(\phi_{d|z})$.

This process captures the notion that users trigger a given transition (s, d) when they are interested in gene z. Iterative sampling from this process can generate an estimate of $\boldsymbol{\mathcal{X}}^-$. However, since we are interested in the artist-to-artist transition matrices, we can define $\mathbf{P}^-(s, d)$ as:

$$\mathbf{P}^-(s, d) = \sum_{z \in |\mathcal{Z}|} p(z|s)p(d|z) \tag{8.5}$$

where $p(z|s) \propto p(s|z)p(z)$. We can also capture the individual user transition matrices as:

$$\mathbf{P}_u^-(s, d) = \frac{\sum_{z \in |\mathcal{Z}|} p(z|u)p(s|z)p(d|z)}{\sum_{z \in |\mathcal{Z}|} p(z|u)p(s|z)}. \tag{8.6}$$

Gibbs Sampling We use a collapsed Gibbs sampler [58] to estimate matrices $\mathbf{\Theta}$, $\mathbf{\Phi}_s$,$\mathbf{\Phi}_d$, and vector $\mathbf{z}$. That is, we sample from the posterior defined by the product $\theta_{z|u}\phi_{s|z}\phi_{d|z}$. We fix hyper-parameters $\alpha = \frac{50}{|\mathcal{Z}|}$, and $\beta_s = \beta_d = 0.001$, although similar results were produced with other values as well. We execute the sampler for 800 iterations with 300 being discarded as burn-in.

MDL Cost As in Chapter 7, we apply the minimum description length (MDL) principle [60] to determine the number of genes $k = |\mathcal{Z}|$. As discussed, MDL captures how good a model $\mathcal{M}$ ($\mathbf{P}^-$ in our case) represents the data. This is done by taking into account the trade-off between the "goodness" (or likelihood) and the complexity (or generality) of the model. MDL is strongly tied to our arguments of not using maximum likelihood estimates since we want a good, but also more general, recovery of matrix $\mathbf{P}^-$.

To apply MDL we first define the likelihood of the data given the model $\mathcal{M}$. Given $n_{sd} = n_{\cdot sd}$ the number of transitions from s to d by all users, the log likelihood of matrix $\mathbf{P}^-$ is given by $\sum_{s,d|s \neq d} n_{sd} log(p(d|s))$[11]. The MDL cost of model $\mathcal{M}$ is given by the sum: $Cost(\mathbf{P}^- \mid \mathcal{M}) + Cost(\mathcal{M})$.

$Cost(\mathbf{P}^- \mid \mathcal{M})$, defined as the negative log-likelihood, captures the likelihood of the data given the model: lower-values imply on better (but less general) recoveries of $\mathbf{P}^-$. $Cost(\mathcal{M})$ captures the complexity of the model as:

$$\begin{aligned} Cost(\mathcal{M}) = log^*(|\mathcal{A}|) + log^*(|\mathcal{Z}|) + \sum_{s,d,z} [log^*(\lceil p(d|z)n_{\cdot\cdot} \rceil) \\ + log^*(\lceil p(s|z)n_{\cdot\cdot} \rceil) + log^*(\lceil p(z)n_{\cdot\cdot} \rceil)] \end{aligned}$$

where $log*$ is the universal coding cost (number of bits) for integers [60]. $Cost(\mathcal{M})$ represents the cost of coding each matrix in the model in integer representation with precision $n_{\cdot\cdot}$ (the total number of transitions)[12].

8.3.3 Fixation Model

Users' bursty repeated consumption of artists requires modeling this behavior with a stochastic process that has memory. Markov modulated processes are a class of models that are particularly versatile for this task [120]. Our goal here is not only to model user behavior but also, through the use of intuitive parameters, understand how users repeatedly consume artists. Most importantly, we want to reproduce user attention giving rise to both exponential and power law distributions observed in our datasets.

Our fixation model, which captures the intra-artist transitions, is a Markov modulated process where we use an infinite number of states, an approach widely used to model systems with bursty behavior [120]. Figure 8.3 illustrates our model (only the initial states). The "start" circle represents the initial transition from the Inter-artist model. From state zero we are interested in how long it takes to exit from the

[11] The likelihood is the product of $p(d|s)$ for all n_{sd} transitions [35].

[12] Since we deal with counts of events, the smallest probability value is $(1/n_{\cdot\cdot})$.

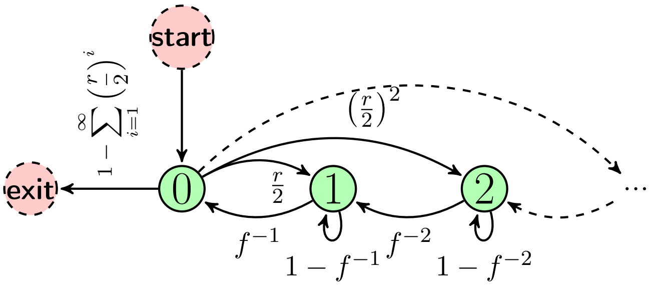

start
exit
$1 - \sum_{i=1}^{\infty} \left(\frac{r}{2}\right)^i$
$\left(\frac{r}{2}\right)^2$
0
$\frac{r}{2}$
1
2
...
f^{-1}
$1 - f^{-1}$
f^{-2}
$1 - f^{-2}$

Figure 8.3: Infinite Markov modulated Attention Model Showing the Transitions Between the First Three States.

"exit" transition. Thus, circles "start" and "exit" in Figure 8.3 are not states but rather entrance (exit) transitions from (to) the inter-artist model. The states of the model capture the affinity of the user for the artist, that is how much the user is willing to repeatedly listen the artist's songs. There is a fixed residency time Δt on each state. Thus, higher states represent that the user has a higher affinity and thus dedicates more play-time to the artist.

The model has parameters $0<r<1$ and $1\leq f<2/r$[13]. Parameter r models the user "rush", capturing how users get excited as they hear their "favorite" artist song and decide to hear more (e.g., an entire album). Parameter f models user "fixation", representing how long it takes for users to get over their initial impression, which is also a function of the artist's song inventory size. A large value of f implies that users quickly get over their initial impression or happens because the artist has just a few songs. For example, a one-hit wonder is expected to have large values of r and f.

We can fit the Fixation model, varying r and f, to the complementary cumulative distribution function (CCDF) of the time users dedicate to an artist using the Levenberg-Marquardt algorithm. The CCDF will define the probability of the residency time in the chain. The infinite number of states can be captured by using a sufficient number of states (100 in our datasets). We evaluate the algorithm on the mean squared error of the real data and the residency times generated by the model. As we shall show empirically in our results, this model is capable of generating both power-law and exponential residency times as also discussed in [120].

8.4 Results

In the previous section we introduced A-FLUX. In this section we focus on understanding the user attention flows on our datasets using A-FLUX, aiming at finding

[13] The limit of $2/r$ is required as described in the original paper of the model [120]

evidence that can shed some light into the questions: *Are OMSSs zero-sum attention markets? Which artists cooperate and which compete for attention?* However, before delving into these questions, we present the challenges of attention flow mining and how A-FLUX meets these challenges. We defer the validation of A-FLUX to the next section.

8.4.1 Applying A-FLUX to the Datasets

We first separate the inter-artist tensor $\boldsymbol{\mathcal{X}}$ from the intra-artist repeated consumption transitions. We run the inter-artist attention flow model of A-FLUX on $\boldsymbol{\mathcal{X}}^-$, and the fixation model of A-FLUX on the intra-artist transitions. We focus on the artists which had at least five plays by five users. In total, the Last.FM-Groups dataset has over 3M plays of such artists, while Last.FM-1k has roughly 176k plays. We note that even after filtering the data, there is still a significant number of rare transitions as 44% of the inter-artist transitions happen less than ten times.

For the inter-artist attention flow model we decide the number of genes (latent factors) k using the MDL-based criteria described previously, searching in the range $k \in [2, 400]$[14]. We found that, as we increase k, the MDL cost first decreases and then rapidly increases, reaching global minimum at k=40. Thus, we experiment with our model using a genome with 40 genes in both datasets.

We first illustrate the power of A-FLUX by showing in Table 8.2 four different genes (latent factors) extracted from the Last.FM-Groups dataset. For each gene, the table shows the top 7 source s and destination d artists. We also selected the top 50 users which have attention flow transitions within each gene. The table also shows how those users are distributed across different nationalities, considering only those who self-declared this information in their Last.FM webpages, as well as statistics of their ages. Discovered genes have a strong tendency to keep user attention flows within their source and destination artists: even for the users with the most diverse musical tastes in our largest dataset. That is, we counted the number of transitions within the same gene by considering the transitions where the gene z has the highest value of $p(z|s)$ and $p(d|z)$. We found that, 96% of all of their inter-artist transitions are within the same gene.

We cross-referenced the top artists in each gene with the AllMusic guide[15], finding that the genes automatically discovered by the algorithm are semantically meaningful. For example, gene $z = 18$ is predominately formed by female pop/rock singers as

[14] We searched $k \in \{2, 4, 8, 10, 20, 30, 40, 50, 100, 200, 300, 400\}$.

[15] http://www.allmusic.com/

	Gene=18 (BR/US pop artists)		Gene=20 (metal artists)		Gene=23 (electronic dance artists)		Gene=39 (Pop artists)	
	Sources	Destinations	Sources	Destinations	Sources	Destinations	Sources	Destinations
Artists	Britney Spears	Britney Spears	Nightwish	Nightwish	Daft Punk	Daft Punk	Britney Spears	Britney Spears
	Wanessa	Wanessa	Within Temptation	Within Temptation	David Guetta	Deadmau5	Madonna	Christina Aguilera
	Christina Aguilera	Christina Aguilera	Evanescence	Evanescence	Skrillex	Skrillex	Christina Aguilera	Madonna
	t.A.T.u.	t.A.T.u.	Epica	Epica	Deadmau5	David Guetta	Rihanna	Rihanna
	Katy Perry	Katy Perry	Korn	Korn	The Prodigy	The Prodigy	Lady Gaga	Lady Gaga
	Claudia Leitte	Claudia Leitte	Disturbed	Disturbed	Tiesto	Pendulum	Katy Perry	Katy Perry
	Lady Gaga	Pitty	Marilyn Manson	Marilyn Manson	Pendulum	Tiesto	Kesha	Kesha
Users	BR=98%, NL=2% Age: 19; 21;24 (min 16, max 34)		DE = 18%, PL = 16%, US = 12% FI = 8%, UK = 6% Age: 21; 24; 29 (min 13; max 60)		US = 18%, BR = 10%, PL = 10%, UK = 10% Age: 20; 22; 25 (min 17; max 33)		BR=78%, US=10%, PL=5% Age: 19; 22; 25 (min 16; max 41)	

Table 8.2: Genes Extracted From the Last.FM-Groups Dataset. Top source and destination artists, and demographics of top-50 users. Sources, destinations, and users are sorted by probabilities $p(s|z)$, $p(d|z), p(u|z)$. BR = Brazil, US = USA, NL = Netherlands, DE = Germany, PL = Poland, FI = Finland. Age statistics: 1^{st}, 2^{nd} and 3^{rd} quartiles, minimum and maximum values.

both sources and destinations. This is not the only gene with similar pop singers, as exemplified by gene $z = 39$. Yet, the presence of Brazilian pop artists (e.g., *Wanessa*, *Claudia Leitte*, and *Pitty*) in gene 18 explains why the vast majority (98%) of the top users in this gene are Brazilians (BR). Gene $z = 20$ in turn is mostly focused on different sub-genres of metal (e.g., goth-metal and rap-metal). A large fraction of the top-50 users of the "heavy metal" gene are from Germany and Poland. Finally, gene $z = 23$ represents users of different nationalities (American being the most frequent one) who like to listen to electronic dance music, often transitioning between different artists of that genre. *Note that even in a dataset mostly comprised of pop artists fans (Last.FM-Groups),* A-FLUX *is able to extract the attention flows of heavy metal and electronic music fans.*

8.4.2 User Attention Evolution and Gene Persistence

As we have discussed so far, A-FLUX breaks artist and users into distinct attention flow patterns. Anecdotally, many of us have had the experience in which we are browsing our music collection and come upon a long forgotten track belonging to a "class of artists" (attention gene) that we used to listen together but we have not listened them in a while. Listening to this forgotten classic gets us hooked on those artists again, changing our attention flow. The rediscovery of *attention gene* may happen less serendipitously, through online ads, media exposure, album releases, or even in response of recommendations by the OMSS. Yet, A-FLUX is designed to cope with this type of rediscovery behavior by mining attention persistence in attention genes.

To illustrate how A-FLUX captures this rediscovery and stickiness behavior of genes, Figure 8.4 shows the time series of a user broken down into the user's four top attention genes. The y-axis is the cumulative sum of the attention paid to the four top attention genes. The attention user u pays to gene g is obtained through: $A(g, u, t) = \sum_s m_s(u, t)\mathbf{1}(g = \arg\max_z p(z|s, u))$, where the value $m_s(u, t)$ is the number of plays of artist s by u during month t.

As shown in the figure, from 2010 until mid 2011 the user goes through a strong Pop music phase – most representative (top) artists in the gene labeled "U.S. Pop (1)" are *Madonna, Nelly Furtado,* and *Alicia Keys*, and top artists in the "U.S. Pop (2)" gene are *Britney Spears, Leona Lewis,* and *Kelly Clarkson.* We also note some interest in 70-80's Rock overtones – top artists being *Queen, Michael Jackson,* and *The Beatles.* After mid 2011 the user moves away from Pop artists towards a "Classic Rock" gene, with *The Beatles, Pink Floyd,* and *Nirvana* as top artists.

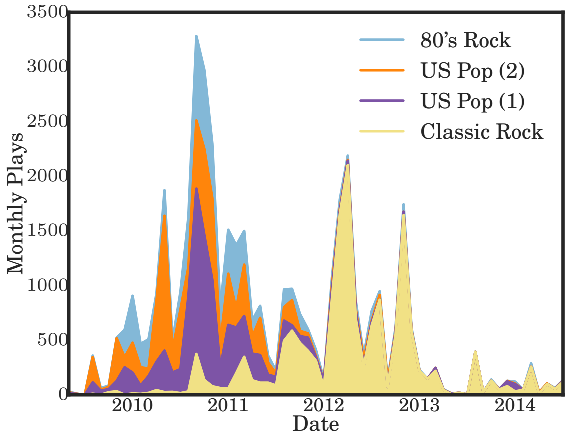


Figure 8.4: Time Series of Monthly Plays of a User Broken Down by Attention Genes. User attention is too complex for user-artist latent factor analysis.

Factor models that rely on user-artist latent factorization – such as SVD and LDA – would consider the user illustrated in Figure 8.4 to have a diverse musical taste. Later on, we perform a more in-depth analysis of the advantages of A-FLUX over this user-artist factor analysis. But, based on the example in Figure 8.4, we see that the attention flows of a user shows the user not as diverse but rather as evolving.

In fact, our data shows that intra-gene attention flows are strong. For instance, we looked at the intra-gene attention flows – removing repeated artist consumption – of the 15 users with the highest attention gene entropy and over 60,000 played tracks. While these users could be thought of as having "diverse tastes", 96% of their inter-artist transitions are to artists within the same gene. The user shown in Figure 8.4 has the same behavior. Given that attention flows tend to stay within the same gene, and that A-FLUX is able to correctly capture these flows, in what follows we use A-FLUX to find evidence of competition and collaboration effects that a new artist release has on the attention genome.

8.4.3 Attention Elasticity and Competition

The question now to answer is whether *a boost in attention of one artist means a boost or a decrease in attention of other artists.* If attention in OSMSs is inelastic, a zero-sum, then any increase in attention of an artist necessarily means decrease in attention of other artists. If, however, OSMS attention is elastic, an increase in attention of an artist may also boost the attention another artists without decreasing anyone's attention. One of the challenges of this analysis is the existence of external attention shocks.

External attention shocks – such as album releases and concert tours – are one of the main challenges to mining inter-gene and intra-genre attention flows. A sudden increase of attention to an artist may be due to an external shock rather than attention flows from other genes. To measure an artist's potential influence over the attention flows of a gene we take two precautions:

(a) We explicitly take album releases into account, looking at how an album release of an artist affects the attention dedicated to all genes over a two-month time window. We focus on genes rather than specific artists to avoid false correlations and variabilities that come from analyzing two sample points.

(b) We aggregate the cumulative effects of attention over all releases of the same artist to ensure that increases or decreases of attention of a gene are more robust to statistical variability. More established artists often have five or more releases. Aggregating their effects helps filtering out noise.

Our precautions are only aimed at cleaning up spurious correlation effects rather than ascertain causation. For example, once an artist releases a new album users that have not listened to songs of that artists in the last 60 days increase their attention to the artist in average 1.66 hours over 60 days when compared to similar users before the album release. For users that already listen to the releasing artist, some increase their attention, others decrease attention. Overall the 30% of the regulars actually decrease their attention, however the overall effect is an increase of 4 hours in average over 60 days. But despite the obvious attention increase towards the artist that releases a new album, measuring attention flows to other artists proved to be a challenge.

Measuring Attention Elasticity and Competition.

Mindful of these challenges, we quantify the impact a source artist s on gene z at day t over a subset of artists $\mathcal{B} \subseteq \mathcal{A}$ by counting the total attention given to gene z at time t to all artists in $\mathcal{B}$, weighted by the probability that s belongs to gene z:

$$A_{\mathcal{B}}(s,z,t) = \sum_{d \in \mathcal{B}} \sum_{u \in \mathcal{U}} p(z|s) p(d|z) F_t(d,u) \tag{8.7}$$

where $p(z|s)$ is the A-FLUX probability of a transition from source s to gene z, and $p(d|z)$ the probability of a transition from gene z to destination d. F_t captures the total attention (or fixation) time user u dedicates to artist d on day t. We estimate the fixation time dedicated to an artist d by summing up the time intervals between consecutive plays of d. To account for pauses or user logouts, we set the maximum playtime of a song to 6 minutes[16].

The total attention dedicated exclusively to artist s on gene z at day t is $A_{\{s\}}(s, z, t)$. The total attention dedicated to all artists other than s is $A_{\mathcal{A}\setminus\{s\}}(s, z, t)$. Figure 8.5 compares the evolution of daily gene attention without an artist $A_{\mathcal{A}\setminus\{s\}}(s, z, t)$ (line with squares) and with the artist $A_{\mathcal{A}}(s, z, t)$ (line with circles) for artist $s \in \{$*Lana Del Rey*, *Katy Perry*, *Miley Cyrus*$\}$. In what follows we look into the genes with the largest overall change in attention[17] following the largest album release (captured by the shock size) by each artist (shown as a vertical line in the graph).

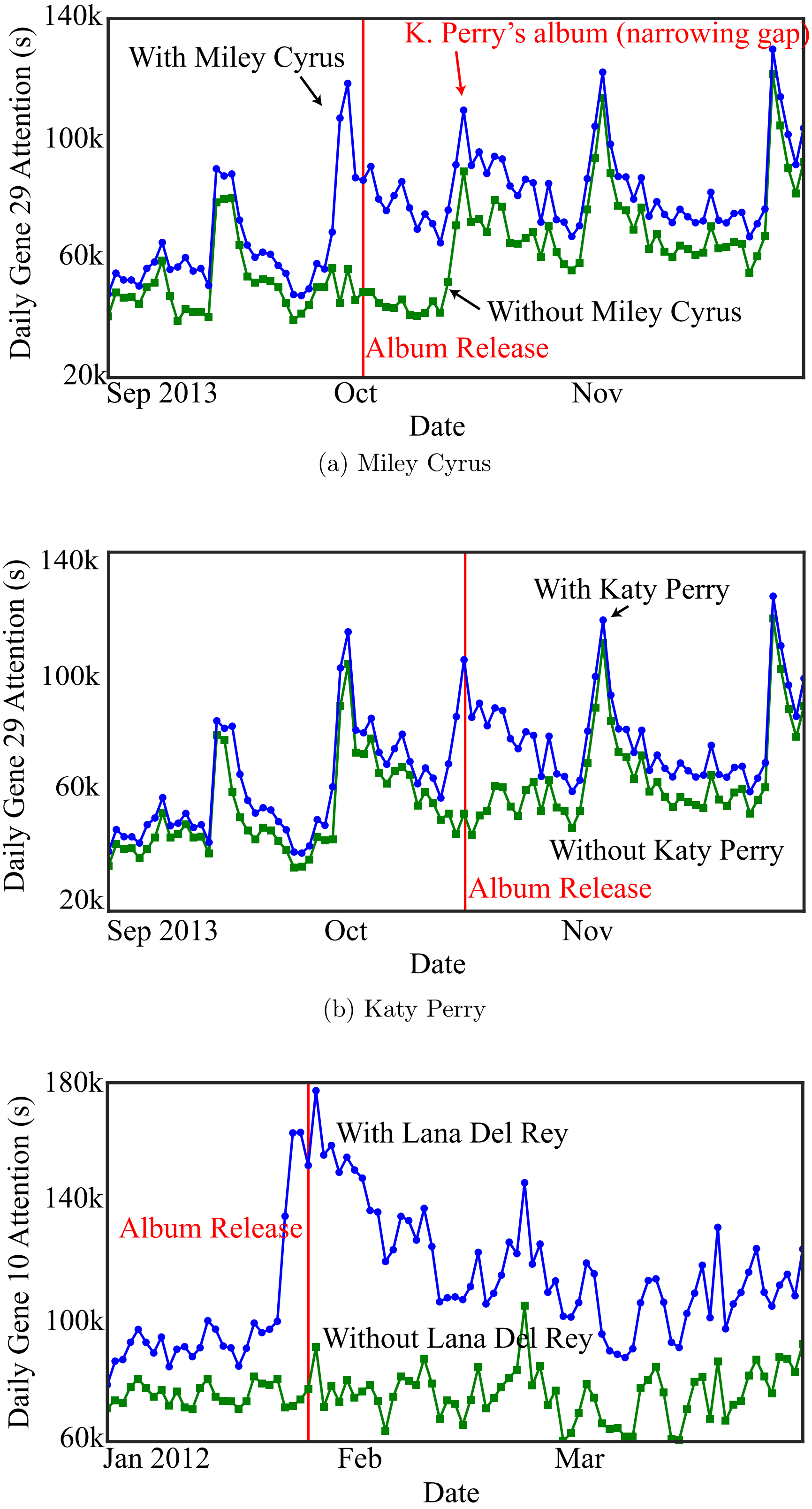
Figure 8.5(a) shows the daily attention towards gene 29 (that includes artists such as *Madonna*, *Avril Lavigne*, and *Katy Perry*) before and after *Miley Cyrus*'s largest album release in our dataset. We observe that the sudden spike of attention towards *Cyrus* does not impact the overall attention towards other artists on gene 29. This is an evidence of attention elasticity, where users dedicate more overall attention to *Cyrus* without decreasing their overall attention to other artists on gene 29. Note that *Cyrus*'s attention gap soon evaporates after artist *Katy Perry* releases a new album on October 18. Figure 8.5(b) shows the evolution of the attention at gene 29 before and after *Katy Perry*'s album release. Note that *Perry*'s new album also has little impact on the overall attention on gene 29 except for the attention towards *Miley Cyrus*, which shows a significant reduction (evidenced by the narrowing gap in Figure 8.5(a)). This suggests that competition between *Miley Cyrus* and *Katy Perry* happens at the "elastic" attention space.

Generally, without competing releases, attention is elastic and tends to be long lasting. *Lana Del Rey* is a good example. Figure 8.5(c) shows the daily attention towards gene 10 (that includes artists such as *Lady Gaga* and *Britney Spears*) before and after artist *Lana Del Rey*'s largest album release in our dataset. Note that *Del Rey* has a lasting increase in attention (comparing the steady state of the time series before and after the release), overall slightly reducing the average attention towards other artists on gene 10. Other pop artists such as *Kanye West*, *Britney Spears*, and *Rihanna* also display similar attention elasticity, though not lasting as long.

So far we have looked at single genes and single album releases. In what follows we look into the aggregate attention flows of all genes over all album releases.

[16] According to MillionMusic, only 12% of the songs have duration exceeding 6 minutes.

[17] We identified these artists manually, later we discuss our Equation 8.8 how to measure competition



(a) Miley Cyrus

(b) Katy Perry

(c) Lana Del Rey

Figure 8.5: Evolution of Gene Activity Showing Attention Elasticity and Artist Competition.

Overall Attention Flows.

Next we define the aggregate the cumulative effects of attention over *all* releases of the same artist to ensure that increases or decreases of attention of a gene are more robust to statistical variability. Established artists often have five or more releases. Aggregating their effects helps filtering out noise. Through function A we can define the total attention change score in an interval of time of size 2δ as follows. Let $\mathcal{R}_s$ be the set of all releases available in our datasets from artist s. The total attention change score is

$$CnC(s,z) = \frac{1}{|\mathcal{R}_s|} \sum_{r_s \in R_s} \left(\sum_{t'=t_{r_s}}^{t_{r_s}+\delta} A_{\mathcal{A}\setminus\{s\}}(s,z,t) - \sum_{t'=t_{r_s}-\delta-1}^{t_{r_s}-1} A_{\mathcal{A}\setminus\{s\}}(s,z,t') \right), \tag{8.8}$$

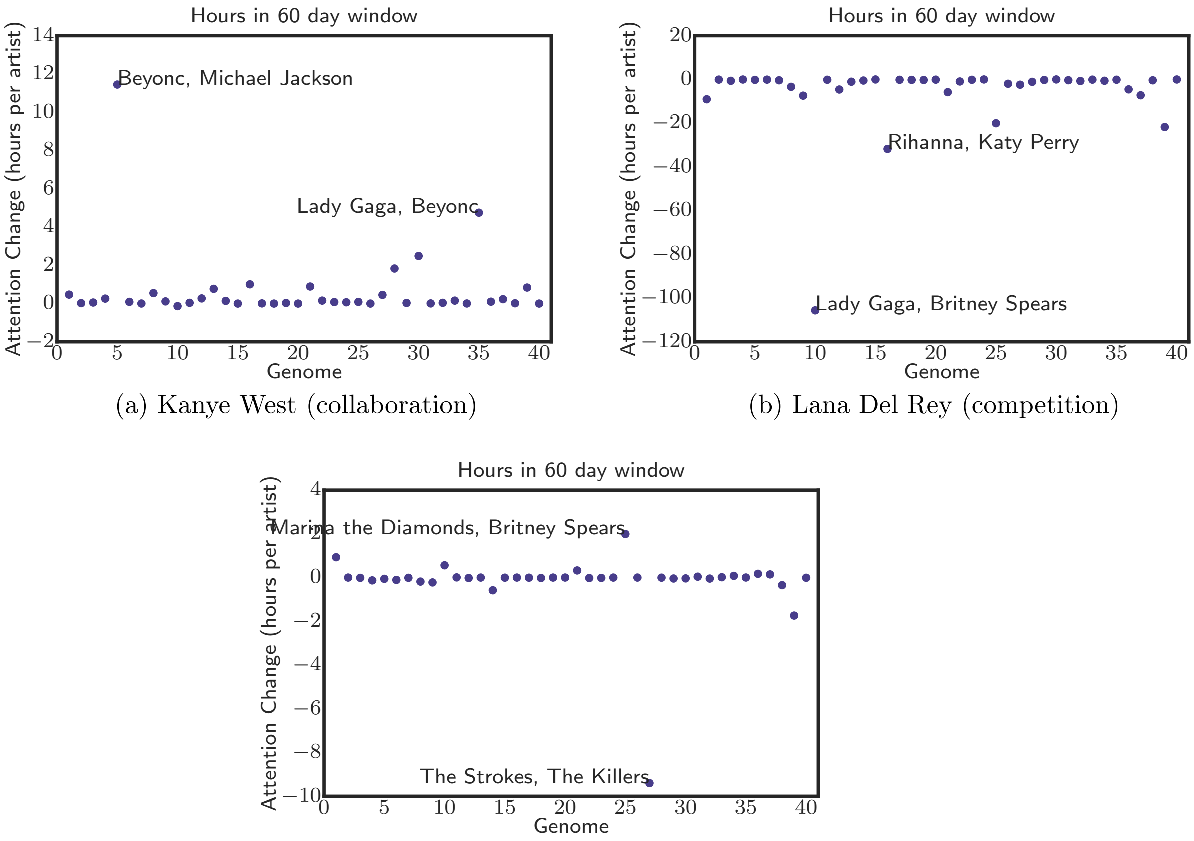
where t_{r_s} is the time of release r_s and δ is an observation window before and after the release (60 days in our analysis). $A_{\mathcal{A}}$ is defined in Equation 8.7.

The CnC score captures the change in the attention flows from source s to destination d in gene z (for all artists in $|\mathcal{A}|$, see Equation 8.7.) after the release r_s, with respect to the group of users represented by z. A positive score provides evidence of a collaboration correlation between s and the collection of artists in z. A negative score points to a possible competition, as artist s may be stealing attention away from the collection of artists in z. Once again, we cannot claim any causation relationship, but rather provide evidence that such interactions might be happening.

We note however that this computation might be affected by external events, such as two artists with releases around the same time or overall increased attention towards a gene. In these case, a positive score could be mistakenly taken as a collaboration. However, as a gene contains all artists with different weights, the effect of a single artist is somewhat limited due to averaging.

We computed the average CnC over all genes $z \in Z$ for 11 artists with documented releases in our datasets. For conciseness we restrict our exposition to the three artists shown in Figure 8.6. Each figure shows the CnC score (y-axis) and the attention gene (x-axis). For some genes that experienced a large change in attention, we also point out the (destination) artists that were most affected, weighted by their gene weight ($p(d|z)$).

Evidence of Attention Collaborations: Kanye West (Figure 8.6(a)) shows elasticity in user attention, as the artist seems to collaborate with other artists by boosting attention of many genes, most markably gene 5. This gene is dominated by attention flows between R&B and pop/rock artists, such as *Beyonce* and *Michael Jackson*, consisting mostly of self-declared U.S. users.



(a) Kanye West (collaboration)

(b) Lana Del Rey (competition)

(c) Arctic Monkeys (mixed)

Figure 8.6: Evidence of competition and cooperation between attention genes.

Evidence of Attention Competitions: Of all 11 artists analyzed, *Lana Del Rey* (Figure 8.6(b)) is the only one that brings no attention elasticity, but rather competition with some genes dominated by female pop singers (genes 10 and 16). *Lana Del Rey* is a new female singer classified as Pop/Rock and Alternative/Pop. A possible reason for this competition is that *Lana Del Rey* is the only non-established artist of all 11 artists analyzed. *Del Rey*'s growth seems to come at the attention cost of similar but more established artists.

Mixed Cooperation/Competition: Arctic Monkeys (Figure 8.6(c)) has a mixture of cooperation with gene 24, with similar performers, and competition with gene 26, where the most affected artists are *The Strokes* and *The Killers*.

We note that the decomposition of A-FLUX shows the same artist with distinct attention flow patterns in distinct genes. For instance, take an international artist like *Britney Spears*. Let's define that a user belongs to gene group g if g is the user's strongest gene, $g = \arg\max_{g'} p(z = g'|u)$, and then rank users according to $p(z = g|u)$. We then see that *Britney Spears* is the preferred artist of multiple gene groups.

Table 8.2 shows two such user gene groups, namely 18 and 39. Note that the groups have similar user demographics and top artists. Yet, these similarities are superficial as A-FLUX uncovers a large difference in the attention flows of both groups. This difference becomes apparent when we look at the increase in attention right after *Britney Spears* releases an album. Despite having *Britney Spears* as the top artist in the group, gene 18 sees a relatively small increase (18%) in the total attention flow to the gene after the release, while attention flows to gene 39 sees a much larger increase (45%).

In sum, using A-FLUX we were able to uncover evidence that user attention can be elastic and that artists seem to compete and collaborate for user attention. Thus, OMSSs are *not* necessarily zero-sum attention markets and benefit from new album releases rather than being indifferent to album releases if attention was inelastic. Yet, we also found evidence of specific artists that compete for user attention, as a release by one of them steals attention of genes with similar artists.

8.5 Model Validation

In this section we validate both fixation and inter-artist attention flow models. In some cases we restrict the presentation to only the larger Last.FM-Groups dataset, although results are qualitatively similar on both sets of data.

8.5.1 Inter-Artist Attention Flow Model

One key aspect of our inter-artist attention flow model is the representation of data based on a user-artist-artist tensor. Next, we compare it against alternative data representations.

Why not decompose the time dimension?

Other tensor decompositions that take time as a tensor mode are designed to capture synchronous behavior. Yet, we found that the average difference between the registration dates of the top 50 users in each gene is 339 days. This result serves as evidence that even though such users have similar attention flow patterns, their overall behavior, may not be synchronous.

To provide further evidence in support of our method, we show that unlike temporal tensor approaches, our model is able to more accurately recover the musical preferences of each user. Since a ground truth of user preferences is not available, we built

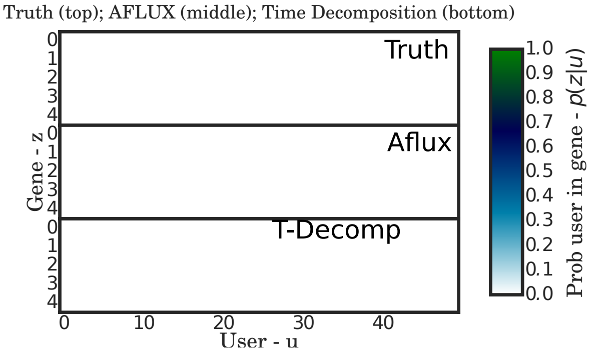


Figure 8.7: Comparison With Temporal Tensors (Truth, A-FLUX, Time Decomposition).

	Last.FM-Groups			Last.FM-1k		
	A-FLUX Att. Flow	Play Count	Trans. Count	A-FLUX Att. Flow	Play Count	Trans. Count
MDL	1.6×10^8	2×10^8	3×10^8	6×10^6	9×10^6	15×10^6
DKL	2.56	4.13	3.78	1.72	3.00	2.77

Table 8.3: Comparison With Other Data Representations.

a synthetic dataset with 50 users, split into 5 homogeneous groups with pre-defined distributions of musical preferences (artist popularity)[18]. For each user, we produced a sequence of plays with exponentially distributed inter-play times. For each play, we sampled an artist from the preference distribution of the user group with probability 0.99. We sampled it from a different distribution with probability 0.01 (to introduce some noise). We simulated a total of 5 days, and each user performs 100 plays per day. We simulated users joining the system at 1 or 2 days apart from each other by starting their sequences of plays at different times.

We applied A-FLUX and a temporal decomposition model [90] on our synthetic dataset, aiming at recovering the preference groups. Figure 8.7 plots the matrices $\mathbf{\Theta}$ produced by both methods and the ground truth (i.e, user ids $[0, 10)$ in group 0, $[10, 19)$ in group 1, etc.). A-FLUX (middle) is able to recover the different user groups (blocks in the matrix), despite the asynchronous behavior of users within each group. In contrast, the temporal decomposition (bottom) fails, as it mixes users of different groups together.

What about other data representations?

We also compare our approach with two models based on other data representations. In the first one, LDA is used to learn a latent space where each factor defines a topic

[18] We modeled the artist popularity of each group using a different Lognormal distribution.

of artists based on the user play counts to the artist. Each user is characterized by a set of topics (e.g., musical preferences). Making a parallel to the term-document representation of LDA, documents are mapped to users, and terms are mapped to artists. We then train LDA to learn the probabilities $p(a|z)$ and $p(z|u)$. Although the inter-artist transition is not explicitly captured, it can be estimated as $\mathbf{P}^{-}(s,d) = p(z|a=s)p(a=d|z)$. We refer to this approach as Play-Count.

The second strategy consists of representing each transition between a pair of artists separately. LDA is then used to learn a latent probability space that defines $p(x_{[s,d]}|z)$, where $x_{[s,d]}$ is a random variable capturing the number of times a transition between artist s and artist d occurred. To define $p(s|z)$, we take the average of $p(x_{[s,d]}|z)$ by fixing s and varying d. We use a similar heuristic to define $p(d|z)$[19]. We refer to this strategy as Transition-Count.

We compare A-FLUX with Play-Count and Transition-Count in terms of their ability to reconstruct transition matrix $\mathbf{P}^{-}$. Table 8.3 shows the MDL scores and the average Kullback-Leibler divergence[20] between the rows of the original transition matrix and the recovered one (lower is better for both metrics) for all models and datasets. A-FLUX is superior to the two other approaches in both datasets, and in terms of both metrics. Play-Count does not capture the inter-artist transition patterns, while Transition-Count, unlike A-FLUX, models transitions between specific pairs of artists separately, thus being more susceptible to sparsity issues.

8.5.2 Fixation Model

We validate our fixation model by showing how it fits the time users spend listening to different artists on any given day (referred to as daily fixation time). Figure 8.8 shows the fitted and empirical complementary cumulative distribution functions (CCDF) of the daily fixation time for two particular example artists, namely *Radiohead*, and *T.I. feat. Justin Timberlake* (a collaboration between two artists). This example was extracted from the Last.FM-Groups dataset.

The distribution for *Radiohead* clearly has long tails, and is similar to the distributions for most artists. In contrast, the distribution for the *T.I. feat. Justin Timberlake* collaboration has a much shorter tail, approaching an exponential distribution. Unlike for the other artists, there is only one song by this artist collaboration in our dataset, which might explain why users tend to spend less time listening to them. Yet, our

[19] We tried other heuristics, such as filtering very low/high values before computing the mean, but the results were similar.

[20] Kullback-Leibler for probability distributions p and q is defined as: $DKL(p,q) = \sum_x p(x)log(\frac{p(x)}{q(x)})$

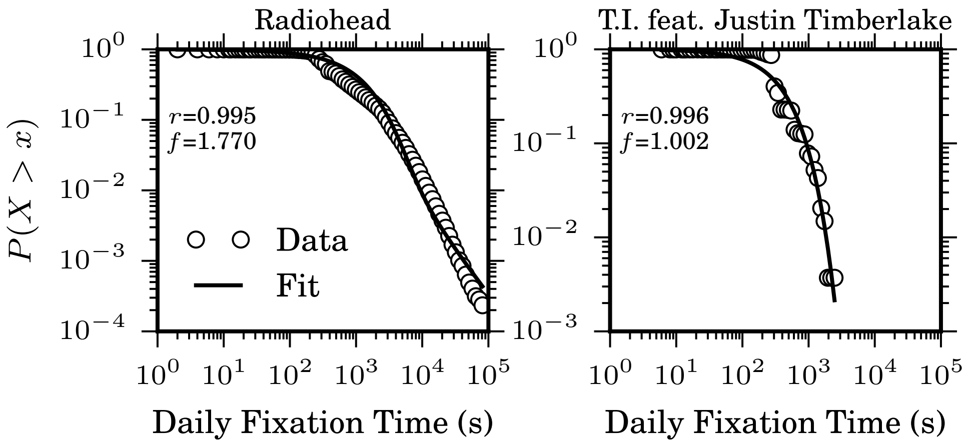


Figure 8.8: Validation of Fixation Model for Example Artists.

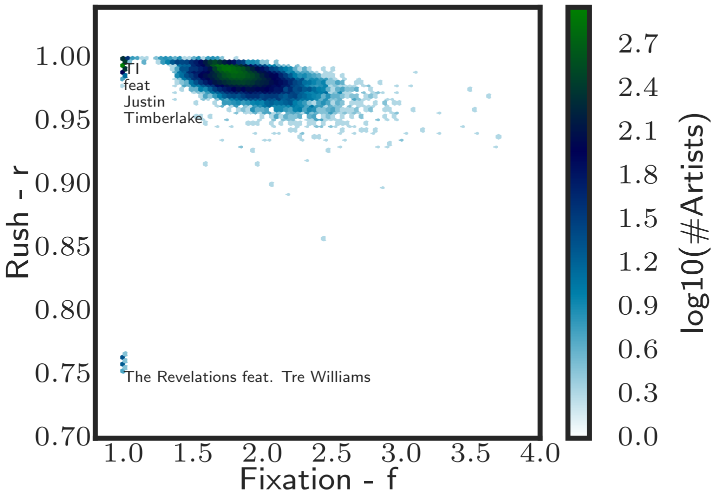


Figure 8.9: Rush vs Fixation

fixation model provides close fittings for both distributions, capturing both long and short tails. Interestingly, we can also use the model parameters r and f to distinguish between these artists: compared to *Radiohead*, the *T.I. feat. Justin Timberlake* collaboration has a slightly higher rush parameter ($r = 0.996$) but a much lower fixation parameter ($f = 1.002$). Despite the higher initial surge of attention, users lose interest more quickly in them.

We fitted our model to the daily fixation times of 36,344 and 2,570 artists in the Last.FM-Groups and Last.FM-1k datasets respectively (artists with more than 5 plays by at least 5 users). In Figure 8.9 we show a scatter plot of the fixation versus rush scores for the Last.FM-Groups dataset. Based on these fits. We found that, the vast majority of the artists have very high values of rush r (above 0.95) and values of fixation f between 1.5 and 2.5. There were also two other small groups of artists with very low (near 1) fixation. Looking into these groups, we found many collaborations between artists, such as the aforementioned TI feat Justin Timberlake. Also, our fitting errors are very small in most cases. The average Mean Squared Errors (MSE) of each fitted distribution for artists in Last.FM-Groups is only of 0.02, whereas in the Last.FM-1k dataset it was of 0.03. The standard deviations were of 0.02 and 0.04 for the Last.FM-Groups and Last.FM-1k datasets, respectively.

8.6 Summary

In this chapter we presented the A-FLUX model to mine user attention flows in social media applications. A-FLUX is a detailed user model which, based on a robust data representation, combines a modulated Markov model to capture the fixation of user attention at particular artists, with a graphical model to represent the inter-artist attention flows. We applied A-FLUX on real datasets, collected from a popular social media OMSS, observing excellent fits and superior results compared to alternative strategies.

More importantly, we used A-FLUX to uncover interesting findings from our datasets and tackle our motivation question: *Are OMSSs zero-sum user attention markets?* Our results unveil that OMSSs have elastic attention – i.e., are *not* zero-sum markets – and benefit from new album releases as user attention to the OMSS increases. This is further reflected in the various artists which we show that seem to cooperate for user attention. Yet, we also observed evidence of competition as some artists steal user attention from others (notably newcomers). Competition and collaboration impact how the popularity evolution of social media objects, artists in this case, takes place over time. As we have shown, a shock in one artist due to a release can reflect on similar artists.

This chapter concludes our work on mining user activities. We have developed two different approaches, the PHOENIX-R and the A-FLUX to capture both the revisit behavior and attention flows of users. Both of our approaches on mining user activities were founded upon our initial studies on understanding popularity evolution and feature importance (RG1). On those studies, we showed that social media content follows different trends and user perceptions of content can be related to popularity evolution. Our models in RG3 help explain why such trends occur, and how users perceive content using our latent parameters (e.g., revisits, rush and fixation). The next chapter concludes the dissertation.

Chapter 9

Conclusions and Future Research Directions

In this chapter we summarize the main achievements of this dissertation. Moreover, we also present a discussion on future research directions, as well as the list of publications derived from this dissertation. We break down this section in one subsection for each research goal (Sections 9.1, 9.2, and 9.3) containing: a brief summary of the goal, the obtained results, as well as a small discussion on future work on that goal. Next, we present a broader discussion on future research directions and open research issues. Finally, we present the list of publication in Section 9.5.

9.1 Research Goal 1 - Understanding Feature Importance to Popularity Evolution in Social Media Objects

On Research Goal 1 (RG1) we focused on understanding the evolution of popularity of UGC, how such popularity is related to different textual and incoming link features, as well as to the users perception of content. Our current results can be summarized as:

- We showed that videos that are considered the most popular ones (i.e., the Top dataset) achieve most of their views early on their *lifespan*. Moreover, a large fraction of this popularity is concentrated on a single day or week. Moreover, videos on the Random dataset exhibit a more linear like growth pattern.
- We showed that the top 10 referrers are responsible for a small fraction (up to 35%) of views and happen early in a videos' lifespan. One hypothesis for this

behavior is that early referrers will accumulate more views due to the rich-gets-richer phenomenon, but the sum of all other referrers (not captured) in our dataset exceeds the top ones.

- Using the KSC algorithm, we showed that 4 main popularity trends govern the dynamics of YouTube videos. This result confirms previous empirical evidence of these 4 trends [34].

- We also showed that early on a video's lifespan static features, such as it's category or referrers are beneficial for popularity prediction. As time passes by, popularity features become more important. This use of static features was shown useful in our results in Chapters 5 and 6, where we combined static features and early popularity features to create a novel popularity prediction methods.

- We proposed an experimental methodology on how to assess how users' perceptions of content relate with the popularity of social media objects.

- Our results showed that while a consensus between users on their perception of content is somewhat rare, whenever users do agree on which content they perceive as more interesting (i.e., the one they prefer in terms of taste, social sharing, or global perceptions of popularity), that piece of information is usually the most popular one. This observation provides evidence on the importance of content to popularity evolution.

Future Work: Although focused on YouTube videos, our work in this front could be extended to tackle other types of content. For instance, as future work we aim at further tackling questions such as: Is the consumption of content for different kinds of events(e.g., real world events such as holidays, or different types of referrers) largely different? What are the most important blogs or personalities that drive attention to different events? How does content diffusion in one service, say Twitter or Facebook, impact the popularity of videos on YouTube? In particular, comparing how popularity evolves across different media types and the factors that are responsible for this evolution could be used by content producers and marketeers to choose the applications on which they should focus. Another interesting direction for future work is the study of user popularity (as opposed to content popularity). Recent findings [128,139,141] show that the amount of subscribers a user has plays a large role in the popularity of the content shared by her. We intend to extend our study to investigate the factors that impact user popularity on social media applications, as well as the inter-dependencies that might exist between user and content popularity.

9.2 Research Goal 2 - Predicting Object Popularity

Towards achieving RG2, we focused on developing novel methods of popularity prediction. Unlike previous work, our approach focuses on predicting both *trends* (i.e., classes) and *views* of social media content (Chapter 5). We also focused on the natural trade-off between accurate predictions and the remaining interest after prediction (Chapter 6). We initially made use of the KMeans and KSC algorithm [146] to extract popularity trends of objects. Then, we then combined traditional machine learning techniques (e.g., classification and regression models) in order to perform popularity trend and views predictions. Features used in these tasks were defined by the distance between popularity time series and *previously* extracted trends. Classification was initially performed based on early popularity time series and static features. Then, specific regression [110] models were used to predict the popularity value of social media objects. Our current results are promising, showing significant improvements in prediction accuracy when compared to baseline methods [18, 110, 112] before considerable interest in the content has diminished.

Future Work: As future work, we plan to further investigate how our prediction methods can be applied to different kinds of social media (e.g., blogs and Flickr photos). We also intend to further work on improving TrendLearner's accuracy, and evaluate its effectiveness for different tasks. In details, one another important task for future work is on outlier detection. That is, predicting that a content that has attracted little interest will suddenly burst in popularity.

9.3 Research Goal 3 - Mining User Activities

On our third research goal, we focused on understanding how users activities shape social media popularity. Initially, we performed a characterization of the revisit behavior of users to social media content. Based on this characterization, we developed the Phoenix-R model, a novel time series model which captures the revisit behavior of users and multiple propagation cascades to a single object. The Phoenix-R model is a scalable approach to understand how users visit content and to predict the future popularity of content.

Next, we shifted our attention to the competitive nature of objects. User attention is a scarce and vied commodity of most social media applications. Using the A-Flux model we modeled how musical artists compete and collaborate for user attention. The Bayesian nature of A-Flux makes the model robust to biased datasets (e.g., heavy tailed counts or crawling biases). Our results show that when major releases occur, user attention for music is mostly elastic. That is, users will increase the time they spend on the application to listen to new releases. However, in some cases, such as new artists emerging in the music scene, artist releases may prey on the user attention of more well established previous artists. Also, A-Flux can capture the amount of time users spend listening to artists, as well as cluster artists based on their inherent features of attracting and keeping user attention (rush and fixation parameters).

Future Work: As future work on the Phoenix-R model we intend on extending the model to deal with: (1) interacting populations between shocks; (2) multiple cascades from a single population; and, (3) fitting on multiple time series at once (e.g., audience and revisits). In the case of A-Flux, future work includes investigating other applications of A-Flux in OMSSs (e.g., prediction) as well as in other domains. Moreover, we are also exploiting extensions of the model to deal with four mode tensors which capture: the time domain, user, source artist and destination artist. This four mode tensor analysis will enable us to identify important cascades (e.g., album releases that impact the application) directly from the user transitions dataset, and not through the use of external data sources.

9.4 Future Research Direction

In addition the specific investigations proposed as future work in the previous sections, we also discuss a broader goal as a possible future research direction we plan on pursuing. In details, we plan on studying the applicability of our results to online advertisement. A common setting in online advertising is the pay-per-click approach. In this setting, advertisers pay content providers per click that generates traffic to their website [77]. Here, we plan to study if the prediction of popularity measures (i.e., hits) and trends (evolution) can be used in conjunction with simple models of revenue estimation [52]. In simple terms, we want to verify if, in the case where click prices are determined by popularity predictions, these predictions are good enough in order to generate revenue. To this end, we plan to compare if our predictions generate revenues as good as the ones expected from the actual popularity evolution of each video. We intend to use revenue estimation models that assume fixed revenue per click for all videos, fixed revenue per click for each video (i.e., the revenue per click varies with the video's popularity) as well as models that assume revenue per click evolving based on temporal and popularity information [52]. Our evaluation will be performed in several synthetic (but relevant) scenarios, built from the model parameters [52]. We are specially interested in the work of Fu *et al.* [49] which shows that in some kinds of auctions [101], revealing data, such as a popularity prediction, is always preferable to ommiting data in order to maximize the profit of the *auctioneer*. Two other interesting models are the ones by Abraham *et al.* [2] and Mahdian *et al.* [53], which consider the case where *bidders* create their own private sources of information (e.g., popularity prediction based on a local dataset) in order to create an asymmetric market and maximize their revenues.

9.5 List of Publications

The results of this dissertation were published, or are being reviewed, in the following publications:

- Figueiredo, F., Almeida, J., Benevenuto, F., Gonçalves, M. "TrendLearner: Early Prediction of Popularity Trends of User Generated Content", *Elsevier Information Sciences*, second review round
- Figueiredo, F., Ribeiro, B., Almeida, J., Faloutsos, C. "TribeFlow: Mining & Predicting User Trajectories" In *Proceedings of the ACM World Wide Web Conference - WWW*, 2016, under review
- Figueiredo, F., Ribeiro, B., Almeida, J., Faloutsos, C. "Mining User Attention Flows in Online Music Streaming Services" In *ACM Transactions on Intelligent Systems and Technology*, under review
- Figueiredo, F., Almeida, J., Benevenuto, F., Gonçalves, M. "On the Dynamics of Social Media Popularity: A YouTube Case Study", *ACM Transaction on Internet Technology*, Vol. 14, Issue 4, December 2014.
- Figueiredo, F., Almeida, J., Gonçalves, M. "Improving the Effectiveness of Content Popularity Prediction Methods using Time Series Trends" In *ECML/PKDD Predictive Analytics Challenge*, 2014.
- Figueiredo, F., Matsubara, Y., Ribeiro, B., Almeida, J., Faloutsos, C. "Revisit Behavior in Social Media: The Phoenix-R Model and Discoveries" In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery - ECML/PKDD*, 2014.
- Figueiredo, F., Almeida, J., Benevenuto, F., Gummadi, K. "Does Content Determine Information Popularity in Social Media? A Case Study of YouTube Videos' Content and their Popularity" In *Proc. ACM CHI Conference on Human Factors in Computing Systems - CHI*, 2014.

- Figueiredo, F. “On the Prediction of Popularity of Trends and Hits for User Generated Videos” In *Proc. ACM Conference on Web Search and Data Mining - WSDM*, 2013.
- Figueiredo, F., Benevenuto, F. and Almeida, J. “The Tube over Time: Characterizing Popularity Growth of YouTube Videos”, In *Proc. ACM Conference on Web Search and Data Mining - WSDM*, 2011.

During the course of this dissertation, the following papers were published in collaboration with other authors.

- Gonçalves, G., Figueiredo, F., Almeida, J., Gonçalves, M. “Characterizing Scholar Popularity: A Case Study in the Computer Science Research Community“ In *ACM/IEEE Joint Conference on Digital Libraries - JCDL*, 2014.
- Figueiredo, F., Belém, F., Pinto, H., Almeida, J., Gonçalves, M., Fernandes, D. and Moura, E. “Assessing the Quality of Textual Features in Social Media”, *Information Processing & Management*, Vol. 49, pp. 222 - 247, 2013.
- Santos-Neto, E., Figueiredo, F., Almeida, J., Abilio, N., Andrade, N., and Ripeanu, M. “Assessing the Value of Peer-Produced Information for Exploratory Search”, *Journal of Information Science*, under review.
- Santos-Neto, E., Figueiredo, F., Almeida, J., Mowbray, M., Gonçalves, M. and Ripeanu, M. “Assessing the Value of Contributions in Tagging Systems”, In *Proc. IEEE International Symposium on Social Intelligence and Networking - SIN*, 2010.

Bibliography

[1] Abdi, H. The Bonferonni and Šidák Corrections for Multiple Comparisons. In *Encyclopedia of Measurement and Statistics*, N. Salkind, Ed. SAGE, 2001.

[2] Abraham, I., Athey, S., Babaioff, M., and Grubb, M. Peaches, lemons, and cookies: designing auction markets with dispersed information. In *Proc. EC* (2013).

[3] Adamic, L. A. *Network Dynamics: The World Wide Web.* PhD thesis, 2001.

[4] Ahmed, M., Spagna, S., Huici, F., and Niccolini, S. A Peek Into the Future: Predicting the Evolution of Popularity in User Generated Content. In *Proc. WSDM* (2013).

[5] Anderson, A., Kumar, R., Tomkins, A., and Vassilvitski, S. Dynamics of Repeat Consumption. In *Proc. WWW* (2014).

[6] Batista, G. E. A. P. A., Keogh, E. J., Tataw, O. M., and Souza, V. M. A. CID: An Efficient Complexity-Invariant Distance for Time Series. *Data Mining and Knowledge Discovery* (Apr. 2013).

[7] Bauckhage, C., Kersting, K., and Hadiji, F. Mathematical Models of Fads Explain the Temporal Dynamics of Internet Memes. In *Proc. ICWSM* (2013).

[8] Bellogín, A., de Vries, A. P., and He, J. Artist Popularity: Do Web and Social Music Services Agree? In *Proc. ICWSM* (2013).

[9] Bernstein, M. S., Bakshy, E., Burke, M., and Karrer, B. Quantifying the invisible audience in social networks. In *Proc. CHI* (2013).

[10] Blei, D. M. Introduction to Probabilistic Topic Modeling. *Communications of the ACM 55* (2012), 77–84.

[11] Bogdanov, P., Busch, M., Moehlis, J., Singh, A. K., and Szymanski, B. K. The social media genome: Modeling individual topic-specific behavior in social media. In *ASONAM* (2013), IEEE.

[12] Boll, S. MultiTube–Where Web 2.0 and Multimedia Could Meet. *IEEE Multimedia 14*, 1 (Jan. 2007), 9–13.

[13] Borghol, Y., Ardon, S., Carlsson, N., Eager, D., and Mahanti, A. The Untold Story of the Clones: Content-agnostic Factors that Impact YouTube Video Popularity. In *Proc. KDD* (2012).

[14] Borghol, Y., Mitra, S., Ardon, S., Carlsson, N., Eager, D., and Mahanti, A. Characterizing and Modeling Popularity of User-Generated Videos. *Performance Evaluation 68*, 11 (Nov. 2011), 1037–1055.

[15] Brodersen, A., Scellato, S., and Wattenhofer, M. YouTube Around the World. In *Proc. WWW* (2012).

[16] Broxton, T., Interian, Y., Vaver, J., and Wattenhofer, M. Catching a Viral Video. *Journal of Intelligent Information Systems* (Dec. 2011), 1–19.

[17] Carrascosa, J. M., González, R., Cuevas, R., and Azcorra, A. Are trending topics useful for marketing? In *Proc. COSN.* (2013).

[18] Castillo, C., El-Haddad, M., Pfeffer, J., and Stempeck, M. Characterizing the life cycle of online news stories using social media reactions. In *Proc. CSCW* (2014).

[19] Cattuto, C., Loreto, V., and Servedio, V. D. P. A Yule-Simon process with memory. *Europhysics Letters (EPL) 76*, 2 (Oct. 2006), 208–214.

[20] Celma, O. *Music Recommendation and Discovery in the Long Tail*, 1 ed. Springer, 2010.

[21] Cha, M., Benevenuto, F., Ahn, Y.-Y., and Gummadi, K. P. Delayed information cascades in Flickr: Measurement, analysis, and modeling. *Computer Networks 56*, 3 (Feb. 2012), 1066–1076.

[22] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., and Moon, S. Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems. *IEEE/ACM Transactions on Networking 17*, 5 (Oct. 2009), 1357–1370.

[23] Cha, M., Mislove, A., Adams, B., and Gummadi, K. P. Characterizing social cascades in flickr. In *Proc. WOSP* (2008).

[24] Cha, M., Mislove, A., and Gummadi, K. P. A Measurement-Driven Analysis of Information Propagation in the Flickr Social Network. In *Proc. WWW* (2009).

[25] Chatfield, C. *The Analysis of Time Series: An Introduction*, vol. 59 of *Texts in Statistical Science.* Chapman & Hall / CRC, 2004.

[26] Chatzopoulou, G., Sheng, C., and Faloutsos, M. A First Step Towards Understanding Popularity in YouTube. In *Proc. Infocom Workshops.* (2009).

[27] Chen, G. H., Nikolov, S., and Shah, D. A Latent Source Model for Nonparametric Time Series Classification. In *Proc. NIPS* (2013).

[28] Clauset, A., Shalizi, C. R., and Newman, M. E. J. Power-Law Distributions in Empirical Data. *SIAM Review 51*, 4 (Nov. 2009), 661–703.

[29] Coates, A., and Ng, A. Learning Feature Representations with K-Means. *Neural Networks: Tricks of the Trade* (2012), 561–580.

[30] Conover, M. D., Ferrara, E., Menczer, F., and Flammini, A. The digital evolution of occupy wall street. *PloS one 8*, 5 (Jan. 2013), e64679.

[31] Cormode, G., and Krishnamurthy, B. Key Differences Between Web1.0 and Web2.0. *First Monday 13*, 6 (2008).

[32] Cover, T. M., and Thomas, J. A. *Elements of Information Theory*, vol. 6 of *Wiley Series in Telecommunications.* Wiley, 1991.

[33] Cox, D. Regression Models and Life-Tables. *Journal of the Royal Statistical Society Series B Methodological 34*, 2 (1972), 187–220.

[34] Crane, R., and Sornette, D. Robust Dynamic Classes Revealed by Measuring the Response Function of a Social System. *Proceedings of the National Academy of Sciences 105*, 41 (Oct. 2008), 15649–53.

[35] Csiszár, I., and Shields, P. C. The consistency of the bic markov order estimator. *The Annals of Statistics 28*, 6 (12 2000), 1601–1619.

[36] da Cunha Recuero, R. Information Flows and Social Capital in Weblogs. In *Proc. HT* (2008).

[37] Drèze, X., and Zufryden, F. Measurement of Online Visibility and Its Impact on Internet Traffic. *Journal of Interactive Marketing 18*, 1 (Jan. 2004), 20–37.

[38] Du, P., Kibbe, W. A., and Lin, S. M. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics 22*, 17 (2006), 2059–2065.

[39] Duong, Q., Goel, S., Hofman, J., and Vassilvitskii, S. Sharding social networks. In *Proc. WSDM* (2013).

[40] Džeroski, S., and Ženko, B. Is Combining Classifiers with Stacking Better than Selecting the Best One? *Machine Learning 54*, 3 (Mar. 2004), 255–273.

[41] Easley, D., and Kleinberg, J. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, 1 ed. Cambridge University Press, July 2010.

[42] Farahat, A., and Bailey, M. C. How Effective is Targeted Advertising? In *Proc. WWW.* (2012).

[43] Ferrara, E., Varol, O., Menczer, F., and Flammini, A. Traveling trends: social butterflies or frequent fliers? In *Proc. COSN* (2013).

[44] Figueiredo, F., Benevenuto, F., and Almeida, J. The Tube Over Time: Characterizing Popularity Growth of YouTube Videos. In *Proc. WSDM* (2011).

[45] Figueiredo, F., Pinto, H., Belém, F., Almeida, J., Gonçalves, M., Fernandes, D., and Moura, E. Assessing the Quality of Textual Features in Social Media. *Information Processing & Management* (Apr. 2012).

[46] Filippova, K., and Hall, K. B. Improved Video Categorization from Text Metadata and User Comments. In *Proc. SIGIR* (2011).

[47] Fleiss, J. L., and Levin, B. *Statistical Methods for Rates and Proportions*, 3 ed. Wiley-Interscience, 2003.

[48] Flickr. Flickr: Advertising sollutions.

[49] Fu, H., Jordan, P., Mahdian, M., Nadav, U., Talgam-Cohen, I., and Vassilvitskii, S. Algorithmic Game Theory. In *Algorithmic Game Theory: Lecture Notes in Computer Science*, Lecture Notes in Computer Science. Springer, 2012, pp. 168–179.

[50] Fu, T.-c. A Review on Time Series Data Mining. *Engineering Applications of Artificial Intelligence 24*, 1 (Feb. 2011), 164–181.

[51] Geurts, P., Ernst, D., and Wehenkel, L. Extremely Randomized Trees. *Machine Learning 63*, 1 (2006), 3–42.

[52] Ghose, A., and Yang, S. An Empirical Analysis of Search Engine Advertising: Sponsored Search in Electronic Markets. *INFORMS: Management Science* (2010).

[53] Ghosh, A., Mahdian, M., McAfee, R. P., and Vassilvitskii, S. To Match or Not to Match: Economics of Cookie Matching in Online Advertising. *ACM Transactions on Economics and Computation 3*, 2 (Apr. 2015).

[54] Gill, P., Arlitt, M., Li, Z., and Mahanti, A. Youtube Traffic Characterization: A View From the Edge. In *Proc. IMC* (2007).

[55] Gill, P., Erramilli, V., Chaintreau, A., Krishnamurthy, B., Papagiannaki, D., and Rodriguez, P. Follow the Money: Understanding Economics of Online Aggregation and Advertising. In *Proc. IMC* (2013).

[56] Golbandi, N. G., Katzir, L. K., Koren, Y. K., and Lempel, R. L. Expediting Search Trend Detection via Prediction of Query Counts. In *Proc. WSDM* (2013).

[57] Golder, S. A., and Hubberman, B. Usage Patterns of Collaborative Tagging Systems. *Journal of Information Science 32*, 2 (Apr. 2006), 198–208.

[58] Griffiths, T. Gibbs sampling in the generative model of latent dirichlet allocation. Tech. rep., 2002.

[59] Gulotta, R., Faste, H., and Mankoff, J. Curation, provocation, and digital identity: risks and motivations for sharing provocative images online. In *Proc. CHI* (2012).

[60] Hansen, M. H., and Yu, B. Model Selection and the Principle of Minimum Description Length, 2001.

[61] Harvey, M., Ruthven, I., and Carman, M. J. Improving social bookmark search using personalised latent variable language models. In *Proc. WSDM* (2011).

[62] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 27 of *Springer Series in Statistics*. Springer, 2009.

[63] Hauger, D., Schedl, M., Kosir, A., and Tkalci, M. The Million Musical Tweets Dataset: What Can we Learn from Microblogs. In *Proc. ISMIR* (2013).

[64] Hethcote, H. W. The Mathematics of Infectious Diseases. *SIAM Review 42*, 4 (2000), 599–653.

[65] Hu, Q., Wang, G., and Yu, P. S. Deriving Latent Social Impulses to Determine Longevous Videos. In *Proc. WWW* (2014).

[66] Huang, C., Li, J., and Ross, K. W. Can Internet Video on Demand be Profitable? In *Proc. SIGCOMM* (2007).

[67] Huberman, B. A., and Adamic, L. A. Internet: Growth Dynamics of the World-Wide Web. *Nature 401*, 6749 (Sept. 1999), 131–2.

[68] Islam, M. A., Eager, D., Carlsson, N., and Mahanti, A. Revisiting Popularity Characterization and Modeling of User-generated Videos. In *Proc. Mascots* (2013).

[69] Jain, R. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* Wiley, 1991.

[70] Jiang, L., Miao, Y., Yang, Y., Lan, Z., and Hauptmann, A. G. Viral Video Style: A Closer Look at Viral Videos on YouTube. In *Proc. ICMR* (2014).

[71] Kamath, K. Y., Caverlee, J., Lee, K., and Cheng, Z. Spatio-temporal dynamics of online memes: a study of geo-tagged tweets. In *Proc WWW.* (2013).

[72] Kang, J.-H., and Lerman, K. LA-CTR: A Limited Attention Collaborative Topic Regression for Social Media. In *Proc. AAAI* (2013).

[73] Kang, J.-h., Lerman, K., and Getoor, L. LA-LDA: A Limited Attention Model for Social Recommendation. In *Social Computing, Behavioral-Cultural Modeling and Prediction* (2013), Springer Berlin Heidelberg.

[74] Kaplan, A. M., and Haenlein, M. Users of the World, Unite! The Challenges and Opportunities of Social Media. *Business Horizons 53*, 1 (Jan. 2010), 59–68.

[75] Kasneci, G., Ramanath, M., Suchanek, F., and Weikum, G. The YAGO-NAGA approach to knowledge discovery. *ACM SIGMOD Record 37*, 4 (2009), 41.

[76] Khosla, A., Sarma, A. D., and Hamid, R. What Makes an Image Popular. In *Proc. WWW* (2014).

[77] Lacerda, A., Cristo, M., Gonçalves, M. A., Fan, W., Ziviani, N., and Ribeiro-Neto, B. Learning to Advertise. In *Proc. SIGIR* (2006).

[78] Lakkaraju, H., McAuley, J., and Leskovec, J. What's in a Name? Understanding the Interplay between Titles, Content, and Communities in Social Media. In *Proc. ICWSM* (2013).

[79] Lee, J. G., Moon, S., and Salamatian, K. An Approach to Model and Predict the Popularity of Online Contents with Explanatory Factors. In *Proc. WIC* (2010), vol. 1.

[80] Lehmann, J., Gonçalves, B., Ramasco, J. J., and Cattuto, C. Dynamical classes of collective attention in twitter. In *Proc. WWW* (2012).

[81] Lerman, K., and Hogg, T. Using a Model of Social Dynamics to Predict Popularity of News. In *Proc. WWW* (2010).

[82] Lerman, K., and Jones, L. Social Browsing on Flickr. In *Proc. ICWSM* (2006).

[83] Leskovec, J. Social Media Analytics. In *Proc. WWW* (2011).

[84] Leskovec, J., Backstrom, L., and Kleinberg, J. Meme-Tracking and the Dynamics of the News Cycle. In *Proc. KDD* (2009).

[85] Lin, J., Keogh, E., Wei, L., and Lonardi, S. Experiencing SAX: A Novel Symbolic representation of Time Series. *Data Mining and Knowledge Discovery 15* (2007), 107–144.

[86] Mader, H. M., Coles, S. G., Connor, C. B., and Connor, L. J. *Statistics in Volcanology.* Geological Society of London, 2006.

[87] Manchanda, P., Dubé, J.-P., Goh, K. Y., and Chintagunta, P. K. The Effect of Banner Advertising on Internet Purchasing. *The Effect of Banner Advertising on Internet Purchasing 43*, 1 (2006), 98–108.

[88] Marlow, C., Naaman, M., Boyd, D., and Davis, M. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Prog. HT* (2006).

[89] Matsubara, Y., Sakurai, Y., and Faloutsos, C. The Web as a Jungle: Non-Linear Dynamical Systems for Co-evolving Online Activities. In *Proc. WWW* (2015).

[90] Matsubara, Y., Sakurai, Y., Faloutsos, C., Iwata, T., and Yoshikawa, M. Fast mining and forecasting of complex time-stamped events. In *Proc. KDD* (2012).

[91] Matsubara, Y., Sakurai, Y., Prakash, B. A., Li, L., and Faloutsos, C. Rise and Fall Patterns of Information Diffusion. In *Proc. KDD* (2012).

[92] Menascé, D., and Almeida, V. *Capacity Planning for Web Services: Metrics, Models, and Methods.* Prentice Hall, 2002.

[93] Mennecke, B., Roche, E. M., Bray, D. A., Konsynski, B., Lester, J., Rowe, M., and Townsend, A. M. Second Life and Other Virtual Worlds: A Roadmap for Research. In *Proc. ICIS* (2007).

[94] Mestyán, M., Yasseri, T., and Kertész, J. Early prediction of movie box office success based on Wikipedia activity big data. *PloS one 8*, 8 (Jan. 2013), e71226.

[95] Meyer, C. D. Stochastic complementation, uncoupling markov chains, and the theory of nearly reducible systems. *SIAM Review 31* (1989), 240–272.

[96] Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B. Measurement and Analysis of Online Social Networks. In *Proc. IMC* (2007).

[97] Mitzenmacher, M. Dynamic Models for File Sizes and Double Pareto Distributions. *Internet Mathematics 1*, 3 (2004), 305–333.

[98] Moat, H. S., Curme, C., Avakian, A., Kenett, D. Y., Stanley, H. E., and Preis, T. Quantifying Wikipedia Usage Patterns Before Stock Market Moves. *Scientific Reports 3* (May 2013).

[99] Mohri, M. *Foundations of Machine Learning (Adaptive Computation and Machine Learning series).* MIT Press, 2012.

[100] Myers, S. A., and Leskovec, J. Clash of the Contagions: Cooperation and Competition in Information Diffusion. In *Proc. ICDM* (2012), IEEE.

[101] Myerson, R. B. Optimal Auction Design. *Mathematics of Operations Research 6*, 1 (1981), 58–73.

[102] Nannen, V. A Short Introduction to Model Selection, Kolmogorov Complexity and Minimum Description Length (MDL). *Complexity*, Mdl (2010), 20.

[103] Newman, M. W. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics 46* (Feb. 2006), 323–351.

[104] Nielsen. Nielsen Entertanment and Billboard's 2014 Music Industry Report. Tech. rep., 2014.

[105] Nielsen Institute. Nielsen Entertanment and Bullboard's 2014 MID-Year Music Industry Report. Tech. rep., 2014.

[106] Nigam, K., and Ghani, R. Analyzing the Effectiveness and Applicability of Co-training. In *Proc. CIKM* (2000).

[107] Nikolov, S. *Trend or No Trend: A Novel Nonparametric Method for Classifying Time Series.* PhD thesis, MIT, 2012.

[108] Nowak, R. Investigating the interactions between individuals and music technologies within contemporary modes of music consumption. *First Monday 19*, 10 (2014), Online.

[109] Oliveira, R. D., Cherubini, M., and Oliver, N. Looking at Near-Duplicate Videos from a Human-Centric Perspective. *ACM Transactions on Multimedia Computing, Communications, and Applications 6*, 3 (Aug. 2010), 1–22.

[110] Pinto, H., Almeida, J., and Gonçalves, M. Using Early View Patterns to Predict the Popularity of YouTube Videos. In *Proc. WSDM* (2013).

[111] Preis, T., Moat, H. S., and Stanley, H. E. Quantifying trading behavior in financial markets using Google Trends. *Scientific reports 3* (Jan. 2013), 1684.

[112] Radinsky, K., Svore, K., Dumais, S., Teevan, J., Bocharov, A., and Horvitz, E. Behavioral Dynamics on the Web: Learning, Modeling, and Prediction. *ACM Transactions on Information Systems 32*, 3 (2013), 1–37.

[113] Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., and Keogh, E. Searching and Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12* (New York, New York, USA, 2012), ACM Press.

[114] Ratkiewicz, J., Flammini, A., and Menczer, F. Traffic in social media I: paths through information networks. In *Proc. SIN* (2010).

[115] Reed, W. J., and Hughes, B. D. From Gene Families and Genera to Incomes and Internet File Sizes: Why Power Laws are so Common in Nature. *Physical review. E 66*, 6 (2002).

[116] Rendle, S., Freudenthaler, C., and Schmidt-Thieme, L. Factorizing personalized Markov chains for next-basket recommendation. *Proc. WWW* (2010).

[117] Reshef, D., Reshef, Y., Finucane, H. K., Sharon R. Grossman, McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. Detecting Novel Associations in Large Data Sets. *Science 334*, 6062 (2011), 1518–1524.

[118] Ribeiro, B. Modeling and Predicting the Growth and Death of Membership-based Websites. In *Proc. WWW* (2014).

[119] Ribeiro, B., and Faloutsos, C. Modeling Website Popularity Competition in the Attention-Activity Marketplace. In *Proc. WSDM* (2015).

[120] Robert, S., and Boudec, J.-Y. L. On a Markov modulated chain exhibiting self-similarities over finite timescale. *Performance Evaluation 27-28* (1996), 159–173.

[121] Saez-Trumper, D., Comarela, G., Almeida, V., Baeza-Yates, R., and Benevenuto, F. Finding trendsetters in information networks. In *Proc. KDD* (2012).

[122] Salganik, M. J., Dodds, P. S., and Watts, D. J. Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market. *Science 311*, 5762 (2006), 854–856.

[123] Shamma, D. A., Kennedy, L., and Churchill, E. F. Peaks and persistence: Modeling the shape of microblog conversations. In *Proc. CSCW* (2011).

[124] Shao, J. Linear model selection by cross-validation. *Journal of the American Statistical Association 88*, 422 (1993), pp. 486–494.

[125] Shumway, R. H., and Stoffer, D. S. *Time Series Analysis and Its Applications With R Examples*, vol. 102 of *Springer Texts in Statistics.* Springer, 2006.

[126] Sinha, S., and Pan, R. K. How a "Hit" is Born: The Emergence of Popularity from the Dynamics of Collective Choice. In *Econophysics and Sociophysics: Trends and Perspectives*. Wiley, 2007, p. Online.

[127] Stone, M. An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society. Series B (Methodological) 39*, 1 (1977), pp. 44–47.

[128] Susarla, A., Oh, J.-H., and Tan, Y. Social Networks and the Diffusion of User-Generated Content: Evidence from YouTube. *Information Systems Research 23*, 1 (2011), 1–19.

[129] Szabo, G., and Huberman, B. A. Predicting the Popularity of Online Content. *Communications of the ACM 53*, 8 (2010), 80–88.

[130] Thomson, D. J. Jackknifing multiple-window spectra. In *Proceedings of ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing* (1994), vol. vi, IEEE.

[131] Vakali, A., Giatsoglou, M., and Antaris, S. Social networking trends and dynamics detection via a cloud-based framework design. In *Proc. WWW* (2012).

[132] Valera, S., Gomez-Rodriguez, M., and Gummadi, K. P. Modeling Adoption of Competing Products and Conventions in Social Media. In *NIPS Workshop in Networks* (2014).

[133] van Zwol, R. Flickr: Who is Looking? In *Proc. WI* (2007), IEEE.

[134] Vimeo. Advertise on Vimeo, 2012.

[135] Vintsyuk, T. K. Speech discrimination by dynamic programming. *Cybernetics 4*, 1 (1972), 52–57.

[136] Wang, C., and Huberman, B. A. Long trend dynamics in social media. *EPJ Data Science 1*, 2 (2012).

[137] Wang, X., and McCallum, A. Topics over time. In *Proc. KDD* (2006).

[138] WANG, Y.-C., BURKE, M., AND KRAUT, R. E. Gender, topic, and audience response: an analysis of user-generated content on facebook. In *Proc. CHI* (2013).

[139] WATTENHOFER, M., WATTENHOFER, R., AND ZHU, Z. The YouTube social network. In *Proc. ICWSM* (2012).

[140] WATTS, D. J., AND DODDS, P. S. Influentials, Networks, and Public Opinion Formation. *Journal of Consumer Research 38*, 4 (2007), 441–458.

[141] WENG, L., FLAMMINI, A., VESPIGNANI, A., AND MENCZER, F. Competition among memes in a world with limited attention. *Nature Scientific reports 2* (Jan. 2012), 335.

[142] WOOLDRIDGE. *Introductory Econometrics: A Modern Approach.* South-Western, 2013.

[143] XIONG, L., CHEN, X., HUANG, T.-K., SCHNEIDER, J., AND CARBONEL, J. G. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proc. SDM* (2010).

[144] YAN, X., GUO, J., LAN, Y., AND CHENG, X. A biterm topic model for short texts. 1445–1456.

[145] YANG, J., AND LESKOVEC, J. Modeling Information Diffusion in Implicit Networks. In *Proc. ICDM* (2010).

[146] YANG, J., AND LESKOVEC, J. Patterns of temporal variation in online media. In *Proc. WSDM* (2011).

[147] YANO, T., AND SMITH, N. A. What's Worthy of Comment? Content and Comment Volume in Political Blogs. In *Proc. ICWSM* (2010).

[148] YE, L., AND KEOGH, E. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Mining and Knowledge Discovery 22*, 1-2 (June 2011), 149–182.

[149] YIN, H., CUI, B., CHEN, L., HU, Z., AND ZHANG, C. Modeling Location-based User Rating Profiles for Personalized Recommendation. *ACM Transactions on Knowledge Discovery from Data To Appear*.

[150] YIN, P., LUO, P., WANG, M., AND LEE, W.-C. A straw shows which way the wind blows: Ranking Potentially Popular Items from Early Votes. In *Proc. WSDM* (2012).

[151] YOUTUBE. YouTube press statistics, 2012.

[152] Yu, H., Xie, L., and Sanner, S. Exploring the Popularity Phases of YouTube Videos: Observations, Insights, and Prediction. In *Proc. ICWSM* (2015).

[153] Zeng, D., Chen, H., Lusch, R., and Li, S.-H. Social Media Analytics and Intelligence. *IEEE Intelligent Systems 25*, 6 (Nov. 2010), 13–16.

[154] Zhao, C., Hinds, P., and Gao, G. How and to whom people share: The role of culture in self-disclosure online communities. In *Proc. CSCW* (2012).

[155] Zhou, R., Khemmarat, S., and Gao, L. The impact of YouTube Recommendation System on Video Views. In *Proc. IMC* (2011).