A Vocabulary for Growth: Topic Modeling of Content Popularity Evolution

Rosario G. Garroppo, *Member IEEE,* Mohamed Ahmed, Saverio Niccolini, and Maurizio Dusi

***Abstract*—In this paper, we present a novel method to predict long-term popularity of User Generated Content (UGC). At first, the method clusters the dynamics of UGC popularity into a vocabulary of growth in popularity (sequence) by using a mixture model. Eventually, the method assigns to each sequence a topic model to describe the dynamics of the sequence in a compact way. We then use this topic model to identify similar patterns of growth in popularity of newly observed UGC. The proposed method has two key features: 1) it considers the historical dynamics of the UGC popularity, and 2) it provides long-term popularity prediction. Results on real dataset of UGC show that the proposed method is flexible, and able to accurately forecast the complete growth in popularity of a given UGC.**

***Index Terms*—Gaussian Mixture Model, Latent Dirichlet Allocation model, Topic model, popularity prediction.**

I. Introduction

The economic sustainability of the sites hosting User Generated Content (UGC) is strictly related to a deep understanding of the popularity patterns of the hosted UGC. Early detection of popular UGC allows publishers to maximise their revenues through better advertisement placement. Moreover, content-distribution networks can exploit popularity prediction methods to forecast users' future demand and proactively allocate resources.

In this scenario, recent research has been focused on defining the characteristics of the growth path of UGC popularity studying wide-spread platforms such as YouTube, Facebook, Twitter, and Vimeo. Every minute, users around the world send more than 350,000 tweets, share more than 680,000 pieces of content on Facebook, and upload 100 hours of video on YouTube [1]. Hence, lots of work on UGC popularity estimation are focused on combining multi-domain data to either classify or predict observations.

The design of algorithms for detecting popular UGC is a challenging task [2], especially when only one source of data is considered. Even when considering data coming from multiple domains, their combination is difficult given that the data are usually sparse or sampled with different time intervals.

When we consider data within a single domain, popularity dynamics can be seen as timeseries. As shown in [2], there exist different methods that achieve a good trade-off between complexity and prediction accuracy, even though the prediction is only on short-term. Note that the adjectives "long-term" and "short-term" refer to the temporal distance of the prediction from the present. As concerning the time-series, "short-term" predictors predict values few steps ahead, whereas "long-term" several steps ahead.

Long-term popularity prediction is a challenging task. In [3], we find attempts to long-term prediction, based on regression methods. However, the authors of [4] have observed from YouTube data that videos with very similar popularity values at early stages may have distinct future patterns. Furthermore, in [5], the authors show that popularity fluctuations are more likely to be attributed to their historical dynamics than to their historical values. For example, videos that at an early stage increases their popularity rapidly usually become more popular than those with slow growth. More accurate prediction methods, e.g., [4] and [6], include information on the observed popularity paths.

A. Motivations and contributions

In this study, we aim at a method for predicting the UGC popularity taking into account an application-agnostic feature space. We consider the number of cumulative views of a UGC over time as our only feature, as this feature is the most basic and accessible piece of information. Several works also exploit this feature in regression-based models to infer the long-term popularity of a given content, i.e., to predict the popularity in 48 hours after observing the views it gets within the first hour. However, regression-based approaches require to define the time instant of the prediction before starting the training phase of the model itself.

We instead aim at forecasting the complete growth in popularity of a content rather than a single time instant, which is set during the training phase. Moreover, our model is designed to improve the prediction accuracy as new cumulative views observations of the considered UGC become available.

In our previous work [6], we designed a technique to reveal and cluster distinct paths of popularity growth based on the cumulative views. A transition graph then helps us predict which clusters a given content will likely belong to. Results on a real dataset have shown that this method significantly outperforms the baseline regression-based methods. However, there were two limitations in our approach, which we try to overcome in this paper: 1) the Affinity Propagation algorithm used for clustering is unsuitable for an online mechanism, and 2) the prediction accuracy is very sensitive to the estimation of the transition probabilities between the clusters.

The main contributions of this paper are the following.

- A novel method for predicting the popularity of UGC based only on the number of cumulative views of the

Rosario G. Garroppo is with Dipartimento di Ingegneria dell'informazione, University of Pisa.

Mohamed Ahmed was at NEC Laboratories Europe, Heidelberg, Germany, during the writing of this work.

Maurizio Dusi is at Copan Group, Brescia, Italy.

Saverio Niccolini is at NEC Laboratories Europe, Heidelberg, Germany

content over time. The choice of this feature allows us to take into account the correlation between the observation time and the number of cumulative views in a bidimensional model used to represent the popularity evolution over the time. The prediction of the popularity is obtained sampling its model. The proposed method is composed of two steps. We first model the training data via mixture models, thus capturing regions of possible dynamics of growths, or segments of growth. We denote each segment of growth as *pseudo-word*: the prefix "pseudo" is used to remind the reader that our approach does not consider textual features of the UGC, while "word" recalls the use of Latent Dirichlet Allocation (LDA) model for representing the sequence of segments of growth (i.e. the sequence of pseudo-words) that form the popularity path. For simplicity, we do not use the prefix "pseudo" in the definition of "topic" and "vocabulary". Each segment of growth (pseudo-word) is characterised by a set of distribution parameters. The set of sequences obtained during the training phase represents our vocabulary, which we use to infer future pseudo-words, i.e. segments of the popularity growth of a new UGC, given its available observations.

- We overcome the two key limitations of our previous work in two ways. First, by using the Factorized Asymptotic Bayesian (FAB) Gaussian Mixture Model (GMM) clustering algorithm [7], which similarly to the Affinity Propagation does not require to set the number of clusters before running the algorithm. Second, by using a "bag of words" analogy to define a topic model (LDA) for evaluating the likelihood of possible transitions between known pseudo-words. The result is a topic model that reflects possible popularity paths observed in the training set.
- Results shows that with respect to other solutions based on the same feature space, our method has two advantages: 1) it takes into account the historical dynamics at the early stage of the UGC popularity, and 2) it provides long-term popularity prediction.

The rest of the paper is organised as follows. In Section II we discuss the related work. In Section III we describe our method, and in Section IV its implementation. Section V contains a description of the datasets used in the performance analysis and Section VI shows our results. Finally, Section VII concludes the paper.

II. Related work

The idea of mapping data points from the original space into another, sparse, space is not new. For instance, the authors of [8] propose a solution based on hashing techniques for computing similarities between images. Furthermore, the importance of video UGC has led researchers to study different aspects of the systems based on this kind of UGC. For instance, in [9] the authors propose a video recommendation system able to track users' click-through rate and provide personalised recommendation. Within the purpose of estimating the popularity of UGC, several studies have appeared in the literature, starting from the preliminary works of Gill *et al.* [10] and Zink *et al.* [11]. Makaroff *et al.* [12] present a survey on measurement studies on the characteristics of YouTube and related sites.

Borghol *et al.* [13], [14] studied the evolution of popularity of YouTube videos over time. They showed that most videos achieve their peak in popularity within less than six weeks since their upload date and that there is a correlation between the current and future popularity of a video. They identify the total view count as the simplest, yet strongest predictor for the popularity of a video, the only exception being videos uploaded shortly before starting observing the popularity trend due to lack of information for building the prediction model. Our study builds upon this observation for selecting cumulative view count as feature. The authors also acknowledge that content-agnostic factors (e.g., social media) help explain popularity dynamics.

Figueiredo *et al.* [15] report on the burst nature of UGC: popular videos usually experience a huge number of views on a single peak day or week. Furthermore, they study the popularity patterns for videos that are popular, videos that were deleted and randomly selected videos. Videos that were deleted for copyright violation tend to get most of their views much earlier in their (short) lifetime than other videos. Moreover, they investigated how users reach each given video (e.g., by searching on YouTube or following a link on other websites), to shed light on the mechanisms that contribute to the video's popularity. Their analysis is based on the total view count over time of the YouTube videos as returned by Google charts API, i.e., one hundred data points over the entire lifetime of the video, regardless of the upload time of the video. The details of the popularity pattern are indeed limited by this coarse-grained representation.

Chowdhury *et al.* [16] break down YouTube videos by categories and investigate their evolution independently. Their results suggest that sophisticated techniques (e.g., based on timeseries clustering) are required to predict future popularity since from an early stage.

In a recent survey [2], Tatar *et al.* collect findings from several works aimed at predicting the popularity evolution of UGC.

1) Prediction based on the observed domain: The following methods only model a user's past behaviour or an item's history individually without accounting for external signals (e.g., from social media). Our method falls under this category.

Both Szabo *et al.* [3] and Pinto *et al.* [4] analyse the popularity growth of YouTube videos and Digg stories, and show a strong correlation between the (logarithmically-transformed) past popularity and current popularity by using a univariate and multivariate model, respectively. However, their approaches are designed to predict the growth of aggregate items, and may not be accurate when targeting an individual item.

Yin *et al.* [17] rank potentially popular items from early votes, instead of focusing on the trend of popularity growth. They propose a Conformer-Maverick (CM) probabilistic model to rank potentially popular items in online content sharing systems. Different people have different distributions of these two personalities (conformer or maverick), which can

be learned according to the observed voting history. Lee *et al.* [18] propose a model based on survival analysis to evaluate the probability that a given content will receive more than a given number of hits; Hu *et al.* [19] present an additive and a multiplicative timeseries model for the popularity evolution patterns of hot topics, which takes into account the seasonality of the data. With a similar goal, Radinsky *et al.* [20] propose a temporal modeling framework adapted from physics and signal processing.

In [21] and [22], the authors show that popularities change over time at various levels of temporal granularity. For example, popularity patterns of street snap on Flickr are observed to depict distinctive fashion styles at specific time scales, such as season-based periodic fluctuations for Trench Coat or one-off peak in days for Evening Dress. The authors present a study to incorporate multiple time-scale dynamics into predicting online popularity. They propose Multi-scale Temporalization, a computational framework for estimating popularity based on multi-scale decomposition and structural reconstruction in a tensor space of user, post, and time by joint low-rank constraints. Their solution exploits contextual information of individual popularities. Contextual features are built from three main perspectives: user influence, visual content, and post metadata.

The studies of Hong *et al.* [23], Lakkaraju *et al.* [24], Gao *et al.* [25], and Kong *et al.* [26], focus on predicting the popularity of Twitter messages and the probability of retweeting with a multi-class classification approach. They use k-nearest neighbour and Support Vector Machine for the prediction.

Recently, Wu *et al.* [27] proposed EvoModel, a stochastic fluid model, to capture the different evolution patterns of a given video. It is based on a two-step approach: the information spreading and the user reaction process. However, the authors do not explore the application of their model to the prediction of video popularity. Tan *et al.* [28] propose a novel timeseries model for popularity prediction based on the correlation between early and future popularity series. Instead of inferring the precise view counts for a video, they focus on accurately identifying the most popular videos based on the predicted popularity.

Li *et al.* [29] propose a model that can capture the popularity dynamics based on early popularity evolution pattern and future popularity burst prediction. The model is motivated by the following features highlighted by the data analysis: 1) the strong correlation between video's early popularity and long-term popularity; 2) the great impact of popularity evolution pattern on the long-term popularity; 3) the possibility of popularity bursts in the middle of a video's lifetime. Lymperopoulos [30] models the popularity evolution by interlacing linear and non-linear growth terms, and predicts the popularity of online content through extrapolation.

2) Prediction based on multi-domain information: The following methods take into account information from multiple domains for their prediction. For instance, they include online social network data collected from the same website hosting the UGC.

Studies have shown that some external factors are correlated with the popularity of a video over time, such as the number of users' subscription to a given YouTube channel [31], the referrers that lead users to videos [32], user comments [33] and user voting behavior [34].

Jamali *et al.* [35] use users' comments to predict the popularity of online content linked at Digg. In particular, they compare the prediction performance obtained using the decision tree classifier, the nearest neighbour classifier, and support vector machines for performing the classification.

Vallet et al. [36] study the correlation between the virality of a video content on Twitter and the popularity of such content on YouTube. The analysis highlights the unique properties of content that is both popular and viral, i.e., attracts a high number of views on YouTube and achieves fast propagation on Twitter. Then, they propose a framework for predicting videos that are likely to be popular, viral, and both. The proposed framework achieves a high degree of accuracy with a low amount of training data.

Roy et al. [37] propose a model for predicting the popularity of YouTube videos. The model extracts popular and trending topics on Twitter, which are linked to the corresponding YouTube videos. They achieve 70% higher accuracy of significant popularity growth prediction when compared to the single-domain models that only use data from YouTube.

Trzciński et al. [38] propose to use Support Vector Regression with Gaussian Radial Basis Functions to predict the popularity of online video content. Their results suggest that using only visual features computed before the publication of the video can help predict future video popularity. Nevertheless, higher prediction accuracy is achieved when adding temporal features, such as view counts or social features.

Although these works show that considering external factors can lead to a gain when inferring popularity information, we argue that (i) such factors are not always available for every content – with the consequent problem of building models with lacking data – and (ii) the temporal correlation between external and internal factors is yet to be shown.

III. The Proposed Method

We first introduce some notations used throughout the paper. Let o_i be the observed content i, and $\mathcal{D}(o_i)$ the collected set of points (t, d_t^i) describing its popularity evolution. Each point is composed of t, the time elapsed since when we start observing o_i, and d_t^i, the cumulative number of downloads (or weblog) of o_i up to time t. It is worth noting that t is a relative timestamp, given that the observation of each object begins independently from one another. We indicate the whole dataset as $\mathcal{O} = \{\mathcal{D}(o_1), \ldots, \mathcal{D}(o_i), \ldots, \mathcal{D}(o_N)\}$, and use $\mathcal{O}$ to solve the following problem.

Problem Formulation: Let o_{new} be a new object and O_b a set of observations (t, d_t^{new}) with $t \in [t_a, t_b]$. Provide the popularity prediction of o_{new} for $t > t_b$.

Note that when $t_a = 0$, it means that we consider all the view that the object received since its upload time. When instead $t_a > 0$, data observed up to t_a are not considered when performing the prediction.

As introduced in Section I-A, considering the timeseries as a set of bidimensional points (t, d_t^i) allows us to take into account the correlation between t and d_t^i when modeling $\mathcal{D}(o_i)$. The timeseries $\mathcal{D}(o_i)$ can then be generated by sampling its model.

A. The training phase

The training phase is composed of two steps:

- the definition of a vocabulary, as shown in Figure 1,
- the modelling of sequences of pseudo-words via the LDA model, as shown in Figure 2.

In the first step, we model the training set $\mathcal{O}_\mathcal{T} = \{\mathcal{D}(o_1), \ldots, \mathcal{D}(o_i), \ldots, \mathcal{D}(o_\tau)\}$ as a bi-dimensional mixture model, denoted as $M_\mathcal{O}$:

$$f(t, d_t^i) = \sum_{c=1}^{C_\mathcal{O}} \alpha_c p_c((t, d_t^i)|\Phi_c) \tag{1}$$

The model $M_\mathcal{O}$ is characterized by the $C_\mathcal{O}$ components and corresponding weights α_c. Each component is described by a distribution p_c of parameters Φ_c and represents a cluster of segments of growth. The set $C_\mathcal{O}$ denotes a reduced space where we can map the timeseries $\mathcal{D}(o_i)$ for all $o_i \in \mathcal{O}$.

$M_\mathcal{O}$ is related to the particular dataset $\mathcal{O}$ of content we choose, e.g., YouTube video popularity, or web page popularity. Furthermore, the model can be updated, for instance whenever new available data show low performance in the prediction phase.

$M_\mathcal{O}$ represents an estimation of the density of $\mathcal{O}$, which enables us to partition the observed samples over the space of possible values. However, the model itself does not help understand how different $\mathcal{D}(o_i)$ evolve. To this end, we propose a strategy based on the mapping of the points (t, d_t^i) of $\mathcal{D}(o_i)$ into the reduced space $C_\mathcal{O}$: by using this procedure, we summarise a set of (t, d_t^i) pairs with a particular component of $C_\mathcal{O}$.

To understand the mapping mechanism and the approximation introduced by this strategy, we need to recall Equation (1). To classify an observation of $\mathcal{D}(o_i)$ at the time t, we compute the component of $C_\mathcal{O}$ with highest likelihood of generating the point (t, d_t^i):

$$w_t^i = \arg\max_{c \in C_\mathcal{O}} \ \alpha_c p_c(t, d_t^i|\Phi_c) \tag{2}$$

For a given t, this operation maps each sample (t, d_t^i) to a pseudo-word, i.e., a component of $C_\mathcal{O}$. Thus, each timeseries $\mathcal{D}(o_i)$ is mapped to a sequence of pseudo-words w_t^i, that we denote as $\mathcal{W}(o_i)$. Each pseudo-word w_t^i gives a probabilistic representation of (t, d_t^i), which includes information on the correlation between time and observation, i.e., between t and d_t^i. Due to this, we assume that starting from the pseudo-words distribution (i.e. the coefficients of the mixture distribution) individual pseudo-words can be generated independently from one another. We can then obtain new approximate samples of the original $\mathcal{D}(o_i)$ by sampling the mixture defined by $\mathcal{W}(o_i)$. $\mathcal{W}(o_i)$ defines a new mixture that approximates $f(t, d_t^i)$, i.e.,

$$\hat{f}(t, d_t^i) = \sum_{c \in C^{\mathcal{W}(o_i)}} \alpha_c^{\mathcal{W}(o_i)} p_c(X|\Phi_c) \tag{3}$$

where $C^{\mathcal{W}(o_i)}$ is the subset of $C_\mathcal{O}$ in $\mathcal{W}(o_i)$. Since the components in (3) are a subset of $C_\mathcal{O}$, the weights $\alpha_c^{\mathcal{W}(o_i)}$ must be calculated by normalising it to the sum of the frequency of each component in $C^{\mathcal{W}(o_i)}$. The weights $\alpha_c^{\mathcal{W}(o_i)}$ and $C^{\mathcal{W}(o_i)}$ represent the mixture parameters of the $\mathcal{W}(o_i)$ model. The new representation of $\mathcal{W}(o_i)$ can be used to i) assess the similarity between different timeseries (for clustering) and ii) predict the expected values for the timeseries, by inferring future pseudo-words.

In the second step, we consider $\mathcal{W}(o_i)$ for all $o_i \in \mathcal{O}_\mathcal{T}$ and define a model that represents the sequence of pseudo-words in $\mathcal{W}(o_i)$. To this aim, we exploit the LDA model, which has been proposed in natural language processing for detecting hidden relationships between words and documents in large text corpora. LDA assumes that words and documents are being generated from a hidden mixture of topics that captures the semantic structure of documents. This approach allows us to work with a fixed number of topics which documents are generated from. The details on the LDA theory and inference techniques can be found in [39] and [40]. In our scenario, we assume that each $\mathcal{W}(o_i)$ is a document of the text corpora $\mathcal{O}_\mathcal{T}$. The LDA represents the set of $\mathcal{W}(o_i)$ by looking at hidden structures in the sequence of segments of growth of each timeseries. The topics of the LDA model represent this hidden structure. They can be used for finding similarities between UGC popularity patterns and for predicting popularity evolution. The LDA model returns the set of topics, $T_{\mathcal{O}_\mathcal{T}}$, which generates $\mathcal{W}(o_i)$ for all $o_i \in \mathcal{O}_\mathcal{T}$. For each $\mathcal{W}(o_i)$, the LDA estimates its topic model given by the distribution of the subset of topics belonging to $T_{\mathcal{O}_\mathcal{T}}$. Finally, the topic model is used to estimate the mixture model based on $C_\mathcal{O}$, which can then be used to generate the samples (t, d_t^i).

This approach assumes that the growth in popularity of $\mathcal{D}(o_i)$ can be described by relatively few latent patterns (topics), which define a mixture over a set of components. As such, the growth of the popularity of a given content can be viewed as the transition through various paths of growth (topics). The similarity between individual contents can finally be assessed by comparing the distance between their associated topic models.

B. The prediction phase

As shown in Figure 3, the prediction phase of an object o_{new} starts by mapping the available observations $(t, d_t^{o_{new}})$ for $t \in [t_a, t_b]$ into $C_\mathcal{O}$. The result is the sequence of pseudo-words $\mathcal{W}(o_{new})$ in the time interval $t \in [t_a, t_b]$. The LDA model obtained with $\mathcal{O}_\mathcal{T}$ infers the topic model of $\mathcal{W}(o_{new})$. By means of this model, we calculate the probability associated to each pseudo-word, i.e., to each mixture component $C_\mathcal{O}$. In other words, from the topic model of $\mathcal{W}(o_{new})$ we estimate the parameters $\alpha_c^{\mathcal{W}(o_{new})}$ and $C^{\mathcal{W}(o_{new})}$ of the

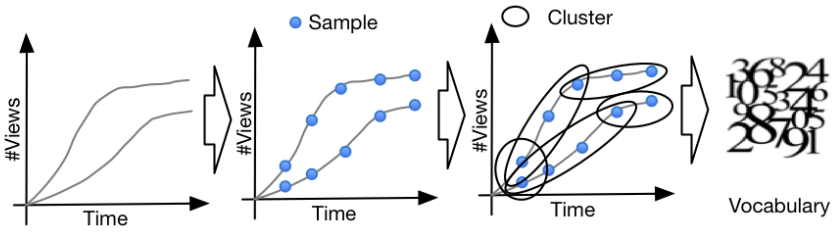


Fig. 1: From the training set $\mathcal{O}_\mathcal{T}$ to the vocabulary.

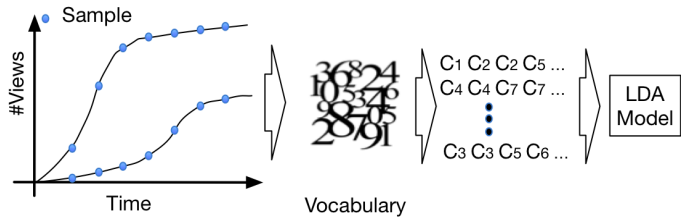


Fig. 2: From the timeseries $\mathcal{D}(o_i)$ to the LDA model of the "pseudo-words" sequences $\mathcal{W}(o_i)$.

mixture representing o_{new}. The predicted values of o_{new} for $t > t_b$ are then generated by sampling the estimated mixture.

There are different approaches for inferring the topic model of $\mathcal{W}(o_{new})$. The simplest one is based on applying the same LDA model parameters calculated during the training phase. Another approach starts from the previous one and appropriately weights the topic model of the K-Nearest Neighbor (K-NN) objects in $\mathcal{O}_\mathcal{T}$. The distance between two objects takes into account only the parameters of their topic models. Section IV expands on this.

Note that this prediction mechanism takes into account the entire structure of the available samples of an object. Hence, higher is the observation period, the more accurate is the prediction, as supported by our results.

IV. An implementation of the method

In the first step of our implementation, we select the GMM model for partitioning the dataset $\mathcal{O}_\mathcal{T}$ and for finding the sparse code $C_\mathcal{O}$. Note that any mixture model or clustering mechanism can be used to define the sparse code. The GMM model parameters can be estimated using a Bayesian model selection approach, which is based on the evaluation of marginal log-likelihood. Since exact evaluation is often computationally and analytically intractable, a number of approximation algorithms have been studied, such as the one named Variational Bayesian (VB) [41]. However, these approaches require to set the number of components in order to estimate the parameters of the GMM. To overcome this issue, we use the FAB algorithm, as presented in [7]. The FAB algorithm automatically selects the number of components, thus mitigating overfitting problems. Furthermore, this algorithm addresses the non-identifiability issue in mixture modelling and outperforms the VB method in terms of performance when selecting the model, and in terms of computational efficiency. Given $C_\mathcal{O}$, we simply use the relation (2) to map each sample of $\mathcal{D}(o_i)$ into a pseudo-word of the sequence $\mathcal{W}(o_i)$. Before estimating the parameters of the LDA model, we perform a compression of the sequences $\mathcal{W}(o_i)$ for all $o_i \in \mathcal{O}_\mathcal{T}$. The compression consists in summarising in one pseudo-word a sequence of equal pseudo-words. For example, assuming that c_i indicates the i-th component of $C_\mathcal{O}$, the following sequences

$$\mathcal{W}(o_1) = [c_1, c_1, c_1, c_4, c_4, c_4, c_5, c_5, c_5, c_5],$$
$$\text{and } \mathcal{W}(o_2) = [c_1, c_2, c_2, c_1, c_3, c_3, c_4, c_4, c_3, c_3].$$

can be compressed such that

$$\mathcal{W}_c(o_1) = [c_1, c_4, c_5],$$
$$\text{and } \mathcal{W}_c(o_2) = [c_1, c_2, c_1, c_3, c_4, c_3]$$

The obtained corpus given by the set $\mathcal{W}_c(o_i)$ for all $o_i \in \mathcal{O}_\mathcal{T}$ represents the input to the variational procedure as described in [39]. The procedure outputs the parameters of the LDA model: these parameters provide a probabilistic description of each observed pseudo-words sequence $\mathcal{W}_c(o_i)$ for all $o_i \in \mathcal{O}_\mathcal{T}$. In particular, for each $\mathcal{D}(o_i) \in \mathcal{O}_\mathcal{T}$ we have a topic model that gives a compressed representation of the popularity evolution of $\mathcal{D}(o_i)$ approximated using the sparse code $C_\mathcal{O}$.

Note that after the first step, the prediction problem could be solved modelling the transition probabilities among the components of $C_\mathcal{O}$. However, this approach has different computational problems that are solved by the compact representation of the growth paths given by the topic model of LDA.

A. Inference

Given a set of observations $(t, d_t^{o_{new}})$ in $t \in [t_a, t_b]$ for the object o_{new}, the inference procedure classifies the observations against $C_\mathcal{O}$, via the relation (2). The resulting $\mathcal{W}(o_{new})$ is then compressed. The new sequence $\mathcal{W}_c(o_{new})$ and the LDA

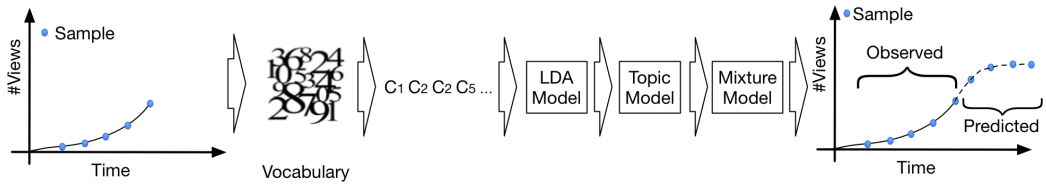


Fig. 3: From the observed data $\mathcal{D}(o_{new})$ in the interval $[t_a, t_b]$ to the predicted values of o_{new} for $t > t_b$.

parameters are used to obtain the topic model of o_{new} for the available observations in $[t_a, t_b]$. We denote this topic model as $TMOI(o_{new})$, i.e., the Topic Model based on an Observation Interval of (o_{new}).

The simplest prediction method performs the sampling of the mixture model from $TMOI(o_{new})$. By using the topic distribution and the component distribution over each topic, we calculate the probability to observe each $C_{\mathcal{O}}$ component in o_{new}. As a result, we have an estimated mixture model for o_{new} that we use to predict samples of o_{new} for $t > t_b$. The weights of the mixture model are obtained by normalising the probability related to each component as given by the LDA model. For instance, assuming that $\mathcal{W}_c(o_{new}) = [c_4]$, the LDA returns $Topic_1 \in T_{\mathcal{O}_\mathcal{T}}$ with a probability of 0.75. Supposing that in the LDA model $Topic_1 = [0.1c_1, 0.5c_4, 0.4c_5]$, where the coefficient of c_i indicates its probability, the components and the related probabilities of o_{new} are given by $0.75Topic_1$, i.e., c_1 with probability 0.075, c_4 with 0.375, and c_5 with 0.3. By normalising the probability associated to these three components of $C_{\mathcal{O}}$, we obtain the weights, $\alpha_c^{\mathcal{W}(o_{new})}$, of $C^{\mathcal{W}(o_{new})} = \{c_1, c_4, c_5\}$. In other words, referring to relation (3) we have the parameters of the mixture model that represents the estimated path of o_{new}, given the observations $(t, d_t^{o_{new}})$ for $t \in [t_a, t_b]$.

We obtained the best prediction results by the alternative method described in the following. In particular, we consider the $TMOI(o_{new})$ for finding the K-Nearest Neighbor (K-NN) objects in $\mathcal{O}_\mathcal{T}$, in terms of topic model parameters. These K-NN objects are then used to estimate a topic model of o_{new}, denoted as $KTM(o_{new})$ (K-NN Topic Model). In particular, during the training phase, we obtain for each $\mathcal{D}(o_i) \in \mathcal{O}_\mathcal{T}$ the associated list of topics and their corresponding probabilities. This information is summarised in a matrix, where each row refers to $\mathcal{D}(o_i)$ and the column refers to each observed topic in the training phase. Each cell reports the probability of observing the topic, indicated in the column index, in the timeseries $\mathcal{D}(o_i)$, associated with the row index. Having the topics and related probabilities for the considered o_{new}, we can find the K-NN objects that have similar topics structure. We use the cosine similarity as measure. Given two vectors of attributes, $A = a_1, \cdots, a_m$ and $B = b_1, \cdots, b_m$, the cosine similarity, $d_{cos}(A, B)$, is defined as

$$d_{cos}(A, B) = \frac{\sum_{i=1}^{m} a_i b_i}{\sqrt{\sum_{i=1}^{m} a_i^2} \sqrt{\sum_{i=1}^{m} b_i^2}} \tag{4}$$

We average the topic model of the K-NN objects to obtain the estimation of $KTM(o_{new})$. Starting from $KTM(o_{new})$ and the pseudo-words distribution over each topic of $KTM(o_{new})$, we then compute the probability to observe each component of $C_{\mathcal{O}}$. After the normalisation of the probabilities associated with each component, we obtain the mixture model of o_{new}. By sampling this model, we finally get the whole path estimated for the o_{new}, taking into account the observations $(t, d_t^{o_{new}})$ for $t \in [t_a, t_b]$.

Note that the prediction values are obtained by sampling the mixture model estimated with one of the presented procedures. This approach gives the estimation of the popularity evolution of o_{new} at any time t, after the observation interval $[t_a, t_b]$.

V. Datasets

We evaluated the accuracy of our algorithm on two datasets. The first dataset is publicly available and includes data from real-time analytics engine Chartbeat. Taking into account 30,000 URLs posted on the web during 2013 and having at least 10 visits, this dataset contains a timeseries of the number of pageviews in time slots of 5 minutes of those URLs and the number of messages posted on Twitter and Facebook that include those URLs. For a full description of this dataset, please refer to [42]. From the original dataset, we extracted for each URL the cumulative number of pageviews sampled every 5 minutes as feature.

The second dataset includes the popularity evolution of a randomly selected set of 20,000 YouTube videos. All the videos have been observed for 100 days. For each video, we considered the number of cumulative views sampled each day since their upload as feature.

For each dataset, we randomly assigned half of the observations to the training set and the rest to the testing set. We used the training set to gather the model following the procedure described in Section IV. Then, we used the test set for evaluating the performance of the proposed prediction procedure.

As performance parameter, we use both Root Mean Square Error (RMSE) and R^2 value, i.e. the square of the Pearson's correlation coefficient R.

The RMSE represents a measure of accuracy and is computed as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M} (\log(1 + \hat{d}_s^i) - \log(1 + d_s^i))^2}{M}} \tag{5}$$

where d_s^i and $\hat{d}_s^i$ are the actual and the estimated value of the popularity of o_i at the considered time s; M is the cardinality of the considered testing set $\mathcal{O}_{Test}$.

We choose the R^2 parameter for comparing the prediction performance of our method with other works presented in the literature, with particular respect to the ones presented in [43] during the Predictive Web Analytics challenge where we got the Chartbeat dataset from. Unless otherwise reported, all the experiments presented in the rest of the paper are carried out on the Chartbeat dataset.

VI. Performance analysis

At first, we compare the performance of our GMM-LDA implementation with different design options available for our method. Specifically, we first discuss how alternative formulations of the sparse code impacts on the performance of the predictor. The considered alternative formulation uses K-means clustering to partition the space. Second, we try to replace the LDA and apply a first-order Markov Chain to infer the future popularity of contents. Lastly, we analyse the prediction accuracy of our proposal against the regression method proposed by Pinto *et al.* [4].

A. Topic modelling of data

At first, we prepare and clean up of the raw data before to apply the technique as detailed in Section IV.

1) Parameterizing the model: Figure 4 shows the analysis we carried out to evaluate the number of mixture components necessary to partition the dataset. The figure shows the number of mixture components obtained for different sample sizes (e.g., number of (t, d_t^i) pairs): the number of components required for convergence, as automatically computed by the FAB algorithm, is in the order of 40, and saturates quickly.

Figure 5 reports on the structure of the co-occurrence matrix formed by the transitions between components. The figure shows that for each component (in the x-axis) we have a non-null "log density" only towards a subset of components reported in the y-axis. This observation indicates a sparse transition matrix and supports our initial hypothesis that the evolution of content popularity has an underlying structure.

When we analyse the fitting of the pseudo-word sequences with the LDA, we find that the number of required topics with both datasets starts to converge at around 70-80 topics. As shown in Figure 6, the Chartbeat dataset indicates a better fit: however, it is worth mentioning that Chartbeat dataset has 8 times more samples than YouTube dataset.

The fact that the number of topics converges suggest that we should expect little difference in adding more data to build the mixture model or selecting more topics. However, in experimenting with the different sample sizes and number of topics, we consistently observed better results when using 858,644 samples, which is approximately 5% of the entire dataset, and 100 topics. Therefore, we will use this setup in the rest of the paper.

2) Validating the approach: In this section, we look at the impact of replacing the GMM mixture model and LDA with K-Means and Markov Chain. Using K-means to create sparse code in a similar method to ours is popular in the graphics and imagine retrieval community. We report here the best results, that we achieved with K set to 150, and number of topics equal

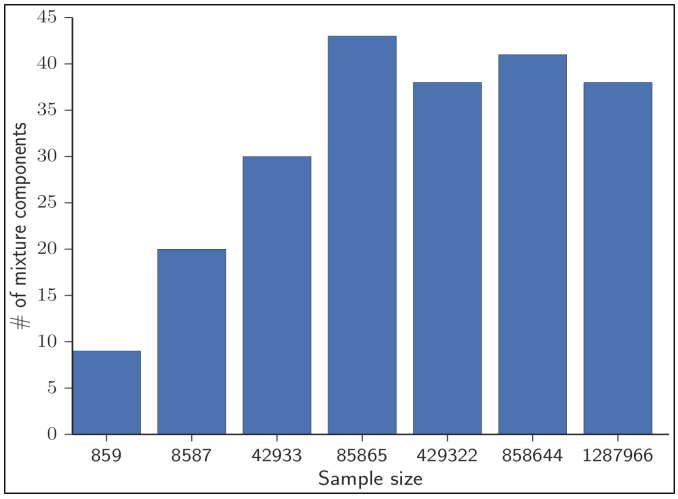


Fig. 4: Number of components in the GMM - Chartbeat dataset. The last value of sample size value corresponds to about 10% of the whole dataset.

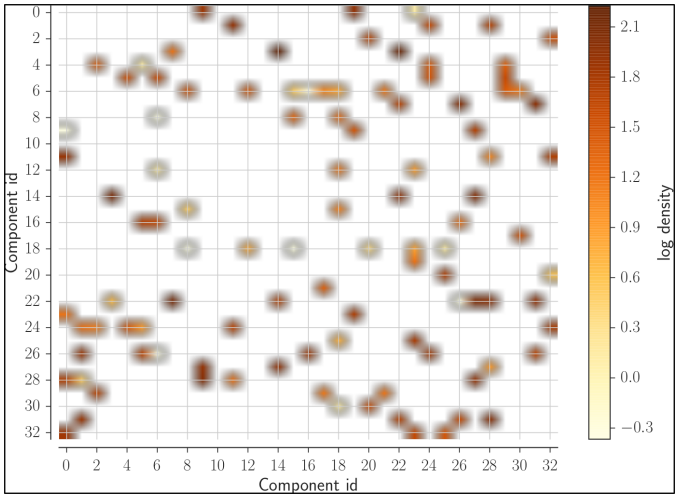


Fig. 5: Label co-occurrence matrix- Chartbeat dataset.

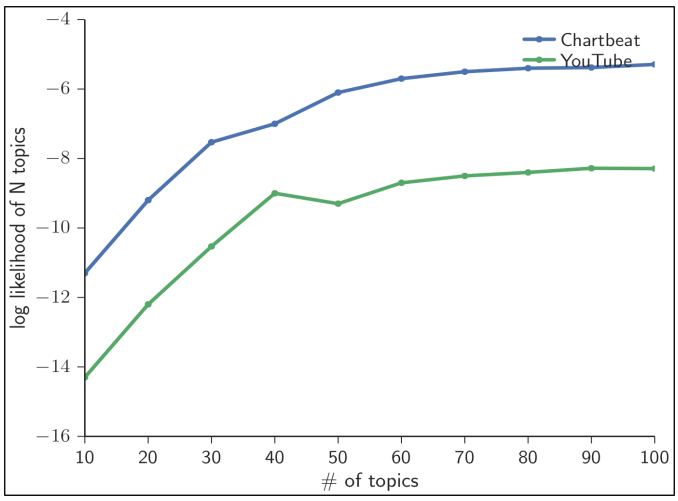


Fig. 6: Convergence of the number of topics required to model the data.

to 100. On the other hand, the Markov Chain is motivated by the results in Figure 5, which show the high preferential attachment between components and suggest that modelling these transitions should be fruitful.

Table I summarises the results of these experiments by using R^2 score as metric, for different observation intervals (i.e., $[0, t^*]$) and prediction horizon (i.e., the prediction time is $s = t^*$+*prediction horizon*). The analysis considers the set of values $(1h, 8h, 12h, 16h)$ for t^* and the prediction horizon. Our results show that the combination of GMM and LDA achieves better performance both in the long and the short prediction horizon. We noted that the errors in the K-means test came predominantly from the inaccuracy of the clustering as opposed to the GMM.

When using Markov Chain for predicting the transitions among components, we observed that the error in prediction of the next-component is low on average: the error is dominated by the transition probabilities of popular components. This result can be explained considering that the transitions in the Markov Chain are based only on the components of the previous step, and the probability of transitions among different components are time independent: this leads the Markov Chain to error when there are multiple paths to choose from. In contrast, LDA considers the entire history.

Table II shows that the R^2 scores for the YouTube dataset are in line with the ones for the Chartbeat dataset, given that in the Chartbeat dataset, 12 samples are associated to $1h$, whereas in the YouTube dataset we have 1 sample per day.

Finally, we compare the results with the regression technique presented in [43]. In that paper, the authors develop a model with many variables, including information regarding the attention an article receives from social media, as opposed to our approach based only on one feature. They report $R^2 = 0.72$ for predicting the number of visits 1 hour after publication: our model achieves instead $R^2 = 0.95$. On the other hand, we have comparable numbers when predicting 6-8 hours in advance using 6-8 hours of observation window.

		Predicted (GMM + LDA)			
	R^2	+1 hour	+8 hours	+12 hours	+16 hours
Observed	1 hour	0.95	0.57	0.61	0.60
	8 hours	0.99	0.83	0.72	0.69
	12 hours	0.99	0.87	0.84	0.83
	16 hours	0.99	0.98	0.97	0.96
		Predicted (K-Means + LDA)			
	R^2	+1 hour	+8 hours	+12 hours	+16 hours
Observed	1 hour	0.83	0.55	0.46	0.50
	8 hours	0.91	0.68	0.67	0.66
	12 hours	0.92	0.77	0.79	0.77
	16 hours	0.91	0.86	0.82	0.81
		Predicted (GMM + Markov chain)			
	R^2	+1 hour	+8 hours	+12 hours	+16 hours
Observed	1 hour	0.9	0.4	0.3	0.2
	8 hours	0.91	0.57	0.30	0.30
	12 hours	0.88	0.44	0.52	0.49
	16 hours	0.92	0.62	0.60	0.70

TABLE I: R^2 values for the different alternatives explored - Chartbeat dataset.

		Predicted			
	R^2	+1 day	+7 days	+14 days	+30 days
Observed	1 day	0.7	0.65	0.54	0.39
	7 days	0.96	0.82	0.83	0.76
	14 days	0.95	0.87	0.82	0.83
	30 days	0.96	0.86	0.85	0.88

TABLE II: R^2 values for YouTube predictions

B. The prediction of content popularity

Based on our study, we found that while topics are capable of capturing the general trend of the growth process, sampling from the components of the mixture often leads to higher variance in the estimates and can be dominated by few components. For instance, Figure 7 lists the topics and neighbours associated with two given objects (samples are omitted for clarity). In Figure 7(a), we see that two components only can capture the behaviour of the object (number 4 and 15 in the figure) when considering just a single topic. The object in Figure 7(b) has more active components, which are related to two topics and cover a wider range. On the other hand, the K-nearest neighbours, which in both cases is set to 10, show much more homogeneity in their growth paths.

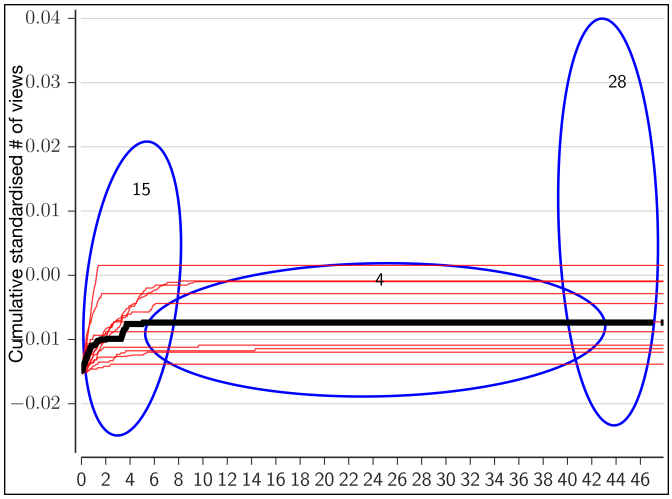
Figure 8 reports the RMSE values at different time. Each curve refers to a different value of the observation interval $[0, t^*]$ (t^* has been set in the range $[0.5h, 42h]$). Interestingly enough, our method is very accurate in the short-term prediction (a few hours ahead), due to the object transition between few components: $RMSE < 1$, with 1 hour of observations and predicting up-to 10 hours in advance. We remind that in the Chartbeat dataset 1h corresponds to 12 samples of the timeseries. Longer observation time (i.e., higher t^*) leads to a significant improvement: with $t^* = 6h$, the $RMSE$ is less than 0.5 in all cases. In particular, we observe this low $RMSE$ value even when we consider a prediction horizon of 40 h (i.e., 480 samples ahead in the timeseries), showing the accuracy of the proposed method for long-term prediction.

For comparison, Figure 9 displays the accuracy of our method with respect to the multiple variables regression method given in [4]. Our approach significantly outperforms the regression method: the regression method requires at least 3-4 more observations to achieve the same RMSE as our model in predicting the number of views at the time $48h$.

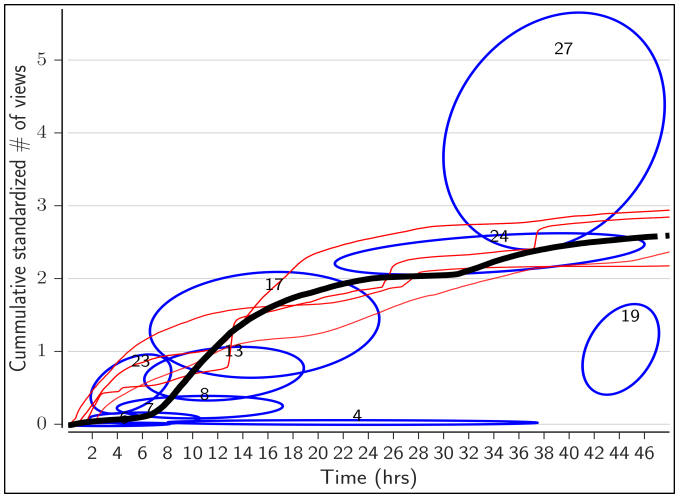
VII. Conclusion

In this paper, we presented a novel method for long-term prediction of the UGC popularity. The method is composed of two steps. First, we model the dynamics of the popularity of the content via a GMM mixture model. Each component of the mixture represents a segment of the popularity growth path. The different paths are then modeled as encoded sequences. Second, we infer topics for each popularity growth path via LDA. This model is then used to find similarities among different growth paths and to predict the future path of a newly observed objects.

Our results show that the combination of GMM and LDA outperforms solutions based on K-means and Markov Chain, as well as the baseline regression method for long-term prediction.



(a) Sample object 1



(b) Sample object 2

Fig. 7: Example growth paths for two objects of the Chartbeat dataset. In each plot, the ellipses give the significant ($p \geq 1e-3$) topic components of the object, the bold black line represents its popularity path, while the red lines represent the top 10 nearest neighbours. Note that in 7(b), we have less red lines because we found two groups of 4 nearest neighbours having similar popularity path.

Our method has the advantage to forecast the whole path of growth in popularity, rather than just having to set the prediction horizon as parameter of the training phase.

Finally, our method is modular and any alternative combination of density partition and sequence learning approaches can be explored.

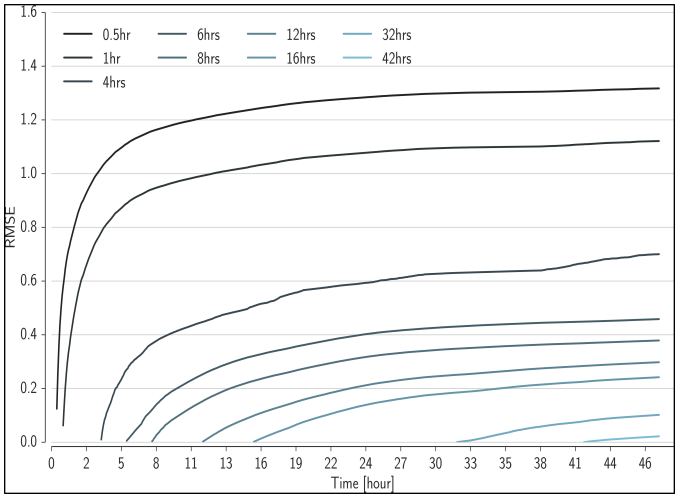


Fig. 8: RMSE error of predictions using the Chartbeat dataset. Each curve refers to a particular observation interval, reported in the legend. The abscissa shows the prediction time.

REFERENCES

[1] "Internet live statistics." http://www.internetlivestats.com, 2016.

[2] A. Tatar, M. de Amorim, S. Fdida, and P. Antoniadis, "A survey on predicting the popularity of web content," *Journal of Internet Services and Applications*, vol. 5, no. 1, 2014.

[3] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Communications of the ACM*, vol. 53, no. 8, pp. 80–88, August 2010.

[4] H. Pinto, J. Almeida, and M. Goncalves, "Using early view patterns to predict the popularity of youtube videos," in *Proceedings of the sixth ACM international conference on Web search and data mining (WSDM 2013)*. ACM, February 2013, pp. 365–374.

[5] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos, "Rise and fall patterns of information diffusion: Model and implications," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 6–14.

[6] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini, "A peek into the future: Predicting the evolution of popularity in user generated content," in *Proceedings of the sixth ACM international conference on Web search and data mining (WSDM 2013)*. ACM, February 2013, pp. 607–616.

[7] R. Fujimaki and S. Morinaga, "Factorized asymptotic bayesian inference for mixture modeling," in *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*. JMLR, April 2012, pp. 400–408.

[8] C. Yan, H. Xie, D. Yang, J. Yin, Y. Zhang, and Q. Dai, "Supervised hash coding with deep neural network for environment perception of intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[9] T. Mei, B. Yang, X.-S. Hua, and S. Li, "Contextual video recommendation by multimodal relevance and user feedback," *ACM Transactions on Information Systems*, vol. 29, no. 2, 2011.

[10] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterisation: A view from the edge," in *Proceedings of the ACM SIGCOMM conference on Internet Measurement (IMC)*. ACM, October 2007, pp. 15–28.

[11] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network - measurements, models, and implications," *Computer Networks*, vol. 53, no. 11, pp. 501–514, March 2009.

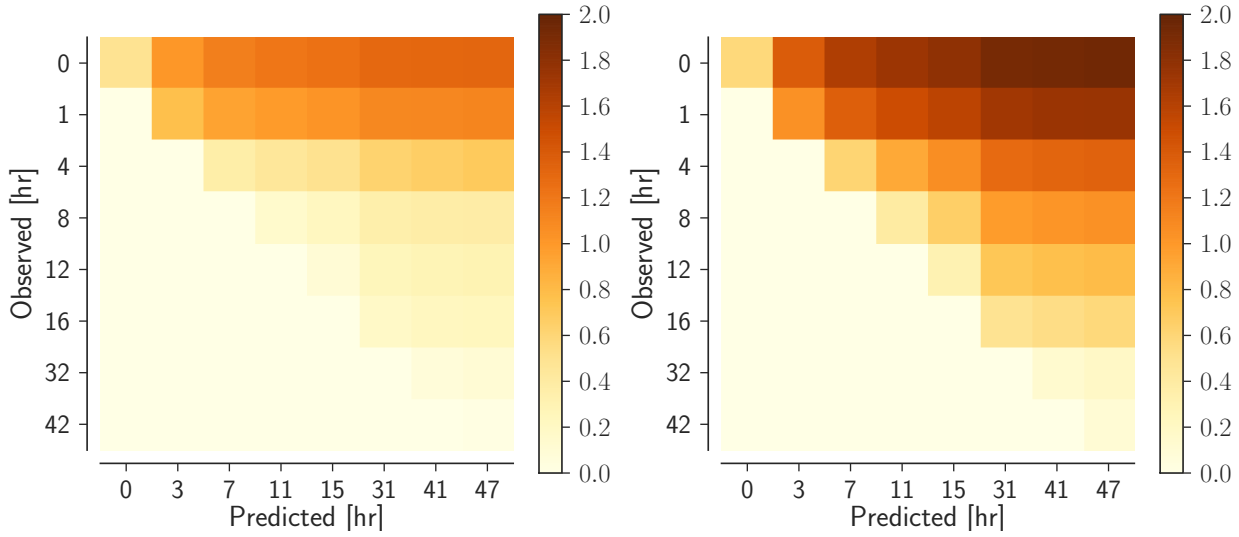
[12] C. S. Alam and D. J. Makaroff, "Characterizing videos and users in youtube: A survey," in *Seventh International Conference on Broadband, Wireless Computing, Communication and Applications*. IEEE, November 2012, pp. 244–254.

[13] Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti, "Characterizing and modelling popularity of user-generated videos," *Performance Evaluation*, vol. 68, no. 11, pp. 1037–1055, November 2011.

[14] ——, "The untold story of the clones: Content-agnostic factors that impact youtube video popularity," in *Proc. of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 12)*, August 2012, pp. 1186–1194.

[15] F. Figueiredo, F. Benevenuto, and J. Almeida, "The tube over time: Characterizing popularity growth of youtube videos," in *Proc. of the 4th ACM International Conference of Web Search and Data Mining (WSDM'11)*. ACM, February 2011, pp. 745–754.

[16] S. Chowdhury and D. Makaroff, "Popularity growth patterns of youtube videos: A category-based study," in *WEBIST 2013 - Proceedings of*



(a) Our method

(b) The Pinto *et al.* model

Fig. 9: Comparative accuracy of the Chartbeat dataset: The RMSE in predicting the number of pageviews at the time reported in the x-axis, given the observation interval $[0, t^*]$ reported in the y-axis.

the 9th International Conference on Web Information Systems and Technologies. SCITEPRESS Digital Library, May 2013, pp. 233–242.

[17] P. Yin, P. Luo, M. Wang, and W.-C. Lee, "A straw shows which way the wind blows: Ranking potentially popular items from early votes," in *Proceedings of the fifth ACM international conference on Web search and data mining (WSDM 2012)*. ACM, February 2012, pp. 623–632.

[18] J. Lee, S. Moon, and K. Salamatian, "Modeling and predicting the popularity of online contents with cox proportional hazard regression model," *Neurocomputing*, vol. 76, no. 1, pp. 134–145, 2012.

[19] C. Hu, Y. Hu, W. Xu, P. Shi, and S. Fu, "Understanding popularity evolution patterns of hot topics based on time series features," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8710 LNCS, pp. 58–68, 2014.

[20] K. Radinsky, K. Svore, S. Dumais, J. Teevan, A. Bocharov, and E. Horvitz, "Modeling and predicting behavioral dynamics on the web," in *Proceedings of the 21st Annual Conference on World Wide Web (WWW 2012)*, April 2012, pp. 599–608.

[21] B. Wu, W.-H. Cheng, Y. Zhang, and T. Mei, "Time matters: Multi-scale temporalization of social media popularity," in *MM 2016 - Proceedings of the 2016 ACM Multimedia Conference*, 2016, pp. 1336–1344.

[22] B. Wu, T. Mei, W.-H. Cheng, and Y. Zhang, "Unfolding temporal dynamics: Predicting social media popularity using multi-scale temporal decomposition," in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016, pp. 272–278.

[23] L. Hong, O. Dan, and B. Davison, "Predicting popular messages in twitter," in *Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011*, 2011, pp. 57–58.

[24] H. Lakkaraju and J. Ajmera, "Attention prediction on social media brand pages," in *International Conference on Information and Knowledge Management, Proceedings*, 2011, pp. 2157–2160.

[25] S. Gao, J. Ma, and Z. Chen, "Popularity prediction in microblogging network," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8709 LNCS, pp. 379–390, 2014.

[26] S. Kong, F. Ye, and L. Feng, "Predicting future retweet counts in a microblog," *Journal of Computational Information Systems*, vol. 10, no. 4, pp. 1393–1404, 2014.

[27] J. Wu, Y. Zhou, D. Chiu, and Z. Zhu, "Modeling dynamics of online video popularity," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1882–1895, 2016.

[28] Z. Tan, Y. Wang, Y. Zhang, and J. Zhou, "A novel time series approach for predicting the long-term popularity of online videos," *IEEE Transactions on Broadcasting*, vol. 62, no. 2, pp. 436–445, 2016.

[29] C. Li, J. Liu, and S. Ouyang, "Characterizing and predicting the popularity of online videos," *IEEE Access*, vol. 4, pp. 1630–1641, 2016.

[30] I. Lymperopoulos, "Predicting the popularity growth of online content: Model and algorithm," *Information Sciences*, vol. 369, pp. 585–613, 2016.

[31] M. Wattenhofer, R. Wattenhofer, and Z. Zhu, "The youtube social network," in *Sixth International AAAI Conference on Weblogs and Social Media (ICWSM 2012)*. AAAI, June 2012, pp. 354–361.

[32] F. Figueiredo, "On the prediction of popularity of trends and hits for user generated videos," in *Proceedings of the sixth ACM international conference on Web search and data mining (WSDM 2013)*. ACM, February 2013, pp. 741–746.

[33] X. He, M. Gao, M.-Y. K. Y. Liu, and K. Sugiyama, "Predicting the popularity of web 2.0 items based on user comments," in *SIGIR 2014 - Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, July 2014, pp. 233–242.

[34] K. Lerman and T. Hogg, "Using a model of social dynamics to predict popularity of news," in *Proceedings of the World Wide Web Conference (WWW 2010)*, April 2010, pp. 621–630.

[35] S. Jamali and H. Rangwala, "Digging digg: Comment mining, popularity prediction, and social network analysis," in *Proceedings of International Conference on Web Information Systems and Mining (WISM 2009)*. IEEE, November 2009, pp. 32–38.

[36] D. Vallet, S. Berkovsky, S. Ardon, A. Mahanti, and M. Kaafar, "Characterizing and predicting viral-and-popular video content," in *International Conference on Information and Knowledge Management, Proceedings*, 2015, pp. 1591–1600.

[37] S. Roy, T. Mei, W. Zeng, and S. Li, "Towards cross-domain learning for social video popularity prediction," *IEEE Transactions on Multimedia*, vol. 15, no. 6, pp. 1255–1267, 2013.

[38] T. Trzcinski and P. Rokita, "Predicting popularity of online videos using support vector regression," *IEEE Transactions on Multimedia*, vol. 19, no. 11, pp. 2561–2570, 2017.

[39] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, no. 3, pp. 993–1022, 2003.

[40] T. Griffiths and M. Steyvers, "Finding scientific topics," *The National Academy of Sciences*, no. 101, pp. 5228–5235, 2004.

[41] C. M. Bishop, *Pattern Recognition and Machine Learning*. Reading, Massachusetts: Springer-Verlag, 2006.

[42] "Predictive web analytics." https://sites.google.com/site/predictivechallenge2014, 2014.

[43] C. Castillo, M. El-Haddad, J. Pfeffer, and M. Stempeck, "Characterizing the life cycle of online news stories using social media reactions," in *Computer-Supported Cooperative Work and Social Computing (CSCW)*, 2014.