Timely Video Popularity Forecasting based on Social Networks

Jie Xu, Mihaela van der Schaar
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, CA, USA
Email: jiexu@ucla.edu, mihaela@ee.ucla.edu

Jiangchuan Liu, Haitao Li
School of Computing Science
Simon Fraser University
Burnaby, Canada
Email: jcliu@cs.sfu.ca, haitaol@sfu.ca

***Abstract*—This paper presents Pop-Forecast, a systematic method for accurately forecasting the popularity of videos promoted through social networks. Pop-Forecast aims to optimize the forecasting accuracy and the timeliness with which forecasts are issued, by explicitly taking into account the dynamic propagation of videos in social networks. The forecasting is performed online and requires no training phase or a priori knowledge. We analytically bound the performance loss of Pop-Forecast as compared to that obtained by an omniscient oracle and prove that the bound is sublinear in the number of video arrivals, thereby guaranteeing its fast rate of convergence as well as its asymptotic convergence to the optimal performance. We validate the performance of Pop-Forecast through extensive experiments using real-world data traces collected from the videos shared in RenRen, one of the largest online social networks in China. These experiments show that our proposed method outperforms existing approaches for popularity prediction (which do not take into account the propagation in social network) by more than 30% in terms of prediction rewards.**

I. Introduction

Social networks are recently being used to inform predictions and decisions in a variety of application domains, ranging from live or on-demand event broadcasting, to security and surveillance [1], to disaster management [2], to economic forecasting [3]. In all these applications, forecasting the popularity of the content shared in a social network is vital due to a variety of reasons. For network and cloud service providers, accurate forecasting facilitates prompt and adequate reservation of computation, storage, and bandwidth resources [4], thereby ensuring smooth and robust content delivery at low costs. For advertisers, accurate and timely popularity prediction provides a good revenue indicator, thereby enabling targeted ads to be composed for specific videos and viewer demographics. For content producers and contributors, attracting a high number of views is paramount for attracting potential revenue through micro-payment mechanisms.

While popularity prediction is a long-lasting research topic [5] [6] [7] [8], understanding how social networks affect the popularity of the media content and using this understanding to make better forecasts poses significant new challenges. Conventional prediction tools have mostly relied on the history of the past view counts, which worked well when the recipients were generally passive. In contrast, social network users are proactive in terms of the content they watch and are heavily influenced by their social network interactions; for instance, the recipient of a certain media content may further forward it or not, depending on not only its attractiveness, but also the social network in which this content was generated and propagated [9]. For example, the latest measurement on Twitter's Vine, a highly popular short mobile video sharing service, has suggested that the popularity of a short video indeed depends less on the content itself, but more on the contributor's position in the social network [10]. Hence, considering the content initiator's information and the friendship network of the sharers can clearly improve the accuracy of the popularity forecasts. However, critical new questions need to be answered: how to use the information extracted from social networks to improve popularity forecasts and how to utilize the dynamically changing and evolving information about the social networks to update the popularity forecasts?

As social networks become increasingly more ubiquitous and influential, users' sharing behavior dynamically change and evolve as well. Offline prediction tools [5] [11] [12] [13] [8] depend on specific training datasets, which may be biased or outdated, and hence may not accurately capture the real-world propagation patterns in social networks. Moreover, popularity forecasting is a *multi-stage* rather than a single-stage task since each video may be propagated through a cascaded social network for a relatively long time and thus, the forecast (and its subsequent decision such as ad placement and cache reservation) can be made at any time while the video is being propagated. A fast prediction has important economic and technological benefits; however, too early a prediction may lead to a low accuracy that is less useful or even damaging (e.g. investment in videos that will not actually become popular). The timeliness of the prediction has yet to be considered in existing works [5] [11] [12] [13] [8] which solely focus on maximizing the accuracy at a given reference time point. Hence, we strongly believe that developing a systematic methodology for accurate and timely popularity forecasting is essential.

In this paper, we propose for the first time a systematic methodology and associated online algorithm for forecasting popularity of videos promoted by social networks, which jointly consider the accuracy and timeliness of the forecast.

To this end, we explicitly consider the unique situational conditions that affect the video propagated in social networks, and demonstrate how this *context information* can be incorporated to improve the accuracy of the forecasts. The unique features of Pop-Forecast as well as our key contributions are summarized below:

- We rigorously formulate the online popularity forecasting problem as a multi-stage sequential decision and online learning problem. Our solution, the Pop-Forecast algorithm, makes popularity prediction in an online fashion, requiring no *a priori* training phase or dataset. It exploits the dynamically changing and evolving video propagation patterns through social networks to maximize the prediction reward. The algorithm is easily tunable to enable tradeoffs between the accuracy and timeliness of the forecasts as required by various applications, entities and/or deployment scenarios.
- We analytically quantify the regret of Pop-Forecast, that is, the performance gap between its expected reward and that of the best prediction policy which can be only obtained by an omniscient oracle having complete statistical knowledge of the video popularity trends. We prove that the regret is sublinear in the number of video arrivals, which implies that the expected prediction reward asymptotically converges to the optimal expected reward. The upper bound on regret also gives a lower bound on the convergence rate to the optimal average reward.
- We validate Pop-Forecast's performance through extensive experiments with real-world data traces from RenRen (the largest Facebook-like online social network in China). The results show that significant improvement can be achieved by exploiting the situational meta-data associated with the video and its propagation through the social network.

The rest of the paper is organized as follows. Section II discusses related works. In Section III, we describe the system model and rigorously formulate the online popularity forecast problem. In Section IV, we propose the online learning algorithm and prove that it achieves sublinear regret bounds. Section V discusses the experimental results and our findings. Section VI concludes this paper.

II. RELATED WORKS

In this section, we review the representative related works from both the application and the theoretical foundation perspectives.

A. Popularity Prediction for Online Content

Popularity prediction of online content has been extensively studied in the literature. Early works have focused on predicting the future popularity of content (e.g. video) on conventional websites such as YouTube. Various solutions are proposed based on time series models like ARIMA (Autoregressive integrated moving average) [14] [15], regression models [16] [17] and classification models [16] [18]. In [5], training sets are used to establish log-linear correlations between early views and the later views. An improved algorithm is proposed in [8] by incorporating the views up to a reference date rather than simply using the total view count. These methods are generally view-based, meaning that the prediction of the future views is solely based on the early views. While they provide satisfactory performance for YouTube-like accesses, their performance is largely unacceptable [9] when applied to predicting popularity in the social network context.

Recently, different social network features are exploited to assist popularity prediction [12] [13] [19] [9] [20]. For instance, in [12], the retweets prediction on Twitter is modeled as a classification problem, and a variety of context-aware features are investigated. For predicting the popularity of news in Digg, such aspects as website design have been incorporated [13], and for predicting the popularity of short messages, the structural characteristics of social networks have been used [19]. For video sharing in social networks, [9] has identified a series of context-aware factors which influence the propagation patterns. In [20], knowledge from social streams is utilized to predict sudden popularity bursts in online content.

Our work in this paper is motivated by these studies, but it is the first systematic solution for forecasting video popularity by exploiting social network characteristics. First, existing works are mostly measurement-based and their solutions generally work offline, requiring existing training data sets. Instead, Pop-Forecast operates entirely online and does not require any a priori gathered training data set. Second, Pop-Forecast inherently adapt on-the-fly to the underlying social network structure and user sharing behavior. Last but not least, unlike the early empirical studies which employ only simulations to validate the performance of their predictions, we can rigorously prove performance bounds for Pop-Forecast.

B. Contextual Bandits Learning

Our forecasting algorithm is closely related to the contextual bandits framework [21] [22] [23] [24] [25] but with significant differences due to the unique features of the online prediction problem. First, most of the prior work on contextual bandits is focused on an agent making a single-stage decision based on the provided context information for each incoming instance. Instead, in this paper, the agent needs to make a sequence of decisions at multiple stages for each video. The reward obtained by selecting an action at one stage depends on the actions chosen at other stages and thus, rewards and actions at different stages are coupled. Second, unlike bandit problems in which the estimated rewards of an action can be updated only after the action is selected, since the prediction action does not affect the underlying popularity evolution, rewards can be computed and updated even for actions that are not selected. In particular, we update the reward of an action as if it was selected. Therefore, actions with the best estimated rewards are always selected, thereby improving the learning performance.

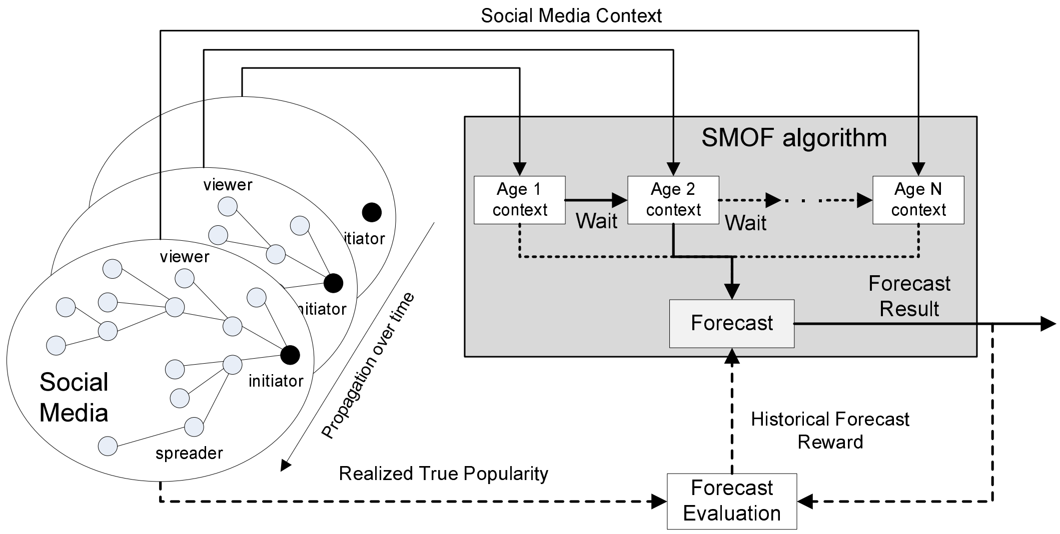

Fig. 1. System diagram.

III. System Model

A. Sharing Propagation and Popularity Evolution

We consider a generic Web 2.0 information sharing system in which videos are shared by users through social networks (see Figure 1 for a system diagram). We assign each video with an index $k \in \{1, 2, ..., K\}$ according to the absolute time t_{init}^k when it is initiated[1]. Once a video is initiated, it will be propagated through the social network for some time duration. We assume a discrete time model where a period can be minutes, hours, days, or any suitable time duration. A video is said to have an age of $n \in \{1, 2, ...\}$ periods if it has been propagated through the social network for n periods. In each period, the video is further shared and viewed by users depending on the sharing and viewing status of the previous period. The propagation characteristics of video k up to age n are captured by a d_n-dimensional vector $\boldsymbol{x}_n^k \in \mathcal{X}_n$ which includes information such as the total number of views and other contextual information such as the characteristics of the social network over which the video was propagated. The specific characteristics that we use in this paper will be discussed in Section V. In this section, we keep $\boldsymbol{x}_n^k$ in an abstract form and call it succinctly the *context information* at age n.

Several points regarding the context information are noteworthy. First, the context space $\mathcal{X}_n$ can be different at different ages n. In particular, $\boldsymbol{x}_n^k$ can include all history information of video k's propagation characteristics up to age n and hence $\boldsymbol{x}_n^k$ includes all information of $\boldsymbol{x}_m^k, \forall m < n$. Thus the type of contextual information is also age-dependent. Second, $\boldsymbol{x}_n^k$ can be taken from a large space, e.g. a finite space with a large number of values or even an infinite space. For example, some dimensions of $\boldsymbol{x}_n^k$ (e.g. the Sharing Rate used in Section V) take values from a continuous value space and $\boldsymbol{x}_n^k$ may include all the past propagation characteristics (e.g. $\boldsymbol{x}_m^k \in \boldsymbol{x}_n^k, \forall m < n$). Third, at age n, $\boldsymbol{x}_m^k, \forall m > n$ are not yet revealed since they represent future information which is yet to be realized. Hence, given the context information $\boldsymbol{x}_n^k$ at age n, the future context information $\boldsymbol{x}_m^k, \forall m > n$ are random variables.

We are interested in predicting the future popularity status of the video by the end of a pre-determined age N, and we aim to make the prediction as soon as possible. The choice of N depends on the specific requirements of the content provider, the advertiser and the web hosts. In this paper, we will treat N as given[2]. Thus, the context information for video k during its lifetime of N periods is collected in $\boldsymbol{x}^k = (\boldsymbol{x}_1^k, \boldsymbol{x}_2^k, ..., \boldsymbol{x}_N^k)$. For expositional simplicity, we also define $\boldsymbol{x}_{n^+} = (\boldsymbol{x}_{n+1}, ..., \boldsymbol{x}_N)$, $\boldsymbol{x}_{n^-} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_{n-1})$ and $\boldsymbol{x}_{-n} = (\boldsymbol{x}_{n^-}, \boldsymbol{x}_{n^+})$.

Let $\mathcal{S}$ be the popularity status space, which is assumed to be finite. For instance, $\mathcal{S}$ can be either a binary space {Popular, Unpopular} or a more refined space containing multiple levels of popularity such as {Low Popularity, Medium Popularity, High Popularity} or any such refinement. We let s^k denote the popularity status of video k by the end of age N. Since s^k is realized only at the end of N periods, it is a random variable at all previous ages. However, the conditional distribution of s^k will vary at different ages since they are conditioned on different context information. In many scenarios, the conditional distribution at a higher age n is more informative for the future popularity status since more contextual information has arrived. Nevertheless, our model does not require this assumption to hold.

[1] It is easy to assign unique identifiers if multiple videos which are generated/initiated at the same time.

[2] This assumption is generally valid given that the video sharing events have daily and weekly patterns, and the active lifespans of most shared videos through social networks are quite limited

B. Prediction Reward

For each video k, at each age $n = 1, ..., N$, we can make a prediction decision $a_n^k \in \mathcal{S} \cup \{\text{Wait}\}$. If $a_n^k \in \mathcal{S}$, we predict a_n^k as the popularity status by age N. If $a_n^k = \text{Wait}$, we choose to wait for the next period context information to decide (i.e. predict a popularity status or wait again). For each video k, at the end of age N, given the decision action vector $\boldsymbol{a}^k$, we define the *age-dependent reward* r_n^k at age n as follows,

$$r_n^k = \begin{cases} U(a_n^k, s^k, n), & \text{if } a_n^k \in \mathcal{S} \\ r_{n+1}^k, & \text{if } a_n^k = \text{Wait} \end{cases} \tag{1}$$

where $U(a_n^k, s^k, n)$ is a reward function depending on the accuracy of the prediction (determined by a_n^k and the realized true popularity status s^k) and the timeliness of the prediction (determined by the age n when the prediction is made).

The specific form of $U(a_n^k, s^k, n)$ depends on how the reward is derived based on the popularity prediction based on various economical and technological factors. For instance, the reward can the ad revenue derived from placing proper ads for potential popular videos or the cost spent for adequately planning computation, storage, and bandwidth resources to ensure the robust operation of the video streaming services. Even though our framework allows any general form of the reward function, in our experiments (Section VI), we will use a reward function that takes the form of $U(a_n^k, s^k, n) = \theta(a_n^k, s^k) + \lambda\psi(n)$ where $\theta(a_n^k, s^k)$ measures the prediction accuracy, $\psi(n)$ accounts for the prediction timeliness and $\lambda > 0$ is a trade-off parameter that controls the relative importance of accuracy and timeliness.

Let n^* be the first age at which the action is not "Wait" (i.e. the first time a forecast is issued). The *overall prediction reward* is defined as the $r^k = r_{n^*}^k$. According to equation (1), when the action is "Wait" at age n, the reward is the same as that at age $n+1$. Thus $r_1^k = r_2^k = ... = r_{n^*}^k$. This suggests that the overall prediction reward is the same as the age-dependent reward at age 1, i.e. $r^k = r_1^k$. For age $n > n^*$, the action a_k^n and the age-dependent reward r_n^k do not affect the realized overall prediction result since a prediction has already been made. However, we still select actions and compute the age-dependent reward since it helps learning the best action and the best reward for this age n which in turn will help decide whether or not we should wait at an early age. Figure 2 provides an illustration on how the actions at different ages determine the overall prediction reward.

Remark: The prediction action itself does not generate rewards. It is the action (e.g. online ad investment) taken using the prediction results that is rewarding. In many scenarios, this action can only be taken once and cannot be altered afterwards. This motivates the above overall reward function formulation in which the overall prediction reward is determined by the first non-"Wait" action. Nevertheless, our framework can also be easily extended to account for more general overall reward functions which may depend on all non-"Wait" actions. For instance, the action may be revised when a more accurate later prediction is made. In this case, the reward function

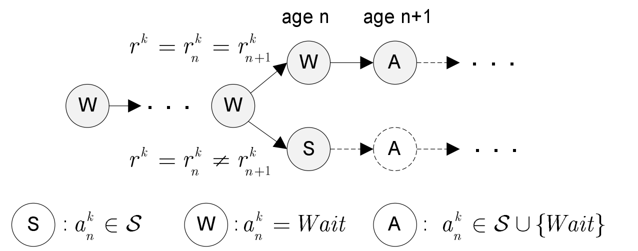

Fig. 2. An illustration for the multi-stage decision making. The first $n-1$ action is "Wait". If the age-n action is "Wait", then $r_n^k = r_{n+1}^k$ which depends on later actions. If the age-n action is not "Wait", then $r_n^k \neq r_{n+1}^k$ and r^k does not depend on later actions. However, we can still learn the reward of action at age $n+1$ as if all actions before $n+1$ were "Wait".

$U(a_n^k, s^k, n)$ in (1) will depend on not only the current prediction action $a_n^k \in \mathcal{S}$ but also all non-"Wait" actions after age n. We will use the reward function in (1) because of its simplicity for the exposition but our analysis also holds for general reward functions.

C. Prediction Policy

In this paper, we focus on prediction policies that depend on the current contextual information. Let $\pi_n : \mathcal{X}_n \to \mathcal{S} \cup \{\text{Wait}\}$ denote the prediction policy for a video link of age n and $\pi = (\pi_1, ..., \pi_N)$ be the complete prediction policy. Hence, a prediction policy π prescribes actions for all possible context information at all ages. For expositional simplicity, we also define $\pi_{n^+} = (\pi_{n+1}, ..., \pi_N)$ as the policy vector for ages greater than n, $\pi_{n^-} = (\pi_1, ..., \pi_{n-1})$ as the policy vector for ages smaller than n and $\pi_{-n} = (\pi_{n^-}, \pi_{n^+})$. For a video with context information $\boldsymbol{x}^k$, the prediction policy π determines the prediction action at each age and hence the overall prediction reward, denoted by $r(\boldsymbol{x}|\pi)$, as well as the age-dependent rewards $r_n(\boldsymbol{x}|\pi), \forall n = 1, ..., N$. Let $f(\boldsymbol{x})$ be the probability distribution function of the video context information, which also gives information of the popularity evaluation patterns. The expected prediction reward of a policy π is therefore,

$$V(\pi) = \int_{\boldsymbol{x} \in \mathcal{X}} r(\boldsymbol{x}|\pi) f(\boldsymbol{x}) d\boldsymbol{x} \tag{2}$$

Note that the age-n policy π_n will only use the context information $\boldsymbol{x}_n$ rather than $\boldsymbol{x}$ to make predictions since $\boldsymbol{x}_{n^+}$ has not been realized at age n.

Our objective is to determine the optimal policy π^{opt} that maximizes the expected prediction reward, i.e. $\pi^{opt} = \arg\max_{\pi} V(\pi)$. If the distribution $f(\boldsymbol{x})$ were known, then we can compute the optimal policy π^* using (2). However, in practice $f(\boldsymbol{x})$ represents unknown information and thus, we will propose a systematic methodology and associated algorithms that find the optimal policy online.

IV. Learning the Optimal Forecasting Policy

In this section, we develop a learning algorithm to determine the optimal prediction policy without any prior knowledge of the underlying context distribution $f(\boldsymbol{x})$. In the considered scenario, videos arrive to the system in sequence[3] and we will make popularity prediction based on past experiences by exploiting the similarity information of videos. We will first focus mainly on learning the period-n policy π_n for fixed policies π_{-n} for other periods. Then using a backward induction argument, we can show that the complete optimal policy can be learned asymptotically.

A. Performance Metric

In this subsection, we define the performance metric of our learning algorithm. Let σ_n be a learning algorithm of π_n which takes action $\sigma_n^k(\boldsymbol{x}_n^k)$ at instance k. We will use learning regret to evaluate the performance of a learning algorithm. Since we focus on π_n, we will use simplified notations in this section by neglecting π_{-n}. However, keep in mind that the age-n prediction reward depends on actions at all later ages a_{n^+} besides a_n when $a_n =$ Wait. Let $\mu_n(\boldsymbol{x}_n|a_n)$ denote the expected reward when age-n context information is $\boldsymbol{x}_n$ and the algorithm takes the action $a_n \in \mathcal{S} \cup \{\text{Wait}\}$.

The optimal action given a context $\boldsymbol{x}_n$ is therefore, $a^*(\boldsymbol{x}_n) = \arg\max_{a_n} \mu_n(\boldsymbol{x}_n|a_n)$ (with ties broken deterministically) and the optimal expected reward is $\mu_n^*(\boldsymbol{x}_n) = \mu_n(\boldsymbol{x}_n|a_n^*)$. Let $\Delta = \max_{\boldsymbol{x}_n \in \mathcal{X}_n}\{\mu_n^*(\boldsymbol{x}_n) - \mu_n(\boldsymbol{x}_n|a_n \neq a_n^*)\}$ be the maximum reward difference between the optimal action and the non-optimal action over all context $\boldsymbol{x}_n \in \mathcal{X}_n$. Finally, we let $r_n(\boldsymbol{x}_n^k|\sigma_n^k)$ be the realized age-n reward for video k by using the learning algorithm σ. The expected regret by adopting a learning algorithm σ_n is defined as

$$R_n(K) = \mathbb{E}\{\sum_{k=1}^{K} \mu_n^*(\boldsymbol{x}_n^k) - \sum_{k=1}^{K} r_n(\boldsymbol{x}_n^k|\sigma_n^k)\} \tag{3}$$

Our online learning algorithm will estimate the prediction rewards by selecting different actions and then choose the actions with best estimates based on past experience. One way to do this is to record the reward estimates without using the contextual information. However, this could be very inefficient since for different contexts, the optimal actions can be very different. Another way is to maintain the reward estimates for each individual context $\boldsymbol{x}_n$ and select the action only based on these estimates. However, since the context space $\mathcal{X}_n$ can be very large, for a finite number K of video instances, the number of videos with the same context $\boldsymbol{x}_n$ is very small. Hence it is difficult to select the best action with high confidence. Our learning algorithm will exploit the similarity of contexts, adaptively partition the context space into smaller subspaces and learn the optimal action within each subspace.

B. Online Popularity Prediction with Adaptive Partition

In this subsection, we propose the online prediction algorithm with adaptive partition (Adaptive-Partition) that adaptively partitions the context space according to the context arrivals. This will be the key module of the Pop-Forecast algorithm. For analysis simplicity, we normalize the context space to be $\mathcal{X}_n = [0,1]^d$. We call a d-dimensional hypercube which has sides of length 2^{-l} a level l hypercube. Denote the partition of $\mathcal{X}_n$ generated by level l hypercubes by $\mathcal{P}_l$. We have $|\mathcal{P}_l| = 2^{ld}$. Let $\mathcal{P} := \cup_{l=0}^{\infty}\mathcal{P}_l$ denote the set of all possible hypercubes. Note that $\mathcal{P}_0$ contains only a single hypercube which is $\mathcal{X}_n$ itself. For each instance arrival, the algorithm keeps a set of hypercubes that cover the context space which are mutually exclusive. We call these hypercubes *active* hypercubes, and denote the set of active hypercubes at instance k by $\mathcal{A}_k$. Clearly, we have $\cup_{C\in\mathcal{A}_k} = \mathcal{X}_n$. Denote the active hypercube that contains $\boldsymbol{x}_n^k$ by C_k. Let $M_{C_k}(k)$ be the number of times context arrives to hypercube C_k by instance k. Once activated, a level l hypercube C will stay active until the first instance k such that $M_{C_k}(k) \geq A2^{pl}$ where $p > 0$ and $A > 0$ are algorithm design parameters. When a hypercube C_k of level l becomes inactive, the hypercubes of level $l+1$ that constitute C_k, denoted by $\mathcal{P}_{l+1}(C_k)$, are then activated.

When a context $\boldsymbol{x}_n^k$ arrives, we first check to which active hypercube $C_k \in \mathcal{A}_k$ it belongs. Then we choose the action with the highest reward estimate $a_n = \arg\max_a \bar{r}_{a,C_k}(k)$, where $\bar{r}_{a,C_k}(k)$ is the sample mean of the rewards collected from action a in C_k which is an activated hypercube at instance k. When the prediction reward is realized for instance k (i.e. at the end of age N), we perform a *virtual update* for the reward estimates for all actions. The reason why we can perform such a virtual update for actions which are not selected is because the context transition over time is independent of our prediction actions and hence, the reward by choosing any action can still be computed even though it is not realized.

Algorithm 1 provides a formal description for the Adaptive-Partition algorithm. Figure 3 illustrates the adaptive partition process of Adaptive-Partition algorithm. Next, we bound the regret by running the Adaptive-Partition algorithm.

Algorithm 1 Adaptive-Partition Algorithm

```
Initialize A_1 = P_0, M_C(0) = 0, r̄_{a,C}(0) = 0, ∀a, ∀C ∈ P.
for each video instance k do
    Determine C ∈ A_k such that x_n^k ∈ C.
    Select a_n = arg max_a r̄_{a,C}(k).
    After the prediction reward is realized, update r̄_{a,C}(k+1) for all a.
    Set M_C(k) ← M_C(k−1) + 1.
    if M_C(k) ≥ A2^{pl} then
        Set A_{k+1} = (A_k \ C) ∪ P_{l+1}(C)
    end if
end for
```

C. Theoretical Performance Result

To facilitate our analysis, we make a widely adopted assumption [22] [23] [24] that the expected reward of an action is similar for similar contextual and situational information; we formalize this in terms of (uniform) Lipschitz condition.

[3]To simplify our analysis, we will assume that one video arrives at one time. Nevertheless, our framework can be easily extended to scenarios where multiple videos arrive at the same time.

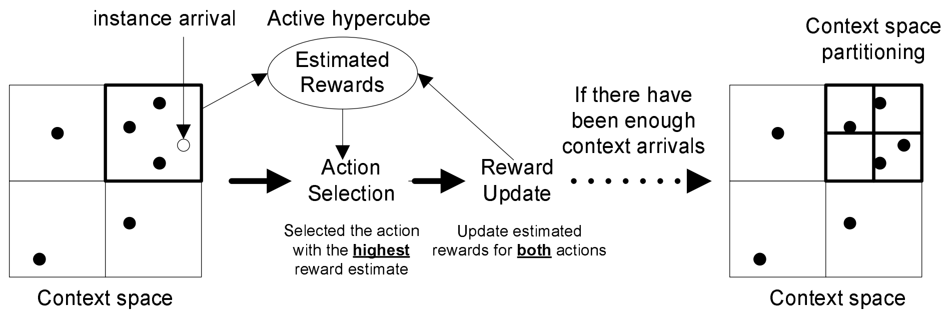

Fig. 3. The context space partitioning of the Adaptive-Partition algorithm.

Assumption. *(Lipschitz) For each $a_n \in \mathcal{S} \cup \{Wait\}$, there exists $L > 0, \alpha > 0$ such that for all $\boldsymbol{x}_n, \boldsymbol{x}'_n \in \mathcal{X}_n$, we have $|\mu(\boldsymbol{x}_n|a_n) - \mu(\boldsymbol{x}'_n|a_n)| \leq L\|\boldsymbol{x}_n, \boldsymbol{x}'_n\|^\alpha$.*

In order to get the regret bound of the Adaptive-Partition algorithm, we need to consider how many hypercubes of each level is formed by the algorithm up to instance K. The number of such hypercubes explicitly depends on the context arrival process. Therefore, we investigate the regret for different context arrival scenarios.

Definition. *We call the context arrival process the* **the worst-case arrival process** *if it is uniformly distributed inside the context space, with minimum distance between any two context samples being $K^{-1/d}$, and* **the best-case arrival process** *if $\boldsymbol{x}^k \in C, \forall k$ for some level $\lceil(\log_2(K)/p\rceil + 1$ hypercube C.*

In the theorem below, we determine the finite time, uniform regret bound for the Adaptive-Partition algorithm [4].

Theorem.
- *For the worst case arrival process, $R_n(K) = O(K^{\frac{d+\alpha/2+\sqrt{9\alpha^2+8\alpha d}/2}{d+3\alpha/2+\sqrt{9\alpha^2+8\alpha d}/2}})$.*
- *For the best case arrival process, then $R_n(K) = O(K^{2/3})$.*

The regret bounds proved in Theorem 1 are sublinear in K which guarantee convergence in terms of the average reward, i.e. $\lim_{K\to\infty} \mathbb{E}[R_n(K)]/K = 0$. Thus our online prediction algorithm makes the optimal predictions as sufficiently many videos instances have been seen. More importantly, the regret bound tells how much reward would be lost by running our learning algorithm for any finite number K of videos arrivals. Hence, it provides a rigorous characterization on the learning speed of the algorithm.

D. Learning the Complete Policy π

We now present in Algorithm 2 the Pop-Forecast algorithm that learns the complete policy. Pop-Forecast learns all age-dependent policies $\pi_n, \forall n$ simultaneously. For a given age n, since π_{-n} is not fixed to be the optimal policy π_{-n}^{opt} during the learning process, the learned policy π_n may not be the global optimal π_n^{opt}. However, in order to determine π_n^{opt}, only the policies for ages greater than n, i.e. $\pi_{n^+}^{opt}$ need to be determined. Thus even though we are learning $\pi_n, \forall n$ simultaneously, the learning problem of π_N is not affected and hence, π_N^{opt} will be learned with high probability after a sufficient number of video arrivals. Once π_N^{opt} is learned with high probability, π_{N-1}^{opt} can also be learned with high probability after an additional number of video arrivals. By this induction, such a simultaneous learning algorithm can still learn the global optimal complete policy with high probability. In the experiments we will show the performance of this algorithm in practice.

[4]The complete regret analysis and proofs can be found in the online appendix at https://www.dropbox.com/s/53mo6tgyx7e244m/Infocom_proof.pdf

Algorithm 2 Pop-Forecast Algorithm

```
for each video instance k do
    for each age n = 1 to N do
        Get context information x_n^k.
        Select a_n^k according to Adaptive-Partition.
        Perform context partition using Adaptive-Partition.
    end for
    Popularity status s^k is realized.
    for each age n = 1 to N do
        Compute the age-dependent reward r_n^k.
        Update reward estimates using Adaptive-Partition.
    end for
end for
```

E. Complexity of Pop-Forecast

For each age of one video instance arrival, Pop-Forecast needs to do one comparison operation and one update operation on the estimated reward of each forecast action. It also needs to update the counting of context arrivals to the current context subspace and perform context space partitioning if necessary. In sum, the time complexity has the order $O(|\mathcal{S}|N)$ for each video instance and $O(|\mathcal{S}|NK)$ for K video arrivals. Since the maximum age N of interest and the popularity status space is given, the time complexity is linear in the number of video arrivals K. The Pop-Forecast algorithm maintains for each *active* context subspace reward estimates of all forecast actions. Each partitioning creates $2^d - 1$ more *active* context subspaces and the number of partitioning is at most K/A. Thus the space complexity for K video arrivals is at most $O(2^d NK/A)$. Since the context space dimension d

and the algorithm parameter A are given and fixed, the space complexity is at most linear in the number of video arrivals K.

V. Video Propagation Characteristics

To instantiate the parameters of the proposed Pop-Forecast algorithm in real world, we have collaborated closely with RenRen, which is a highly popular social network in China, to examine the unique propagation characteristics of videos promoted by social networks. Today RenRen owns 160 million registered users, attracting 31 million monthly user accesses. RenRen shares many features with Facebook and also enables micro-blog features similar to those in Twitter, and its data structure allows for tracing shared video information.

A RenRen user can post a link to a video taken by him/herself or from an external video sharing website such as Youtube. The user, referred to as an *initiators*, then starts the sharing process. The friends of these initiators can find this video in their "News Feed". Some of them may watch this video and some may re-share the video to their own friends. We call the users who watched the shared video *viewers* and those who re-shared the video *spreaders*. Since spreaders generally watched the video before re-shared it, most of them are also viewers. In the experiment, we will use two characteristics of videos promoted by social networks as the context information for our algorithm. The first is the initiator's *Branching Factor (BrF)*, which is the number of viewers who directly follow the initiator. The second is the *Share Rate (ShR)*, which is the ratio of the viewers that re-share the video after watching it. Note that the values of these two parameters are changing during the propagation period of the corresponding video. Figure 4 shows the evolution of the number of views, the BrF and the ShR for three representative videos over 40 periods. Among these three videos, video 1 is an unpopular video while video 2 and video 3 are popular videos. We analyze the differences between popular and unpopular videos as follows.

- *Video 1 vs Video 2*. The ShRs of both videos are similar. The BrF of video 2 is much larger than that of video 1. This indicates that video 2 may be initiated by users with a large number of friends, e.g. celebrities and pubic accounts. Thus, videos with larger BrF potentially will achieve popularity in the future.
- *Video 1 vs Video 3*. The BrFs of both videos are low (at least before video 3 becomes popular). Video 3 has a much larger ShR than video 1. This indicates that video 3 is being shared with high probability and thus, videos with larger ShR will potentially become popular in the future.

The above analysis shows that BrF and ShR are good metrics for videos promoted by social networks. Therefore we will use these two metrics in addition to the total and per-period view counts as the input to our proposed online prediction algorithms. Nevertheless, our algorithms are general enough to take other metrics to further improve the prediction

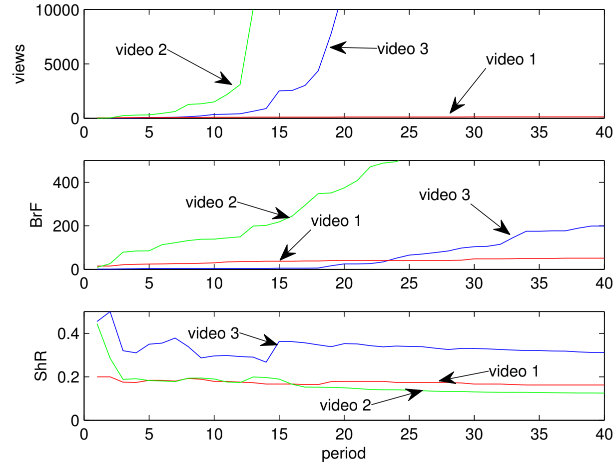

Fig. 4. Popularity evolution of 3 representative videos.

performance, e.g. the type of the videos, the number of spreaders, and other metrics reflecting the propagation topology etc.

VI. Experiments

A. Experiment set-up

In this section we evaluate the performance of the proposed Pop-Forecast algorithm using the metrics discussed in Section V based on the RenRen data. We set one period to be 5 hours and are interested in predicting the video popularity by 40 periods after the initiation of the video. In most of our experiments, we will consider a binary popularity status space {Popular, Unpopular} where "Popular" is defined for videos whose total number of views exceeds a predetermined threshold. This predetermined threshold will be varied in experiments. However, we also conduct experiments on a more refined popularity status space in Section VI(C).

The prediction reward function that we use is $U(a_n^k, s^k, n) = \theta(a_n^k, s^k) + \lambda\psi(n)$. For the case of binary popularity status space, the accuracy reward function θ is chosen as follows

$$\theta(a_n^k, s^k) = \begin{cases} 1, & \text{if } a_n^k = s^k = \text{Unpopular} \\ w, & \text{if } a_n^k = s^k = \text{Popular} \\ 0, & \text{if } a_n^k \neq s^k \end{cases} \quad (4)$$

where $w > 0$ is fixed reward for correctly predicting popular videos and hence controls the relative importance of true positive and true negative. The timeliness reward function ψ is simply taken as $\psi(n) = N - n$. Recall that the prediction reward function is a combination of the two and we use $\lambda > 0$ to trade-off accuracy and timeliness. In the experiments, we will vary both w and λ to investigate their impacts on the prediction performance. Note that we use these specific reward functions in this experiment but other reward functions can easily be adopted in our algorithm.

We will compare the performance of our online prediction algorithm with two benchmark solutions. The first benchmark is proposed by Szabo and Huberman [5], labeled as SH. The second benchmark is proposed by Pinto, Almedia and

TABLE I
COMPARISON OF NORMALIZED PREDICTION REWARD WITH VARYING w

w	SH5	SH10	SH15	PAG5	PAG10	PAG15	PF
1	0.82	0.80	0.78	0.71	0.79	0.78	0.92
2	0.82	0.80	0.80	0.77	0.82	0.82	0.93
3	0.81	0.80	0.81	0.80	0.84	0.85	0.94

TABLE II
COMPARISON OF PREDICTION ACCURACY WITH VARYING w

w	SH5	SH10	SH15	PAG5	PAG10	PAG15	PF
2	0.80	0.83	0.84	0.64	0.81	0.85	0.94

Goncalves [8], labeled as PAG. We normalized the prediction rewards of these algorithms to the perfect prediction scheme that makes accurate prediction at age 1 for all videos.

B. Performance comparison

In this subsection, we compare the prediction performance of our proposed Pop-Forecast algorithm (labeled as PF for short) with the benchmarks. This set of experiments are carried out on a set of 5000 video links. The videos were initiated in sequence and thus, initially we do not have any knowledge of the videos or video popularity evolution patterns. For the SH and PAG algorithms, we use three versions, labeled as SH5 (PAG5), SH10 (PAG10), SH15 (PAG15), in which the forecast is made at age 5, 10, 15, respectively.

Table I records the normalized forecast rewards obtained by our proposed algorithm and the benchmarks for $\lambda = 0.01$ and $w = 1, 2, 3$ when the popularity threshold is 10000 views. The trade-off parameter λ for accuracy and timeliness is set to be small because the lifetime N is large. Table II further shows the forecast accuracy obtained by Pop-Forecast and the benchmark solutions for $\lambda = 0.01$ and $w = 2$. We have the following observations: 1) The accuracies of both SH and PAG are increasing in the reference age when the forecast is made. It implies that having more information is helpful for the prediction. The prediction rewards of both SH and PAG are relatively insensitive to the time when the forecast is made. This is because even though accuracy improves when the reference age is large, the prediction timeliness decreases. These two effects almost balance out in our experiments. 2) The proposed Pop-Forecast algorithm significantly outperforms SH and PAG in terms of both accuracy and reward. This shows that Pop-Forecast is able to predict accurately and in a timely manner.

Next, we fix w and vary λ. Table III records the normalized prediction rewards obtained by our proposed algorithm and the benchmarks for $w = 1$ and $\lambda = 0.01, 0.015, 0.02$. Several points are worth discussing: 1) The rewards obtained by both SH and PAG are decreasing in λ. This suggests that the rewards are mainly derived from prediction accuracy but they are not able to make the prediction in a timely manner. 2) Our proposed Pop-Forecast algorithm significantly outperforms all other benchmark algorithms and achieves close-to-optimal rewards for all values of λ. Thus, it makes optimal tradeoff between accuracy and timeliness.

TABLE III
COMPARISON OF NORMALIZED PREDICTION REWARD WITH VARYING λ

λ	SH5	SH10	SH15	PAG5	PAG10	PAG15	PF
0.01	0.82	0.80	0.78	0.71	0.79	0.78	0.92
0.015	0.82	0.79	0.76	0.73	0.79	0.76	0.92
0.02	0.83	0.79	0.75	0.74	0.78	0.75	0.91

TABLE IV
COMPARISON OF NORMALIZED PREDICTION REWARD AND ACCURACY WITH VARYING POPULARITY THRESHOLD (IN EACH ENTRY, THE FIRST NUMBER IS THE REWARD, THE SECOND NUMBER IS THE ACCURACY)

	SH10	PAG10	PF
1e4	0.80, 0.83	0.79, 0.81	0.92, 0.94
3e4	0.81, 0.82	0.86, 0.90	0.94, 0.96
5e4	0.84, 0.87	0.85, 0.89	0.95, 0.96

We then vary the popularity threshold. Table IV reports the prediction rewards and accuracies for different thresholds 10000, 30000, 50000 for SH10, PAG10 and Pop-Forecast by fixing $\lambda = 0.01$ and $w = 1$. As the popularity threshold increases, the rewards and accuracies obtained by the Pop-Forecast algorithm and the benchmark solutions all increase. In particular, the PAG algorithm has a significant increase in the prediction accuracy. This suggests that these benchmark algorithms have better accuracy in videos with a large number of views.

C. More refined popularity prediction

In the previous experiments, we considered a binary popularity status space. Nevertheless, our proposed popularity prediction methodology and associated algorithm can also be applied to predict popularity in a more refined space. In this experiment, we consider a refined popularity status space {High Popularity, Medium Popularity, Low Popularity} where "High Popularity" is defined for videos with more than 50000 views, "Medium Popularity" for videos with views between 10000 and 50000, and "Low Popularity" for videos with views below 10000. Table V illustrates the normalized rewards obtained by different algorithms for $\lambda = 0.01, 0.015, 0.02$. Table VI reports the predication accuracy for $\lambda = 0.02$. It can been seen from the table that the rewards obtained by all algorithms decrease compared with the binary popularity status case since prediction becomes more difficult. However, the performance improvement of Pop-Forecast against SH and PAG becomes even larger. This suggests that our algorithm, which explicitly considers the contextual information associated with the social network, is able to achieve a higher performance gain against the benchmark approaches for more refined popularity prediction.

VII. CONCLUSION

In this paper, we have proposed a novel, systematic and highly-efficient online popularity forecasting algorithm for videos promoted by social networks. We have shown that by incorporating the contextual information of the social network, the forecasts can be significantly more accurate than those

TABLE V
COMPARISON OF NORMALIZED PREDICTION REWARD FOR TERNARY POPULARITY LEVELS.

λ	SH5	SH10	SH15	PAG5	PAG10	PAG15	PF
0.01	0.71	0.72	0.70	0.66	0.74	0.73	0.89
0.015	0.73	0.72	0.69	0.69	0.74	0.71	0.88
0.02	0.75	0.72	0.68	0.71	0.74	0.70	0.87

TABLE VI
COMPARISON OF PREDICTION ACCURACY FOR REFINED POPULARITY SPACE.

SH5	SH10	SH15	PAG5	PAG10	PAG15	PF
0.63	0.70	0.73	0.55	0.74	0.77	0.90

obtained based on the existing approaches which disregard this information and only consider the number of times that videos have been viewed so far. The proposed Pop-Forecast algorithm can operate easily and successfully in online, dynamically-changing environments such as social networks. We have systematically proven sublinear regret bounds on the performance loss incurred by our algorithm due to online learning, thereby guaranteeing its asymptotic convergence to the optimal performance as well as its fast learning rate. Importantly, Pop-Forecast can easily be extended to make popularity predictions for various types of content and using the information of various social media.

ACKNOWLEDGMENT

J. Xu and M. van der Schaar's research is supported by the US Air Force Office of Scientific Research under the DDDAS Program. J. Liu and H. Li's research is supported by an NSERC Discovery Grant.

REFERENCES

[1] Daniel Trottier, "Social media as surveillance," *Farnham: Ashgate*, 2012.
[2] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 851–860.
[3] Hyunyoung Choi and Hal Varian, "Predicting the present with google trends," *Economic Record*, vol. 88, no. s1, pp. 2–9, 2012.
[4] Hongqiang Harry Liu, Ye Wang, Yang Richard Yang, Hao Wang, and Chen Tian, "Optimizing cost and performance for content multihoming," in *Proceedings of the ACM SIGCOMM 2012*. ACM, 2012, pp. 371–382.
[5] Gabor Szabo and Bernardo A Huberman, "Predicting the popularity of online content," *Communications of the ACM*, vol. 53, no. 8, pp. 80–88, 2010.
[6] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon, "I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM conference*. ACM, 2007, pp. 1–14.
[7] Tingyao Wu, Michael Timmers, Danny De Vleeschauwer, and Werner Van Leekwijck, "On the use of reservoir computing in popularity prediction," in *Evolving Internet (INTERNET), 2010 Second International Conference on*. IEEE, 2010, pp. 19–24.
[8] Henrique Pinto, Jussara M Almeida, and Marcos A 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*. ACM, 2013, pp. 365–374.
[9] Haitao Li, Xiaoqiang Ma, Feng Wang, Jiangchuan Liu, and Ke Xu, "On popularity prediction of videos shared in online social networks," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013, pp. 169–178.
[10] Lei Zhang, Feng Wang, and Jiangchuan Liu, "Understand instant video clip sharing on mobile platforms: Twitter's vine as a case study," in *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*. ACM, 2014, p. 85.
[11] Wojciech Galuba, Karl Aberer, Dipanjan Chakraborty, Zoran Despotovic, and Wolfgang Kellerer, "Outtweeting the twitterers-predicting information cascades in microblogs," in *Proceedings of the 3rd conference on Online social networks*. USENIX Association, 2010, pp. 3–3.
[12] Liangjie Hong, Ovidiu Dan, and Brian D Davison, "Predicting popular messages in twitter," in *Proceedings of the 20th international conference companion on World wide web*. ACM, 2011, pp. 57–58.
[13] Kristina Lerman and Tad Hogg, "Using a model of social dynamics to predict popularity of news," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 621–630.
[14] Di Niu, Zimu Liu, Baochun Li, and Shuqiao Zhao, "Demand forecast and performance prediction in peer-assisted on-demand streaming systems," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 421–425.
[15] Gonca Gursun, Mark Crovella, and Ibrahim Matta, "Describing and forecasting video access patterns," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 16–20.
[16] Zhi Wang, Lifeng Sun, Chuan Wu, and Shiqiang Yang, "Guiding internet-scale video service deployment using microblog-based prediction," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2901–2905.
[17] Matthew Rowe, "Forecasting audience increase on youtube," in *International Workshop on User Profile Data on the Social Semantic Web*, 2011.
[18] Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro, "How useful are your comments?: analyzing and predicting youtube comments and comment ratings," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 891–900.
[19] Peng Bao, Hua-Wei Shen, Junming Huang, and Xue-Qi Cheng, "Popularity prediction in microblogging network: a case study on sina weibo," in *Proceedings of the 22nd international conference on World Wide Web companion*, 2013, pp. 177–178.
[20] S Roy, Tao Mei, Wenjun Zeng, and Shipeng Li, "Towards cross-domain learning for social video popularity prediction," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1255–1267, 2013.
[21] Cem Tekin, Simpson Zhang, and Mihaela van der Schaar, "Distributed online learning in social recommender systems," *arXiv preprint arXiv:1309.6707*, 2013.
[22] Aleksandrs Slivkins, "Contextual bandits with similarity information," *arXiv preprint arXiv:0907.3986*, 2009.
[23] Miroslav Dudik, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang, "Efficient optimal learning for contextual bandits," *arXiv preprint arXiv:1106.2369*, 2011.
[24] John Langford and Tong Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," in *Advances in neural information processing systems*, 2008, pp. 817–824.
[25] Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire, "Contextual bandits with linear payoff functions," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 208–214.