Content Popularity Prediction Towards Location-Aware Mobile Edge Caching

Peng Yang, Ning Zhang, Shan Zhang,
Li Yu, Junshan Zhang, and Xuemin (Sherman) Shen

Abstract

Mobile edge caching enables content delivery within the radio access network, which effectively alleviates the backhaul burden and reduces response time. To fully exploit edge storage resources, the most popular contents should be identified and cached. Observing that user demands on certain contents vary greatly at different locations, this paper devises location-customized caching schemes to maximize the total content hit rate. Specifically, a linear model is used to estimate the future content hit rate. For the case where the model noise is zero-mean, a ridge regression based online algorithm with positive perturbation is proposed. Regret analysis indicates that the proposed algorithm asymptotically approaches the optimal caching strategy in the long run. When the noise structure is unknown, an H_∞ filter based online algorithm is further proposed by taking a prescribed threshold as input, which guarantees prediction accuracy even under the worst-case noise process. Both online algorithms require no training phases, and hence are robust to the time-varying user demands. The underlying causes of estimation errors of both algorithms are numerically analyzed. Moreover, extensive experiments on real world dataset are conducted to validate the applicability of the proposed algorithms. It is demonstrated that those algorithms can be applied to scenarios with different noise features, and are able to make

P. Yang and L. Yu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China (e-mail: yangpeng@hust.edu.cn; hustlyu@hust.edu.cn).

N. Zhang is with the Department of Computing Sciences, Texas A & M University-Corpus Christi, Corpus Christi, TX 78412, USA (e-mail: ning.zhang@tamucc.edu).

S. Zhang is with Beijing Key Laboratory of Computer Networks, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: zhangshan18@buaa.edu.cn).

J. Zhang is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA (e-mail: junshan.zhang@asu.edu).

X. Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

This paper was accepted in part by IEEE GLOBECOM 2017 [23].

adaptive caching decisions, achieving content hit rate that is comparable to that via the hindsight optimal strategy.

Index Terms

Mobile edge computing, dynamic content caching, popularity prediction, location awareness.

I. Introduction

The past decade has witnessed a significant growth of mobile traffic. Such growth puts tremendous pressure on the paradigm of Cloud-based service provisioning, since moving a large volume of data into and out of the cloud wirelessly requires substantial spectrum resources, and meanwhile may incur large latency. *Mobile Edge Computing* (MEC) emerges as a new paradigm to alleviate the capacity concern of mobile networks [1]. Residing on the network edge, MEC makes abundant storage and computing resources available to mobile users through low-latency wireless connections, facilitating a number of mobile services like local content caching, augmented reality, and cognitive assistance [2].

Among these services, content caching at the network edge is garnering much attention [3]-[17]. In particular, with the prevalence of social media, multimedia contents are spreading among mobile users in a viral fashion, putting high pressure on the network backhaul [9], [11]. It is pointed out that, by caching contents on network edge, up to 35% traffic on the backhaul can be reduced [2]. Unfortunately, compared with the increasing content volume, the storage size at edge node (EN) is always limited. It is impossible to cache all the contents locally. Hence, identifying the optimal set of contents that maximizes cache utilization becomes crucial.

Content popularity is an effective measure for making caching decisions. Extensive works have been devoted to popularity-based content caching. According to the features of content popularity profile, those works on content caching can be classified into three categories: 1) known popularity profile [11]-[13]; 2) fixed but unknown popularity profile [16], [17]; and 3) time-varying and unknown popularity profile [19], [20]. In case of fixed and unknown popularity profile, learning algorithms have been proposed under different network settings. In case of time-varying and unknown popularity profile, context information of the request, including system states and user characteristics, is exploited to make content hit rate predictions. To improve the accuracy of popularity prediction, the context space needs to be subtly designed since there is endless context information that could be taken into consideration. It is often difficult to directly

identify the factors that influence content popularity. More importantly, using user information for context differentiation is subject to privacy regulations and may not be applicable in practice.

In this paper, we investigate mobile edge caching with time-varying and unknown popularity profile. Instead of relying on user information for context differentiation, we explore location features of each EN to improve the accuracy of popularity prediction, with the rationale outlined as follows. First, locations can be divided into categories with distinct social functions, such as residential area and business district. Meanwhile, users in different places have diverse interests [22]. As indicated by real-world measurement studies [35], the distribution of content popularity for even adjacent Wi-Fi APs and cellular base stations are different, and existing content caching schemes do not take such fine-grained popularity difference into consideration [36]. To further improve the content distribution in mobile context, it is crucial to investigate content popularity with location awareness. Given that there is no established model to characterize location features and user demands, we take some initial steps to devise a model where user demand of a certain content is treated as a linear combination of content features and location characteristics with unknown noise. It follows that, the popularity prediction problem boils down to the estimation of location feature vector of each EN in the presence of noise. In practice, the noise process is affected by various factors. Firstly, it is affected by location-dependent factors, such as user interests, the number of users and the social function of the coverage area of each EN. Secondly, it is also affected by content-dependent factors, which include genre, length, and frame quality for video contents. Unfortunately, it is often difficult for content providers and edge servers to understand the statistical nature of the underlying noise process in such complicated context space. To solve the location feature estimation problem, two online prediction algorithms are proposed for different scenarios.

To start with, we consider the tractable zero-mean noise scenario as the first step. A ridge regression based prediction algorithm (RPUC) is proposed to estimate the location feature vector. To account for the impact of noise, a positive perturbation is added to the result as the correction of the prediction. By comparing to the hindsight optimal caching policy, theoretical analysis shows that the RPUC algorithm achieves sublinear regret, i.e., it asymptotically approaches the optimal strategy in the long-term.

Furthermore, we consider practical cases where noise structure is unknown *a priori*. To ensure robust prediction, we resort to the H_∞ filter technique, which enables us to obtain guaranteed accuracy even in the worst-case scenario. In particular, taking a prescribed accuracy threshold

as an input, we propose an H_∞ based prediction algorithm (HPDT), which is robust as long as the noise amplitude is finite.

Both RPUC and HPDT require no training phases, and hence are adaptive to the time-varying user demand. Numerical analysis indicates that, the regret of RPUC originates from the bias and variance of ridge regression, as well as the artificial perturbation. Note that the HPDT algorithm is conservative in that it makes no assumption on the noise. Yet, it is still able to make unbiased estimation on the location feature vector. Extensive simulations on real world traces demonstrate that those two algorithms can be applied to scenarios with different noise features, and both of them are able to make adaptive caching decisions, achieving content hit rate that is comparable to that using the hindsight optimal strategy. The contributions of this work on mobile edge caching are three-fold:

- We propose to exploit the diversity of content popularity over different locations. We establish a linear model for content popularity prediction, taking into account both content and location features.
- We develop two popularity prediction algorithms that deal with different noise models. Both algorithms are able to make location-aware caching decisions. Moreover, they require no training phases, and hence can adapt to dynamic user demand.
- We demonstrate the effectiveness of the proposed algorithms through theoretical analysis. It is proved that performance of the RPUC algorithm asymptotically approaches that using the hindsight optimal strategy, while performance of the HPDT algorithm hinges upon noises. Experiments on real dataset crawled from YouTube show that, the long-term content hit rates of the proposed algorithms are comparable to that via the hindsight optimal strategy.

The remainder of the paper is organized as follows. Section II reviews related works on content caching in wireless networks. Section III describes the system model, including the mobile edge caching architecture and the formal problem formulation. In Section IV, we propose the RPUC caching algorithm for the case of zero-mean noise, and give the detailed performance analysis. For the case of unknown noise model, we present the HPDT algorithm as well as detailed regret analysis in Section V. Numerical analysis and experimental results of the two algorithms are provided in Section VI, followed by concluding remarks in Section VII.

II. Related Work

Mobile user's capacity is greatly augmented in the era of MEC. As a result, mobile service provisioning is expected to have further improved quality of experience (QoE) [2]. To this end, various mobile edge architectures have been proposed. Tandom *et al.* proposed to deploy edge resources within radio access networks. They characterized the relationship between latency and caching size, as well as latency and fronthaul capacity, from an information-theoretic perspective [3]. Yang *et al.* introduced an edge resource provisioning architecture based on cloud radio access network (C-RAN), and devised a cloud-edge interoperation scheme via software defined networking techniques [4]. Tong *et al.* designed a hierarchical edge architecture, aiming at making efficient use of edge resources when serving the peak loads from mobile users [5]. As the 5G wireless network is expected to incorporate diverse access technologies, in this paper, we consider edge caching in the context of heterogeneous networks. Potential EN deployment can be capacity-augmented base stations, WiFi access points and other devices with excess resources.

As an effective approach to improving QoE in 5G systems, edge caching has received extensive attention [6]. Specifically, various works have been done on video content caching, since video contents are forecast to be dominant in 5G systems [7]-[10]. A vast amount of other works simply focus on generalized content caching. Zhang *et al.* investigated the cache-enabled vehicular networks with energy harvesting, aiming at minimizing network deployment costs with QoE guarantees [12]. Ao *et al.* explored distributed content caching and small cell cooperation to accelerate content delivery [13]. Device-to-device (D2D) communication is another promising solution to improve the QoE of mobile content dissemination [14]. Different from conventional content unicast from cellular base stations, D2D communication has the potential to significantly boost system throughput by multicasting. Ji *et al.* provided a comprehensive summary on D2D caching networks, incorporating throughput scaling law and coded caching in D2D networks [15]. In the above works, content popularity profile was assumed to be completely known. However, in practice, content popularity may be unknown *a priori*. To address this issue, various learning-based approaches have been proposed to predict content popularity. Bharath *et al.* proposed a learning method that achieves desired popularity accuracy in finite training time [16]. Blasco *et al.* modeled content caching with unknown popularity as a multi-armed bandit problem [17]. By carefully balancing exploration and exploitation in the learning phase, they proposed three algorithms that quickly learn content popularity under various system settings.

Unfortunately, often times content popularity profile can not only be unknown *a priori*, but also time-varying. This is because user's interests change constantly, and meanwhile new contents are being created [22]. As a result, learning-based caching algorithms should be designed in an online fashion, i.e., requiring no training phase, and adaptive to popularity fluctuations. To this end, Roy *et al.* proposed to predict video popularity by utilizing knowledge from the social streams [18]. Müller *et al.* introduced context-aware proactive caching [19]. By constructing context space based on user information, they proposed an online algorithm that first learns context-specific user demands, and then updates cached contents accordingly. Other information has also been used for context differentiation, such as content features and system states [20]. The prediction accuracy of those solutions is highly dependent on the information used for context differentiation. To content service providers, however, user information is extremely sensitive and often unavailable. In addition, it is also impossible for them to get detailed system or network information when making caching decisions.

In this paper, we exploit locational features for context differentiation. Locational information can be easily obtained, for example, users attached to different ENs are naturally divided into geographical groups. Based on which we investigate the location-aware caching problem with unknown and time-varying content popularity profile. By modeling user demand as linear combination of location features and content attributes, our previous work has addressed the content popularity prediction problem with the assumption that the model noise is zero-mean [23]. As an extension, this paper additionally considers the practical scenario, where noise structure is unknown *a priori*. Specifically, a robust prediction algorithm is proposed with detailed theoretical caching performance analysis. The proposed algorithm is robust and practical as it guarantees prediction accuracy regardless of the noise statistics. Additionally, numerical analysis and comparison on the root causes of estimation errors of both algorithms are presented. Much extensive experiments are conducted to validate the performance of the proposed algorithms.

It is worth noting that, in the mobile context, fetching content from the edge cache significantly reduces the delay, compared with that from conventional content distribution network (CDN). Moreover, existing content pushing strategies in CDN do not consider the fine-grained popularity differentiation in neighbouring Wi-Fi APs and cellular base stations [35]. With the consideration of location awareness, this paper further models and predicts the dynamics of content popularity, which is constantly varying with time.

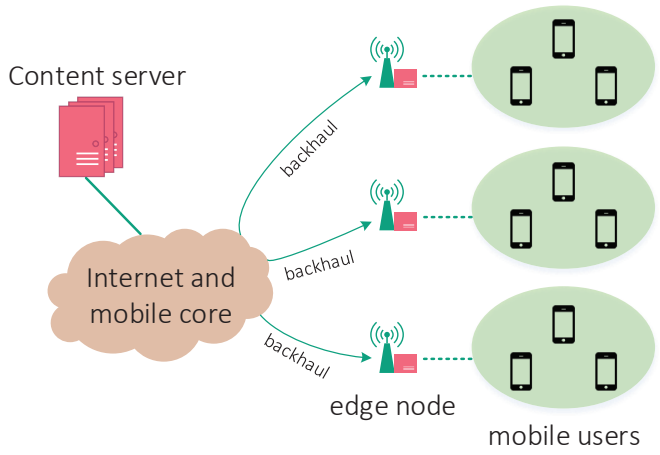


Fig. 1. Network model of mobile edge caching.

III. System Model and Problem Formulation

In this section, we present the system model and formulate the caching problem in mobile edge networks.

A. Network Model

Mobile edge computing can enhance mobile user's capacity by provisioning storage, computing and networking resources in their proximity. Capacity-augmented base stations, WiFi access points and other devices with excess capacity can be exploited for edge node deployment [1]. In this paper, the storage resources at edge nodes are harnessed for content caching services. Specifically, as shown in Fig. 1, a set of edge nodes $\mathcal{N} = \{1, 2, \ldots, N\}$ is deployed with separated backhaul links connecting to the mobile core network. Online contents are dynamically pushed to edge nodes so that user's content requests can be processed with reduced latency. Each edge node serves a disjoint set of mobile users.

B. Content Popularity and Location Diversity

A simple yet effective caching strategy is to push the most popular contents to the network edge. Hence, local content hit rate is maximized and user's requests are served with reduced latency and improved QoE. Extensive works have been done on the popularity of contents, especially video files [9], [22], [25]. According to the statistics we crawled from YouTube, as illustrated in Fig. 2, the popularity profile of a video file varies in two-fold. 1) The daily

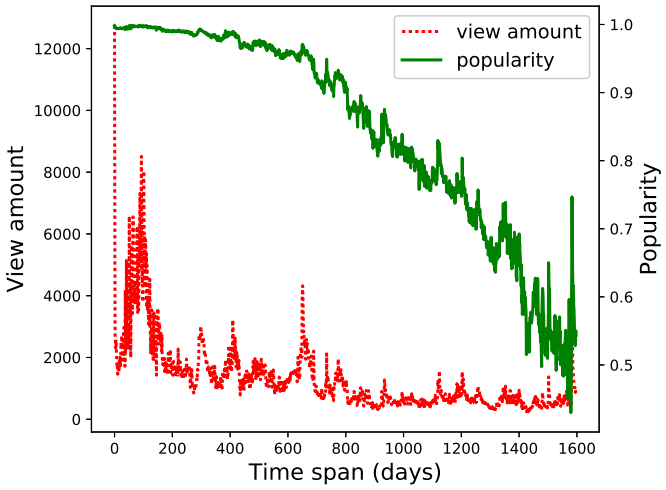


Fig. 2. The daily view amount and popularity trends of a YouTube video since uploaded. The popularity score equals to the ratio of the video's daily view amount to the total daily view amount of all the videos. Note the statistics are based on a set of randomly crawled videos.

view amount is time-varying. 2) As other videos' daily view amounts are also varying and new videos are uploaded, the popularity of a video file is constantly fluctuating [21]. Moreover, location diversity also affects the content popularity. As a result, general caching strategies based on fixed popularity profile are not optimal in practice.

Let $\boldsymbol{x}_{f,n} \in \mathbb{R}^d$ be a d-dimensional attribute vector of file f associated with EN n. For example, the attributes of video contents may include video quality, genre, length, and historical view statistics. Then, the hit rate[1] of file f at EN n, denoted by $d_{f,n}$, can be expressed as the following noisy linear combination

$$d_{f,n} = \boldsymbol{x}_{f,n}^{\top} \boldsymbol{\theta}_n^* + w_n, \tag{1}$$

where $\boldsymbol{\theta}_n^* \in \mathbb{R}^d$ is the unknown location feature vector associated with EN n. Further, it also represents the location characteristics of EN n, which is time-invariant. w_n is the random noise associated with EN n, which may be affected by various locational features, including social function of the area around EN n, the number of users served by EN n, the frequency of content update (e.g., hourly or daily). As a result, contents with the same attribute vector are expected to have different view amounts at different ENs. This linear prediction model is widely used in other areas, such as signal processing and financial engineering [26]. It provides a method to

[1]We define *hit rate* as the number of content requests rather than a ratio.

predict future hit rate and it is essential when exploiting location diversity for popular-unknown content caching. Without loss of generality, let $||\boldsymbol{\theta}_n^*|| \leq \zeta$, $||\boldsymbol{x}_{f,n}|| \leq \eta$ and $d_{f,n} \leq \gamma$ for all $f \in \mathcal{F}$ and $n \in \mathcal{N}$, where $||\boldsymbol{x}|| = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$ denotes the Euclidean norm of $\boldsymbol{x}$, ζ, η and γ are positive constants. Also, for notational simplicity, define $||\boldsymbol{x}||_{\boldsymbol{V}} \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{V} \boldsymbol{x}}$ as the weighted (by a matrix $\boldsymbol{V} \in \mathbb{R}^{d \times d}$) Euclidean norm of $\boldsymbol{x}$.

C. Problem Formulation

Consider a set of files $\mathcal{F} = \{1, 2, \ldots, F\}$ that can be cached at ENs, and let $c < F$ be the caching size of each EN. We assume that all the contents are of equal size[2] and the size is normalized to 1, i.e., each EN can cache up to c contents. As indicated by Fig. 2, content popularity is time-varying. Therefore, contents with higher popularity should be proactively identified and cached at the ENs, and the less popular ones should be evicted so as to improve the local content hit rate. Considering a sequence of time slots $\mathcal{T} = \{1, 2, \ldots, T\}$, and let $\mathcal{F}_{n,t}$ denote the set of contents cached at EN n during time slot $t \in \mathcal{T}$, and $d_{f,n,t}$ be the amount of user demand on file f at EN n during time slot t. The objective of a caching policy is to maximize the time-averaged hit rate. Formally, it can be formulated as the following time-averaged hit rate maximization (THRM) problem[3]:

$$\begin{aligned} (\underline{\text{THRM}}): \max \ & \tfrac{1}{T} \textstyle\sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}_{n,t}} d_{f,n,t} \\ \text{Subject to:} \ & |\mathcal{F}_{n,t}| \leq c, \ \forall n \in \mathcal{N}, \ t \in \mathcal{T}. \end{aligned} \tag{2}$$

As the amount of user demand, i.e., the hit rate of contents at each EN, is unknown *a priori*, the decision variables $\mathcal{F}_{n,t}$ in problem (2) is intractable directly. For convenience, denoting the optimal caching strategy $\mathcal{F}_{n,t}^*$ for EN n at time t, we have

$$\mathcal{F}_{n,t}^* = \underset{|\mathcal{F}_{n,t}| \leq c}{\arg\max} \sum_{f \in \mathcal{F}_{n,t}} d_{f,n,t}, \ \forall n \in \mathcal{N}, \ t \in \mathcal{T}. \tag{3}$$

Define the time-averaged caching *regret* of a solution respect to the optimal caching strategy as

$$R(T) \triangleq \frac{1}{T} \mathbb{E} \left[\sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \left(\sum_{f \in \mathcal{F}_{n,t}^*} d_{f,n,t} - \sum_{f \in \mathcal{F}_{n,t}} d_{f,n,t} \right) \right]. \tag{4}$$

[2]In case contents are of different sizes, they are split into smaller ones of equal size. For example, the widely used DASH (Dynamic Adaptive Streaming over HTTP) protocol breaks contents into small segments before transmission. This assumption is used to simplify the theoretical analysis, and a similar assumption has been made in [20], [24]. Location-aware edge caching with different content sizes deserves further investigation.

[3]Without loss of generality, we assume that the underlying process is ergodic.

Then, the THRM problem can be reformulate as a time-averaged *regret* minimization (TRM) problem:

$$\begin{aligned} (\underline{\text{TRM}}): \min \quad & R(T) \\ \text{Subject to:} \quad & |\mathcal{F}_{n,t}| \leq c, \ \forall n \in \mathcal{N}, \ t \in \mathcal{T}. \end{aligned} \tag{5}$$

Given that the optimal set $\mathcal{F}_{n,t}^*$ is unknown *a priori*, our goal is to develop a caching policy that constantly makes good estimation of the optimal set $\mathcal{F}_{n,t}^*$, and therefore minimizes the time-averaged caching regret. As indicated by Eq. (3), the LRM problem boils down to estimating user demands of different contents at each EN. Given the linear model in Eq. (1), if the location feature vector $\boldsymbol{\theta}_n^*$ can be found in the presence of noise, we can make an accurate prediction on user demand. Unfortunately, there is no established statistical model on the noise processes that impinges the prediction of user demand. In what follows, we propose two online content popularity prediction algorithms by making dynamic estimations on the location feature vectors for different noise processes. In particular, the first algorithm achieves near-optimal performance with the assumption that the model noise is zero-mean, while the second algorithm is designed to provide robust performance guarantees in the case of unknown noise statistics.

IV. Ridge Regression based Content Popularity Prediction and Edge Caching

In this section, as the first attack on the TRM problem, we present a caching algorithm when noise is zero-mean.

A. Location Feature Vector Estimation

When the noise is zero-mean, according to Eq. (1), we have

$$\mathbb{E}[d_{f,n}|\boldsymbol{x}_{f,n}] = \boldsymbol{x}_{f,n}^\top \boldsymbol{\theta}_n^*. \tag{6}$$

It can be interpreted that, at time slot t, given the attribute vector $\boldsymbol{x}_{f,n,t}$, the hit rate of file f at EN n is predicted to be the linear combination of its attributes, which provides a feasible way to predict the content hit rate. Since the location feature vector $\boldsymbol{\theta}_n^*$ of EN n is time-invariant, a good estimation of $\boldsymbol{\theta}_n^*$ will lead to accurate prediction of the content hit rate.

Let the attribute matrix $\boldsymbol{\Phi}_{f,n} \in \mathbb{R}^{m \times d}$ be the historical data up to time slot t, where m is the frequency of file f being cached at EN n up to time slot t, and the m-th row of $\boldsymbol{\Phi}_{f,n}$ is the corresponding attribute vector $\boldsymbol{x}_{f,n,m}$. Denote by $\boldsymbol{y}_{f,n} \in \mathbb{R}^m$ the m-time empirical hit rate of file f at EN n. By applying the standard ordinary least square linear regression, i.e.,

$\boldsymbol{\theta}_n^* = \arg\min_{\boldsymbol{\theta}_n} ||\boldsymbol{y}_{f,n} - \boldsymbol{\Phi}_{f,n}\boldsymbol{\theta}_n||^2$, we can obtain the unique solution $\boldsymbol{\theta}_n^* = (\boldsymbol{\Phi}_{f,n}^\top \boldsymbol{\Phi}_{f,n})^{-1}\boldsymbol{\Phi}_{f,n}^\top \boldsymbol{y}_{f,n}$, which is unbiased. However, when there are correlated variables in the attribute vector, the matrix $\boldsymbol{\Phi}_{f,n}^\top \boldsymbol{\Phi}_{f,n}$ may not be invertible. As a result, the estimated $\boldsymbol{\theta}_n$ can be poorly determined and will exhibit high variance.

In contrast to the unbiased estimation, ridge regression makes biased estimation by adding a control parameter that "penalizes" the magnitude of estimated $\boldsymbol{\theta}_n$, which helps to improve estimation stability. Specifically, ridge regression aims at minimizing a penalized sum

$$\boldsymbol{\theta}_n^* = \arg\min_{\boldsymbol{\theta}_n} \left(||\boldsymbol{y}_{f,n} - \boldsymbol{\Phi}_{f,n}\boldsymbol{\theta}_n||^2 + \mu||\boldsymbol{\theta}_n||^2 \right), \tag{7}$$

where $\mu > 0$ controls the size of $\boldsymbol{\theta}_n$: the larger the value of μ, the greater the shrinkage of the magnitude of $\boldsymbol{\theta}_n$ [27]. Consequently, the estimation of $\boldsymbol{\theta}_n^*$ can be explicitly given as

$$\tilde{\boldsymbol{\theta}}_n = (\boldsymbol{\Phi}_{f,n}^\top \boldsymbol{\Phi}_{f,n} + \mu \boldsymbol{I}_d)^{-1} \boldsymbol{\Phi}_{f,n}^\top \boldsymbol{y}_{f,n}\,, \tag{8}$$

where $\boldsymbol{I}_d \in \mathbb{R}^{d\times d}$ is the identity matrix. The accuracy of the estimation depends on the amount of data and the selection of μ. For convenience, let $\boldsymbol{V}_{f,n} = \boldsymbol{\Phi}_{f,n}^\top \boldsymbol{\Phi}_{f,n} + \mu \boldsymbol{I}_d$ for all $f \in \mathcal{F}$ and $n \in \mathcal{N}$. The following lemma, which is slightly manipulated from [31], gives an upper bound on the estimation error of ridge regression.

Lemma 1. *If $||\boldsymbol{\theta}_n^*|| \leq \zeta$ for all $n \in \mathcal{N}$, then $\forall \delta > 0$, the estimation error of ridge regression can be upper bounded as*

$$|\boldsymbol{x}_{f,n}^\top \tilde{\boldsymbol{\theta}}_n - \boldsymbol{x}_{f,n}^\top \boldsymbol{\theta}_n^*| \leq (\delta + \zeta\mu)||\boldsymbol{x}_{f,n}||_{\boldsymbol{V}_{f,n}^{-1}} \tag{9}$$

with probability at least $1 - 2e^{-2\delta^2}$.

Please refer to Appendix A for the proof. The probabilistic upper bound of estimation error provided in Lemma 1 indicates that, the true hit rate $\boldsymbol{x}_{f,n}^\top \boldsymbol{\theta}_n^*$ falls into the confidence interval around the estimation $\boldsymbol{x}_{f,n}^\top \tilde{\boldsymbol{\theta}}_n$ with high probability. The righthand side of Eq. (9) gives the length of the confidence interval, which is crucial to the following content popularity prediction and caching algorithm.

B. RPUC Caching Algorithm

The location-aware edge caching algorithm is sketched in Algorithm 1. After initialization, the algorithm iteratively performs the following three phases.

Algorithm 1 RPUC: **R**idge Regression **P**rediction with **U**pper **C**onfidence for Location-Aware Edge Caching

Input: $\mu > 0$.

Output: Set of files to be cached in each EN.

Initialization: Cache files in every EN and get the initial attribute vectors $\boldsymbol{x}_{f,n,0}$ of all file-EN pairs.
$\boldsymbol{V}_n \leftarrow \mu \boldsymbol{I}_d$, $\boldsymbol{h}_n \leftarrow \boldsymbol{0}_d$, $\forall n \in \mathcal{N}$
for $t = 1, 2, \ldots, T$ **do**
  for each EN $n \in \mathcal{N}$ **do**
    $\tilde{\boldsymbol{\theta}}_{n,t} \leftarrow \boldsymbol{V}_n^{-1} \boldsymbol{h}_n$
    for each file $f \in \mathcal{F}$ **do**
      Obtain attribute vector $\boldsymbol{x}_{f,n,t}$
      $\tilde{d}_{f,n,t} \leftarrow \boldsymbol{x}_{f,n,t}^{\top} \tilde{\boldsymbol{\theta}}_{n,t}$
      Compute the perturbation $p_{f,n,t}$ in Eq. (11)
      $\hat{d}_{f,n,t} \leftarrow \tilde{d}_{f,n,t} + p_{f,n,t}$
    end for
    $\mathcal{F}_{n,t} = \arg\max_{\mathcal{F}_n \subseteq \mathcal{F},\, |\mathcal{F}_n| \leq c} \sum_{f \in \mathcal{F}_n} \hat{d}_{f,n,t}$
    Cache all the files in set $\mathcal{F}_{n,t}$ on EN n
    Observe empirical demands $d_{f,n,t}$ of cached files
    Update $\boldsymbol{V}_n$ and $\boldsymbol{h}_n$ based on $\boldsymbol{x}_{f,n,t}$ and $d_{f,n,t}$ of all cached files:
    $\boldsymbol{V}_n \leftarrow \boldsymbol{V}_n + \boldsymbol{x}_{f,n,t} \boldsymbol{x}_{f,n,t}^{\top}$
    $\boldsymbol{h}_n \leftarrow \boldsymbol{h}_n + \boldsymbol{x}_{f,n,t} d_{f,n,t}$
  end for
end for

1. *Predict*: During each time slot t, the location feature vector $\tilde{\boldsymbol{\theta}}_{n,t}$ is firstly updated according to the demand information observed in time slot $t-1$. Then, based on the linear prediction model, the estimated demand $\tilde{d}_{f,n,t}$ is obtained. Considering the impact of random noises, a perturbation is added to the estimation, i.e., the ultimate hit rate is predicted to be

$$\hat{d}_{f,n,t} = \boldsymbol{x}_{f,n,t}^{\top} \tilde{\boldsymbol{\theta}}_{n,t} + p_{f,n,t}\,, \tag{10}$$

where the perturbation $p_{f,n,t}$ is given by

$$p_{f,n,t} = \alpha_t ||\boldsymbol{x}_{f,n,t}||_{\boldsymbol{V}_{f,n}^{-1}}, \tag{11}$$

and $\alpha_t = \left[\ln(tF^{\frac{1}{2}})\right]^{\frac{1}{2}} + \mu\zeta$.

2. *Optimize and cache*: Based on the predicted hit rate $\hat{d}_{f,n,t}$ of each content, a set of contents $\mathcal{F}_{n,t}$ that maximizes the content hit rate at EN n during time slot t is identified and cached respectively. Note that, certain contents may be cached in multiple ENs simultaneously.
3. *Observe and update*: At the end of time slot t, the empirical hit rate information of cached files $\mathcal{F}_{n,t}$ on each EN is recorded, which is then used to update the parameter matrices for subsequent estimation and prediction.

The rationale of the perturbation is that Eq. (6) only gives a mean value of the hit rate which omits the potential random fluctuation, while Lemma 1 provides a probabilistic upper bound of the demand estimation error. The perturbation given in Eq. (11) is inline with the righthand side of Eq. (9) and can be regarded as the optimism in face of uncertainty, or equivalently, the upper confidence of the demand estimation. By adding a perturbation according to Eq. (11), we have $\delta = \left[\ln(tF^{\frac{1}{2}})\right]^{\frac{1}{2}}$. According to Lemma 1, the upper bound holds with probability at least $1 - 2F^{-1}t^{-2}$, which approximates to 1 rapidly as t increases.

C. Regret Analysis

The content hit rate of the RPUC algorithm highly depends on the accuracy of prediction. This subsection gives a theoretical upper bound on its time-averaged caching regret $R(T)$.

In mobile edge caching, let c be the caching size of each EN, and F be the size of ground file set. Note that content hit rate satisfies the linear model, and content attribute vectors and user demands are bounded by $||\boldsymbol{x}_{f,n,t}|| \leq \eta$ and $d_{f,n,t} \leq \gamma$ for all $f \in \mathcal{F}$, $n \in \mathcal{N}$ and $t \in \mathcal{T}$, we have the following theorem.

Theorem 1. *If the noise is zero-mean, the RPUC algorithm achieves near-optimal performance, i.e., the time-averaged regret $R(T)$ is of order $O\left(cN\sqrt{\frac{d(\ln T)\ln(\mu+T\eta^2/d)}{T}}\right)$, and $R(T) \to 0$ when $T \to \infty$.*

The proof has been relegated to Appendix B. Basically, the root-cause of regret is two-fold: the estimation error and the perturbation. Particularly, the estimation error consists of the linear model error and the intended bias incurred by $\mu > 0$ in ridge regression. The perturbation term is

well managed by the time-varying control parameter α_t. Theorem 1 indicates that under RPUC algorithm, the content hit rate asymptotically approaches the optimal caching policy in the long term.

V. Robust Content Popularity Prediction and Edge Caching

In the previous section, we proposed the online caching algorithm RPUC based on the linear prediction model given in Eq. (6). A biased estimation of the location feature vector $\boldsymbol{\theta}_n^*$ for each EN is obtained by ridge regression. Further, a perturbation is added to the estimation of content hit rate to account for uncertainty. However, this caching algorithm would not work well when the noise is not zero-mean. Even worse, it is likely that we are unable to get detailed noise statistics, as it is affected by various location features, such as population, social function or even weather condition. Therefore, robust prediction algorithm that could handle noise uncertainty is desirable. In this section, by resorting to the H_∞ filter technique, we propose a popularity prediction algorithm that provides guaranteed accuracy in the case of unknown noise structures.

A. Noisy Model for Content Popularity

Denoted by $w_{n,t}$ the additive noise added to the linear model. The linear model is rewritten as

$$d_{f,n,t} = \boldsymbol{x}_{f,n,t}^\top \boldsymbol{\theta}_n^* + w_{n,t}. \tag{12}$$

If the noise process $w_{n,t}$ follows white Gaussian distribution and its mean and correlation are always known, Kalman filtering technique can be applied to estimate $\boldsymbol{\theta}_n^*$, which achieves the smallest possible standard deviation of the estimation error [28]-[30]. Since there is no established model on the statistics of the noise structure, robust estimators on the location feature vector that can tolerate noise uncertainty are needed. Next, we will introduce the H_∞ filtering technique for location feature vector estimation, which requires no *a priori* information on the noise process. The only assumption is that, the magnitude of the noise process is finite, which is true since the total demand $d_{f,n,t}$ is always finite in reality.

B. An H_∞ Filter Approach

When locational features are time-invariant, the true location feature vector $\boldsymbol{\theta}_n^*$ remains the same across the time span. For notational simplicity, in this subsection, we focus on a specific

content on a certain EN and neglect indices f and n. Based on Eq. (12), the location feature vector estimation problem is reformulated as

$$\begin{cases} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t \\ d_t &= \boldsymbol{x}_t^\top \boldsymbol{\theta}_t + w_t. \end{cases} \tag{13}$$

Different from Kalman filter, we aim at providing a uniformly small estimation error, $e_t = |d_t - \tilde{d}_t|$, for any form of noise process. Notice that estimation of the location feature vector $\boldsymbol{\theta}^*$ is crucial to minimizing the estimation error e_t. Then, we define the following cost function of estimation [28]:

$$\mathcal{J}_0 \triangleq \frac{\sum_{t=0}^{T} ||\boldsymbol{\theta}^* - \tilde{\boldsymbol{\theta}}_t||^2}{||\boldsymbol{\theta}_0 - \tilde{\boldsymbol{\theta}}_0||^2_{\boldsymbol{P}_0} + \sum_{t=0}^{T} |w_t|^2}, \tag{14}$$

where $\boldsymbol{P}_0$ is a symmetric positive definite matrix reflecting the confidence of the *a priori* knowledge of the initial state. A 'smaller' choice of $\boldsymbol{P}_0$ indicates larger uncertainty of the initial condition and vice versa. The objective is to make a sequence of estimation on $\tilde{\boldsymbol{\theta}}_t$ such that the above cost is minimized. The denominator of the cost function can be regarded as a combined norm of all possible initial states and noises affecting the system. Given that there is no established stochastic model for w_t, the cost function in Eq. (14) allows us to make robust estimation of content popularity from game-theoretical perspective. Suppose that there is an unrestricted adversary, who can control the initial state $\boldsymbol{\theta}_0$ and the magnitude of w_t to maximize the error of our estimation. While we focus on minimizing the numerator of Eq. (14), the adversary may incur infinite magnitude of disturbances. The form of $\mathcal{J}_0$ prevents the adversary from using brute force to maximize $||\boldsymbol{\theta}^* - \tilde{\boldsymbol{\theta}}_t||$. Instead, the adversary needs to carefully choose $\boldsymbol{\theta}_0$ and w_t as it tries to maximize $||\boldsymbol{\theta}^* - \tilde{\boldsymbol{\theta}}_t||$. Formally, this game can be generalized as a minmax problem, and the optimal estimation on $\tilde{\boldsymbol{\theta}}_t$ achieves the minimal cost $\mathcal{J}_0^*$ as

$$\mathcal{J}_0^* = \min_{\tilde{\boldsymbol{\theta}}_t} \max_{\boldsymbol{\theta}_0,\, w_t} \mathcal{J}_0. \tag{15}$$

Given the cost function in Eq. (14), directly minimizing $\mathcal{J}_0$ is challenging. In practice, a better approach for H_∞ filtering is to seek a sub-optimal estimation that meets a given threshold. Specifically, one can try to find $\tilde{\boldsymbol{\theta}}_t$ such that the optimal estimate of $\boldsymbol{\theta}_t$ among all possible $\tilde{\boldsymbol{\theta}}_t$ (even including the worst-case performance measure) should satisfy

$$\sup\, \mathcal{J}_0 < \psi^2, \tag{16}$$

where $\psi > 0$ is the prescribed performance bound. Eq. (16) indicates that H_∞ filter guarantees the smallest estimation error over all possible finite disturbances of the noise magnitude [28]. Rearranging Eq. (16) gives the following equivalent minmax problem

$$\min_{\tilde{\boldsymbol{\theta}}_t} \max_{\boldsymbol{\theta}_0,\, w_t} \mathcal{J} \triangleq -\frac{1}{2}\psi^2 ||\boldsymbol{\theta}_0 - \tilde{\boldsymbol{\theta}}_0||^2_{\boldsymbol{P}_0} + \frac{1}{2}\sum_{t=0}^{T}\Big[||\boldsymbol{\theta}^* - \tilde{\boldsymbol{\theta}}_t||^2 - \psi^2 |w_t|^2\Big]. \tag{17}$$

This minmax problem can be interpreted as a zero-sum game against the adversary. With a given ψ, our goal is to find an estimation that wins the game (i.e., achieve a negative cost $\mathcal{J} < 0$). By resorting to the Lagrange multiplier method for the dynamic constrained optimization problem (17), the H_∞ filter approach results in the following iterative algorithm to find the optimal estimates $\tilde{\boldsymbol{\theta}}_t$ for all $t \in \mathcal{T}$:

$$\tilde{\boldsymbol{\theta}}_{t+1} = \tilde{\boldsymbol{\theta}}_t + \boldsymbol{R}_t(d_t - \boldsymbol{x}_t^\top \tilde{\boldsymbol{\theta}}_t), \tag{18}$$

where $\tilde{\boldsymbol{\theta}}_t$ is initialized as $\tilde{\boldsymbol{\theta}}_0 = \mathbf{0}_d$, $\boldsymbol{R}_t$ is the H_∞ filter gain, given by

$$\boldsymbol{R}_t = \boldsymbol{M}_t \boldsymbol{x}_t (1 + ||\boldsymbol{x}_t||^2_{\boldsymbol{M}_t})^{-1}, \tag{19}$$

and

$$\boldsymbol{M}_{t+1}^{-1} = \boldsymbol{M}_t^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top - \psi^{-2}\boldsymbol{I}_d, \tag{20}$$

with $\boldsymbol{M}_t^{-1}$ initialized by $\boldsymbol{M}_0^{-1} = \boldsymbol{P}_0 - \psi^{-2}\boldsymbol{I}_d$. The detailed proof of this solution can be found in [30]. With the aid of H_∞ filter, we are able to make performance-guaranteed estimation on the location feature vector $\boldsymbol{\theta}^*$, regardless of the detailed noise structure. Based on the estimation of location feature vector, content popularity prediction and caching algorithm can be further devised.

C. From Determining ψ to the HPDT Caching Algorithm

The performance of H_∞ filter highly depends on the prescribed threshold ψ. A smaller ψ results in a smaller estimation error. However, if ψ is too small, $\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top - \psi^{-2}\boldsymbol{I}_d$ in Eq. (20) may be singular, which renders the iterative solution infeasible. Hence, the value of ψ should be carefully selected. An adaptive scheme for threshold selection was proposed in [28], which makes online iterative prediction possible. Denote by ψ_{t+1} the threshold on the $(t+1)$-th iteration, it should be properly chosen to guarantee $\boldsymbol{M}_{t+1}^{-1}$ is positive definite, i.e.,

$$\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top - \psi_{t+1}^{-2}\boldsymbol{I}_d \succ 0. \tag{21}$$

Algorithm 2 HPDT : $\boldsymbol{H}_\infty$ filter **P**rediction with **D**ynamic **T**hreshold for Location-aware Edge Caching

Input: ξ close to but larger than 1.

Output: Set of files to be cached in each EN.

Initialization: Cache files in every EN and get the initial attribute vectors $\boldsymbol{x}_{f,n,0}$ of all file-EN pairs;
Choose symmetric positive definite $\boldsymbol{P}_{n,0} \succ 0$ for all $n \in \mathcal{N}$;
Choose the smallest possible value of $\psi_{n,0}$, so that $\boldsymbol{P}_{n,0} - \psi_{n,0}^{-2}\boldsymbol{I}_d$ is nonsingular;
$\boldsymbol{M}_{n,0}^{-1} \leftarrow \boldsymbol{P}_{n,0} - \psi_{n,0}^{-2}\boldsymbol{I}_d$, $\tilde{\boldsymbol{\theta}}_{n,0} \leftarrow \mathbf{0}_d$ for all $n \in \mathcal{N}$.
for $t = 1, 2, \ldots, T$ **do**
  for each EN $n \in \mathcal{N}$ **do**
    for each file $f \in \mathcal{F}$ **do**
      Obtain the attribute vector $\boldsymbol{x}_{f,n,t}$
      Compute the estimated user demand:
      $\tilde{d}_{f,n,t} \leftarrow \boldsymbol{x}_{f,n,t}^\top \tilde{\boldsymbol{\theta}}_{n,t}$
    end for
    $\mathcal{F}_{n,t} \leftarrow \arg\max_{\mathcal{F}_n \subseteq \mathcal{F},\, |\mathcal{F}_n| \le c} \sum_{f \in \mathcal{F}_n} \tilde{d}_{f,n,t}$
    Cache the set of files $\mathcal{F}_{n,t}$ in EN n
    Observe user demand $d_{f,n,t}$ of cached files
    Update $\tilde{\boldsymbol{\theta}}_{n,t}$ based on $\boldsymbol{x}_{f,n,t}$ and $d_{f,n,t}$ of all cached files: $\tilde{\boldsymbol{\theta}}_{n,t} \leftarrow \tilde{\boldsymbol{\theta}}_{n,t-1} + \boldsymbol{R}_{n,t}(d_{f,n,t} - \boldsymbol{x}_{f,n,t-1}^\top \tilde{\boldsymbol{\theta}}_{n,t-1})$, where
    $\boldsymbol{R}_{n,t} = \boldsymbol{M}_{n,t}\boldsymbol{x}_{f,n,t}(1 + ||\boldsymbol{x}_{f,n,t}||^2_{\boldsymbol{M}_{n,t}})^{-1}$
    $\boldsymbol{M}_{n,t+1}^{-1} = \boldsymbol{M}_{n,t}^{-1} + \boldsymbol{x}_{f,n,t}\boldsymbol{x}_{f,n,t}^\top - \psi_{n,t+1}^{-2}\boldsymbol{I}_d$
    $\psi_{n,t+1}^{-2} = \xi^{-2}\lambda_{\min}(\boldsymbol{M}_{n,t}^{-1} + \boldsymbol{x}_{f,n,t}\boldsymbol{x}_{f,n,t}^\top)$
  end for
end for

Denote by $\lambda_{\max}(\boldsymbol{A}) = \lambda_1(\boldsymbol{A}) \ge \cdots \ge \lambda_k(\boldsymbol{A}) \cdots \ge \lambda_d(\boldsymbol{A}) = \lambda_{\min}(\boldsymbol{A})$ the eigenvalues of $d \times d$ matrix $\boldsymbol{A}$, and $\lambda_k(\boldsymbol{A})$ is the k-th largest eigenvalue. According to the min-max theorem on matrix eigenvalues, the adaptive threshold ψ_{t+1} should satisfy

$$\lambda_k(\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top) > \lambda_k(\psi_{t+1}^{-2}\boldsymbol{I}_d)\,, \tag{22}$$

for all $k \in \{1, 2, \cdots, d\}$. Since $\lambda_k(\psi_{t+1}^{-2}\boldsymbol{I}_d) = \psi_{t+1}^{-2}$ holds for all k, equivalently, we have $\lambda_{\min}(\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top) > \psi_{t+1}^{-2}$. We may let

$$\psi_{t+1}^{-2} = \xi^{-2}\lambda_{\min}(\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top), \tag{23}$$

where ξ is a constant very close to but larger than one, so that $\lambda_{\min}(\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top - \psi_{t+1}^{-2}\boldsymbol{I}_d)$ is guaranteed to be positive and hence $\boldsymbol{M}_{t+1}^{-1}$ is nonsingular. Meanwhile, the magnitude of ψ_t is also suppressed.

With the aid of H_∞ filter technique, we are able to make performance-guaranteed estimation on the location feature vectors. Therefore, more precise prediction on content popularity can be made, and hence differentiated caching policies can be devised on each EN. The corresponding prediction and caching algorithm is sketched in Algorithm 2. Similar to Algorithm 1, the iteration of the HPDT algorithm can be generalized into the following three steps after initialization.

1. *Predict*: During time slot t, estimation of the location feature vector $\boldsymbol{\theta}_n^*$ of each EN is predicted based on the updated location feature vector by the end of time slot $t-1$.
2. *Optimize and cache*: Based on the predicted content hit rate profile, the set of contents with maximized predicted content hit rate are cached on each EN respectively. Certain contents may be cached in multiple ENs simultaneously.
3. *Observe and update*: At the end of time slot t, the empirical hit rate of the cached files on each EN is observed, which is then used to update the input of the H_∞ filtering process, yielding the updated location feature vector.

The adaptive adjustment of ψ in Eq. (23) is crucial to the online HPDT algorithm. It is tuned to its minimum at each iteration, so that $\boldsymbol{M}_t$ is guaranteed to be positive definite, and meanwhile the upper bound of the cost function J_0 is minimized.

D. Regret Analysis

The H_∞ filter technique provides a robust estimation of the location feature vector regardless of the statistical model of the noise process. However, this approach is also conservative since it needs to accommodate the disturbances of all kinds of noise processes. In this subsection, the performance bound of H_∞ filter based prediction and caching algorithm is given.

Note that the prescribed performance threshold is crucial to the prediction accuracy, the adaptive threshold ψ_t in HPDT algorithm is firstly characterized in this subsection. Note that $\boldsymbol{P}_0$ is initialized as a $d \times d$ symmetric and positive definite matrix, and $\boldsymbol{M}_t^{-1}$ is also guaranteed

to be symmetric and positive definite with the help of ψ_t. According to Weyl's monotonicity theorem [33], the smallest eigenvalue of $\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top$ can be bounded as:

$$\lambda_d(\boldsymbol{M}_t^{-1}) \leq \lambda_d(\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top) \leq \lambda_d(\boldsymbol{M}_t^{-1}) + \lambda_1(\boldsymbol{x}_t\boldsymbol{x}_t^\top), \tag{24}$$

where $\lambda_1(\boldsymbol{x}_t\boldsymbol{x}_t^\top)$ is the largest eigenvalue of $\boldsymbol{x}_t\boldsymbol{x}_t^\top$, and can be simple bounded by matrix trace as $\lambda_1(\boldsymbol{x}_t\boldsymbol{x}_t^\top) \leq \mathrm{tr}(\boldsymbol{x}_t\boldsymbol{x}_t^\top) = ||\boldsymbol{x}_t||^2 \leq \eta^2$. As $\boldsymbol{M}_{t+1}^{-1}$ is positive definite, we have

$$\xi^{-2}\lambda_d(\boldsymbol{M}_t^{-1}) \leq \psi_{t+1}^{-2} \leq \xi^{-2}(\lambda_d(\boldsymbol{M}_t^{-1}) + \eta^2)\,, \tag{25}$$

where ξ^{-2} is very close to but smaller than one. According to Eq. (20), the smallest eigenvalue of $\boldsymbol{M}_{t+1}^{-1}$ equals to $(1-\xi^{-2})\lambda_d(\boldsymbol{M}_t^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top)$, which is suppressed to be small but positive. Hence, $\boldsymbol{M}_{t+1}^{-1}$ is guaranteed to be nonsingular. Iteratively, $\boldsymbol{M}_t^{-1}$ is positive definite for all $t \in \mathcal{T}$.

In the following, a theorem is given to provide a bound on the caching regret of the HPDT algorithm. Let c be the caching size of each EN, and F be the cardinality of the ground file set. Suppose content hit rate satisfies the linear model given by Eq. (13), and note that the attribute vectors are bounded as $||\boldsymbol{x}_{f,n,t}|| \leq \eta$ for all $f \in \mathcal{F}$, $n \in \mathcal{N}$ and $t \in \mathcal{T}$. Let Θ and w be the upper bound of $||\boldsymbol{\theta}_{n,0} - \tilde{\boldsymbol{\theta}}_{n,0}||^2_{\boldsymbol{P}_{n,0}}$ and $w_{n,t}$ for all $n \in \mathcal{N}$ and $t \in \mathcal{T}$, respectively. Then, we have the following theorem.

Theorem 2. *The time-averaged regret $R(T)$ of the HPDT algorithm is of order $O\Big(c\eta\psi N\sqrt{\frac{\Theta}{T} + w^2}\Big)$.*

Please refer to Appendix C for the proof. Theorem 2 indicates that, if the linear model is free of noises, the time-averaged regret of the HPDT algorithm tends to zero as T grows to infinity. Otherwise, the HPDT algorithm may not approach the optimal solution, and its performance depends on the noise magnitude. This is due to the characteristics inherited from the H_∞ filter. Since H_∞ filter makes no assumption on the noise feature, to minimize the worst-case estimation error, it needs to accommodate all possible noise processes, which turns out to be over-conservative. However, when the noise is zero-mean, the regret of exploiting HPDT algorithm reduces to the order of $O(c\eta\psi N\sqrt{\frac{\Theta}{T}})$, which is smaller than that using the RPUC algorithm. In the next section, we will further evaluate the proposed two algorithms by numerically decomposing the estimation errors, and examine the algorithms by experiments on real dataset.

VI. Numerical Analysis and Experimental Results

To validate performance of the proposed caching algorithms, numerical analysis is firstly performed in this section. Afterwards, an experiment based on real-world dataset from YouTube is conducted to further illuminate the performance of the proposed algorithms in practical scenarios.

A. Numerical Analysis

The proposed two algorithms can be used for content caching with different user demand features. In essence, they are estimating the location feature vector $\boldsymbol{\theta}_n^*$, which specifies the location characteristics and user preferences on each EN. Given the linear model of user demand, the performance of the proposed algorithms highly depends on the accuracy of the estimation. We use mean square error (MSE) to evaluate the accuracy of the proposed algorithms. For notational simplicity, the indices n and t are omitted in this section. Let $\boldsymbol{\theta}$ be the underlying feature vector, and $\tilde{\boldsymbol{\theta}}$ be the estimation of $\boldsymbol{\theta}$ by using the proposed algorithms. Denote $\mathbb{E}[\tilde{\boldsymbol{\theta}}] = \bar{\boldsymbol{\theta}}$, then, the MSE can be defined in Euclidean norm as

$$\begin{aligned}\mathsf{MSE}_{\tilde{\boldsymbol{\theta}}} &= \mathbb{E}\big[||\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}||^2\big] \;= \mathbb{E}\Big[\big|\big|\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}} + \bar{\boldsymbol{\theta}} - \boldsymbol{\theta}\big|\big|^2\Big] \\ &= \mathbb{E}\big[||\tilde{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}||^2\big] + ||\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}||^2 \,. \end{aligned} \tag{26}$$

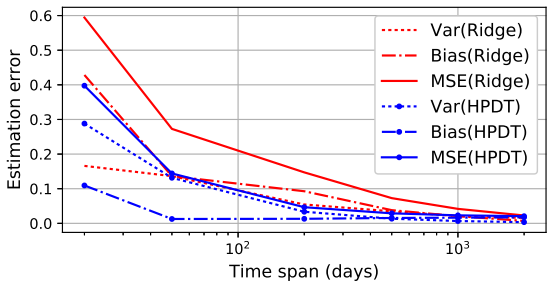
The two terms in Eq. (26) turn out to be the variance and bias of $\tilde{\boldsymbol{\theta}}$, respectively.

Note that the RPUC algorithm is based on ridge regression. Unlike the ordinary least square linear regression, which makes an unbiased estimation on the feature vector, ridge regression intentionally introduces bias so as to reduce variance of the estimation. Moreover, the RPUC algorithm adds a perturbation to the estimation of ridge regression to account for noise uncertainty, which further increases the bias.

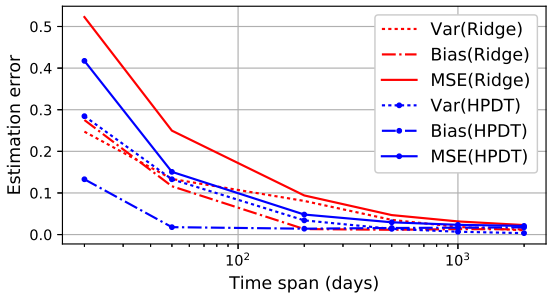
In contrast, the HPDT algorithm makes no assumption on the statistical model of the underlying noise process. It is able to meet the prescribed performance threshold even if the noise process leads to the worst case. Moreover, the HPDT algorithm makes unbiased estimation on the feature vector. This can be observed from the definition of the cost function $\mathcal{J}_0$ in Eq. (14). By decomposing the numerator of $\mathcal{J}_0$, we have

$$\sum_{t=0}^{T} ||\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*||^2 = (T+1) \cdot \mathsf{MSE}_{\tilde{\boldsymbol{\theta}}} \geq T||\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^*||^2. \tag{27}$$

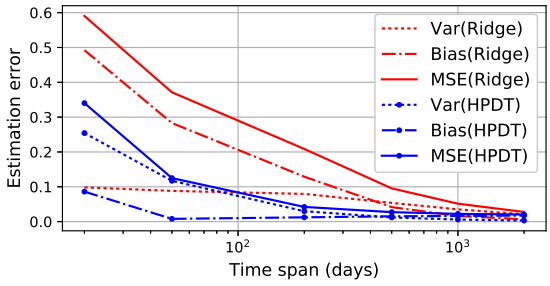
As H_∞ filter makes robust estimation over all kinds of noise structures and initial conditions, given any $\epsilon > 0$, there exists a combination of initial condition $\boldsymbol{\theta}_0$ and noise $\{w_t\}_{t=0}^T$ such that



(a)



(b)



(c)

Fig. 3. Comparison of estimation errors of ridge regression and HPDT algorithm under varying sample rate and noise structure. a) Zero-mean noise; b) Non-zero-mean noise of uniform distribution (with smaller mean value); c) Non-zero-mean noise of normal distribution (with larger mean value).

$\epsilon > ||\boldsymbol{\theta}_0 - \tilde{\boldsymbol{\theta}}_0||^2_{\boldsymbol{P}_0} + \sum_{t=0}^{T} |w_t|^2 \neq 0$. Since this is true for all $\epsilon > 0$, according to Eq. (27), the cost function $\mathcal{J}_0 \geq T||\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^*||^2/\epsilon$, which will grow linearly if $||\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^*||^2 \neq 0$. Consequently, any algorithm that bounds the cost function $\mathcal{J}_0$ must be unbiased, i.e., $||\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^*||^2 = 0$.

On the other hand, the performance of the HPDT algorithm also depends on the *a priori* confidence on the estimation of the initial state, i.e., the selection of $\boldsymbol{P}_0$. A smaller matrix (eigenvalue) should be chosen if the estimation of initial condition is made with larger uncertainty, and vice versa.

TABLE I

COMPARISON OF ESTIMATION VARIANCE AND BIAS

Time Span	Ridge Regression			$H_\infty(\lambda_{\min}(\boldsymbol{P}_0) = 5)$			$H_\infty(\lambda_{\min}(\boldsymbol{P}_0) = 10)$			$H_\infty(\lambda_{\min}(\boldsymbol{P}_0) = 30)$		
	Variance	Bias	MSE	Variance	Bias	MSE	Variance	Bias	MSE	Variance	Bias	MSE
20	0.1659	0.4284	0.5943	0.2881	0.1094	0.3975	0.2687	0.1045	0.3732	0.2647	0.1036	0.3683
50	0.1364	0.1364	0.2728	0.1315	0.0126	0.1441	0.1244	0.0116	0.1360	0.1228	0.0115	0.1343
200	0.0545	0.0927	0.1472	0.0335	0.0131	0.0466	0.0318	0.0129	0.0447	0.0314	0.0129	0.0444
500	0.0347	0.0381	0.0728	0.0134	0.0155	0.0289	0.0127	0.0154	0.0282	0.0126	0.0154	0.0280
1000	0.0211	0.0207	0.0418	0.0067	0.0164	0.0231	0.0064	0.0164	0.0227	0.0063	0.0164	0.0227
2000	0.0157	0.0070	0.0227	0.0034	0.0174	0.0207	0.0032	0.0174	0.0206	0.0032	0.0174	0.0205

We conduct simulation based on synthesized time sequences, which is generated according to a prescribed linear model with zero-mean noise and non-zero-mean noise, respectively. Since the proposed RPUC algorithm is perturbed intentionally, we only present the comparison between ridge regression and the HPDT algorithm. Fig. 3 shows that, under all scenarios, ridge regression provides a more stable estimation as it achieves smaller variance than that using HPDT algorithm when the amount of historical data is small. Such stability advantage of ridge regression benefits from its intentional penalty. However, the HPDT algorithm performs much better in terms of bias and MSE under varying sampling rate in all scenarios, which proves the robustness of the HPDT algorithm. Moreover, with the increase of noise magnitude, the gain of HPDT over ridge regression also increases.

To demonstrate the impact of $\boldsymbol{P}_0$ on the performance of HPDT, estimation results of the HPDT algorithm with different initialization matrices are presented. $\boldsymbol{P}_0$ is initialized as diagonal matrix with positive elements on the main diagonal. When the confidence on the initial state is small, a larger $\boldsymbol{P}_0$ with bigger eigenvalues is used. As shown in Table I, $\boldsymbol{P}_0$ with larger eigenvalue achieves better performance than the others, which means that the initial guess $\boldsymbol{\theta}_0$ is close to the prescribed vector. In practice, the matrix $\boldsymbol{P}_0$ can be selected according to the prior information regarding the initial condition.

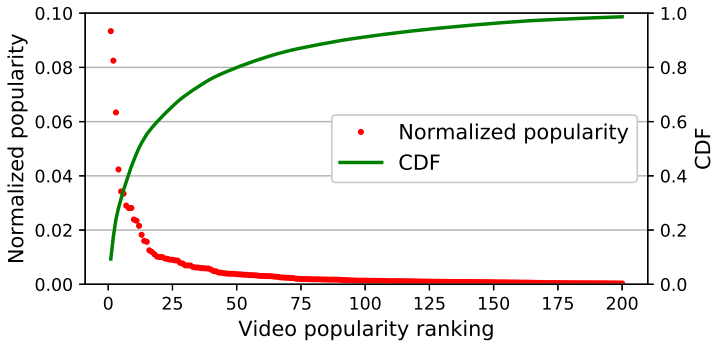
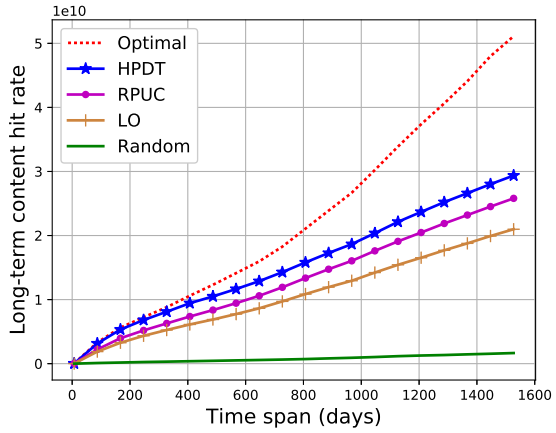


Fig. 4. Popularity skewness of the video set in our experiment. Note that the videos are randomly crawled from YouTube, which may not reflect the overall skewness of video popularity.

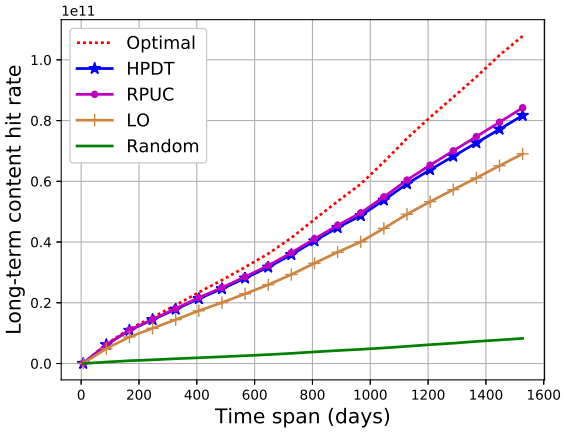
B. Real Dataset Experiment

1) Experiment Setup: To further demonstrate the advantages of the proposed algorithms in practical scenarios, we conduct an experiment on the dataset crawled from YouTube. On YouTube, some video owners made their video view statistics open to public. Among other information, the view amount information is recorded on a daily basis. To obtain such information, a Python-based crawling program is written, and the request record of each video is crawled into a `.json` file. Based on which we conducted the rest Python-based experiments . In total, 800 videos are randomly crawled, which were uploaded before January 2013, with full view statistics till May 2017. The most popular video has been watched over 2.85 billion times by the end of the timespan, while the least popular one has been rarely viewed across the time span. Fig. 4 shows the statistics of video popularity skewness. The popularity of the most popular 25 videos is highly skewed, and the most popular 50 videos account for almost 80% of the total view amount.

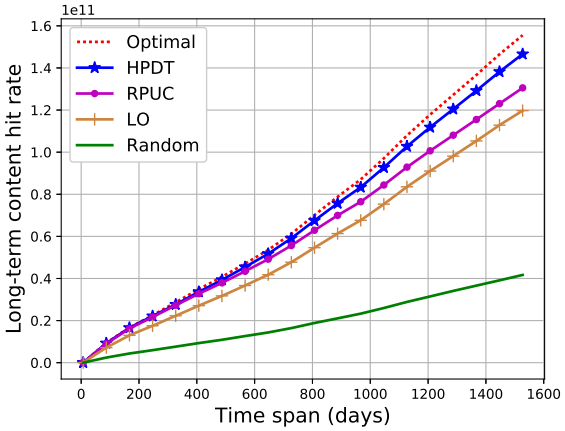
Note that the dataset only contains the global statistics of each video record (with the recent update of YouTube webpage, even the global view statistics is inaccessible), while the view statistics of most online video content providers in a local area is unavailable. To emulate the video request processes on different locations, the original statistic of each video is shifted and scaled randomly over the time span. For the record of a certain video, by shifting, the request record of the original global data of each content is moved backward and forward on the timespan. By scaling, each content request record is then randomly scaled up and down. After shifting and scaling, the original global request statistics are transformed and treated as



(a)



(b)



(c)

Fig. 5. The content hit rate comparison between the proposed algorithm and other benchmarks with varying caching size, where the total number of files is 800 and the caching sizes of six figures are 5, 20, and 100 respectively.

the requests from different locations. The key point is that the pattern of the record remains valid after the above transformation. In this way, we are able to characterize the location diversity based on the emulated view statistics.

Specifically, consider the content library containing those 800 videos, each video can be cached on 3 ENs, each with caching size c. Content refreshing is performed upon the network traffic pattern. For example, wireless traffic presents regular peak and valley every day. Hence, content refreshing can be performed during the off-peak period with minimal impact on normal network activity. A video can be characterized from several aspects, including video quality, genre, length, and historical view statistic. In this experiment, we use view amount information in the past 7 days as the attribute vector of each content, i.e., $d = 7$. Based on the attribute vectors and an initial guess on the content popularity, the algorithms gradually select contents that are predicted to be more popular than the others, and cache them on each EN accordingly. The long-term content hit rates of the proposed algorithms are shown by comparing with the following benchmark algorithms. 1) Hindsight optimal. By analyzing the full view record over the time span at each EN, the most popular videos are selected and cached respectively. Note that this benchmark requires future information and hence cannot be implemented in practice. 2) Location oblivious (denoted by LO). During each time slot, the historical demands of all the contents from all ENs are analyzed, afterwards the ones that are predicted (by ridge regression) to have the highest demands in the next time slot are identified. Then, all ENs will cache the same set of contents without location differentiation. 3) Random. A random set of videos is selected to update the ENs during each time slot.

2) Experimental Results: As shown in Fig. 4, the popularity of YouTube video is highly skewed, and the most popular 10% videos have attracted almost 90% of user requests. The skewness of video popularity has also been validated in [25]. The popularity of this dataset can be roughly divided into three levels: highly skewed (popularity of the top 15 videos); medium skewed (popularity of videos ranking from 16 to 50); and less skewed (the rest ones).

Figure 5 shows the comparison of long-term content hit rates of different algorithms with varying EN caching sizes. The performance of the caching algorithms is affected by the skewness of the popularity profile. However, the proposed location-based approaches always outperform the location-oblivious scheme in varying caching size scenarios. Specifically, when the caching size falls into the highly skewed area and less skewed area, the proposed caching algorithms RPUC and HPDT outperform other benchmarks considerably. In particular, the HPDT algorithm

performs better than the RPUC. For the highly skewed area, the top 5 videos present much higher variance than the rest. As a result, the noise mean of those records is also significant. Since RPUC algorithm is designed for zero-mean noises, their performance is limited when noise amplitude is significant. In contrast, when the caching size falls into the less skewed area, the algorithms need to corporate various noise types of different videos, which may not always be zero-mean. RPUC and HPDT perform equally well when the caching size falls into the medium skewed area (Fig. 5(b)). The H_∞ filter is utilized to provide guaranteed performance even when noise type lead to the worst case for estimation. As a result, the HPDT algorithm is conservative yet robust.

Figure 4 also indicates that content popularity is long-tailed, i.e., the less popular contents attract almost vanishing requests compared to the popular ones. As a result, the total hit rate of different caching schemes in Fig. 5 does not increase linearly with the caching size. Note that, both algorithms run iteratively in an online fashion. During each iteration, the most computationally intensive execution is the n times sorting of the estimated demands of F contents, which has a typical computational complexity of $O(nFlogF)$. Complexity of the value assignments and matrix update could be neglected compared with sorting. As a result, both algorithms are of low time complexity.

C. Discussions

1) Another dimension of the prediction: This work focuses on the estimation of location feature vector. Actually, the selection of video attribute vector $\boldsymbol{x}_{f,n}$ also influences the prediction accuracy. As mentioned before, other factors, such as video quality, length and genre, can also be used to characterize video contents. If such labeling information is available, by reducing the dimension as well as training the dataset, we can identify influential features that affect content popularity. On the other hand, for the location feature vector $\boldsymbol{\theta}_n^*$, both RPUC and HPDT algorithms are designed with the precondition that $\boldsymbol{\theta}_n^*$ is time-invariant, as indicated in both Eq. (6) and (13). Actually, the HPDT algorithm can be directly extended to time-variant scenario if the state equation is also linear, i.e., $\boldsymbol{\theta}_{t+1} = \boldsymbol{A}\boldsymbol{\theta}_t + v_t$, where $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ is the transition matrix, and $v_t \in \mathbb{R}^d$ is the state noise vector. By resorting to H_∞ technique, the adaptive estimation on $\boldsymbol{\theta}_{n,t}$ can be made with guaranteed accuracy [28].

2) The applicability in practical video streaming: The proposed popularity prediction approach can be applied to the delivery of various types of contents. As video content consumes the

most bandwidth, it deserves to be in-depth investigated. Practical video streaming protocols (such as HTTP-based Adaptive Streaming, HAS) divide a video content into several chunks/segments, each with multiple bitrates and quality versions [38]. Those pull-based streaming protocols dynamically change the quality of the streamed video according to the observed network conditions on a per-fragment basis. Most of the research works on adaptive video streaming (both server-side bitrate switching [37] and client-side switching [39]) strive to predict the network condition when transmitting the next video segment. In contrast, with the aid of edge storage resources, our work focuses on push-based content distribution. In other words, estimating the available bandwidth is out of the scope of this work, and content updates are scheduled to off-peak periods, where streaming bandwidth is sufficient.

When streaming video contents based on HAS, whether to cache individual segments or the whole quality representation depends on both the available bandwidth and the content popularity. In particular, for the popular contents, users tend to keep requesting them regardless of the received video quality. Hence, it is more appropriate to store the whole representation of the video so as to fulfill users' requirements via dynamic bandwidth. As the popularity profile of contents are highly skewed (shown in Fig. 4), the streaming provider only needs to store the full quality representation of the most popular videos (the amount of which really depends on the caching size budget). For the less popular ones, the provider may choose to cache the individual segments that are of moderate quality, so as to save bandwidth and meanwhile be responsive to user requests. The rationale behind such decision is the content popular profile and the available caching resources, which is the merit of our work. In this sense, the proposed location-based popularity prediction approaches are crucial in HAS-based streaming system, and the prediction of content popularity and network condition will collectively contribute to improved video streaming.

VII. Conclusion

In this paper, we investigate popularity prediction for mobile edge caching, with special focus on location awareness. We model the content popularity profile by a linear model and propose online algorithms to deal with different statistical models of the noise process. The proposed RPUC algorithm achieves content hit rate that asymptotically approaches the optimal solution when the noise is zero-mean. Noticing that the noise may not necessarily be zero-mean, we resort to the H_∞ filter technique and propose the HPDT algorithm for popularity prediction. This

algorithm can achieve guaranteed prediction accuracy even when the worst-case noise occurs. Both algorithms can be implemented without training phases. Numerical analysis shows how the performance of the proposed algorithms is affected by different types of noises, the amount of historical data, and the initial state. Extensive experiments on real dataset demonstrate the advantage of the proposed algorithm, which helps to make customized caching decisions in practical scenarios. For future works, we will exploit locational features of neighboring ENs to make better caching decisions.

APPENDIX A
PROOF OF LEMMA 1

Let $\boldsymbol{h}_{f,n} = \boldsymbol{\Phi}_{f,n}^{\top}\boldsymbol{y}_{f,n}$, based on Eq. (8), the estimation error can be rewritten as

$$\begin{aligned} & |\boldsymbol{x}_{f,n}^{\top}\tilde{\boldsymbol{\theta}}_n - \boldsymbol{x}_{f,n}^{\top}\boldsymbol{\theta}_n^*| \\ = \; & |\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{h}_{f,n} - \boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}(\boldsymbol{\Phi}_{f,n}^{\top}\boldsymbol{\Phi}_{f,n} + \mu\boldsymbol{I}_d)\boldsymbol{\theta}_n^*| \\ = \; & |\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{\Phi}_{f,n}^{\top}(\boldsymbol{y}_{f,n} - \boldsymbol{\Phi}_{f,n}\boldsymbol{\theta}_n^*) - \mu\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{\theta}_n^*|. \end{aligned}$$

Since $||\boldsymbol{\theta}_n^*|| \leq \zeta$, Hölder's inequality indicates that $|\mu\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{\theta}_n^*| \leq \zeta\mu||\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}||$. Then, the estimation error is bounded as

$$\begin{aligned} |\boldsymbol{x}_{f,n}^{\top}\tilde{\boldsymbol{\theta}}_n - \boldsymbol{x}_{f,n}^{\top}\boldsymbol{\theta}_n^*| \; \leq \; & |\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{\Phi}_{f,n}^{\top}(\boldsymbol{y}_{f,n} - \boldsymbol{\Phi}_{f,n}\boldsymbol{\theta}_n^*)| \\ & + \; \zeta\mu||\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}||. \end{aligned} \tag{28}$$

The right-hand side of above inequality decomposes the estimation error into two parts, where the first (variance term) specifies the error caused by linear model, and the second (bias term) is the bias incurred by ridge regression parameter μ. According to Eq. (6), we have $\mathbb{E}[\boldsymbol{y}_{f,n} - \boldsymbol{\Phi}_{f,n}\boldsymbol{\theta}_n^*] = 0$. The Azuma's inequality gives a probabilistic upper bound on the variance term of Eq. (28):

$$\begin{aligned} & \mathbb{P}\left\{\left|\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{\Phi}_{f,n}^{\top}(\boldsymbol{y}_{f,n} - \boldsymbol{\Phi}_{f,n}\boldsymbol{\theta}_n^*)\right| > \delta||\boldsymbol{x}_{f,n}||_{\boldsymbol{V}_{f,n}^{-1}}\right\} \\ \leq \; & 2\exp\Big(-\frac{2\delta^2||\boldsymbol{x}_{f,n}||_{\boldsymbol{V}_{f,n}^{-1}}^2}{||\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{\Phi}_{f,n}^{\top}||^2}\Big) \; \leq \; 2e^{-2\delta^2}, \end{aligned} \tag{29}$$

where the last inequality is due to the fact that

$$\begin{aligned} ||\boldsymbol{x}_{f,n}||_{\boldsymbol{V}_{f,n}^{-1}}^2 &= \boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}(\boldsymbol{\Phi}_{f,n}^{\top}\boldsymbol{\Phi}_{f,n} + \mu\boldsymbol{I}_d)\boldsymbol{V}_{f,n}^{-1}\boldsymbol{x}_{f,n} \\ &\geq \boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{\Phi}_{f,n}^{\top}\boldsymbol{\Phi}_{f,n}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{x}_{f,n} \\ &= ||\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{\Phi}_{f,n}^{\top}||^2. \end{aligned} \tag{30}$$

Hence, the variance term of Eq. (28) can be bounded by $\delta||\boldsymbol{x}_{f,n}||_{\boldsymbol{V}_{f,n}^{-1}}$ with probability at least $1-2e^{-2\delta^2}$. Further, The bias term of Eq. (28) can be bounded as

$$\begin{aligned} ||\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}|| &= \sqrt{\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}\boldsymbol{I}_d\boldsymbol{V}_{f,n}^{-1}\boldsymbol{x}_{f,n}} \\ &\leq \sqrt{\boldsymbol{x}_{f,n}^{\top}\boldsymbol{V}_{f,n}^{-1}(\mu\boldsymbol{I}_d+\boldsymbol{\Phi}_{f,n}^{\top}\boldsymbol{\Phi}_{f,n})\boldsymbol{V}_{f,n}^{-1}\boldsymbol{x}_{f,n}} \\ &= ||\boldsymbol{x}_{f,n}||_{\boldsymbol{V}_{f,n}^{-1}}. \end{aligned} \tag{31}$$

By substituting Eq. (29) and (31) into Eq. (28), the probabilistic bound in Eq. (9) directly follows.

APPENDIX B
PROOF OF THEOREM 1

The total regret depends on the RUPC algorithm's accuracy of estimation on content hit rate, which is elaborated in Lemma 1. According to this lemma, the true hit rate of file f at EN n lies in the confidence interval around the predicted hit rate

$$\mathcal{I}_{f,n,t}=[\boldsymbol{x}_{f,n,t}^{\top}\tilde{\boldsymbol{\theta}}_{n,t}-p_{f,n,t},\ \boldsymbol{x}_{f,n,t}^{\top}\tilde{\boldsymbol{\theta}}_{n,t}+p_{f,n,t}] \tag{32}$$

with high probability.

Let $\mathcal{X}_{n,t}=\{\exists f\in\mathcal{F}:|d_{f,n,t}-\tilde{d}_{f,n,t}|\geq p_{f,n,t}\}$ be the event that there exists at least one file whose true hit rate lies outside its confidence interval. Let $\bar{\mathcal{X}}_{n,t}$ be the complementary event of $\mathcal{X}_{n,t}$, i.e., all files' true hit rates fall into their confidence interval. Let $r_{n,t}$ be the instant regret of a caching algorithm in EN n at time slot t. According to Eq. (4), the total regret depends on the difference between the set of files chosen by the Algorithm and the optimum set, i.e., $\mathcal{F}_{n,t}$ and $\mathcal{F}_{n,t}^{*}$, thus

$$r_{n,t}=\textstyle\sum_{f\in\mathcal{F}_{n,t}^{*}}d_{f,n,t}-\sum_{f\in\mathcal{F}_{n,t}}d_{f,n,t}, \tag{33}$$

and the time-averaged regret can be rewritten as

$$\begin{aligned} R(T)=&\frac{1}{T}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}r_{n,t} \\ =&\frac{1}{T}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\mathbb{1}_{\{\mathcal{X}_{n,t}\}}r_{n,t}+\frac{1}{T}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\mathbb{1}_{\{\bar{\mathcal{X}}_{n,t}\}}r_{n,t}, \end{aligned} \tag{34}$$

where $\mathbb{1}_{\{\mathcal{X}_{n,t}\}}$ is an indicator variable that equals to 1 if event $\mathcal{X}_{n,t}$ happens, and equals to 0 otherwise. To proceed, the two terms in Eq. (34) are bounded separately.

Firstly, consider the case when event $\mathcal{X}_{n,t}$ happens. With the setting of α_t in Eq. (11), for a file f and EN n at time t, we have $\mathbb{P}\{|d_{f,n,t} - \tilde{d}_{f,n,t}| \geq p_{f,n,t}\} \leq 2F^{-1}t^{-2}$. As a result, the frequency of event $\mathcal{X}_{n,t}$ happens on all ENs over the time span can be bounded as:

$$\begin{aligned}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}} \mathbb{1}_{\{\mathcal{X}_{n,t}\}} &\leq \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\sum_{f\in\mathcal{F}} \mathbb{P}\big\{|d_{f,n,t} - \tilde{d}_{f,n,t}| \geq p_{f,n,t}\big\} \\ &\leq \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\sum_{f\in\mathcal{F}} 2F^{-1}t^{-2} = 2N\sum_{t\in\mathcal{T}} t^{-2} \\ &\leq 2N\sum_{t=1}^{\infty} t^{-2} \leq \frac{\pi^2}{3}N. \end{aligned} \tag{35}$$

As content hit rate $d_{f,n,t} \leq \gamma$, according to Eq. (33), a coarse upper bound of the instant regret is $r_{n,t} \leq c\gamma$, where $c = |\mathcal{F}^*_{n,t}|$ is the caching size of each EN. Therefore, the first term of Eq. (34) can be bound as

$$\frac{1}{T}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}} \mathbb{1}_{\{\mathcal{X}_{n,t}\}} r_{n,t} \leq \frac{\pi^2 c\gamma N}{3T}. \tag{36}$$

Then, consider the case when event $\bar{\mathcal{X}}_{n,t}$ happens, all files' true hit rates falls into the confidence interval around their estimation $\tilde{d}_{f,n,t}$. Hence, $|d_{f,n,t} - \tilde{d}_{f,n,t}| \leq p_{f,n,t},\ \forall f \in \mathcal{F}$. With $\hat{d}_{f,n,t} = \tilde{d}_{f,n,t} + p_{f,n,t}$, we have

$$\hat{d}_{f,n,t} - d_{f,n,t} \leq 2p_{f,n,t}. \tag{37}$$

By Eq. (33) and (37), when event $\bar{\mathcal{X}}_{n,t}$ happens, the instant regret $r_{n,t}$ can be bounded as

$$\begin{aligned} r_{n,t}|_{\bar{\mathcal{X}}_{n,t}} &= \textstyle\sum_{f\in\mathcal{F}^*_{n,t}\setminus\mathcal{F}_{n,t}} d_{f,n,t} - \sum_{f\in\mathcal{F}_{n,t}\setminus\mathcal{F}^*_{n,t}} d_{f,n,t} \\ &\leq \textstyle\sum_{f\in\mathcal{F}^*_{n,t}\setminus\mathcal{F}_{n,t}} \hat{d}_{f,n,t} - \sum_{f\in\mathcal{F}_{n,t}\setminus\mathcal{F}^*_{n,t}} d_{f,n,t} \\ &\leq \textstyle\sum_{f\in\mathcal{F}_{n,t}\setminus\mathcal{F}^*_{n,t}} \left(\hat{d}_{f,n,t} - d_{f,n,t}\right) \\ &\leq \textstyle 2\sum_{f\in\mathcal{F}_{n,t}\setminus\mathcal{F}^*_{n,t}} p_{f,n,t}, \end{aligned} \tag{38}$$

where inequality (38) is due to fact that since the algorithm selects files in $\mathcal{F}_{n,t} \setminus \mathcal{F}^*_{n,t}$ rather than $\mathcal{F}^*_{n,t} \setminus \mathcal{F}_{n,t}$, hence the collective upper confidence bound hit rate satisfies $\sum_{f\in\mathcal{F}_{n,t}\setminus\mathcal{F}^*_{n,t}} \hat{d}_{f,n,t} \geq \sum_{f\in\mathcal{F}^*_{n,t}\setminus\mathcal{F}_{n,t}} \hat{d}_{f,n,t}$. We will need two lemmas from [32] for the subsequent analysis.

Lemma 2. *(Lemma 11 of [32]). Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T \in \mathbb{R}^d$ and $\boldsymbol{V}_t = \mu\boldsymbol{I}_d + \sum_{s=1}^{t} \boldsymbol{x}_s\boldsymbol{x}_s^\top$. If $\forall t \in \mathcal{T}$, $||\boldsymbol{x}_t|| \leq \eta$ holds, then for some $\mu > 0$, when $\mu \geq \max(1, \eta^2)$, we have*

$$\sum_{s=1}^{t}(\boldsymbol{x}_s^\top \boldsymbol{V}_t^{-1}\boldsymbol{x}_s) \leq 2\ln\frac{\det(\boldsymbol{V}_t)}{\mu}. \tag{39}$$

Lemma 3. *(Determinant-Trace Inequality, Lemma 10 of [32]). Suppose $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T \in \mathbb{R}^d$, and $\forall t \in \mathcal{T}$ it holds that $||\boldsymbol{x}_t|| \leq \eta$. Let $\boldsymbol{V}_t = \mu \boldsymbol{I}_d + \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}_s^\top$, then for $\mu > 0$, we have*

$$\det(\boldsymbol{V}_t) \leq (\mu + t\eta^2/d)^d. \tag{40}$$

Based on those two lemmas, the second term in Eq. (34) can be bounded as

$$\begin{aligned}
\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}} r_{n,t}|_{\bar{\mathcal{X}}_{n,t}} &\leq 2\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\sum_{f\in\mathcal{F}_{n,t}\setminus\mathcal{F}^*_{n,t}} p_{f,n,t} \\
&\leq 2c\alpha_T \sum_{n\in\mathcal{N}}\sum_{t\in\mathcal{T}} ||\boldsymbol{x}_{f,n,t}||_{\boldsymbol{V}^{-1}_{f,n}} &(41)\\
&\leq 2c\alpha_T \sum_{n\in\mathcal{N}} \sqrt{T\sum_{t\in\mathcal{T}} ||\boldsymbol{x}_{f,n,t}||^2_{\boldsymbol{V}^{-1}_{f,n}}} &(42)\\
&\leq 2c\alpha_T N\sqrt{2T\ln\frac{(\mu+T\eta^2/d)^d}{\mu}}, &(43)
\end{aligned}$$

where Eq. (41) is due the fact that α_t increases with t, Eq. (42) holds because the arithmetic mean of a set of values is smaller than their root-mean square and Eq. (43) is based on the above two lemmas.

By substituting Eq. (43) and (36) into Eq. (34), and together with $\alpha_T = \sqrt{\ln(TF^{\frac{1}{2}})} + \mu\zeta$, we have

$$\begin{aligned}
R(T) &\leq 2c\alpha_T N\sqrt{\frac{2}{T}\ln\frac{(\mu+T\eta^2/d)^d}{\mu}} + \frac{\pi^2 c\gamma N}{3T} \\
&= O\Big(cN\sqrt{\frac{d(\ln T)\ln(\mu+T\eta^2/d)}{T}}\Big). &(44)
\end{aligned}$$

Appendix C
Proof of Theorem 2

The typical estimation error on the demand of a certain content f on EN n during time slot t is $|d_{f,n,t} - \tilde{d}_{f,n,t}|$. Based on Hölder's inequality, the estimation error can be bounded as

$$\begin{aligned}
|d_{f,n,t} - \tilde{d}_{f,n,t}| &= |\boldsymbol{x}^\top_{f,n,t}\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{x}^\top_{f,n,t}\boldsymbol{\theta}^*_n| \\
&\leq ||\boldsymbol{x}_{f,n,t}|| \cdot ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n|| \leq \eta||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n||. &(45)
\end{aligned}$$

Define the per-EN per-slot regret as

$$r_{n,t} = \textstyle\sum_{f\in\mathcal{F}^*_{n,t}} d_{f,n,t} - \sum_{f\in\mathcal{F}_{n,t}} d_{f,n,t}. \tag{46}$$

Since the files in $\mathcal{F}_{n,t} \setminus \mathcal{F}^*_{n,t}$ is selected rather than the ones in $\mathcal{F}^*_{n,t} \setminus \mathcal{F}_{n,t}$, the estimated user demands of those files satisfy

$$\sum_{f \in \mathcal{F}^*_{n,t} \setminus \mathcal{F}_{n,t}} \tilde{d}_{f,n,t} \leq \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}^*_{n,t}} \tilde{d}_{f,n,t}. \tag{47}$$

Eq. (45) indicates that $\tilde{d}_{f,n,t} \leq d_{f,n,t} + \eta ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n||$ and $d_{f,n,t} \leq \tilde{d}_{f,n,t} + \eta ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n||$ for all $f \in \mathcal{F}$, $n \in \mathcal{N}$ and $t \in \mathcal{T}$. With Eq. (47), we have

$$\begin{aligned} r_{n,t} &= \sum_{f \in \mathcal{F}^*_{n,t} \setminus \mathcal{F}_{n,t}} d_{f,n,t} - \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}^*_{n,t}} d_{f,n,t} \\ &\leq \sum_{f \in \mathcal{F}^*_{n,t} \setminus \mathcal{F}_{n,t}} (\tilde{d}_{f,n,t} + \eta ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n||) \\ &\quad - \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}^*_{n,t}} (\tilde{d}_{f,n,t} - \eta ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n||) \\ &\leq \sum_{f \in \mathcal{F}^*_{n,t} \setminus \mathcal{F}_{n,t}} \eta ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n|| \\ &\quad + \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}^*_{n,t}} \eta ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n|| \\ &\leq 2c\eta ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n||. \end{aligned}$$

Then, the cumulative mean regret can be rewritten as

$$\begin{aligned} R(T) &= \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} r_{n,t} \leq \frac{2c\eta}{T} \sum_{n \in \mathcal{N}} \sum_{t \in \mathcal{T}} ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n|| \\ &\leq \frac{2c\eta}{T} \sum_{n \in \mathcal{N}} \sqrt{T \sum_{t \in \mathcal{T}} ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n||^2}. \end{aligned} \tag{48}$$

Since the initialization error and noises are bounded as $||\boldsymbol{\theta}_{n,0} - \tilde{\boldsymbol{\theta}}_{n,0}||^2_{\boldsymbol{P}_{n,0}} \leq \Theta$ and $|w_{n,t}| \leq w$ for all $t \in \mathcal{T}$ and $n \in \mathcal{N}$. Based on Eq. (14) and (16), for a certain EN n, we have

$$\begin{aligned} \sum_{t \in \mathcal{T}} ||\tilde{\boldsymbol{\theta}}_{n,t} - \boldsymbol{\theta}^*_n||^2 &\leq \psi^2 (||\boldsymbol{\theta}_{n,0} - \tilde{\boldsymbol{\theta}}_{n,0}||^2_{\boldsymbol{P}_{n,0}} + \sum_{t \in \mathcal{T}} |w_{n,t}|^2) \\ &\leq \psi^2 (\Theta + Tw^2). \end{aligned} \tag{49}$$

By substituting Eq. (49) into Eq. (48), the time-averaged caching regret is finally bounded as:

$$R(T) \leq 2c\eta\psi N \sqrt{\frac{\Theta}{T} + w^2}, \tag{50}$$

which completes the proof.

REFERENCES

[1] B. Liang, "Mobile Edge Computing," in *Key Technologies for 5G Wireless Systems*, Cambridge University Press, 2017.

[2] ETSI Group Specification, "Mobile Edge Computing (MEC); Technical Requirements," ETSI GS MEC 002 V1.1.1, Mar. 2016.

[3] R. Tandon and O. Simeone, "Harnessing Cloud and Edge Synergies: Toward an Information Theory of Fog Radio Access Networks," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 44-50, Aug. 2016.

[4] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. Shen, "Catalyzing Cloud-Fog Interoperation in 5G Wireless Networks: An SDN Approach," *IEEE Netw.*, vol. 31, no. 5, pp. 14-20, Sep./Oct. 2017.

[5] L. Tong, Y. Li, and W. Gao, "A Hierarchical Edge Cloud Architecture for Mobile Computing," in *Proc. of IEEE INFOCOM*, San Francisco, CA, USA, 2016, pp. 1-9.

[6] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the Air: Exploiting Content Caching and Delivery Techniques for 5G Systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131-139, Feb. 2014.

[7] C. Li, P. Frossard, H. Xiong, and J. Zou, "Distributed Wireless Video Caching Placement for Dynamic Adaptive Streaming," in *Proc. of ACM NOSSDAV*, Klagenfurt, Austria, 2016, pp. 1-6.

[8] S. Zhang, N. Zhang, P. Yang, and X. Shen, "Cost-Effective Cache Deployment in Mobile Heterogeneous Networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11264-11276, Dec. 2017.

[9] A. Liu and V. K. N. Lau, "Exploiting Base Station Caching in MIMO Cellular Networks: Opportunistic Cooperation for Video Streaming," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 57-69, Jan. 2015.

[10] T. H. Luan, L. X. Cai, and X. Shen, "Impact of Network Dynamics on User's Video Quality: Analytical Framework and QoS Provision," *IEEE Trans. Multimedia*, vol. 12, no. 1, pp. 64-78, Jan. 2010.

[11] K. Shanmugam *et al.*, "FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402-8413, Dec. 2013.

[12] S. Zhang, N. Zhang, X. Fang, P. Yang, and X. Shen, "Cost-Effective Vehicular Network Planning with Cache-Enabled Green Roadside Units," in *Proc. of IEEE ICC*, Paris, France, 2017, pp. 1-6.

[13] W. C. Ao and K. Psounis, "Distributed Caching and Small Cell Cooperation for Fast Content Delivery," in *Proc. of ACM MobiHoc*, Hangzhou, China, 2015, pp. 127-136.

[14] M. Gregori, J. Gómez-Vilardebó, J. Matamoros, and D. Gündüz, "Wireless Content Caching for Small Cell and D2D Networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5 pp. 1222-1234, May 2016.

[15] M. Ji, G. Caire, and A. F. Molisch, "Wireless Device-to-Device Caching Networks: Basic Principles and System Performance," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176-189, Jan. 2016.

[16] B. N. Bharath, K. G. Nagananda, and H. V. Poor, "A Learning-Based Approach to Caching in Heterogeneous Small Cell Networks," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1674-1686, Apr. 2016.

[17] P. Blasco and D. Gündüz, "Learning-Based Optimization of Cache Content in a Small Cell Base Station," in *Proc. of IEEE ICC*, Sydney, Australia, 2014, pp. 1897-1903.

[18] S. Roy, T. Mei, W. Zeng, and S. Li, "Towards Cross-Domain Learning for Social Video Popularity Prediction," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1255-1267, Oct. 2013.

[19] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-Aware Proactive Content Caching With Service Differentiation in Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 1024-1036, Feb. 2017.

[20] S. Li, J. Xu, M. van der Schaar, and W. Li, "Trend-Aware Video Caching Through Online Learning," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2503-2516, Dec. 2016.

[21] Y. Zhou, L. Chen, C. Yang, and D. M. Chiu, "Video Popularity Dynamics and Its Implication for Replication," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1273-1285, Aug. 2015.

[22] S. Dhar and U. Varshney, "Challenges and Business Models for Mobile Location-based Services and Advertising," *Commun. ACM*, vol. 54, no. 5, pp. 121-128, May 2011.

[23] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen, "Dynamic Mobile Edge Caching with Location Differentiation," in *Proc. of IEEE GLOBECOM*, Singapore, 2017, pp. 1-6.

[24] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-Centric Sparse Multicast Beamforming for Cache-Enabled Cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6118-6131, Dec. 2016.

[25] M. Cha *et al.*, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," in *Proc. of ACM IMC*, San Diego, CA, USA, 2007, pp. 1-14.

[26] Y. Feng and D. P. Palomar, "A Signal Processing Perspective of Financial Engineering," *Foundations and Trends in Signal Processing*, vol. 9, no. 1-2, pp. 1-231, 2015.

[27] T. Hastie, R. Tibshirani, and J. Friedman, "*The Elements of Statistical Learning: Data Mining, Inference, and Prediction*," Second Edition, Springer Series in Statistics, 2008.

[28] J. Cai, X. Shen, and J. W. Mark, "Robust Channel Estimation for OFDM Wireless Communication Systems-An H_∞ Approach," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 2060-2071, Nov. 2004.

[29] W. Zhuang, "Adaptive H_∞ Channel Equalization for Wireless Personal Communications," *IEEE Trans. Veh. Technol.*, vol. 48, no. 1, pp. 126-136, Jan. 1999.

[30] D. Simon, "*Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*," Wiley-Interscience, 2006.

[31] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, "Contextual Bandits with Linear Payoff Functions," in *Proc. of AISTATS*, Fort Lauderdale, FL, USA, 2011, pp. 208-214.

[32] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved Algorithms for Linear Stochastic Bandits," in *Proc. of NIPS*, Granada, Spain, 2011, pp. 2312-2320.

[33] I. C. F. Ipsen and B. Nadler. "Refined Perturbation Bounds for Eigenvalues of Hermitian and Non-Hermitian Matrices," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 1, pp. 40-53, 2009.

[34] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding Performance of Edge Content Caching for Mobile Video Streaming," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1076-1089, May 2017.

[35] W. Hu, Z. Wang, M. Ma, and L. Sun, "Edge Video CDN: A Wi-Fi Content Hotspot Solution," *Journal of Computer Science and Technology*, vol. 31, no. 6, pp. 1072-1086, Nov. 2016.

[36] G. Ma, Z. Wang, M. Chen, and W. Zhu, "APRank: Joint Mobility and Preference-Based Mobile Video Prefetching," in *Proc. of IEEE ICME*, Hong Kong, China, 2017, pp. 7-12.

[37] S. Akhshabi, L. Ananthakrishnan, A. Begen, and C. Dovrolis, "Server-Based Traffic Shaping for Stabilizing Oscillating Adaptive Streaming Players," in *Proc. ACM SIGMM NOSSDAV*, Oslo, Norway, 2013, pp. 19-24.

[38] Z. Lu, S. Ramakrishnan, and X. Zhu, "Exploiting Video Quality Information with Lightweight Network Coordination for HTTP-based Adaptive Video Streaming," *IEEE Trans. Multimedia*, to appear, DOI: 10.1109/TMM.2017.2772802.

[39] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *Proc. ACM SIGCOMM*, New York, NY, USA, 2015, pp. 325-338.