

# Dynamic Mobile Edge Caching with Location Differentiation

Peng Yang\*, Ning Zhang<sup>†</sup>, Shan Zhang<sup>‡</sup>, Li Yu\*, Junshan Zhang<sup>‡</sup>, and Xuemin (Sherman) Shen<sup>†</sup>

\*School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China

<sup>†</sup>Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada

<sup>‡</sup>School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona, USA

Email: \*{yangpeng, hustlyu}@hust.edu.cn, <sup>†</sup>{n35zhang, s372zhan, sshen}@uwaterloo.ca, <sup>‡</sup>junshan.zhang@asu.edu

**Abstract**—Mobile edge caching enables content delivery directly within the radio access network, which effectively alleviates the backhaul burden and reduces round-trip latency. To fully exploit the edge resources, the most popular contents should be identified and cached. Observing that content popularity varies greatly at different locations, to maximize local hit rate, this paper proposes an online learning algorithm that dynamically predicts content hit rate, and makes location-differentiated caching decisions. Specifically, a linear model is used to estimate the future hit rate. Considering the variations in user demand, a perturbation is added to the estimation to account for uncertainty. The proposed learning algorithm requires no training phase, and hence is adaptive to the time-varying content popularity profile. Theoretical analysis indicates that the proposed algorithm asymptotically approaches the optimal policy in the long term. Extensive simulations based on real world traces show that, the proposed algorithm achieves higher hit rate and better adaptiveness to content popularity fluctuation, compared with other schemes.

## I. INTRODUCTION

The soaring mobile traffic has put high pressure on the paradigm of Cloud-based service provisioning, because moving a large volume of data into and out of the Cloud wirelessly consumes 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 access networks [1]. Residing on the network edge, MEC makes storage and computing resources available to mobile users through one-hop wireless connections, facilitating a number of mobile services, such as local content caching, augmented reality, and cognitive assistance [2].

Among other services, content caching on the edge gains increased attention [3]-[9]. In particular, with the prevalence of social media, contents, such as high-resolution videos, are spreading among mobile users in viral fashion, putting tremendous pressure on the network backhaul [10], [11]. It is forecast that, caching contents on network edge can reduce up to 35% traffic demand on the backhaul [2]. Unfortunately, compared with the ever-increasing content volume, the storage on the edge node (EN) is always limited. It is impossible to cache all the contents locally. Hence, it becomes crucial to identify the optimal set of contents that maximizes the cache utilization.

Content popularity is an effective measure for making caching decisions, based on which influential contents can be identified and cached proactively [12]. Yet, content popularity

is unknown *a priori*, it is hard to directly select and cache the most popular ones. Blasco *et al.* developed an online algorithm to learn the content popularity profile [6]. Specifically, it predicts future content hit rate based on the number of instantaneous requests of cached contents, and then makes new caching decisions correspondingly. In practice, however, the popularity profile of content is not only unknown, but also varying since user's interests are constantly changing [13], and meanwhile new contents are being created. To maximize cache utilization under the condition of varying and unknown popularity profile, Müller *et al.* proposed a context-aware caching algorithm based on user information, which includes users' ages, genders or their preferences [9]. However, relying on user information for context differentiation is risky since such information is extremely sensitive and often unavailable. Alternatively, exploiting location features for context differentiation is a feasible approach. Generally, locations can be classified according to their social functions, such as residential area and business district. Users in different areas have diverse interests [13]. Hence, locational statistics, including the number of users and regional content preferences, can be used for context differentiation. Based on which fine-grained caching decisions can be made to improve the cache utilization.

In this paper, we investigate the problem of mobile edge caching with location differentiation. Since the ability of identifying popular contents is crucial, this problem is challenging in the following ways. Firstly, the future hit rate of a content at a certain location is unknown ahead. Secondly, though the popularity diversity among different locations is evident, there is no established model of how location features affect content hit rate. Thirdly, content hit rate is varying continuously, so the caching strategy should dynamically adjust to the changes. To address those issues, we propose a novel learning algorithm that estimates future content hit rate based on a linear prediction model. This model incorporates content feature and location characteristics, with well-balanced stability and estimation accuracy. Considering the impact of random noise, a subtly designed perturbation is added to the prediction of the linear model to account for uncertainty. Theoretical analysis indicates that the proposed algorithm achieves sublinear regret, i.e., it asymptotically approaches the optimal strategy in the long term. Extensive simulations based on real world traces show that the proposed caching algorithm achieves better

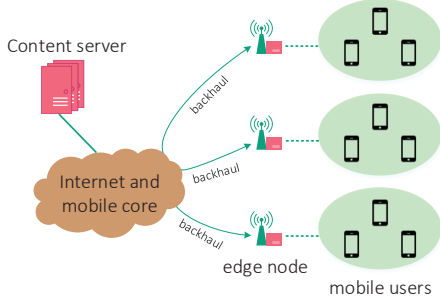


Fig. 1. Network model of mobile edge caching.

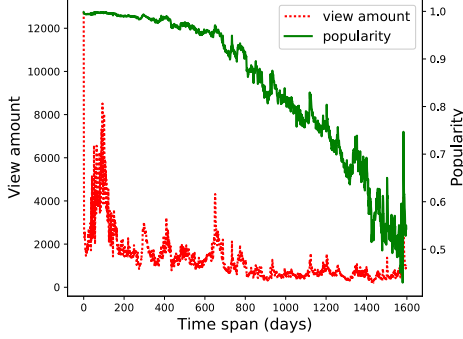


Fig. 2. The daily view amount and popularity curves of a YouTube video since uploaded. Note that the popularity score is calculate based on the statistics of a set of randomly crawled videos.

accuracy on hit rate prediction, and meanwhile adapts steadily to the popularity dynamics.

The remainder of the paper is organized as follows. Section II describes the system model and the caching problem formulation. Section III presents the proposed location differentiated caching algorithm, followed by the theoretical regret analysis in Section IV. Simulation results are shown in Section V and concluding remarks are given in Section VI.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

### A. Network Model

Capacity-augmented base stations, WiFi access points and other devices with excess capacity can be exploited for EN deployment [1]. In this paper, the storage resources on ENs are harnessed for content caching services. Specifically, consider the mobile edge caching model illustrated in Fig. 1, a set of ENs  $\mathcal{N} = \{1, 2, \dots, N\}$  is deployed with separated backhaul links connecting to the mobile core network. Each EN  $n$  is associated with a distinct location and has different characteristics in terms of content requests compared with others. Online contents are dynamically pushed to ENs so that user's content requests can be processed with reduced latency. Each EN serves a disjoint set of mobile users.

### B. Content Popularity with Location Differentiation

A simple yet effective caching strategy is pushing the most popular contents to the network edge. Hence, local content hit

rate is maximized and user requests are served with reduced latency and improved quality of experience. Extensive works have been done on the popularity of contents, especially video files [10], [13], [14]. According to the statistics we crawled from YouTube, as illustrated in Fig. 2, the popularity profile of a video file varies in two-fold. On one hand, the daily view amount is varying. On the other hand, as other videos' daily view amounts are also varying and new videos are being uploaded, the popularity score of a video file is constantly fluctuating. Moreover, location-related characteristics also affect the content popularity. As a result, general caching strategies that based on fixed popularity profile are not optimal in practice.

Consider a set of files  $\mathcal{F} = \{1, 2, \dots, F\}$  that can be cached at ENs, and let  $c < F$  be the caching size of each EN. We assume that all contents are of equal size<sup>1</sup> and the size is normalized to 1, i.e., each EN can cache up to  $c$  contents. We focus on those files' popularity dynamics in a sequence of time slots  $\mathcal{T} = \{1, 2, \dots, T\}$ . Let  $\mathbf{x}_{f,n,t} \in \mathbb{R}^d$  be a  $d$ -dimensional feature vector of file  $f$  associated with EN  $n$  observed before time slot  $t$ . For a certain EN  $n$ , the hit rate<sup>2</sup> of file  $f$  during time slot  $t$ , denoted by  $d_{f,n,t}$ , is statistically linear with respect to its feature vector  $\mathbf{x}_{f,n,t}$ , i.e.,

$$\mathbb{E}[d_{f,n,t} | \mathbf{x}_{f,n,t}] = \mathbf{x}_{f,n,t}^\top \boldsymbol{\theta}_n^*, \quad (1)$$

where  $\boldsymbol{\theta}_n^* \in \mathbb{R}^d$  is the unknown true parameter vector associated with EN  $n$ . The vector  $\mathbf{x}_{f,n,t}$  between file  $f$  and EN  $n$  may contain feature information like the frequency of file  $f$  being cached at EN  $n$  up to time  $t$ , the hit rate of  $f$  at EN  $n$  during the last 5 time slots, 30 time slots etc. The parameter vector  $\boldsymbol{\theta}_n^*$  represents the specific location characteristics at EN  $n$  and determines the particular combination of different features on the expected hit rate. In this way, a certain content is expected to have different hit rates at different ENs. This linear prediction model is widely used in other areas like signal processing and financial engineering [15], [16]. It provides a method to predict content hit rate, which is essential to proactive content caching.

### C. Problem Formulation

As indicated by Fig. 2, the popularity of a content is varying constantly. Hence, we intend to perform dynamic content caching that constantly updates the files on ENs to achieve higher long-term hit rate. To this end, contents with higher popularity at different locations should be proactively identified and cached respectively, and meanwhile the less popular ones should be evicted. Let  $\mathcal{F}_{n,t}$  denote the set of contents cached at EN  $n$  during time slot  $t$ , then the dynamic caching problem can be formulated as the following long-term hit rate maximization (LHRM) problem:

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

<sup>1</sup>In case contents are of different sizes, they can be split into smaller ones of equal size. For example, the widely used DASH (Dynamic Adaptive Streaming over HTTP) protocol breaks content into small segments before transmission.

<sup>2</sup>We define *hit rate* as the number of content requests rather than a ratio.

However, the popularity profile, i.e., the hit rate of contents at each EN, is unknown *a priori*. Hence the decision variables  $\mathcal{F}_{n,t}$  in above optimization problem is intractable directly. For convenience, denote the optimal caching strategy  $\mathcal{F}_{n,t}^*$  for EN  $n$  at time  $t$ . We have that

$$\mathcal{F}_{n,t}^* = \arg \max_{|\mathcal{F}_{n,t}| \leq c} \sum_{f \in \mathcal{F}_{n,t}} d_{f,n,t}, \forall n \in \mathcal{N}, t \in \mathcal{T}. \quad (3)$$

Define the long-term *regret* of a solution respect to the optimal caching strategy as

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

Then, the LHRM problem can be reformulate as a long-term regret minimization (LRM) problem:

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

In the long term, if an algorithm can continuously identify the optimal set  $\mathcal{F}_{n,t}^*$  and cache those files at the corresponding ENs, the algorithm achieves *zero-regret*. Since  $\mathcal{F}_{n,t}^*$  is unknown *a priori*, our goal is to develop a caching algorithm that makes good estimation on future content hit rate, and hence better identifies the popular ones. To this end, we propose an online learning algorithm that dynamically adjusts the estimation of location parameter vectors. The estimation error is carefully bounded so that the proposed algorithm asymptotically approaches the optimal caching policy.

### III. LOCATION DIFFERENTIATED CONTENT CACHING ALGORITHM

To better characterize different locations and make accurate prediction on future content hit rate, we resort to the linear model given by Eq. (1). It can be interpreted that, at time slot  $t$ , given the feature vector  $\mathbf{x}_{f,n,t}$ , the hit rate of file  $f$  at EN  $n$  is predicted to be the linear combination of the features, which gives a feasible way to predict the content hit rate. However, the parameter vector for combination is unknown *a priori*. Therefore, a good estimation of the true parameter vector  $\boldsymbol{\theta}_n^*$  will lead to accurate prediction of the content hit rate.

In this section, we first present a regression-based method to estimate the parameter vector of each EN. Then we present an online caching algorithm that predicts future content hit rate based on the continually evolving parameter vector.

#### A. Predicting Content Hit Rate

Consider a specific file  $f$  and EN  $n$ , the estimation of parameter vector  $\boldsymbol{\theta}_n^*$  can be performed in an online fashion based on file  $f$ 's historical data. Let  $\Phi_{f,n} \in \mathbb{R}^{m \times d}$  be the historical feature vectors of file  $f$ , where  $m$  is the frequency of file  $f$  being cached at EN  $n$  up to time slot  $t$ , and the  $m$ -th row of  $\Phi_{f,n}$  is the corresponding feature vector  $\mathbf{x}_{f,n,m}$ . Denote  $\mathbf{y}_{f,n} \in \mathbb{R}^m$  the  $m$ -time empirical hit rate of file  $f$  at EN  $n$ . By applying the standard ordinary least square linear regression, i.e.,  $\boldsymbol{\theta}_n^* = \arg \min_{\boldsymbol{\theta}_n} \|\mathbf{y}_{f,n} - \Phi_{f,n} \boldsymbol{\theta}_n\|$ , the closed-form

estimation of  $\boldsymbol{\theta}_n^*$  can be derived as  $(\Phi_{f,n}^\top \Phi_{f,n})^{-1} \Phi_{f,n}^\top \mathbf{y}_{f,n}$ . However, considering that feature vectors may be correlated, and hence the matrix  $\Phi_{f,n}^\top \Phi_{f,n}$  could be singular, rendering the estimation of  $\boldsymbol{\theta}_n^*$  fluctuate significantly. Instead of the unbiased estimation made by ordinary least square linear regression, ridge regression makes biased estimation by adding a control parameter that restricts the magnitude of the parameter vector, which helps to improve the estimation stability. By ridge regression, the estimation of  $\boldsymbol{\theta}_n^*$  can be explicitly given as

$$\tilde{\boldsymbol{\theta}}_n = (\Phi_{f,n}^\top \Phi_{f,n} + \lambda \mathbf{I}_d)^{-1} \Phi_{f,n}^\top \mathbf{y}_{f,n}, \quad (6)$$

where  $\mathbf{I}_d \in \mathbb{R}^{d \times d}$  is the identity matrix and  $\lambda > 0$  is the control parameter that guarantees the stability of estimation. The accuracy of estimation depends on the amount of data and  $\lambda$ . For convenience, let  $\mathbf{V}_{f,n} = \Phi_{f,n}^\top \Phi_{f,n} + \lambda \mathbf{I}_d$  for all  $f \in \mathcal{F}$  and  $n \in \mathcal{N}$ . The following lemma, which is slightly manipulated from [16], gives an upper bound on the estimation error of ridge regression.

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

$$\|\mathbf{x}_{f,n}^\top \tilde{\boldsymbol{\theta}}_n - \mathbf{x}_{f,n}^\top \boldsymbol{\theta}_n^*\| \leq (\delta + \zeta \lambda) \sqrt{\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n}} \quad (7)$$

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

Please refer to Appendix for the proof. The upper bound of estimation error provided in Lemma 1 can be interpreted that, the true hit rate falls into the confidence interval around the estimation with high probability. Based on Lemma 1, we propose a dynamic content hit rate prediction and caching algorithm.

#### B. Caching Algorithm

The location differentiated EN caching algorithm is sketched in Algorithm 1. During each time slot, the algorithm first updates the estimated parameter vector  $\tilde{\boldsymbol{\theta}}_{n,t}$ . With the accumulation of historical data,  $\tilde{\boldsymbol{\theta}}_{n,t}$  will finally converge to the true parameter vector  $\boldsymbol{\theta}_n^*$ . Then, based on the instantaneous content feature vector  $\mathbf{x}_{f,n,t}$ , the predicted hit rate  $\tilde{d}_{f,n,t}$  is obtained according to the linear model. Furthermore, a perturbation term  $p_{f,n,t}$  is added to the linear prediction, where

$$p_{f,n,t} = \alpha_t \sqrt{\mathbf{x}_{f,n,t}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n,t}}, \quad (8)$$

and  $\alpha_t = [\ln(tF^{\frac{1}{2}})]^{\frac{1}{2}} + \zeta \lambda$ . The rationale of the perturbation is that Eq. (1) only gives an expectation value of the hit rate while omitting the potential random noises. The perturbation specified by Eq. (8) is inline with Lemma 1 and can be regarded as the optimism in face of uncertainty, or equivalently, the upper confidence of the predicted hit rate. With the above setting of  $\alpha_t$ , we have  $\delta = [\ln(tF^{\frac{1}{2}})]^{\frac{1}{2}}$  in Lemma 1. Hence, as  $t$  increases, the upper confidence bound holds with high probability (at least  $1 - 2F^{-1}t^{-2}$ ). Based on the upper confidence  $\hat{d}_{f,n,t}$  of predicted hit rate, a set of file  $\mathcal{F}_{n,t}$  that is predicted to maximize the content hit rate at EN  $n$  is cached

**Algorithm 1** Location Differentiated Edge Caching Algorithm**Input:**  $\lambda > 0$ .**Output:** Set of files to be cached in each EN.

---

```

1: Initialization: Cache files in all ENs and get the initial
   feature vectors  $\mathbf{x}_{f,n,0}$  of all file-EN pairs.
2:  $\mathbf{V}_n \leftarrow \lambda \mathbf{I}_d$ ,  $\mathbf{h}_n \leftarrow \mathbf{0}_d$ ,  $\forall n \in \mathcal{N}$ 
3: for  $t = 1, 2, \dots, T$  do
4:   for each EN  $n \in \mathcal{N}$  do
5:      $\tilde{\boldsymbol{\theta}}_{n,t} \leftarrow \mathbf{V}_n^{-1} \mathbf{h}_n$ 
6:     for each file  $f \in \mathcal{F}$  do
7:       Obtain feature vectors  $\mathbf{x}_{f,n,t}$ 
8:        $\tilde{d}_{f,n,t} \leftarrow \mathbf{x}_{f,n,t}^\top \tilde{\boldsymbol{\theta}}_{n,t}$ ,  $\hat{d}_{f,n,t} \leftarrow \tilde{d}_{f,n,t} + p_{f,n,t}$ 
9:     end for
10:     $\mathcal{F}_{n,t} = \arg \max_{\mathcal{F}_n \subseteq \mathcal{F}, |\mathcal{F}_n| \leq c} \sum_{f \in \mathcal{F}_n} \hat{d}_{f,n,t}$ 
11:    Cache all the files in set  $\mathcal{F}_{n,t}$  on EN  $n$ 
12:    Observe the empirical hit rate  $d_{f,n,t}$  of cached files
13:    Update  $\mathbf{V}_n$  and  $\mathbf{h}_n$  based on  $\mathbf{x}_{f,n,t}$  and  $d_{f,n,t}$  of all
       cached files:
        $\mathbf{V}_n \leftarrow \mathbf{V}_n + \mathbf{x}_{f,n,t} \mathbf{x}_{f,n,t}^\top$ 
        $\mathbf{h}_n \leftarrow \mathbf{h}_n + \mathbf{x}_{f,n,t} d_{f,n,t}$ 
14:   end for
15: end for

```

---

respectively. Afterwards, the empirical hit rate information of all cached files is recorded, which is used to update the database for subsequent estimation and prediction. Note that, a file may be simultaneously cached in multiple ENs.

## IV. REGRET ANALYSIS

The long-term hit rate of the proposed algorithm highly depends on the accuracy of prediction. This section gives a theoretical upper bound on the regret of long-term hit rate of the proposed algorithm.

In mobile edge caching, let  $c$  be the caching size of each EN, and  $F$  be the size of ground file set. Suppose content hit rate satisfies the linear model, and feature vectors are bounded by  $\|\mathbf{x}_{f,n,t}\| \leq \eta$  for all  $f \in \mathcal{F}$ ,  $n \in \mathcal{N}$  and  $t \in \mathcal{T}$ , where  $\|\cdot\|$  denotes the Euclidean norm. We have the following theorem.

**Theorem 1.** *Mobile edge caching Algorithm 1 achieves sub-linear long-term regret. Specifically, the long-term regret  $R(T)$  is at most of order  $O(cN \sqrt{dT}(\ln T) \ln(\lambda + T\eta^2/d))$ .*

*Proof:* The total regret depends on the 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

$$\mathcal{I}_{f,n,t} = [\mathbf{x}_{f,n,t}^\top \tilde{\boldsymbol{\theta}}_{n,t} - p_{f,n,t}, \mathbf{x}_{f,n,t}^\top \tilde{\boldsymbol{\theta}}_{n,t} + p_{f,n,t}] \quad (9)$$

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

$$r_{n,t} = \sum_{f \in \mathcal{F}_{n,t}^*} d_{f,n,t} - \sum_{f \in \mathcal{F}_{n,t}} d_{f,n,t}, \quad (10)$$

and the long-term regret can be rewritten as

$$\begin{aligned} R(T) &= \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} r_{n,t} \\ &= \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \mathbb{1}_{\{\mathcal{X}_{n,t}\}} r_{n,t} + \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \mathbb{1}_{\{\bar{\mathcal{X}}_{n,t}\}} r_{n,t}, \end{aligned} \quad (11)$$

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 bound the long-term regret, the two terms in Eq. (11) are bounded respectively.

Firstly, consider the case when event  $\mathcal{X}_{n,t}$  happens. With the setting of  $\alpha_t$  in Eq. (8), 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 in all ENs across the time span can be bounded as:

$$\begin{aligned} \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \mathbb{1}_{\{\mathcal{X}_{n,t}\}} &\leq \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}} \mathbb{P}\{|d_{f,n,t} - \tilde{d}_{f,n,t}| \geq p_{f,n,t}\} \\ &\leq \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}} 2F^{-1}t^{-2} = 2N \sum_{t \in \mathcal{T}} t^{-2} \\ &\leq 2N \sum_{t=1}^{\infty} t^{-2} \leq \frac{\pi^2}{3} N. \end{aligned} \quad (12)$$

Without loss of generality, let the content hit rate  $d_{f,n,t} \leq \gamma$ ,  $\forall f \in \mathcal{F}$ ,  $n \in \mathcal{N}$  and  $t \in \mathcal{T}$ . According to Eq. (10), a coarse upper bound of the instant regret is  $r_{n,t} \leq c\gamma$ . Therefore, the first term of Eq. (11) can be bound as

$$\sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \mathbb{1}_{\{\mathcal{X}_{n,t}\}} r_{n,t} \leq \pi^2 c \gamma N / 3. \quad (13)$$

Then, consider the case when event  $\bar{\mathcal{X}}_{n,t}$  happens, all files' true hit rates falls in to 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  $\hat{d}_{f,n,t} = d_{f,n,t} + p_{f,n,t}$ , we have

$$0 \leq \hat{d}_{f,n,t} - d_{f,n,t} \leq 2p_{f,n,t}. \quad (14)$$

By Eq. (10) and (14), when event  $\bar{\mathcal{X}}_{n,t}$  happens, the instant regret  $r_{n,t}$  can be bounded as

$$\begin{aligned} r_{n,t} | \bar{\mathcal{X}}_{n,t} &= \sum_{f \in \mathcal{F}_{n,t}^* \setminus \mathcal{F}_{n,t}} d_{f,n,t} - \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}_{n,t}^*} d_{f,n,t} \\ &\leq \sum_{f \in \mathcal{F}_{n,t}^* \setminus \mathcal{F}_{n,t}} \hat{d}_{f,n,t} - \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}_{n,t}^*} d_{f,n,t} \\ &\leq \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}_{n,t}^*} (\hat{d}_{f,n,t} - d_{f,n,t}) \end{aligned} \quad (15)$$

$$\leq 2 \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}_{n,t}^*} p_{f,n,t}. \quad (16)$$

where inequality (15) 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  $\sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}_{n,t}^*} \hat{d}_{f,n,t} \geq \sum_{f \in \mathcal{F}_{n,t}^* \setminus \mathcal{F}_{n,t}} \hat{d}_{f,n,t}$ .

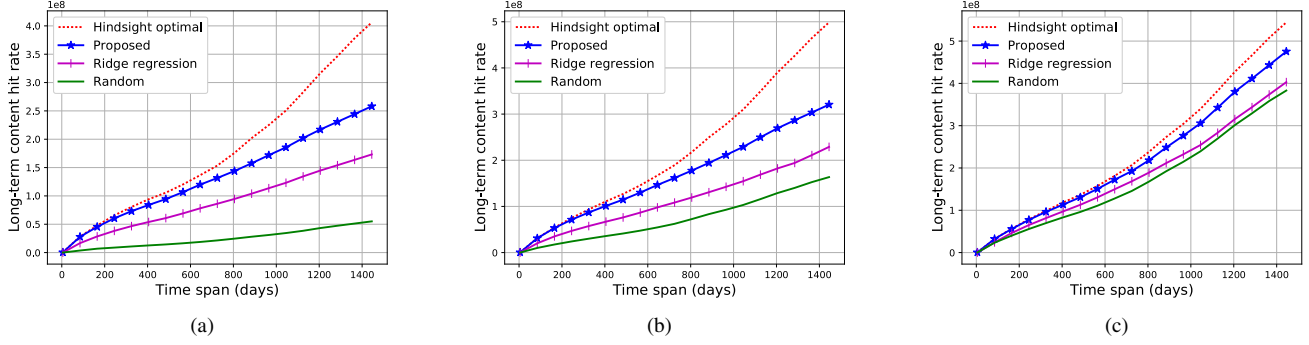


Fig. 3. The content hit rate comparison between the proposed algorithm and other benchmarks with varying caching size, where the total number of videos is 100. (a) EN caching size  $c = 10$ , (b) EN caching size  $c = 30$  and (c) EN caching size  $c = 70$ .

Based on two lemmas from [17] (Lemma 10 and 11), the second term in Eq. (11) can be bounded as

$$\begin{aligned} \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} r_{n,t} |\tilde{\mathcal{X}}_{n,t}| &\leq 2 \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}} \sum_{f \in \mathcal{F}_{n,t} \setminus \mathcal{F}_{n,t}^*} p_{f,n,t} \\ &\leq 2c\alpha_T \sum_{n \in \mathcal{N}} \sum_{t \in \mathcal{T}} \sqrt{\mathbf{x}_{f,n,t}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n,t}} \end{aligned} \quad (17)$$

$$\leq 2c\alpha_T \sum_{n \in \mathcal{N}} \sqrt{T \sum_{t \in \mathcal{T}} \mathbf{x}_{f,n,t}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n,t}} \quad (18)$$

$$\leq 2c\alpha_T N \sqrt{2T \ln \frac{(\lambda + T\eta^2/d)^d}{\lambda}}, \quad (19)$$

where Eq. (17) is due the fact that  $\alpha_t$  increases with  $t$ , Eq. (18) holds because the arithmetic mean of a set of values is smaller than their root-mean square and Eq. (19) is based on the lemmas from [17]. By substituting Eq. (19) and (13) into Eq. (11), and together with  $\alpha_T = \sqrt{\ln(TF^{\frac{1}{2}})} + \zeta\lambda$ , we have

$$\begin{aligned} R(T) &\leq 2c\alpha_T N \sqrt{2T \ln \frac{(\lambda + T\eta^2/d)^d}{\lambda}} + \frac{\pi^2}{3} c\gamma N \\ &= O\left(cN \sqrt{dT(\ln T) \ln(\lambda + T\eta^2/d)}\right), \end{aligned} \quad (20)$$

which concludes the proof. ■

Basically, the regret consists of two parts: estimation error and perturbation. In particular, the estimation error consists of the ridge regression model error and the intended bias incur with  $\lambda > 0$ . The perturbation term is well managed by the varying control parameter  $\alpha_t$ . Theorem 1 indicates that the proposed algorithm achieves sublinear long-term regret, i.e.,  $\frac{R(T)}{T} \rightarrow 0$  when  $T \rightarrow \infty$ , which means that the accumulative content hit rate asymptotically approaches the optimal caching scheme in the long term.

## V. PERFORMANCE EVALUATION

To evaluate the proposed algorithm, we conduct a study on a dataset crawled from YouTube. On YouTube, some video owners made their video view statistics open to public. Among others, the view amount information is recorded on a daily basis. We randomly crawled 100 videos that were uploaded before January of 2013, with full view history till January

of 2017. The most popular video has been watched tens of thousands of times everyday since it was uploaded, while the least popular one has been rarely viewed across the time span.

### A. Simulation Setup

Note that YouTube videos are being watched globally, and we do not have access to the statistics of videos at different locations. To emulate the video view process in different places, we shift the statistics of each video backward and forward on the time span. In this way, we are able to characterize the location-related features based on different view statistics and meanwhile, maintain the temporal feature of each video record. Specifically, we consider the content library containing those 100 videos. Each video can be cached on 3 ENs, each with caching size  $c$ . Content refreshing is performed upon the network status. For example, network traffic presents regular peak and valley every day. Hence, content refreshing can be performed during the off-peak period with minimized impact on the normal network activity. Meanwhile, we use the view amount in the past 5 days as the feature vector, i.e.,  $d = 5$ .

We compare the proposed algorithm with the following benchmarks. 1) Hindsight optimal. Based on the full view record across the time span, the most popular videos are always selected and cached. Note that this benchmark requires future information and cannot be implemented in practice. 2) Ridge regression. As a degraded version of our proposed algorithm, the ridge regression does not account for the random noise of user demand. 3) Random. A random set of videos is selected to update the EN cache at each time slot.

### B. Simulation Result

Figure 3 shows the results of different algorithms in terms of long-term content hit rate with different EN caching sizes. It can be seen that the proposed algorithm outperforms other schemes under all caching schemes. This is because our algorithm chooses to be optimism in face of uncertainty, which helps to better identify the popular contents even under varying popularity profile. As it was found in [14], YouTube video requests are highly skewed, indicating that a small portion of popular contents are attracting the majority of requests. This is confirmed by Fig. 3 since the content hit rate does not

grow linearly with the cache size. The performance gain of our algorithm can be higher if content popularity profile is less skewed, since contents with highly skewed popular profile can also be easily identified by other algorithms. Also note that, hit rate of the optimum stays almost the same from  $c = 30$  to  $c = 70$ . This is due to the fact that content popularity is long-tailed [14], namely, the less popular contents attract almost vanishing requests compared to the popular ones.

## VI. CONCLUSION

This paper proposes an online learning algorithm for dynamic mobile edge caching, by exploiting location related features. The algorithm first estimates the hit rate of a content at a specific location based on a linear model. Noticing that the accuracy may be affected by random noises, a perturbation is added to the estimation to account for uncertainty. Then, according to estimation results, contents that are predicted to maximize hit rate at a certain location are cached respectively. Theoretical analysis indicates that the proposed algorithm achieves sublinear long-term regret when compared to the optimal caching policy. Simulations on real world traces demonstrate the advantage of the proposed algorithm. For future work, we will investigate the impact of location differentiation on coded caching schemes.

## APPENDIX PROOF OF LEMMA 1

Let  $\mathbf{h}_{f,n} = \Phi_{f,n}^\top \mathbf{y}_{f,n}$ , based on Eq. (6), the estimation error can be rewritten as

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

Since  $\|\boldsymbol{\theta}_n^*\| \leq \zeta$ , according to Hölder's inequality,

$$\begin{aligned} |\mathbf{x}_{f,n}^\top \tilde{\boldsymbol{\theta}}_n - \mathbf{x}_{f,n}^\top \boldsymbol{\theta}_n^*| &\leq |\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \Phi_{f,n}^\top (\mathbf{y}_{f,n} - \Phi_{f,n} \boldsymbol{\theta}_n^*)| \\ &\quad + \zeta \lambda \|\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1}\|. \end{aligned} \quad (21)$$

The right-hand side of above inequality decomposes the estimation error into two parts, with the first (variance term) specifies the error caused by linear model, and the second (bias term) is the bias incurred by ridge regression parameter  $\lambda$ . According to Eq. (1), we have  $\mathbb{E}[\mathbf{y}_{f,n} - \Phi_{f,n} \boldsymbol{\theta}_n^*] = 0$ . The Azuma's inequality gives an probabilistic upper bound of the variance term of Eq. (21):

$$\begin{aligned} & \mathbb{P}\left\{|\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \Phi_{f,n}^\top (\mathbf{y}_{f,n} - \Phi_{f,n} \boldsymbol{\theta}_n^*)| > \delta \sqrt{\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n}}\right\} \\ & \leq 2 \exp\left(-\frac{2\delta^2 \mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n}}{\|\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \Phi_{f,n}^\top\|^2}\right) \leq 2e^{-2\delta^2}, \end{aligned} \quad (22)$$

where the last inequality is due to the fact that

$$\begin{aligned} \mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n} &= \mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} (\Phi_{f,n}^\top \Phi_{f,n} + \lambda \mathbf{I}_d) \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n} \\ &\geq \mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \Phi_{f,n}^\top \Phi_{f,n} \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n} \\ &= \|\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \Phi_{f,n}^\top\|^2. \end{aligned} \quad (23)$$

Hence, the variance term of Eq. (21) can be bounded by  $\delta \sqrt{\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n}}$  with probability at least  $1 - 2e^{-2\delta^2}$ . Further, The bias term of Eq. (21) can be bounded as

$$\begin{aligned} \|\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1}\| &= \sqrt{\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \mathbf{I}_d \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n}} \\ &\leq \sqrt{\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} (\lambda \mathbf{I}_d + \Phi_{f,n}^\top \Phi_{f,n}) \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n}} \\ &= \sqrt{\mathbf{x}_{f,n}^\top \mathbf{V}_{f,n}^{-1} \mathbf{x}_{f,n}}. \end{aligned} \quad (24)$$

By substituting Eq. (22) and (24) into Eq. (21), the probabilistic bound in Eq. (7) directly follows.

## ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China under Grant No. 61231010, Research Fund for the Doctoral Program of MOE of China under Grant No. 20120142110015 and the Natural Sciences and Engineering Research Council (NSERC) of Canada. Peng Yang is also financially supported by the China Scholarship Council.

## 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, 2016.
- [4] 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.
- [5] S. Zhang, N. Zhang, P. Yang, and X. Shen, "Cost-Effective Cache Deployment in Mobile Heterogeneous Networks," *IEEE Trans. Veh. Technol.*, DOI: 10.1109/TVT.2017.2724547, to appear.
- [6] 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.
- [7] P. Yang, N. Zhang, Y. Bi, L. Yu, and X. Shen, "Catalyzing Cloud-Fog Interoperation in 5G Wireless Networks: An SDN Approach," *IEEE Netw.*, DOI: 10.1109/MNET.2017.1600078, to appear.
- [8] S. Li, J. Xu, M. van der Schaar, and W. Li, "Popularity-Driven Content Caching," in *Proc. of IEEE INFOCOM*, San Francisco, CA, USA, 2016.
- [9] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Smart Caching in Wireless Small Cell Networks via Contextual Multi-Armed Bandits," in *Proc. of IEEE ICC*, Kuala Lumpur, Malaysia, 2016, pp. 1-7.
- [10] 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, 2015.
- [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, 2013.
- [12] J. Qiao, Y. He, and X. Shen, "Proactive Caching for Mobile Video Streaming in Millimeter Wave 5G Networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 7187-7198, 2016.
- [13] G. Szabo and B. A. Huberman, "Predicting the Popularity of Online Content," *Commun. ACM*, vol. 53, no. 8, pp. 80-88, 2010.
- [14] 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.
- [15] 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, 2016.
- [16] 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.
- [17] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved Algorithms for Linear Stochastic Bandits," in *Proc. of NIPS*, Granada, Spain, 2011, pp. 2312-2320.