Collaborative Edge-Network Content Replication: A Joint User Preference and Mobility Approach

Ge Ma
China Academy of Industrial Internet
Beijing, China
mage@china-aii.com

Qiyang Huang
China Academy of Industrial Internet
Beijing, China
huangqiyang@china-aii.com

Weixi Gu*
China Academy of Industrial Internet
Beijing, China
guweixi@china-aii.com

ABSTRACT

Today's mobile video users have unsatisfactory quality of experience mainly due to the large network distance to the centralized infrastructure. To improve users' quality of experience, content providers are pushing content distribution capacity to the edge-networks. However, existing content replication approaches cannot provide sufficient quality of experience for mobile video delivery. Because they fail to consider the knowledge of user-behavior such as user preference and mobility, which can capture the dynamically changing content popularity. To address the problem, we propose a user-behavior driven collaborative edge-network content replication solution in which user preference and mobility are jointly considered. More specifically, using user-bahavior driven measurement studies of videos and trajectories, we first reveal that both users' intrinsic preferences and mobility patterns play a significant role in edge-network content delivery. Second, based on the measurement insights, it is proposed that a joint user preference- and mobility-based collaborative edge-network content replication solution, namely *APRank*. It is comprised of preference-based demand prediction to predict the requests of video content, mobility-based collaboration to predict the movement of users across edge access points (APs), and workload-based collaboration to enables collaborative replication across adjacent APs. APRank is able to predict the fine-grained content popularity distribution of each AP, handle the trajectory data sparseness problem, and make dynamic and collaborative content replication for edge APs. Finally, through extensive trace-driven experiments, we demonstrate the effectiveness of our design: APRank achieves 20% less content access latency and 32% less workload against traditional approaches.

ACM Reference Format:
Ge Ma, Qiyang Huang, and Weixi Gu. 2020. Collaborative Edge-Network Content Replication: A Joint User Preference and Mobility Approach. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers (UbiComp/ISWC '20 Adjunct), September 12–16, 2020, Virtual Event, Mexico*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3410530.3414593

1 INTRODUCTION

According to Cisco Forecast, mobile video traffic will occupy 78% of the world mobile data traffic by 2021 [1]. Different from traditional PC/laptop-based video streaming, mobile video streaming relies on the usage of mobile devices and wireless networks, allowing people to receive video content on the move. The explosive increase of mobile video streaming is changing the video delivery landscape, and the change has challenged traditional video content delivery, which uses centralized infrastructure (e.g., content delivery network) inside the network backbone for content distribution. To alleviate the load of the network backbone and reduce users' content access latency, Internet service providers (ISPs) and content providers (CPs) are moving content distribution capacity to the edge-networks (e.g., on access points)[2, 3].

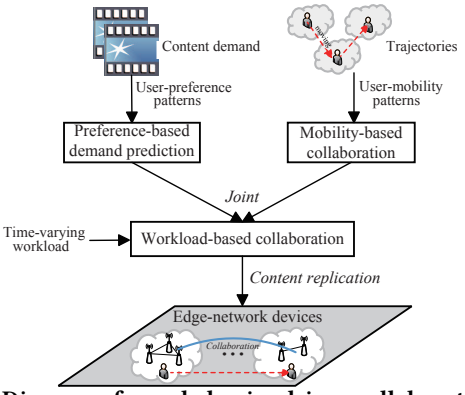


Figure 1: Diagram of user-behavior driven collaborative edge-network content replication.

Edge-network based content delivery platform acts as a crowdsourcing system that offloads content distribution tasks to massive edge devices [4–6]. The uniqueness of content replication in current edge-network lies that both the bandwidth capacity and cache capacity of the edge devices are tightly limited (usually several Mbps and GBs), which are apparently smaller than that of CDN servers. One way to handle the challenge is to introduce collaborative replication in the edge-network, which enables APs to share the cached contents through a backhaul network. When an AP receives a content request, it can fetch the content from adjacent APs that cache the content via the backhual network, instead of fetching it from the CDN servers[7–10]. Therefore, collaboration among APs can reduce the operational cost (e.g., cache and bandwidth cost) of the edge-network [11–13].

There are several limitations making traditional approaches not sufficient for today's mobile video delivery. (1) Traditional approaches cannot reflect the preferences of individuals, which can only be inferred by a collaborative knowledge of requests in different APs. (2) They cannot prefetch content according to user mobility, leading to request missing. For example, a mobile user is generally regarded as non-related users when s/he requests videos at different locations. (3) They cannot reflect the dynamically changing popularity and might cache content that is not needed in the near future.

To address the above challenges, we propose a joint user preference- and mobility-based collaborative edge-network content replication solution, namely *APRank*. More specifically,

▷ First, using large-scale measurement studies of users' preferences of videos and trajectories, we reveal that both users' intrinsic preferences and mobility patterns play a significant role in edge-network content delivery. Our key observations include: (1) user preference (based on the history of requested videos) is relatively

*Corresponding author.

UbiComp/ISWC '20 Adjunct, September 12–16, 2020, Virtual Event, Mexico

ACM ISBN 978-1-4503-8076-8/20/09…$15.00
https://doi.org/10.1145/3410530.3414593

stable over time (Sec. 2.2); (2) users have regular back and forth mobility behaviors patterns, involving 2 – 4 regularly visited APs where they tend to watch videos (Sec. 2.4).

▷ Second, based on the measurement insights, we design user preference and mobility predictive models to capture the popularity distribution of content across different APs. We employ the Markov random fields (MRFs) theory and Gibbs sampling to estimate the probabilities that users watch each video depending on the group-based user preference (Sec. 4). To handle the trajectory data sparseness problem, an iterative algorithm is proposed to predict the movement of users based on the crowd mobility patterns (Sec. 5).

▷ Third, using the previous predictive models, we formulate the collaborative content replication problem as an optimization problem. The objective is to minimize the overall latency and caching cost under limited cache capacity and dynamic workload. We then employ a greedy strategy to practically solve this problem in an online manner (Sec. 6).

We also conduct trace-driven experiments to demonstrate the effectiveness of our design. Compared with traditional content replication approaches, APRank can improve quality of mobile video streaming significantly, e.g., 20% less access latency, 32% less workload of CDN server (Sec. 7).

2 MEASUREMENT

2.1 Datasets

Traces of AP information is provided by a mobile App that asks users to respond to questions on how they use wireless networks. In particular, we have collected over 1 million APs in Beijing city, including the Basic Service Set Identifier (BSSID), timestamp, location and point of interest (PoI) of each AP. Using these traces, we can obtain the geographical distribution of edge APs.

Traces of mobile video sessions is collected by one of the most popular video providers in China. How users watch videos in the mobile video streaming App has been recorded. The dataset was collected in 2 weeks of March 2016, containing 2 million users watching 0.3 million unique videos in Beijing city. In each trace item, the following information is recorded: the user ID, the timestamp and location when and where the user watches the video, the title and duration of the video, etc. Based on these traces, we can study the user preference and mobility.

2.2 User Preference

We study the persistence of user preference of videos, which is generally comprised of multiple kinds of interests, such as TV series, movie and variety show[14]. Hence, users' preferences are not generally binary decisions, e.g., like or dislike of a video, but have a variety of granularities, e.g., "*Movie → Fantasy Movie → Lord of the Rings*". We use a two-level category hierarchy to quantify the preference, shown in Fig. 2. For example, the "*Movie*" category includes the "*Lord of the Rings*" sub-category. All the videos related to "*Lord of the Rings*" are included in the "*Lord of the Rings*" sub-category. For the u-th user, the value a_{uc} of c-th category is the proportion of the number of c-th category requests over the number of total requests. In the analysis, we focus on the users who watch videos at least once a day. For each user, we calculate the average cosine similarity using her/his two-level preference category hierarchy between the first and second week, i.e., $\frac{1}{2}(\boldsymbol{p_1} \cdot \boldsymbol{q_1} + \boldsymbol{p_2} \cdot \boldsymbol{q_2})$, where $\boldsymbol{p_i}$ (resp. $\boldsymbol{q_i}$) is the normalized distribution ($\|\boldsymbol{p_i}\| = \|\boldsymbol{q_i}\| = 1$) of requests of Level i categories in first (resp. second) week. Fig. 3 plots the CDF of similarity coefficient. The result shows that more than 80% users' similarity coefficients are over 0.8, indicating that the preference of users in a consecutive time window (e.g., one week) is relatively stable. *It is possible to design an demand prediction based on the users' preferences.*

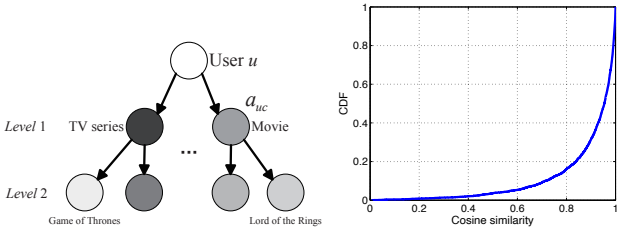


Figure 2: Two-level category hierarchy.

Figure 3: Persistence of user preference.

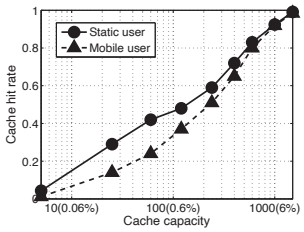


Figure 4: Cache hit rates of different users (the percentage number in brackets is the ratio of cache capacity to the total number of videos).

2.3 Today's Edge Cache Performance

In today's mobile video systems, we measure the cache hit rate for mobile and static users under different cache capacities (we assume the video content size is unit [15, 16], e.g., cache capacity is 10 meaning only 10 videos can be cached in each AP). We let the requests be served by the nearest APs. When a requested video is not cached by an AP, it will be fetched from the distant CDN server . In the measurement, to ensure that each AP has sufficient requests, only the top 10% most popular APs (which receive most requests) are considered. Fig. 4 shows the average cache hit rates using popularity-based replication approach (where videos are prioritized to be replicated or removed according to the videos' historical popularity) under different cache capacities[17]. We observe that the cache hit rate of mobile users is generally lower than that of static users, e.g., the gap between mobile users and static users is about 0.2 with 0.42% cache capacity. So today's edge-network cache performance is not good for mobile video content.

Possible reason 1**:** *Small cache capacity* results in poor caching performance. In Fig. 4, when the cache capacity is small (e.g., less than 100), the gap between static users and mobile users is large, which is up to 0.2. It indicates that collaboration among adjacent APs sharing the cached contents through a backhaul link is a promising solution to improve the cache performance for mobile video content.

Possible reason 2**:** *User mobility* results in poor caching performance. In Fig. 4, when the cache capacity is large (e.g., greater than 100), the gap between static and mobile users also exists. Thus, user mobility probably leads to the gap.

2.4 User Mobility Patterns

Users who watch videos on the move could connect more than a single AP. For the CPs, when performing content replication for the mobile users, they need to strategically determine the APs (that are connected by the same mobile users) where the videos should be replicated. To this end, we investigate the impact and characteristics of user mobility in a mobile video system.

First, we measure the mobility intensity of mobile users. In the analysis, we focus on the behaviors of *active users* who requested at least ten videos daily in our 2-week traces, and record the average result of each day. In Fig. 5, the solid curve plots the relation between

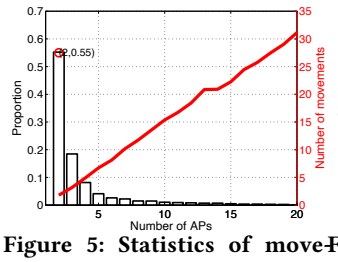


Figure 5: Statistics of movement of mobile users.

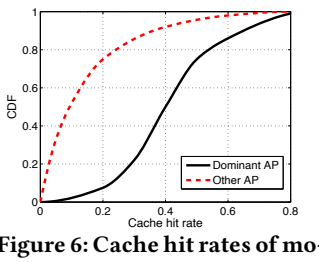


Figure 6: Cache hit rates of mobile users in different APs.

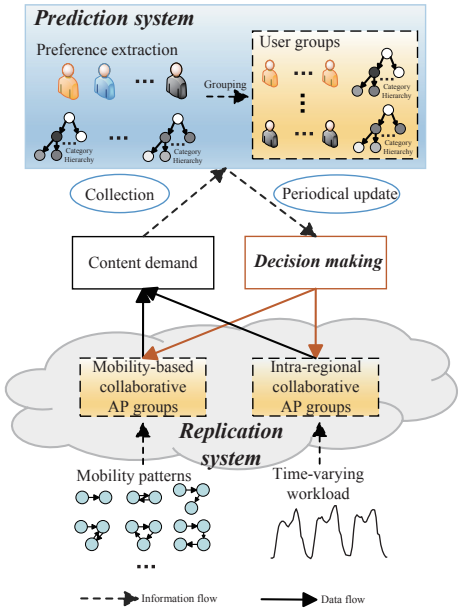


Figure 7: Framework of APRank.

the number of movements and different connected APs in one day. We observe that mobile users issue requests frequently in different APs, but the number of APs (per user) is quite limited. The histogram represents the proportion of mobile users versus the corresponding number of APs. We observe that 55% of users only issue video requests from 2 APs, and 80% of users request videos from less than 4 APs. These results show that mobile users have regular back and forth behaviors, involving $2-4$ regular visited APs where they tend to watch videos.

Next, we investigate why user mobility results in poor caching performance. We focus on the mobile users who access only two APs within one day (occupying 55% of the total mobile users, shown in Fig. 5). We define that a user's *dominant* AP is the AP which receive the most requests of the user. Fig. 6 shows the CDF of cache hit rates of mobile users in their dominant APs and other AP (under cache capacity = 60 and popularity-based replication). It indicates that mobile users generally have volatile quality of experience across different APs.

In summary, *since user mobility affects the performance of content distribution in the edge-network greatly, it is promising to design an mobility-based collaboration to improve the content distribution for mobile users.*

3 APRANK: USER PREFERENCE AND MOBILITY-BASED AP COLLABORATION FRAMEWORK

Based on the measurement insights in mobile video streaming, we design the key components of APRank. APRank jointly utilizes the user-bahavior patterns for content replication based on the regional information (a region is comprised of adjacent APs) in the edge-network, which is illustrated in Fig. 7.

In this figure, the following *three-stage* workflow of APRank is presented.

Stage I: Demand prediction is to predict the possible requests for expected future time in each AP, which provides the main basis of subsequent replication strategy. **User grouping** runs offline periodically in a global view (the view of CPs) based on the two-level category hierarchy of preference. In each time window t (in our experiments, $t = 1h$), content demand prediction of each group runs online based on the history of demand. The output of this stage is an *initialized* local future demand in each AP.

Stage II: Mobility-based collaboration is to predict the movement of users across different APs based on the user mobility patterns. A crowd mobility-based user movement algorithm runs online in a local view (each AP) based on the most recent user movement information and future demand predicted in the first stage. This outputs *final* local future demand by aggregating user mobility with the corresponding local requests from different APs.

Stage III: Intra-regional collaboration is a promising solution to reduce the content access latency and alleviate the workload of network backbone, when the edge APs are equipped with toughly limited cache and bandwidth capacities. **Region partitioning** runs offline periodically in a global view based on the latest location information of APs. In each time window t, a simple and fast content replication algorithm can be computing in parallel across different regions, which does not require the knowledge of global content popularities. It runs online and updates the caches of adjacent APs in each region based on their time-varying workloads and future demands from the previous stage.

In the following sections, we will present the detailed design of key algorithmic pieces.

4 PREFERENCE-BASED DEMAND PREDICTION

Researchers have shown that users with similar preferences are more likely to exhibit the same behaviors in video service, social network and recommendation system [18–21]. So we group users with similar preferences. This enables us to decompose the global process of prediction into separate per-group processes, which reduces the prediction delay.

4.1 User Grouping

First, we quantify user preference using a two-level category hierarchy (Sec. 2.2). We project each user's request history within a period of time (e.g., one week) onto a predefined category hierarchy, where each node is associated with a value a_{uc} representing the proportion of the number of requests of u-th user to the c-th category over the number of total requests. Second, we calculate the TF-IDF value of each node in the hierarchy, where a user's request history is regarded as a document and categories are considered as terms in the document. Intuitively, a user would watch more videos belonging to a category if the user prefers it. Further, if a user watches videos of a category that is rarely watched by other users, the user will like this category more prominently. In particular, a user's preference weight $\mathbf{w}_u = \{w_{uc} : \forall c \in C\}$ is calculated by equation (1), where the first part is the TF value of c-th category in the u-th user's request history and the second part it the IDF vaule of the category.

$$w_{uc} = a_{uc} \times \lg \frac{|\mathcal{U}|}{|\mathcal{U}_c|}, \tag{1}$$

where $\mathcal{U}_c$ is the set of users who have watched the c-th category among all the users $\mathcal{U}$. Next, based on the preference weights of their category hierarchy, we classify users into different groups using a well-known clustering algorithm *K-means* (using Euclidean distance between the coordinates of APs).

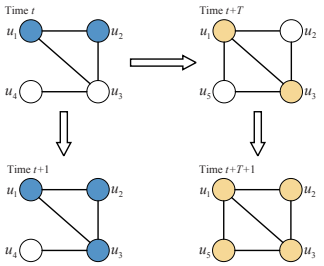


Figure 8: An example (circles represent users, circles connected by an edge represent they are neighbors, and different colors represent different videos have been watch by a user).

4.2 Local Demand Prediction

Based on the previous observation that user preference is relatively stable in a period of time, it is possible to predict which videos a user will watch at time $t+1$ from the states of her/his corresponding group at time t. Each user group g could be regarded as a user-user similarity network, where similar users (min-max normalized cosine similarity coefficient is greater than 0.5) are neighbors of each other. Illustrated in Fig. 8, users u_1, u_2, u_3, u_4 are classified into the same group and users u_1, u_2 have watched the blue video at time t. So the user u_3 will watch the blue video at time $t+1$. It also shows that the network topology of each group may be changed on timescales of weeks (i.e., T). For example, u_1, u_2, u_3, u_5 are classified into the same group and the edges are different at time $t+T$.

We employ the theory of Markov random fields (MRFs) [22, 23] to associate each user with a confidence probability of watching a video. In the MRFs, the problems are how to assign different weights to the parameters and how to estimate the probabilities based on the network.

Let $y_{uv}=1$ (resp. $y'_{uv}=1$) if the u-th user has watched the v-th video at time t (resp. $t+1$) and 0 otherwise (we use $'$ to represent the time $t+1$ unless otherwise noted). For each user, we define his neighbors, $\mathcal{U}_u$, as the set of users similar to u-th user. Using the theory of MRFs, the probability of video labelling is proportional to $e^{-F(y'_{uv})}$.

$$\begin{aligned} &F(y'_{uv}) = -\alpha N_1 - \beta N_{10} - \gamma N_{11} - N_{00}, \\ &N_1 = \sum_{j\in\mathcal{U}_g} y_{jv},\ N_{10} = \sum_{j\in\mathcal{U}_u} (1-y'_{uv})y_{jv} + (1-y_{jv})y'_{uv}, \\ &N_{11} = \sum_{j\in\mathcal{U}_u} y'_{uv}y_{jv},\ N_{00} = \sum_{j\in\mathcal{U}_u} (1-y'_{uv})(1-y_{jv}), \end{aligned} \tag{2}$$

where $\mathcal{U}_g$ is the set of users in group g. In the terminology of MRFs, $F(y)$ is referred as the *potential function*. This function defines a group-based Gibbs distribution of the network, the probability $P(y'_{uv}|\boldsymbol{\theta}) = \frac{1}{Z_v(\boldsymbol{\theta})} e^{-F(y'_{uv})}$, where $\boldsymbol{\theta}=(\alpha,\beta,\gamma)$ are parameters and $Z_v(\boldsymbol{\theta})$ as the partition function in the theory of MRFs is a normalized constant that is calculated by summing over all the configurations, i.e., $Z_v(\boldsymbol{\theta}) = \sum_u e^{-F(y_{uv})}$. To calculate the probability, we use a Gibbs sampler:

$$\begin{aligned} P(y'_{uv}=1|\mathbf{y}_{[-u]v},\boldsymbol{\theta}) &= \frac{P(y'_{uv}=1,\mathbf{y}_{[-u]v}|\boldsymbol{\theta})}{\sum_{k=0}^{1} P(y'_{uv}=k,\mathbf{y}_{[-u]v}|\boldsymbol{\theta})} \\ &= \frac{e^{\alpha+(\beta-1)M_u^0+(\gamma-\beta)M_u^1}}{1+e^{\alpha+(\beta-1)M_u^0+(\gamma-\beta)M_u^1}} \end{aligned} \tag{3}$$

where $\mathbf{y}_{[-u]v} = (y_{1v},\cdots,y_{u-1,v},y_{u+1,v},...,y^t_{|\mathcal{U}_g|v})$, M_u^0 and M_u^1 are the numbers of neighbors of u-th user, labelled with 0 and 1, respectively. It is difficult to use the maximum likelihood estimation method directly to estimate the parameters $\boldsymbol{\theta}=(\alpha,\beta,\gamma)$, because the partition function $Z_v(\boldsymbol{\theta})$ is also a function of the parameters. Thus, we use the quasi-likelihood estimation method based on a basic logistic regression model,

$$\log\frac{P(y'_{uv}=1|\mathbf{y}_{[-u]v},\boldsymbol{\theta})}{1-P(y'_{uv}=1|\mathbf{y}_{[-u]v},\boldsymbol{\theta})} = \alpha+(\beta-1)M_u^0+(\gamma-\beta)M_u^1. \tag{4}$$

Overall, the procedure of our preference-based demand prediction is illustrated in Algorithm 1 (line 1–line 14). In particular, we quantify the user preference (line 1) and construct the user-user similarity networks (line 2–line3). We initialize the parameters (line 5) and update them using a quasi-likelihood estimation method based on a linear logistic model (line 6). Then, in each time windows, we utilize the Gibbs sampling to iteratively obtain the stabilized posterior probabilities (line 7–line 11). Finally, we calculate the local demand of each AP (line 12).

5 MOBILITY-BASED COLLABORATION

The reason we propose an mobility-based collaboration in a mobile video system lies on two fronts. (1) More and more users are likely to watch videos on the move across different APs, resulting in multiple APs are accessed by the same users. These users generally have poor experiences (Sec. 2.3). (2) Users have regular back and forth behaviors for a considerable time, involving $2-4$ regular visited APs where they tend to watch videos (Sec. 2.4). Based on these insights, we design a mobility-based collaboration to predict the movement of users across different APs.

Instead of identifying and modeling personal mobility pattern [20, 24–28], we propose to capture crowd mobility patterns eliminating the uncertainty and randomness of personal mobility to predict the movement of users. For example, there are 100 users in AP l_1 and several users (may be 10 or 20) will move from l_1 to l_2. We want to predict the proportion of users (e.g., 10% or 20%), rather than the exact users (e.g., $u_1, u_3, \cdots$) moving from l_1 to l_2. Under the assumption that the movements of crowd are generally consecutive, not instantaneous (i.e., if there are movements between two APs at time t, it is more likely to exist movements at time $t+1$), we develop a reactive-sensing method (e.g., AP l_2 can sense the number of users coming from AP l_1 at time t and there will be similar proportion of users moving from l_1 to l_2 at time $t+1$). Compared with common proactive-predicting method (e.g., AP can exactly predict the next AP where a user will go to based on extensive historical trajectory data of each user [24, 25, 29]), our reactive-sensing method well captures crowd mobility patterns, having the following advantages: (1) increase the accuracy, and (2) respond to users with less trajectory data.

We consider a general network architecture where a set $\mathcal{L}$ of L edge APs provide video content access to their users. Let $\mathbf{o}_l = (o_{1l},\cdots,o_{Ll})$ denotes the number of users from other APs to l-th AP at time t. The real demand distribution of each AP is given, $\mathbf{d}_l = (d_{l1},\cdots,d_{lv},\cdots,d_{lV})$, where d_{lv} is the request proportion of video v in l-th AP at time t. Our goal is to estimate the change of popularity in each AP due to the user mobility. Without loss of generality, given a specific l-th AP, we want to obtain the future demand distribution $\mathbf{d}'_l$ at time $t+1$. The problem $\mathbf{d}'_l$ can be solved using a crowd mobility-based user movement algorithm derived from PageRank,

$$\mathbf{d}'_l = \mathbf{d}_{l,k} = \frac{\sum_{i\in\mathcal{L}}\sum_{v\in\mathcal{V}} o_{il}d_{iv,k-1}}{\sum_{i\in\mathcal{L}} o_{il}}, \tag{5}$$

where k is the number of iterations.

The details are presented in Algorithm 1 (line 14–line 23). It adopts the value iteration technique, which extends the PageRank and uses L_2 norm to estimate errors. At every iteration, the future demand distribution $\mathbf{d}'_l$ is updated based on the user movement and

previous iteration result (line 18). This process continues until $\mathbf{d}_{l,k}$ begins to converge.

Algorithm 1: Demand Prediction

Input: the set of videos $\mathcal{V}$; category hierarchy $\mathbf{a}_u$ of u-th user; the set of users $\mathcal{U}$; the set of users $\mathcal{U}_c$ of c-th category; video label $\mathbf{y}_u$ of u-th user; local demand $\mathbf{d}_l$ and user movement $\mathbf{o}_l$ of l-th AP.

Output: local demand $\mathbf{d}'_l$ at time $t+1$.

1 calculate the preference weight $\mathbf{w}_u = \{w_{uc} : \forall c \in C\}$ with equation (1)
2 classify users into groups $\mathcal{G}$ with K-means algorithm
3 construct the user-user similarity networks $\mathcal{G}_s$
4 **for** $v \in \mathcal{V}$ **do**
5 initialize the parameters $\boldsymbol{\theta}$, n_1 (e.g., $\mathbf{0}$, 0)
6 estimate $\boldsymbol{\theta}$ using quasi-likelihood approach with equation (4)
7 **for** $m = 1, \cdots, M$ **do**
8 update the value of y'_{uv} with equation (3)
9 $n_1 = n_1 + y'_{uv}$
10 **end**
11 $y'_{uv} = \frac{n_1}{M}$
12 $d_{lv} = \sum_{u \in \mathcal{U}_l} y'_{uv} + d_{lv}$
13 **end**
14 initialize $\mathbf{d}_{l,0} = (d_{lv} : v \in \mathcal{V})$, $\delta = 0$, $k = 1$
15 **while** $\delta \geq \varepsilon$ and $k++$ **do**
16 $\delta = 0$
17 **for all** $l \in \mathcal{L}$ **do**
18 update the value of $\mathbf{d}_{l,k}$ with equation (5)
19 $\delta = \delta + \left\|\mathbf{d}_{l,k} - \mathbf{d}_{l,k-1}\right\|_2$
20 **end**
21 **end**
22 $\mathbf{d}'_l = \mathbf{d}_{l,k}$
23 **return** $\mathbf{d}'_l$

6 INTRA-REGIONAL COLLABORATIVE CONTENT REPLICATION

We enable APs to share the cached contents through a backhaul network. If the requested content is not in the cache of the local AP, it can retrieve the requested content from the caches of adjacent APs instead of from the CDN servers.

6.1 Region Partitioning

Recent works have shown that an optimal local content replication strategy (i.e., content tailored to each local cache) outperforms an optimal global replication strategy [30]. These results illustrate that there is room for improvement from a purely global to local content replication. However, presently there is no universal standard of region partitioning. In general, a city can be partitioned into individual regions based on road network or density information. Thus, we employ a simple and fast clustering algorithm [31] to partition the APs into regions $\mathcal{R}$ based on their longitude and latitude information. Under two assumptions that region centers are characterized by a higher density of APs than their surroundings and by a relatively large distance from other APs with higher densities, this algorithm recognize the regions regardless of their shape and of the dimensionality of the space. Note that other clustering algorithms (e.g., K-means and DBSCAN) also can be used in the procedure of region partitioning.

6.2 Problem Formulation

We formulate the collaborative replication problem with the consideration of minimizing the total content replication and content access latency cost of all edge APs.

Replication cost: In edge-networks, an abundant of APs are scheduled to replicate content, which incurs influential pressure for the CDN server. Thus, we adopt replication cost based on the workload of CDN server. We follow the work in [5, 32] and the replication cost per unit data s_{L+1} can be calculated as follows,

$$s_{L+1} = -\mu \log(1 - \frac{\sigma_{L+1}}{\sigma_{L+1,th}}),$$

where σ_{L+1} is the current server workload and $\sigma_{L+1,th}$ is the server threshold load, and μ is the tuning parameter to guarantee the replication cost is consistent with latency cost. Thus, the rationale behind s_{L+1} is that it is cheap to replicate content when the CDN server is under small utilization.

Latency cost: The average latency per unit data b_l that the l-th AP serves own requests through itself is comprised of a fixed latency and a volatile latency depending on the current workload [5], which can be calculated as below,

$$b_l = b^1 \frac{\sigma_l}{\sigma_{l,th}} + b^0,$$

where σ_l is the current workload, $\sigma_{l,th}$ is the threshold load of l-th AP, b^1 and b^0 are constant variables. Furthermore, the latency b_{lj} that the l-th AP serves own requests through the j-th AP can be computed using the minimum cost path between l-th and j-th AP, and thus satisfies the triangle inequality $b_{lj} \geq b_l + b_j$. In our problem, we simplify $b_{lj} = b_l + b_j$ and $b_{ll} = b_l$. Similarly, the latency of CDN server is that $b_{L+1} = b^1 \frac{\sigma_{L+1}}{\sigma_{L+1,th}} + b^0$.

We assume that the APs across different regions act independently replication strategies. Thus, our problem can be decomposed into $|\mathcal{R}|$ independent sub-problems that minimizes the total cost of each region. The sub-problem can be formulated as the following optimization function:

$$\begin{aligned} \min_{\{\mathbf{x}'_{\mathcal{N}_r}\}} \quad \mathrm{J}(\mathbf{x}'_{\mathcal{N}_r}) = & \sum_{l \in \mathcal{N}_r} \sum_{v \in \mathcal{V}} \sum_{\substack{j \in \mathcal{N}_r \\ \cup \{L+1\}}} d'_{lv} b'_{lj} \lambda'_{lvj} \\ & + \sum_{l \in \mathcal{N}_r} \sum_{v \in \mathcal{V}} x'_{lv}(1 - x_{lv}) s'_{L+1}, \end{aligned} \tag{U}$$

subject to

$$\sum_{\substack{j \in \mathcal{N}_r \\ \cup \{L+1\}}} \lambda'_{lvj} = 1, \quad \forall l, v, \tag{6}$$

$$\lambda'_{lvj} \leq x'_{jv}, \quad j \in \mathcal{N}_r, \forall l, v, \tag{7}$$

$$\sum_{v \in \mathcal{V}} x'_{lv} \leq H_l, \quad \forall l, \tag{8}$$

where λ'_{lvj} indicates whether the users in l-th AP will download v-th content from the j-th AP ($L+1$-th AP is the CDN server) at time $t+1$ ($\lambda'_{lvj} = 1$) or not ($\lambda'_{lvj} = 0$) and $\mathcal{N}_r$ is the set of APs in r-th region, H_l is the cache capacity of l-th AP. The indicator variable x'_{lv} indicates whether v-th content will be cached at time $t+1$ ($x'_{lv} = 1$) or not ($x'_{lv} = 0$). Then the *replication strategy* of l-th AP at time $t+1$ is given by the vector $\boldsymbol{x}'_l = (x'_{l1}, \cdots, x'_{lV})$. In our objective function, the first term is the total latency cost and the second term is the total replication cost. Equation (6) ensures that each request must be served by an AP or the CDN server. Equation (7) ensures that if a request of v-th content is redirected to the j-th AP, the v-th content must be cached in the j-th AP. Equation (8) is needed to satisfy the limitation of cache capacity.

Clearly, our problem is very hard to solve optimally. [15] proves that the joint user redirection and content placement problem which

only minimizes the content access latency is the NP-hardness problem – Helper Decision Problem. It is easy to see that our problem ignoring the replication cost is equivalent to solving the problem in [15]. Thus, our problem J is also NP-hard.

To guarantee a real-time replication strategy, we employ a greedy algorithm working on each region, presented in Algorithm 2. Given the r-th region, the greedy algorithm starts with an empty cache of each AP in r-th region and all the requests are served by the CDN server (line 3); at each iteration, it adds v-th content with the highest marginal value $\{\arg\max_{\forall l,v\ s.t.\ x'_{lv}=0} \Delta \mathrm{J} = \mathrm{J}(x'_{lv}=0) - \mathrm{J}(x'_{lv}=1)\}$ to the corresponding cache of l-th AP ($\mathrm{J}(x'_{lv}=1)$ means we only change the value of x'_{lv} and the other values of $\mathbf{x}'_{N_r}$ are fixed) and then the requests of v-th content can be fetched from l-th AP (line 4 – line 9). Hence, when the highest marginal value $\Delta \mathrm{J} \leq 0$ or the caches of APs are full at one iteration, the algorithm should stop.

Algorithm 2: Replication Strategy

Input: the set of videos $\mathcal{V}$; local demand $\mathbf{d}'_l$; workload σ_l; replication strategy $\mathbf{x}_l$; latency cost b_l and replication cost s_{L+1} of l-th AP.

Output: local replication strategy $\mathbf{x}'_l$ at time $t+1$.

1 cluster the APs into clusters with the algorithm [31]
2 predict the workload σ' of each AP with the SARIMA model
3 initialize $\mathbf{x}'_l = (x'_{lv} = 0 : \forall v), \forall l$
4 **while** $\Delta \mathrm{J} > 0$ and $\sum_l \sum_v x'_{lv} \leq \sum_l H_l$ **do**
5 $\quad (l^*, v^*) = \arg\max_{\forall l,v\ s.t.\ x'_{lv}=0} \{\mathrm{J}(x'_{lv}=0) - \mathrm{J}(x'_{lv}=1)\}$
6 $\quad \Delta \mathrm{J} = \mathrm{J}(x'_{l^*v^*}=0) - \mathrm{J}(x'_{l^*v^*}=1)$
7 $\quad$ **if** $\Delta \mathrm{J} > 0$ and $\sum_v x'_{lv} \leq H_{l^*}$ **then**
8 $\quad\quad x'_{l^*v^*} = 1$
9 $\quad$ **end**
10 **end**
11 **return** $\mathbf{x}'_l$

7 EVALUATION

In this section, we conduct trace-driven experiments to validate the effectiveness of APRank. The results show that (1) the prediction models in APRank have a relatively high precision (Sec. 7.2); (2) APRank can improve quality of mobile video streaming significantly, e.g., 20% less access latency, 32% less workload of CDN servers (Sec. 7.4) with more than 1000 APs and over 1 million video requests.

7.1 Experiment Setup

In our experiments, to ensure each user has enough requests, only the top 10% users with most requests are considered, including a rich collection of 20,047 users, 1,534,966 trace items and 85,063 videos. We use the traces of video sessions in the first week to train our prediction model and the traces in the second week to test our replication approach unless otherwise noted. According to the *traces of AP information*, we redirect these users' requests to the top 0.1% popular APs (about 1000 APs) under the assumption that each request is served by the nearest AP. After the procedure of user grouping in Sec. 4.1, all the users are classified into 100 groups, where the number of users is varying from 81 to 454. Using the method in Sec. 6.1, we divide the Beijing city into 63 regions where the number of APs is varying from 2 to 130. According to [5, 33], the constant variables b^1 and b^0 in latency cost are 0.1 and 0.05 (resp. 0.5 and 0.15) for APs and the CDN servers, respectively. The tuning parameter μ in replication cost is 5. We assume that the content size is unit [15, 16] and all the APs have the same cache capacity, expressed as the percentage of entire video set size, e.g., cache capacity equalling to 0.6% means that each cache can store 500 videos ($85063 \times 0.6\% \approx 500$). The time interval between time t and $t+1$ is 1 h.

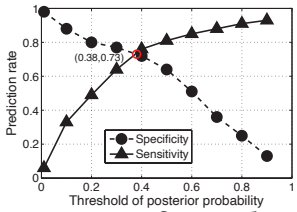


Figure 9: Specificity and sensitivity.

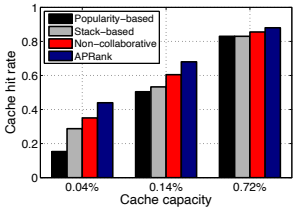


Figure 10: Cache hit ratio versus the cache capacity.

We mainly compare APRank to the following traditional content replication approaches. (1) Popularity-based replication derived from the least frequently used (LFU) replacement algorithm. It replicates (resp. removes) the videos with the highest (resp. lowest) popularities in the recent period. (2) Stack-based replication derived from the least recently used (LRU) replacement algorithm. It replicates (resp. removes) the most (resp. least) recently watched videos in the recent period. (3) Non-collaborative APRank. In contrast to APRank, it replicates videos without the intra-regional collaborative content replication in Sec. 6, i.e., if the requested content is not in the cache of the local AP, users can only retrieve the requested content from the CDN server.

In order to quantify the performance of different content replication approaches, we adopt the following metrics: (1) *cache hit rate*, the proportion of the number of requests served by local cache over the total number of requests; (2) *average content access latency*, the average latency of all the requests formulated in Sec. 6.2; (3) *CDN server load*, the proportion of the number of requests served by CDN server (including local cache missed requests and replicated requests) over the total number of requests.

7.2 Prediction Precision

In this part, we evaluate the effectiveness of our prediction models (Sec. 4). For the demand prediction model, we predict whether a user will watch each video in future and calculate the specificity (true positive rate) and sensitivity (true negative rate). We repeat the experiments for the top 10000 videos in different groups. Fig. 9 shows the relationship between the specificity and sensitivity of our predictor using different thresholds for posterior probabilities. With the threshold equalling to 0.38, the corresponding specificity and sensitivity are the same and equal to 0.73.

7.3 Efficiency of APRank

We first explore the impact of collaboration among the APs on the cache hit rate. Fig. 10 shows the cache hit rates of different content replication approaches under different cache capacity. As expected, increasing the cache capacity can increase the cache hit rate for all approaches. Our observations are as follows. (1) The performance of APRank is always better than that of non-collaborative APRank, which indicates the efficiency of intra-regional collaboration. (2) The non-collaborative APRank consistently outperforms Popularity-based replication and Stack-based replication, which validates the effectiveness of our predictive models based on user preference and mobility. (3) APRank performs better than its counterparts for all the cache capacity values. However, the gap between APRank and other approaches decreases as cache capacity increases, e.g., the gap between APRank and Popularity-based replication (resp. non-collaborative APRank) is 0.28 (resp. 0.09) with 0.04% cache capacity, while the gap is 0.05 (resp. 0.03) with 0.72% cache capacity. It indicates that APRank could significantly improve the cache hit rate with small cache.

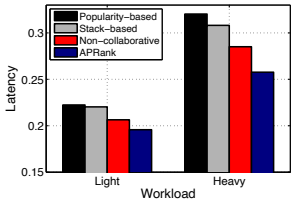


Figure 11: Impact of workload.

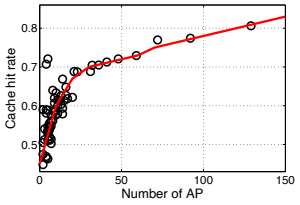
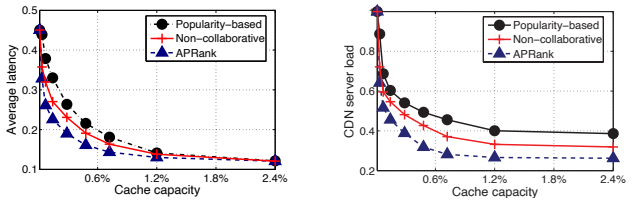


Figure 12: Impact of AP number.



(a) Latency. (b) CDN server load.

Figure 13: Impact of cache capacity.

7.4 Overall Performance: Parameter Impact Analysis

Impact of workload. We analyze the impact of AP workload on average content access latency with 0.14% cache capacity. In the experiment in Fig. 11, we select the top (resp. end) 10% most popular APs as the *heavy* (resp. *light*) workload APs. Compared with Popularity-based replication, APRank can reduce the average latency by 19.5% (resp. 12%) with heavy (resp. light) workload. It indicates APRank can achieve a larger improvement when the workload is heavy.

Impact of AP number. We study the impact of the number of APs in a region on cache hit rate. In the experiment in Fig. 12, there are 63 regions where the number of APs is varying from 2 to 130, and we calculate the cache hit rate of each region with 0.14% cache capacity. As expected, increasing the number of APs can improve the cache hit rate, as more APs collaborate with each other in a region. Moreover, with the number of APs increasing, the performance improvement slows down when the number is already greater than a certain value (e.g., 30 in our experiment).

Impact of cache capacity. Finally, we show the impact of cache capacity on content access latency and CDN server load in Fig. 13. Increasing the cache capacity reduces the average latency and CDN server load for all the approaches (Stack-based replication has a similar performance with Popularity-based replication). APRank can achieve a lower average latency with up to 20% (resp. 12%) reduction and save about 32% (resp. 18%) of CDN server load, compared with Popularity-based replication (resp. non-collaborative APRank). We also observe that increasing cache capacity cannot continuously reduce the average latency and alleviate the CDN server load when the cache is already large (e.g., greater than 1.2%).

8 CONCLUSION

This paper addresses the challenges in the replication of mobile video contents, resulting from the difference of users' preferences, mobility patterns, and diverse workloads in the edge-network. In this paper, we propose the user-behavior driven collaborative edge-network content replication, in which user prference and mobility are jointly considered. First, using large-scale measurement studies, we reveal that both users' intrinsic preferences and mobility patterns play a significant role in edge-network content delivery. Second, we propose APRank, a joint user preference- and mobility-based collaborative edge-network content replication solution. APRank is comprised of preference-based demand prediction using a group-based MRFs method, mobility-based collaboration using a crowd-based iterative method, and workload-based collaboration taking both content access latency and replication cost into consideration. Finally, the extensive trace-driven experiments demonstrate the effectiveness and superiority of our design, which provides high cache hit rate and low access latency for mobile video users, under dramatically changing content popularity and diverse AP workloads.

REFERENCES

[1] Cisco Visual Networking Index, "Global mobile data traffic forecast update, 2016-2021," *San Jose, USA: Cisco White paper*, 2017.

[2] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Network*, vol. 32, no. 1, pp. 80–86, 2018.

[3] G. Ma, Z. Wang, J. Ye, and W. Zhu, "Wireless caching in large-scale edge access points: A local distributed approach," in *MobiCom*. ACM, 2018, pp. 726–728.

[4] M. Ma, Z. Wang, K. Yi, J. Liu, and L. Sun, "Joint request balancing and content aggregation in crowdsourced cdn," in *ICDCS*. IEEE, 2017.

[5] W. Hu, Y. Jin, Y. Wen, Z. Wang, and L. Sun, "Towards wi-fi ap-assisted content prefetching for on-demand tv series: A learning-based approach," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[6] L. Chen, Y. Zhou, M. Jing, and R. TB Ma, "Thunder crystal: a novel crowdsourcing-based content distribution platform," in *NOSSDAV*. ACM, 2015, pp. 43–48.

[7] G. Ma, Z. Wang, M. Chen, and W. Zhu, "Aprank: Joint mobility and preference-based mobile video prefetching," in *ICME*. IEEE, 2017, pp. 7–12.

[8] G. Ma, Z. Chen, J. Cao, Z. Guo, Y. Jiang, and X. Guo, "A tentative comparison on cdn and ndn," in *2014 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2014, pp. 2893–2898.

[9] C. Koch, N. Bui, J. Rückert, G. Fioravantti, F. Michelinakis, S. Wilk, J. Widmer, and D. Hausheer, "Media download optimization through prefetching and resource allocation in mobile networks," in *MMSys*. ACM, 2015, pp. 85–88.

[10] G. Ma and Z. Chen, "Comparative study on ccn and cdn," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2014, pp. 169–170.

[11] H. Rahul, S. Kumar, and D. Katabi, "Jmb: Scaling wireless capacity with user demands," *Communications of the ACM*, vol. 57, no. 7, pp. 97–106, 2014.

[12] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1444–1462, 2014.

[13] A. Gharaibeh, A. Khreishah, B. Ji, and M. Ayyash, "A provably efficient online collaborative caching algorithm for multicell-coordinated systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1863–1876, 2016.

[14] S. Huang, Z. Wang, L. Cui, Y. Jiang, and R. Gao, "Fine-grained fitting experience prediction: A 3d-slicing attention approach," in *MM*. ACM, 2019, pp. 953–961.

[15] N. Golrezaei, K. Shanmugam, A. G Dimakis, A. F Molisch, and G. Caire, "Femto-caching: Wireless video content delivery through distributed caching helpers," in *INFOCOM*. IEEE, 2012.

[16] J. Hachem, N. Karamchandani, and S. Diggavi, "Content caching and delivery over heterogeneous wireless networks," in *INFOCOM*. IEEE, 2015, pp. 756–764.

[17] G. Ma, Z. Wang, M. Zhang, J. Ye, M.a Chen, and W. Zhu, "Understanding performance of edge content caching for mobile video streaming," *IEEE Journal on Selected Areas in Communications*, 2017.

[18] C. A Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems*, vol. 6, no. 4, pp. 13, 2016.

[19] Z. Wang, W. Zhu, M. Chen, L. Sun, and S. Yang, "CPCDN: Content delivery powered by context and user intelligence," *IEEE Transactions on Multimedia*, vol. 17, no. 1, pp. 92–103, 2015.

[20] W. Gu, M. Jin, Z. Zhou, C. J Spanos, and L. Zhang, "Metroeye: Smart tracking your metro trips underground.," in *MobiQuitous*, 2016, pp. 84–93.

[21] M. He, W. Gu, and Y. Kong, "Group recommendation: by mining users' check-in behaviors," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 2017, pp. 65–68.

[22] W. Gu, Z. Yang, L. Shangguan, W. Sun, K. Jin, and Y. Liu, "Intelligent sleep stage mining service with smartphones," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 649–660.

[23] W. Gu, Z. Yang, L. Shangguan, X. Ji, and Y. Zhao, "Toauth: Towards automatic near field authentication for smartphones," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*. IEEE, 2014, pp. 229–236.

[24] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *ICC*. IEEE, 2015, pp. 3358–3363.

[25] K. Poularakis and L. Tassiulas, "Exploiting user mobility for wireless content delivery," in *ISIT*. IEEE, 2013, pp. 1017–1021.

[26] W. Gu, Y. Liu, Y. Zhou, Z. Zhou, C. J Spanos, and L. Zhang, "Bikesafe: bicycle behavior monitoring via smartphones," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 2017, pp. 45–48.

[27] W. Gu, K. Zhang, Z. Zhou, M. Jin, Y. Zhou, X. Liu, C. J Spanos, Z. Max Shen, W. Lin, and L. Zhang, "Measuring fine-grained metro interchange time via smartphones," *Transportation Research Part C: Emerging Technologies*, vol. 81, pp. 153–171, 2017.

[28] Z. Yang, L. Shangguan, W. Gu, Z. Zhou, C. Wu, and Y. Liu, "Sherlock: Micro-environment sensing for smartphones," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3295–3305, 2014.

[29] W. Gu, Y. Zhou, Z. Zhou, X. Liu, H. Zou, P. Zhang, C. J Spanos, and L. Zhang, "Sugarmate: Non-intrusive blood glucose monitoring with smartphones," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 54, 2017.

[30] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot, and A. Ashkan, "Cache content-selection policies for streaming video services," in *INFOCOM*. IEEE, 2016, pp. 1–9.

[31] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[32] G. Ma, Z. Chen, and K. Zhao, "A cache management strategy for content store in content centric network," in *Fourth International Conference on Networking and Distributed Computing*. 2014, IEEE.

[33] H. Flores, R. Sharma, D. Ferreira, V. Kostakos, J. Manner, S. Tarkoma, P. Hui, and Y. Li, "Social-aware hybrid mobile offloading," *Pervasive and Mobile Computing*, vol. 36, pp. 25–43, 2017.