Cache Allocations for Consecutive Requests of Categorized Contents: Service Provider's Perspective

Minseok Choi[†*], Andreas F. Molisch[†], Dong-Jun Han[‡], Joongheon Kim[*], and Jaekyun Moon[‡]
[†]Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, USA
[‡]School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea
[*]School of Electrical Engineering, Korea University, Seoul, South Korea
E-mails: choimins@usc.edu, molisch@usc.edu djhan93@kaist.ac.kr, joongheon@korea.ac.kr, jmoon@kaist.edu

***Abstract*—In wireless caching networks, a user generally has a concrete purpose of consuming contents in a certain preferred category, and requests multiple contents in sequence. While most existing research on wireless caching and delivery has focused only on one-shot requests, the popularity distribution of contents requested consecutively is definitely different from the one-shot request and has been not considered. Also, especially from the perspective of the service provider, it is advantageous for users to consume as many contents as possible. Thus, this paper proposes two cache allocation policies for categorized contents and consecutive user demands, which maximize 1) the cache hit rate and 2) the number of consecutive content consumption, respectively. Numerical results show how categorized contents and consecutive content requests have impacts on the cache allocation.**

I. Introduction

In multimedia services, e.g., on-demand streaming services, a relatively small number of popular contents generally occupies a large portion of the massive global data traffic, and most of user demands are overlapped and repeated [1]. To deal with this issue, wireless caching technologies have been studied, wherein the base station (BS) or the server pushes popular contents for off-peak hours to cache-enabled nodes so that these nodes provide contents directly to nearby mobile users during peak hours [2]. In practice, caching nodes (i.e., caching helpers and/or cache-enabled devices) have finite storage sizes, which leads the content placement problem to determine which content is better to be stored in caching nodes [3].

The goal of the content placement problem is to find optimal caching policies according to the popularity distribution of contents and network topology. In stochastic wireless caching networks, there exist research efforts on probabilistic content placement introduced in [4], [5]. Many probabilistic caching methods have been proposed for various systems, e.g., device-to-device (D2D) networks [6], N-tier hierarchical networks [7], multi-quality dynamic streaming [8], and probabilistic coded caching was also recently proposed in [9].

Previous research optimized content placement for users requesting only one content, i.e., one-shot request, under the assumption that all content requests are independent. In multimedia services, the user typically accesses a service platform with the purpose of consuming a specific category of the content, and generally consumes more than one content. In this case, the sequence of consecutively consumed contents is highly correlated. For example, in video services such as YouTube, the related video list is recommended to the user after the first content is consumed [10]. In particular, the view count of a given video varies in almost the same scale as the average view count of the top referrer videos in the related list [11]; therefore, a user is highly likely to request one of the videos in the related category. Therefore, this paper considers the scenario in which a user consecutively requests multiple contents that are likely to be in its preferred category. In this context, this paper proposes two cache allocation policies for categorized contents and consecutive user demands, which maximize the cache hit rate and the expected number of consecutive content consumptions, respectively.

The previous work of [12] has proposed a caching policy for consecutive user demands with the assumption that the number of content requests is fixed. This assumption does not allow the service provider to maximize user's content consumption; however, in the perspective of service providers, it is important to satisfy as many of the user's requests as possible. In this paper, each user determines whether to continue to consume more contents depending on cache states in its vicinity, and the service provider aims at making users stay in the service longer.

The main contributions are as follows:

- Different from most existing results on wireless caching in which one-shot requests are considered only, consecutive requests of categorized contents are considered. In practice, the sequence of consecutively consumed contents is highly correlated, and an advanced caching scheme is required.
- Based on real data set, the recent work of [13] has modeled the category-based conditional content popularity distribution. This paper uses this measurement-based popularity model to obtain the proposed cache allocation rule for consecutive requests of categorized contents.
- This paper proposes two cache allocation schemes which maximize cache hit rates and the expected number of consecutive content requests, respectively. The iterative algorithm is presented to find the optimal cache allocation

rules and its convergence is proved.
- Numerical results show how 1) the popularity concentration to the preferred category and 2) different numbers of contents in the different categories influence the cache allocation rule.

The rest of the paper is organized as follows. The system model is described in Section II. The cache allocation rules maximizing the cache hit probability and the number of consecutive content consumption are proposed in Sections III and IV, respectively. The numerical results are shown in Section V and Section VI concludes the paper.

II. System Model

A. Wireless Caching Network

This paper assumes that caching nodes are randomly distributed according to a general spatial distribution Φ, and the server which has a content library $\mathcal{N}$ pushes some popular contents to each caching node during off-peak hours. Suppose that a library $\mathcal{N}$ consists of N contents and all contents have a normalized unit size. Let all N contents be grouped into K categories, and N_i contents are in category i denoted by $\mathcal{C}_i$, for all $i \in \mathcal{K} = \{1, \cdots, K\}$, satisfying $\sum_{i=1}^{K} N_i = N$. Also, denote the content index set of $\mathcal{C}_i$ by $\mathcal{N}_i = \{1, \cdots, N_i\}$.

The caching nodes have the finite storage size M, which means only M contents can be cached in each node. Since $N > M$ in practice, caching nodes store a part of contents in $\mathcal{N}$. A user requests the content from caching nodes in its vicinity. If the user finds at least one caching node that stores the desired content, this case is called the cache hit. When the user requests multiple contents, we define the cache hit as the case where *all* of requested contents can be found in nearby caching nodes. When there is no caching node having some of the requested contents, the server can deliver them via a cellular link. However, this paper assumes that the transmission quality of the cellular link is insufficient due to delay and/or congestion that leads to unacceptable video quality, so that henceforth we do not consider direct transmission from the server.

Let the storage size M be divided into K fractions with unequal sizes denoted by α_i for all $i \in \mathcal{K}$, and contents in $\mathcal{C}_i$ are stored within α_i. These fractions will be called cache allocations for categories and satisfy $\sum_{i=1}^{K} \alpha_i \leq M$ and $\alpha_i \leq N_i$. Given all of $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_K)$, how to store individual contents within each category becomes a classical content placement problem, and we consider the probabilistic caching policy for individual contents as shown in [4], [5].

B. Content Popularity Model

This paper focuses on the scenario in which the user requests multiple contents consecutively, different from most of existing caching policies which considered only one-shot requests. A representative example is a video streaming service. For example, a user can access the service platform with a concrete purpose of watching some sports highlight clips. In this case, we can postulate that sports is the user's preferred category, therefore the probability of requesting sports videos in sequence is very high. In contrast, the probability of requesting contents in other categories, e.g., movie trailers, is very small although not zero.

TABLE I
Key notations

K	Number of categories
N_i	Number of contents in category i
k	Index of the preferred category
M	Cache size
α_i	Cache allocation for category i
f_i	Global popularity of category i
p_1	Popularity of the preferred category
r	Rank of category that requests contents
l	Number of consecutive content requests
l_r	Number of requested contents in the r-ranked category
i_r	Category index of the r-ranked category
ϵ	Probability of not requesting the next content

Accordingly, the content request can be modeled by the following steps. First, the user randomly picks one category in $\mathcal{K}$. Each category $i \in \mathcal{K}$ has a global category popularity, which follows the Zipf distribution [4]: $f_i = i^{-\gamma} / \sum_{j=1}^{K} j^{-\gamma}$ where γ denotes the popularity distribution skewness. Then, the selected category has the first rank among all categories; note that the global category popularity is only used for choosing the first rank. Other categories can have any rank except for the first rank, but this paper models all categories that are not the first for this particular user as statistically equivalent; in other words, the relative ranking from $2, \cdots, K$ does not matter. After determining the preferred category, the user chooses one of categories to request the content depending on their ranks. Again, the category rank distribution given the preferred category is assumed to follow the Zipf distribution, i.e., $\Pr\{R = r\} = r^{-\gamma^{\text{out}}} / \sum_{j=1}^{K} j^{-\gamma^{\text{out}}}$, which represents the popularity of the r-th ranked category and γ^{out} is the Zipf factor. We denote the popularity of the preferred category by $p_1 = \Pr\{R = 1\}$. Note that p_1 is the probability of staying within the given preferred category, not the general probability of picking the 1st-ranked category as in [13], which is a different quantity. Here, we also consider the situation in which the user can stop to request contents by itself with small probability of ϵ. Therefore, the probability of requesting any content in the r-th ranked category after consuming the first content becomes $(1 - \epsilon) \cdot \Pr\{R = r\}$.

After choosing the category rank, the user requests the specific content in the category having the chosen rank. According to [13], the category-based conditional popularity distribution of contents in $\mathcal{C}_i$ follows the Mandelbrot-Zipf (M-Zipf) distribution, i.e.,

$$a_{i,n} = \frac{\frac{1}{(n+c^{\text{in}})^{\gamma_i^{\text{in}}}}}{\sum_{m=1}^{N_i} \frac{1}{(m+c^{\text{in}})^{\gamma_i^{\text{in}}}}}, \tag{1}$$

which represents the popularity of the n-th content in $\mathcal{C}_i$ for $n \in \mathcal{N}_i$. γ_i^{in} and c^{in} are the Zipf factor and the plateau factor of $\mathcal{C}_i$, respectively. Here, if γ^{out} is sufficiently large, $p_1 \gg 1 - \epsilon - p_1$ and the popularity of contents in the preferred $\mathcal{C}_k$ is much larger than that of any content in $\mathcal{C}_i$ for all $i \neq k$. Fig. 1 shows popularity distribution of 100 contents given the preferred category, grouped into 5 categories consisting

of 20 contents. This figure is obtained by multiplying the rank probability and the category-based individual content popularity, when $\gamma^{\text{out}} = 5$, $\gamma_i^{\text{in}} = 2.4$ and $c^{\text{in}} = 69$. Among them, contents whose indices are from 1 to 20 belong to the first-ranked category, and their popularity is relatively much larger than others. Therefore, if γ^{out} is sufficiently large, we approximate the popularity of contents outside the preferred $\mathcal{C}_k$ as uniform distribution, i.e., $a_{i,n}^k \approx \frac{1}{N-N_k}$ for all $i \in \mathcal{K} \setminus \{k\}$ and $n \in \mathcal{N}_i$, irrespective of ranks of those categories. When given $\mathcal{C}_k$, the popularity of contents in $\mathcal{C}_k$ is $a_{k,n}^k = a_{k,n}$ for $n \in \mathcal{N}_k$ still. Thus, consideration of two exclusive sets of the preferred category and all other contents is reasonable.

III. Maximization of Cache Hit Probability

This section derives the cache hit probability and proposes a cache allocation rule that maximizes the cache hit probability.

A. Cache hit probability

Suppose that the user request l contents in sequence. Among l contents, let l_r contents belong to the r-th ranked category satisfying $\sum_{r=1}^{K} l_r = l$. Then, when the preferred category $\mathcal{C}_k$ is given, the cache hit probability given l content requests, i.e., the probability that all of l requested contents can be delivered from any caching node, can be expressed as

$$p_{\text{hit}}^{l,k} = \prod_{r=1}^{K} \left[\Pr\{R = r\} h_{i_r}^k(\alpha_{i_r}) \right]^{l_r}, \tag{2}$$

where $i_r \in \mathcal{K}$ is the index of the r-th ranked category and

$$h_i^k(\alpha_i) = 1 - \sum_{n=1}^{N_i} a_{i,n}^k \sum_{j=0}^{\infty} \Pr_{\Phi}\{J = j\}(1 - b_{i,n}(\alpha_i))^j, \tag{3}$$

is the cache hit probability of a content request within $\mathcal{C}_i$ given α_i when $\mathcal{C}_k$ is the preferred category. In Eq. (3), $\Pr_{\Phi}\{J = j\}$ is the probability that there are j caching nodes storing the requested content in the vicinity of the user. Also, $b_{i,n}(\alpha_i)$ is the caching probability of the n-th content in $\mathcal{C}_i$ given α_i. However, computations required for scanning all combinations of l_r values are exponentially increasing as l grows.

When γ^{out} is large, $p_{\text{hit}}^{l,k}$ is simplified by using approximations of $a_{i,n}^k \approx \frac{1}{N-N_k}$, $\forall i \in \mathcal{K} \setminus \{k\}$ and $n \in \mathcal{N}_i$ into

$$p_{\text{hit}}^{l,k} \approx \sum_{m=0}^{l} \binom{l}{m} [p_1 h_k^k(\alpha_k)]^m \cdot [(1 - \epsilon - p_1) q_k(\boldsymbol{\alpha})]^{l-m}, \tag{4}$$

where

$$q_k(\boldsymbol{\alpha}) = 1 - \sum_{\substack{i=1 \\ i \neq k}}^{K} \sum_{n=1}^{N_i} \frac{1}{N - N_k} \sum_{j=0}^{\infty} \Pr_{\Phi}\{J = j\}(1 - b_{i,n}(\alpha_i))^j \tag{5}$$

which is the cache hit probability of a content request outside $\mathcal{C}_k$. Each term in Eq. (4) is the probability that among l requested contents, m are in $\mathcal{C}_k$ and $l-m$ contents are outside $\mathcal{C}_k$, and all of l contents can be found in caching nodes in the vicinity of the user. For simplicity, we will use the notation $h_k(\alpha_k) = h_k^k(\alpha_k)$ in the following sections.

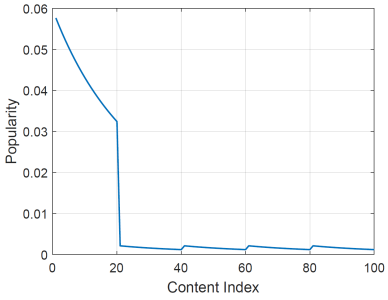

Fig. 1. Popularity of contents given the preferred category

The expected cache hit probability can be finally derived as

$$p_{\text{hit}} = \sum_{k=1}^{K} f_k \sum_{l=1}^{\infty} \Pr\{L = l\} \cdot \sum_{m=0}^{l} \binom{l}{m} [p_1 h_k(\alpha_k)]^m \cdot [(1 - \epsilon - p_1) q_k(\boldsymbol{\alpha})]^{l-m}, \tag{6}$$

where $\Pr\{L = l\}$ is the probability that the user requests l contents in sequence, which is given by

$$\Pr\{L = l\} = \epsilon(1 - \epsilon)^l. \tag{7}$$

Therefore, the cache hit probability is arranged into

$$p_{\text{hit}} = \sum_{k=1}^{K} f_k \sum_{l=1}^{\infty} \epsilon(1 - \epsilon)^l (p_1 h_k(\alpha_k) + (1 - \epsilon - p_1) q_k(\boldsymbol{\alpha}))^l \tag{8}$$

$$= \sum_{k=1}^{K} \frac{f_k (1 - \epsilon)\epsilon (p_1 h_k(\alpha_k) + (1 - \epsilon - p_1) q_k(\boldsymbol{\alpha}))}{1 - (1 - \epsilon)(p_1 h_k(\alpha_k) + (1 - \epsilon - p_1) q_k(\boldsymbol{\alpha}))} \tag{9}$$

In Eq. (9), any caching policy can be utilized within each category given the preferred $\mathcal{C}_k$ and $\boldsymbol{\alpha}$, and $h_k(\alpha_k)$ and $q_k(\boldsymbol{\alpha})$ are determined depending on the caching policy. Then, we can suppose that the caching policy that maximizes the cache hit rate [4], [5] is used for caching of individual contents within every category. Denote the maximum cache hit rates in and outside $\mathcal{C}_k$ by $h_k^*(\alpha_k)$ and $q_k^*(\boldsymbol{\alpha})$, respectively. Then, the cache hit probability of l content requests becomes

$$p_{\text{hit}}^* = \sum_{k=1}^{K} \frac{f_k (1 - \epsilon)\epsilon (p_1 h_k^*(\alpha_k) + (1 - \epsilon - p_1) q_k^*(\boldsymbol{\alpha}))}{1 - (1 - \epsilon)(p_1 h_k^*(\alpha_k) + (1 - \epsilon - p_1) q_k^*(\boldsymbol{\alpha}))} \tag{10}$$

$$= \sum_{k=1}^{K} g_k^*(\boldsymbol{\alpha}, f_k, \epsilon). \tag{11}$$

B. Problem Formulation

The optimal cache allocations of $\boldsymbol{\alpha}^\star = (\alpha_1^\star, \cdots, \alpha_K^\star)$ can be obtained by maximizing (11) as follows:

$$\boldsymbol{\alpha}^\star = \arg\max_{\boldsymbol{\alpha}} \; p_{\text{hit}}^* \tag{12}$$

$$\text{s.t.} \sum_{i=1}^{K} \alpha_i \leq M \tag{13}$$

$$0 \leq \alpha_i \leq \min\{M, N_i\}, \; \forall i \in \mathcal{K}. \tag{14}$$

The constraint (13) is for the storage size of each caching node and the constraint (14) is for the cache allocation for each category. The following key lemmas are used to solve the above problem of (12)–(14).

Lemma 1. $p_1 h_k(\alpha_k) + (1-\epsilon-p_1) q_k(\boldsymbol{\alpha})$ is increasing with α_k.

Proof: In this proof, we simply use the notation of $b_{i,n} = b_{i,n}(\alpha_i),\ \forall i \in \mathcal{K},\ \forall n \in \mathcal{N}_i$. Let $\alpha'_k = \alpha_k + \delta$ and $\delta > 0$. Then, $h^*_k(\alpha'_k) \geq \tilde{h}_k(\alpha'_k)$, where $\tilde{h}_k(\alpha'_k)$ can be the cache hit probability within $\mathcal{C}_k$ of any caching policy $\mathbf{b}'_k = (b'_{k,1}, \cdots, b'_{k,N_k})^T$ satisfying $\sum_{n=1}^{N_k} b'_{k,n} = \alpha'_k$. Let $\mathbf{b}'_k = (b'_{k,1} = b^*_{k,1} + \delta, b'_{k,2} = b^*_{k,2} \cdots, b'_{k,N_k} = b^*_{k,N_k})^T$. Then, since $0 \leq b^*_{k,1} \leq 1$ and $b^*_{k,1}$ is generally much closer to zero than one when the library size of N is large,

$$p_1 h^*_k(\alpha'_k) - p_1 h^*_k(\alpha^*_k) \geq p_1 \tilde{h}_k(\alpha'_k) - p_1 h^*_k(\alpha^*_k)$$
$$= p_1 \sum_{n=1}^{N_k} a_{k,n} \sum_{j=0}^{\infty} \Pr_\Phi\{J=j\}\{(1-b^*_{k,n})^j - (1-b'_{k,n})^j\} \tag{15}$$

$$\approx p_1 \cdot a_{k,1} \sum_{j=0}^{\infty} \Pr_\Phi\{J=j\} \cdot j \cdot \delta \tag{16}$$

is obtained by using the first-order Taylor approximation, i.e., $(1-b^*_{k,n})^j \approx 1 - j \cdot b^*_{k,n}$.

Since the storage size M is fixed, the cache size allocated to all categories except for $\mathcal{C}_k$ is $M - \alpha_k - \delta$. With small δ, there exists any category u for $u \neq k$ such that $\alpha_u \geq \delta$. Then, let $\alpha'_u = \alpha^*_u - \delta$ and $b'_{u,1} = b^*_{u,1} - \eta_1$, $b'_{u,2} = b^*_{u,2} - \eta_2$, $\cdots$, $b'_{u,N_u} = b^*_{u,N_u} - \eta_{N_u}$, where $0 \leq \eta_{u,n} \leq b^*_{u,n}$ for all $n \in \mathcal{N}_u$ and $\sum_{n=1}^{N_u} \eta_{u,n} = \delta$. In this case, cache allocations for other categories can remain unchanged, i.e., $\alpha'_i = \alpha^*_i$ and $b'_{i,n} = b^*_{i,n}$ for all $i \in \mathcal{K},\ i \neq k, u$. Then, similar to before,

$$(1-\epsilon-p_1) q^*_k(\boldsymbol{\alpha'}) - (1-\epsilon-p_1) q^*_k(\boldsymbol{\alpha}) \geq$$
$$\frac{1-\epsilon-p_1}{N-N_k} \Bigg[\sum_{\substack{i=1 \\ i \neq k}}^{K} \sum_{n=1}^{N_i} \sum_{j=0}^{\infty} \Pr\{J=j\} \times$$
$$\left\{ (1-b^*_{i,n})^j - (1-b'_{i,n})^j \right\} \Bigg] \tag{17}$$

$$\approx -\frac{1-\epsilon-p_1}{N-N_k} \sum_{j=0}^{J} \Pr\{J=j\} \cdot j \cdot \delta. \tag{18}$$

Since $p_1 > 1-\epsilon-p_1$ and $a_{k,1} > \frac{1}{N-N_k}$, $p_1 h^*_k(\alpha'_k) + (1-\epsilon-p_1) q^*_k(\boldsymbol{\alpha'}) - p_1 h^*_k(\alpha^*_k) - (1-\epsilon-p_1) q^*_k(\boldsymbol{\alpha}^*) > 0$ and the above lemma is finally proved. ∎

Lemma 2. The optimum vector $\boldsymbol{\alpha}^\star = (\alpha^\star_1, \cdots, \alpha^\star_K)^T$ satisfies $\sum_{i=1}^K \alpha^\star_i = M$.

Proof: Assume that $\sum_{i=1}^K \alpha^\star_i < M$, then $\exists \delta > 0$ such that $\sum_{i=1}^K \alpha^\star_i + \delta \leq M$ and $\alpha^\star_k + \delta < \min\{M, N_k\}$ for certain k. Let $\boldsymbol{\alpha'} = (\alpha^\star_1, \cdots, \alpha^\star_k + \delta, \cdots, \alpha^\star_K)^T$. According to Lemma 1, $g^*_k(\boldsymbol{\alpha}, f_k, \epsilon)$ is increasing with α_k for all $k \in \mathcal{K}$. Thus, $p^*_{\text{hit}}(\boldsymbol{\alpha'}) > p^*_{\text{hit}}(\boldsymbol{\alpha}^\star)$ and it obviously leads to contradiction. ∎

Algorithm 1 Greedy cache allocation algorithm

$\alpha^*_i = \frac{M}{K}$ for all $i \in \mathcal{K}$ and $p^\star_{\text{hit}} = 0$
for $\forall (u,v) \in \mathcal{K} \times \mathcal{K}$ and $u \neq v$ **do**
  $\beta_{u,v} = M - \alpha^*_u - \alpha^*_v$
  Obtain $b_{i,n}\ \forall i \in \mathcal{K}$ and $i \neq u, v$, and $\forall n \in \mathcal{N}_i$ according to [4], [5].
  for $\forall \alpha_u \in \{\max(0, \beta_{u,v} - N_v), \cdots, \min(\beta_{u,v}, N_u)\}$ **do**
    $\alpha_v \leftarrow \beta_{u,v} - \alpha_u$
    Obtain $b_{u,n}$ and $b_{v,m}\ \forall n \in \mathcal{N}_u$ and $\forall m \in \mathcal{N}_v$ according to [4], [5].
    Find p^*_{hit} based on $\boldsymbol{\alpha}$.
    if $p^\star_{\text{hit}} < p^*_{\text{hit}}$ **then** $p^\star_{\text{hit}} = p^*_{\text{hit}}$
    end if
  end for
end for

According to Lemma 2, an inequality constraint (13) can be converted into the equality constraint. The problem of (12)–(14) has K optimization parameters, and the subproblem for finding the optimal $\alpha^\star_u$ and $\alpha^\star_v$ is formulated as follows:

$$\{\alpha^\star_u, \alpha^\star_v\} = \underset{\alpha_u, \alpha_v}{\arg\max}\ \mathcal{M}_{(u,v)} \tag{19}$$

$$\text{s.t. } \alpha_u + \alpha_v = \beta_{u,v} = M - \sum_{i=1, i\neq u,v}^{K} \alpha_i \tag{20}$$

$$0 \leq \alpha_i \leq \min\{M, N_i\},\ \forall i \in \mathcal{K}, \tag{21}$$

where $\mathcal{M}_{(u,v)} = g^*_u(\boldsymbol{\alpha}, f_u, \epsilon) + g^*_v(\boldsymbol{\alpha}, f_v, \epsilon)$. Since $\{\alpha_i\}_{i\neq u,v}$ are fixed, $\alpha_u + \alpha_v$ also becomes a constant $\beta_{u,v}$.

A multivariable function p^*_{hit} can be optimized by iteratively optimizing the subset of variables if the convergence is guaranteed. To find $\boldsymbol{\alpha}^\star = (\alpha^\star_1, \cdots, \alpha^\star_K)$, the subproblem of (19)–(21) can be iteratively applied for all combinations of u and v, for $u, v \in \mathcal{K}$ and $u \neq v$. We find the maximum of the dual-variable problem of (19)–(21) in each iteration, and the sequence of the updated values of $\mathcal{M}_{(u,v)}$ is generated. Since this sequence is non-decreasing and the cache hit probability has a trivial upper bound of 1, i.e., $p^\star_{\text{hit}} \leq 1$, the convergence of the iterative algorithm is guaranteed.

Since $h_k(\alpha_k)$ and $q_k(\boldsymbol{\alpha})$ are obtained by using the bisection method [4], [5], however, the objective function of $\mathcal{M}_{(u,v)}$ is not in closed-form and the problem of (19)–(21) should be numerically handled. Therefore, we consider integer values for cache allocations of $\boldsymbol{\alpha}$ and the greedy algorithm can solve the problem with M not very large. If caching of content partitions is not considered, i.e., only caching of the whole content is allowed, the assumption that α_i is the integer number for all $i \in \mathcal{K}$ is reasonable. The details of the iterative algorithm to solve the problem of (12)–(14) are described in Algorithm 1.

IV. Maximization of Expected Number of Consecutive Content Requests

From the service provider's perspective, it is advantageous for the user to consume as many contents as possible. As

explained in Section II, the user does not request the next content with the probability of ϵ. In addition, we assume that the user stops to consume the next content when no caching node in the vicinity of the user stores the desired content, even though the user requests the next one.

The probability of stopping to consume more is given by

$$p_{\text{stop}}^k = \epsilon + p_1(1 - h_k(\alpha_k)) + (1 - \epsilon - p_1)(1 - q_k(\boldsymbol{\alpha})). \quad (22)$$

In (22), the first term is the probability of not requesting the next content, the second and third terms are probabilities that no caching node stores the requested content when the content belongs to $\mathcal{C}_k$ and is not in $\mathcal{C}_k$, respectively. Then, the expected number of consecutive content consumption is computed as

$$\mathbb{E}[L] = \sum_{l=1}^{\infty} l \cdot \Pr\{L = l\} = \sum_{k=1}^{K} f_k \sum_{l=1}^{\infty} l \cdot (1 - p_{\text{stop}}^k)^l p_{\text{stop}}^k \quad (23)$$

$$= \sum_{k=1}^{K} f_k \frac{1 - p_{\text{stop}}^k}{p_{\text{stop}}^k}. \quad (24)$$

Then, the optimization problem of maximizing the expected number of consecutive video consumption is as follows:

$$\boldsymbol{\alpha}^\star = \arg\max_{\boldsymbol{\alpha}} \mathbb{E}[L] \quad (25)$$

$$\text{s.t.} \sum_{i=1}^{K} \alpha_i \le M \quad (26)$$

$$0 \le \alpha_i \le \min\{M, N_i\}, \ \forall i \in \mathcal{K}. \quad (27)$$

Similar to Lemmas 1 and 2, p_{stop}^k can be proved to be increasing with α_k and the inequality constraint (26) can be converted into the equality constraint, i.e., $\sum_{i=1}^{K} \alpha_i = M$.

Again, the multivariable function $\mathbb{E}[L]$ can be maximized by iteratively optimizing the following dual-variable subproblem:

$$\{\alpha_u^\star, \alpha_v^\star\} = \arg\max_{\alpha_u, \alpha_v} \frac{f_u}{p_{\text{stop}}^u} + \frac{f_v}{p_{\text{stop}}^v} \quad (28)$$

$$\text{s.t. } \alpha_u + \alpha_v = \beta_{u,v} = M - \sum_{i=1, i\neq u,v}^{K} \alpha_i \quad (29)$$

$$0 \le \alpha_i \le \min\{M, N_i\}, \ \forall i \in \mathcal{K}. \quad (30)$$

The sequence of the updated objective values in (28) is nondecreasing, and $\mathbb{E}[L] \le \frac{1}{\epsilon} - 1$ because $p_{\text{stop}}^k \ge \epsilon$. Thus, the algorithm which solves the problem of (25)–(27) by iteratively optimizing the dual-variable problem of (28)–(30) for all combinations of $u, v \in \mathcal{K}$ and $u \neq v$, is guaranteed to converge. The whole algorithm is the same as Algorithm 1 except that p_{hit}^* should be changed into $\mathbb{E}[L]$ in lines 2, 9, 10.

V. Numerical Results

In the subsequent simulations, $N = 100$ contents and $K = 5$ categories are considered. The global category popularity follows $f_i > f_j$ for $i < j$. Caching nodes are distributed according to a Poisson point process with intensity of λ and caching nodes with distances less than $d = 10$ from the user are only considered. In addition, $M = 30$, $\gamma = 1$, $\gamma_i^{in} = 2.4$, and $c_i^{\text{in}} = 69$ are used for all $i \in \mathcal{K}$. We consider three different category structures as follows:

- Case A: $N_1 = N_2 = N_3 = N_4 = N_5 = 20$
- Case B: $N_1 = 35, N_2 = 25, N_3 = 20, N_4 = 15, N_5 = 5$
- Case C: $N_1 = 5, N_2 = 15, N_3 = 20, N_4 = 25, N_5 = 35$.

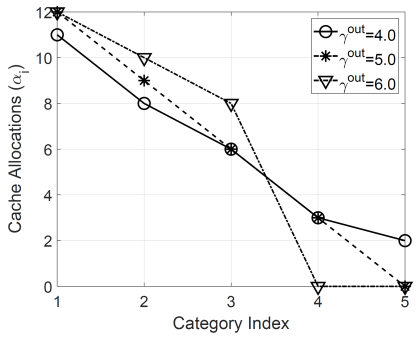

Fig. 2. Cache allocations depending on the skewness factor

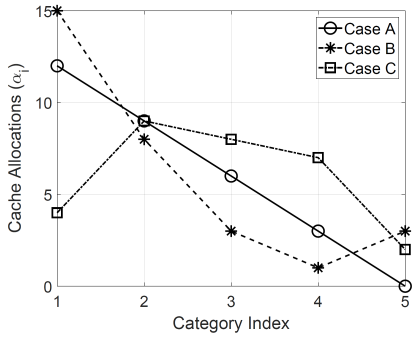

Fig. 3. Cache allocations depending on the number of contents in each category

In Figs. 2 and 3, plots of α_k for every category are shown with $\lambda = 0.02$ and $\epsilon = 0.1$. Case A is considered in Fig. 2. As the skew factor γ^{out} grows, the probability of requesting the content in the preferred category becomes much larger than that of requesting the content in other categories. Therefore, as γ^{out} increases, more cache sizes are allocated to categories having relatively large global category popularities in Fig 2.

In Fig. 3, all plots are obtained with $\gamma^{\text{out}} = 5$. Since all categories in Case A have the same number of contents, cache allocations of Case A depend only on global category popularity. In Case B, the category having a larger global popularity consists of more contents, therefore more cache sizes are allocated, i.e., α_1 in Case B becomes larger than that in Case A. Interestingly, α_5 in Case B is also larger than that in Case A. The reason is that N_5 is the smallest in Case B, i.e., the individual content popularity within $\mathcal{C}_5$ is the largest among all categories. Thus, even though f_5 is smaller than other f_i values, caching multiple contents of $\mathcal{C}_5$ is favorable for consecutive content requests. On the other hand, in Case C, α_1 is smaller than α_2, α_3 and α_4 because $N_1 = 5$ and α_1 should be smaller than N_1. It does not mean that an importance of caching contents in $\mathcal{C}_1$ decreases. Rather, it becomes more important because a portion of contents to be stored in caching

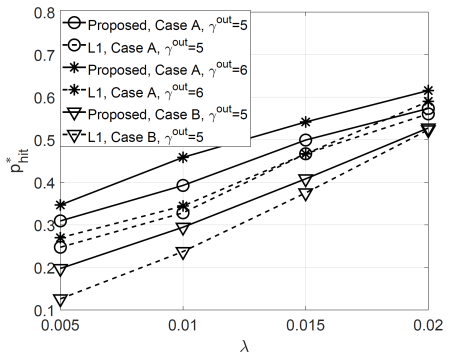

Fig. 4. The expected cache hit probabilities

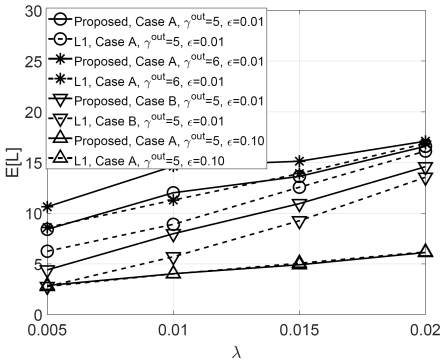

Fig. 5. The expected number of consecutive content consumption

nodes, i.e., $\frac{\alpha_1}{N_1}$, is larger than other cases. By saving the cache size for $\mathcal{C}_1$, a larger cache size can be allocated to other categories with low global popularities compared to Case A. Thus, Figs. 2 and 3 show that the skew factor as well as the number of contents in each category have a strong impact on the proposed cache allocation rule.

Fig. 4 shows plots of cache hit probabilities obtained from the problem of (12)–(14) versus λ. In Fig. 5, the expected numbers of consecutive content consumption obtained from the problem of (25)–(27) are shown. We compared the proposed scheme with the conventional caching method optimized for one-shot content request based on popularity of individual contents in [4], [5]. The comparison scheme is named as 'L1' in the figures. We can easily see in both figures that the proposed scheme outperforms 'L1' with different valuess of γ^{out} and N_i for each category. As λ grows, i.e., as the number of caching nodes in the vicinity of the user grows, the performance improvement of the proposed scheme decreases, because the user becomes more likely to find caching nodes to deliver multiple requested contents even with 'L1'. The performance gain of the proposed scheme over 'L1' is guaranteed when γ^{out} is large. Especially in Fig. 5, when $\epsilon = 0.1$, ϵ dominates the term in (22) representing the probability of stopping to consume contents; therefore, the advantage of the proposed scheme is not remarkable. As ϵ becomes smaller, however, the proposed algorithm is more advantageous for consecutive content consumption than 'L1'. Thus, the service provider can create the opportunity for users to consume more contents and to stay in the service longer by using the proposed scheme.

VI. Concluding Remarks

This paper proposes two optimal cache allocation rules when users request a random number of contents consecutively. The key characteristic that users are likely to consume content highly related to each other consecutively is well captured in the proposed scheme by maximizing the cache hit probability for multiple content requests from the same category. Another cache allocation which maximizes the number of consecutive content consumption is also proposed as it related to the benefits for the service providers. The impacts of categorized contents and consecutive content requests on the cache allocation rule are shown by numerical results.

Acknowledgment

This work was supported by NSF under projects NSF CCF-1423140 and NSF CNS-1816699, and Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00170, Virtual Presence in Moving Objects through 5G).

References

[1] X. Cheng, J. Liu, and C. Dale, "Understanding the Characteristics of Internet Short Video Sharing: A YouTube-based Measurement Study," *IEEE Trans. on Multimedia*, vol. 15, no. 5, pp. 1184–1194, August 2013.

[2] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless Video Content Delivery through Distributed Caching Helpers," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012.

[3] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless Content Delivery Through Distributed Caching Helpers," *IEEE Trans. on Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, December 2013.

[4] B. Blaszczyszyn and A. Giovanidis, "Optimal Geographic Caching in Cellular Networks," in *Proc. IEEE Int'l Conf. on Communications (ICC)*, London, UK, 2015.

[5] Z. Chen, N. Pappas, and M. Kountouris, "Probabilistic Caching in Wireless D2D Networks: Cache Hit Optimal Versus Throughput Optimal," *IEEE Commun. Letters*, vol. 21, no. 3, pp. 584–587, March 2017.

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

[7] K. Li, C. Yang, Z. Chen and M. Tao, "Optimization and Analysis of Probabilistic Caching in N -Tier Heterogeneous Networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1283-1297, Feb. 2018.

[8] M. Choi, J. Kim, and J. Moon, "Wireless Video Caching and Dynamic Streaming Under Differentiated Quality Requirements," *IEEE J. Sel. Areas in Commun.*, vol. 36, no. 6, pp. 1245–1257, June 2018.

[9] D. Ko, B. Hong and W. Choi, "Probabilistic Caching Based on Maximum Distance Separable Code in a User-Centric Clustered Cache-Aided Wireless Network," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1792-1804, March 2019.

[10] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn and S. Moon, "Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems," *IEEE/ACM Trans. Network.*, vol. 17, no. 5, pp. 1357-1370, Oct. 2009.

[11] R. Zhou, S. Khemmarat, and L. Gao, "The impact of YouTube recommendation system on video views" in *Proc. ACM IMC*, New York, NY, USA, 404-410, 2010.

[12] M. Choi, D. Kim, D.-J. Han, J. Kim and J. Moon, "Probabilistic Caching Policy for Categorized Contents and Consecutive User Demands", *IEEE Int'l Conf. on Communications (ICC)*, May 2019.

[13] M. Lee, A. F. Molisch, N. Sastry and A. Raman, "Individual Preference Probability Modeling and Parameterization for Video Content in Wireless Caching Networks," *IEEE/ACM Trans. Network.*, 2019.