Semigradient-based Cooperative Caching Algorithm for Mobile Social Networks

Yecheng Wu*†, Sha Yao†, Yang Yang*†, Zeming Hu‡ and Cheng-Xiang Wang§
*Key Lab for Wireless Sensor network and Communication, SIMIT Chinese Academy of Sciences, Shanghai, 200050, China
†Shanghai Research Center for Wireless Communications, Shanghai, 201210, China
‡ShanghaiTech University, Shanghai, 201210, China
§Institute of Sensors, Signals and Systems, Heriot-Watt University, Edinburgh, EH14 4AS, UK

***Abstract*—Wireless caching at users' devices in mobile social network is considered to be a promising solution to alleviate backhaul overload in future wireless networks. However, most of the current works propose caching schemes based on heuristic reasoning and intuition with poor performance or high complexity which are impractical due to individual devices' computing capacity restriction. In this paper, we design a cooperative caching scheme aimed at maximizing hit ratio, incorporating probabilistic modeling of mobility and user interests patterns from mobile social networks. Furthermore, we reformulate this optimization problem into a submodular function maximization and propose a semigradient-based cooperative caching scheme, while this scheme's efficiency is shown to significantly outperform the greedy caching by 99.6%.**

***Keywords*—Cooperative caching, submodular function, semigradient, hit ratio, mobile social network(MSN).**

I. Introduction

Along with dramatically increasing in mobile devices and services [1], the current network structure (especially backhaul network) cannot practically cope with the explosive data volumn [2]. Despite the unremitting efforts to enhance the network capacity in LTE and LTE-Advanced systems (such as MIMO, CoMP), the utilization efficiency of radio spectrum is notably reaching its theoretical cap [3]. In addition, the reason for the explosive traffic is that a virtual part of mobile traffic is occupied by duplicate downloads of a few popular contents [4]. Therefore, caching those popular contents at network edge [5](BSs and user devices) to reduce duplicate content transmission is considered as one of the most disruptive paradigms in 5G networks [6]. Especially, due to the development of mobile devices (such as caching, communication and computing ability), caching at users' devices where contents are cached and shared locally has attracted more researchers' and engineers' attention.

Traditionally caching has been widely studied in wired networks (which usually serves large areas equipped with great caching and computing capacity), where algorithms such as Most Popular Content (MPC) [2], Least Recently Used (LRU) [2], and Least Frequently Used (LFU) [3] are adopted as efficient content placement strategies. However, due to the unique features of device caching, those caching polices are not suitable for device caching: (i)Each individual device cache is equipped with **limited cache space** so that performance improvement can be enhanced only through elaborate cooperation among users. (ii)In reality, not all users share same interests and **heterogeneous interest pattern** should be considered. (iii)Due to users' mobility and limited communication range, those caches are accessed opportunistically so that **mobility pattern** play important roles on device caching police. (iv)High efficiency caching algorithm should be a priority for the sake of **computing capacity restriction** of individual device.

There are inherent social characteristics that play important roles in the interaction among mobile users. For example, users with higher contact frequency may have similar interests, which means that they have similar high-frequency access contents. Therefore, designing an efficient cooperative caching strategy, by exploiting the social characteristics, can improve the performance dramatically. With the rapid development of mobile social network (MSN), more and more works investigate wireless caching taking specific consideration of the social network patterns. In [7], the non-cooperative and cooperative strategies are compared. The non-cooperative scheme only consider the user's own interest when caching, and the cooperative strategy takes into account both own interests and the interests of other most likely encountered users, thanks to the information of users' interests and mobility patterns provided by the social network modeling. It has been shown that the cooperative strategy outperforms the non-cooperative one so that great improvement will be achieved through effective cooperation. A hierarchical cooperative caching policy is designed in [8], where with the social information on friendship in MSN, friends' interest are given higher priority when designing the content placement scheme by dividing the local cache space of a user into its own space and the space for friends. In [9] the author proposed a novel cooperative caching method based on the concept of close friend set in social network to enable fair sharing of cached contents.

The aforementioned works shows promising improvement of incorporating knowledge on social network patterns to the design of D2D-assisted caching schemes. However, most of the proposed methods are based on heuristic reasoning and intuition which are equipped with poor performance or high complexity. In this paper, we design a cooperative caching scheme aimed at maximizing hit ratio via a probabilistic mobility and user interest modeling. Furthermore, we show that the optimization problem can be reformulated into a submodular function maximization. A semigradient-based cooperative caching algorithm for mobile social network are proposed and it is shown via numerical studies that the proposed algorithms can significantly outperform the existing schemes while guaranteeing performance and maintaining much faster

978-1-5090-1328-9/16/$31.00 ©2016 IEEE

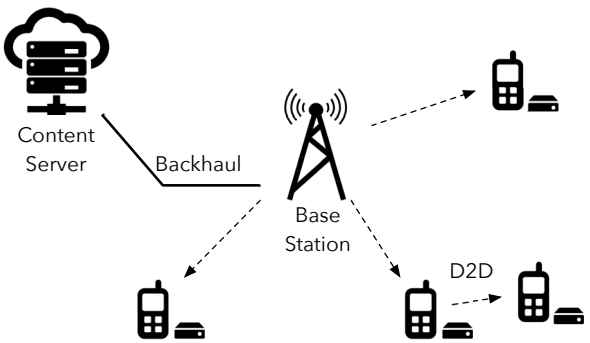


Fig. 1. Illustration of wireless systems considered in our framework, where a single file server is serving multiple cache-enabled mobile users via a base station. The users can share cached content via D2D links with their neighbors.

speed.

The rest of the paper is organized as follows: Section II introduces the system model and the optimization framework based on a generic probabilistic mobility and user interest model. Then we propose our subgradient-based content placement algorithms in Section III. Section IV presents the modeling of interaction between mobility and user interests. The simulation result is given in section V. Section VI concludes the paper.

II. System model

We consider a wireless system consisting of a single base station and M mobile users. The base station is connected to a remote file server via wired/wireless backhaul. The users are interested in accessing the files stored in the remote file server through the base station. Each user device has a local cache of fixed size and can cache part of the content library. We term the set of cache as *virtual cache space* which is associated with user number and each cache size. Meanwhile, if two users are in proximity, D2D communication can be established for file sharing. Ultimately, we are interested in *smart* content placement schemes (which files to store in the caches of the user devices) which can reduces the load of the base station backhaul link. The system is depicted in Fig. 1.

Specifically, the M mobile users are denoted by the set $\mathcal{M} = \{1, 2, 3, \cdots, M\}$. On the file server, there are F files $\mathcal{F} = \{1, 2, \cdots, F\}$ of equal sizes. User m at any discrete moment in time n request one file in $\mathcal{F}$ according to some popularity distribution $p_{m,f} \in \{p_{m,1}, p_{m,2}, \cdots, p_{m,F}\}$, where $p_{m,f}$ denotes the probability of user m requesting file f. The popularity distribution of M users are specified by the popularity distribution matrix $\mathbf{P}_{M\times F}$ with the element $p_{m,f}$. The users are moving around inside the system. We assume a probabilistic mobility model by specifying the pair-wise probability q_{m_1,m_2} as the probability of user m_1 meeting user m_2. Thus, the mobility pattern is fully specified by the encounter probability matrix $\mathbf{Q}_{M\times M}$, with the element q_{m_1,m_2}. Here the encounter of two users means that users are close enough such that efficient D2D connection can be established between the two users.

A caching scheme can be divided into two phases, namely, the content placement phase and the content delivery phase. For content delivery, at any moment n, user m requests a file according to its popularity distribution. The file is accessed from the local cache of the device if it is present. Otherwise, it is accessed from the users in its proximity if it is present in any of their local cache via D2D transmission. If the file is not present in the local cache of the user or its neighbors, it is transmitted by the remote file server via the base station.

Assume that the local cache size of each user is K, i.e each user can cache at most K files. The content placement scheme of user m is specified by the vector $\mathbf{c}_m = [c_{m,1}, c_{m,2}, ..., c_{m,F}]^T$, where $c_{m,f}$ is defined as :

$$c_{m,f} = \begin{cases} 0, & \text{if user } m \text{ doesn't cache content } f, \\ 1, & \text{if user } m \text{ caches content } f. \end{cases} \quad (1)$$

Due to the constraint on cache size, $\sum_{f=1}^{F} c_{m,f} \leq K$. A content placement scheme is thus defined by the matrix $\mathbf{C}_{F\times M} = [\mathbf{c}_1, \cdots, \mathbf{c}_M]$. Throughout of this paper, we consider only *static content placement schemes*, where the cache is filled with content at the beginning and is fixed.

Under a content placement scheme $\mathbf{C}_{F\times M}$, we define the probability that user m can access file f from its local cache or that of its encountered users as the *hit ratio*, denoted by $a_{m,f}$. The hit ratio can be expressed as follows:

$$a_{m,f} = 1 - \prod_i B_{m,i,f} \quad (2)$$

where $B_{m,i,f}$ denotes the probability that user m cannot access content f from user i, and $B_{m,i,f} = 1 - c_{i,f} q_{m,i}$.

The hit ratio of user m can thus be written as:

$$a_m = \sum_{f=1}^{F} a_{m,f} p_{m,f}. \quad (3)$$

For the system with M users and F content files, given the popularity matrix $\mathbf{P}$, the mobility matrix $\mathbf{Q}$ and the cache size limit K, we formulate the problem of finding the optimal content placement scheme $\mathbf{C}$ as the following optimization problem, where the optimality is in the sense of maximizing the average hit ratio $A(C)$ with content placement scheme $\mathbf{C}$ of the system:

$$\underset{\mathbf{C}}{\text{maximize}} \quad A(C) = \frac{1}{M} \sum_{m=1}^{M} a_m \quad (4)$$

$$\text{subject to} \quad \sum_f c_{m,f} \leq K, \quad \forall m \quad (5)$$

$$c_{m,f} \in \{0, 1\}, \quad \forall m, f \quad (6)$$

III. The Algorithms for Content Placement

It is easily shown that the optimization problem as stated in (4), (5) and (6) is NP-hard. In this section, we prove that this utility function is submodular, thus efficient algorithms with low complexity exist to solve the optimization approximately. We reformulate the optimization problem in (4), (5) and (6) as follows:

Define a ground set $\mathcal{V} = \mathcal{M} \times \mathcal{F}$, where $(m, f) \in \mathcal{V}$ represents the configuration that file f is placed in user m's local cache. Thus, ground set $\mathcal{V}$ contains all possible file placement configurations, and our goal is to find a subset of

$\mathcal{V}$ (a group of configurations) which maximize the average hit ratio under the cache size constraint. Denote $g: 2^{\mathcal{V}} \to \mathbb{R}$ as a *discrete set function* on subsets of the ground set $\mathcal{V}$, which is defined as follows:

$$g(\mathcal{X}) = 1 - \frac{1}{M} \sum_{(m,f)\in\mathcal{X}} p_{m,f} \left(\prod_{(m',f)\in\mathcal{X}\cap\mathcal{V}^{(f)}} (1 - q_{m,m'}) \right), \tag{7}$$

where $\mathcal{V}^{(f)} = \{(m,f) : m \in \mathcal{M}\}$ is defined for content file f. Likewise for user m we define $\mathcal{V}_{(m)} = \{(m,f) : f \in \mathcal{F}\}$. The original optimization problem can now be shown to be equivalent to the following problem:

$$\max_{\mathcal{X}\in\mathcal{C}} g(\mathcal{X}), \tag{8}$$

where $\mathcal{C} \in 2^{\mathcal{V}}$ is a family of feasible sets which satisfies the cache size constraint, i.e., $\forall \mathcal{K} \in \mathcal{C}$, $|\mathcal{K} \cap \mathcal{V}_{(m)}| \leq K$ for all $m \in \mathcal{M}$.

Definition 1 [4]: Defining $g(j|\mathcal{S}) \triangleq g(\mathcal{S}\cup j) - g(\mathcal{S})$ as the gain of $j \in \mathcal{V}$ with respect to $\mathcal{S} \subseteq \mathcal{V}$, then g is *submodular* if and only if $g(j|\mathcal{S}) \geq g(j|\mathcal{T})$ for all $\mathcal{S} \subseteq \mathcal{T}$ and $j \notin \mathcal{T}$.

Lemma 1: The discrete set function $g(\mathcal{X})$ as shown in (7) is a monotone non-decreasing submodular function.

Proof: Intuitively it is apparent that $g(\mathcal{X})$ is monotonically non-decreasing since adding one more configuration to a configuration group will only increase the chance of users accessing the file through local caches. The formal proof is as follows:

For $\mathcal{X} \subseteq \mathcal{V}$ and $(m,f) \in \mathcal{V} \setminus \mathcal{X}$, we have

$$\begin{aligned} &g((m,f)\cup\mathcal{X}) - g(\mathcal{X}) \\ =&\frac{1}{M} \sum_{m'\in\mathcal{M}} p_{m',f} q_{m',m} \prod_{(m'',f)\in\mathcal{X}} (1 - q_{m'',m}) \\ \geqslant&0. \end{aligned} \tag{9}$$

To prove that it is also a submodular function, we assume that $\mathcal{X}_1, \mathcal{X}_2 \subseteq \mathcal{V}$, $\mathcal{X}_1 \subseteq \mathcal{X}_2$ and $(m,f) \in \mathcal{V} \setminus \mathcal{X}_2$.

$$\begin{aligned} &g((m,f)|\mathcal{X}_1) - g((m,f)|\mathcal{X}_2) \\ =&\frac{1}{M} \sum_{m'\in\mathcal{M}} \left((1 - q_{m,m'}) \left(1 - \prod_{(m'',f)\in\mathcal{X}_2^f \setminus \mathcal{X}_1^f} (1 - q_{m',m''}) \right) \right. \\ &\left. \times \prod_{(m''',f)\in\mathcal{X}_1^f} (1 - q_{m',m'''}) \right) \\ \geq&0 \end{aligned} \tag{10}$$

Thus, $g(\mathcal{X})$ is a submodular function. ■

Traditionally, there are some algorithms proposed for submodular maximization like [10], but its time complexity is $O(MF)^8$ which is too computational and unacceptable for larger M and F. In [4], the author propose a greedy submodular maximization algorithm with comparatively low complexity whose iteration number is at least MF. However, for individual devices equipped with limited computing capacity in dynamics MSN, those algorithms are obviously unsuitable. As follows, we will introduce the concept the semidifferential to assist the cooperative caching algorithm design.

Definition 2 [11]: The subdifferential $\partial_g(\mathcal{X})$ of a submodular set function $g: 2^{\mathcal{V}} \to \mathbb{R}$ for a set $\mathcal{X} \subseteq \mathcal{V}$, is defined as:

$$\begin{aligned} \partial_g(\mathcal{X}) =&\{y \in \mathbb{R}^{MF} : \\ &g(\mathcal{X}_1) - y(\mathcal{X}_1) \geq g(\mathcal{X}) - y(\mathcal{X}), \text{ for all } \mathcal{X}_1 \in \mathcal{C}\} \end{aligned} \tag{11}$$

We denote a subgradient at $\mathcal{X}$ by $h_{\mathcal{X}} \in \partial_g(\mathcal{X})$. The extreme point of $\partial_g(\mathcal{X})$ can be computed as follows:

Let σ be a permutation of V that assign the elements in $\mathcal{X}$ to the first $|\mathcal{X}|$ positions . Each such permutation defines a chain with elements $S_0^\sigma = \emptyset$, $S_i^\sigma = \{\sigma(1), \sigma(2), ...\sigma(i)\}$ and $S_{|\mathcal{X}|}^\sigma = \mathcal{X}$. This chain defines an extreme point $h_{\mathcal{X}}^\sigma$ with entries

$$h_{\mathcal{X}}^\sigma(\sigma(i)) = g(S_i^\sigma) - g(S_{i-1}^\sigma) \tag{12}$$

And

$$h_{\mathcal{X}}^\sigma(\mathcal{X}_1) = \sum_{\sigma(i)\in\mathcal{X}_1} h_{\mathcal{X}}^\sigma(\sigma(i)) \tag{13}$$

With the above subgradient $h_{\mathcal{X}}^\sigma$, we can define a genetic algorithm. In each iteration, the algorithm optimizes a modular approximation formed via the current solution $\mathcal{X}$. For maximization a lower bound

$$m_{h_{\mathcal{X}}}(\mathcal{X}_1) = g(\mathcal{X}) + h_{\mathcal{X}}(\mathcal{X}_1) - h_{\mathcal{X}}(\mathcal{X}) \leq g(\mathcal{X}_1) \tag{14}$$

This bound is tight at the current solution ,satisfying

$$m_{h_{\mathcal{X}}}(\mathcal{X}) = g(\mathcal{X}) + h_{\mathcal{X}}(\mathcal{X}) - h_{\mathcal{X}}(\mathcal{X}) = g(\mathcal{X}) \tag{15}$$

In most cases, optimizing the modular approximation is much faster than the original cast function $g(\mathcal{X}_1)$ like the greedy algorithms in [4]. Algorithm 1 **SCC** shows our scheme for submodular function maximization.

Algorithm 1 **S**emigradient-based **C**ooperative **C**aching(**SCC**)

Start with $t \leftarrow 0$, any $\mathcal{X}^0 \in \mathcal{C}$
Repeat
(i) Pick a semigradient $h_{\mathcal{X}^t}^\sigma$ at $\mathcal{X}^t$:
Set σ as a permutation of V that assign the element $\mathcal{X}^t$ to be first $|\mathcal{X}|$ positions
$S_0^\sigma = \emptyset$ and $g(S_0^\sigma) = 0$
For $i = 1 : |\mathcal{V}|$
$S_i^\sigma = \{S_{i-1}^\sigma, \sigma(i)\}$
$h_{\mathcal{X}^t}^\sigma(\sigma(i)) = g(S_i^\sigma) - g(S_{i-1}^\sigma)$
End
$h_{\mathcal{X}^t}^\sigma(\mathcal{X}) = \sum_{\sigma(i)\in\mathcal{X}} h_{\mathcal{X}^t}^\sigma(\sigma(i))$
(ii) $\mathcal{X}^{t+1} := \arg\max_{\mathcal{X}\in\mathcal{C}} m_{h_{\mathcal{X}^t}}(\mathcal{X})$
$= \arg\max_{\mathcal{X}\in\mathcal{C}} g(\mathcal{X}^t) + h_{\mathcal{X}^t}^\sigma(\mathcal{X}) - h_{\mathcal{X}^t}^\sigma(\mathcal{X}^t)$
(iii) $t \leftarrow t + 1$
Until We have converged($\mathcal{X}^{t+1} = \mathcal{X}^t$)

Actually, SCC starting with different initial values (such as most popular caching, emptyset and random caching) will lead to solutions converged to different local optimal. Due to the page limit, we don't show the analysis detailedly and just present the conclusion. Under all circumstance, SCC initialing

with most popular caching always perform best among three initial values above. For example, under baseline simulation parameters in Table I, SCC initializing with most popular caching outperforms SCC with other initial values by 10.3% and 10.6%. In addition, the impact of initial value on SCC algorithms grows with larger virtual cache space, steeper request probability distribution and more heterogeneous interest. Therefore, for simplicity, we just show the evaluation results of SCC initializing with most popular caching in Section V.

IV. Social Networks Modeling based on Interest Vector and Interest Similarity

So far, we have assumed generic probabilistic model for users' request patterns and mobility model. In this section, we will model the interaction between request and mobility with results from mobile social network researches.

It has been shown in [12] and [13] that in a social network, people with more common interest have higher chance of meeting each other. To model this effect in social networks, we introduce the *relative interest* of user m towards file f as $s_{m,f} \in [0,1]$. Therefore, the *interest vector* of user m, denoted as $\mathbf{s}_m = [s_{m,1}, s_{m,2}, \cdots, s_{m,F}]^T$, represents the user's relative interest to different files in the library. Naturally we can also define the interest matrix of the system as $\mathbf{S}_{F\times M} = [\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_M]$. To measure the similarity of two users' interests, we define the *interest similarity* of two interest vectors as

$$d_{m_1,m_2} = \frac{1}{F}\sum_{i=1}^{F} w_i\,(1 - |s_{m_1,i} - s_{m_2,i}|), \qquad (16)$$

where, w_i is the weight factor of interest i. For simplicity, we treat all interests equally so that $w_i = 1$ for all i.

A. Interest vector and file request probability

It has been shown in literature that the request pattern of a user usually follows a Zipf-like distribution [13]. In my paper, as in [13], given the interest vector of user m $\mathbf{s}_m$, the corresponding request probability is:

$$p_{m,f} = \frac{\gamma}{rank(m,f)^{\alpha}}, \qquad (17)$$

where α is the parameter that determine the shape of the Zipf distribution, γ is a normalization factor, and $rank(m,f)$ is the order of file f by a descent sort of $\mathcal{F}$ according to the interest $s_{m,f}$.

B. Interest similarity and contact probability

In order to investigate the model of interaction between request and mobility, we analyze the Infocom06 dataset [14]. The Infocom06 dataset contains the contact logs among 79 volunteers in a period 337417 seconds and questionnaire answers regarding their languages, interest topics and other aspects. For any question regarding a specific interest, a user can make multiple choices. Similar to the method used in measuring the "social feature distance" in [15], if two users have made common choices in one of their answers, we view the similarity of this answer (interest) as 1; otherwise it is 0. Eventually, the sum of 10 fields are considered except the name and email. In Fig. 2, the samples show that the

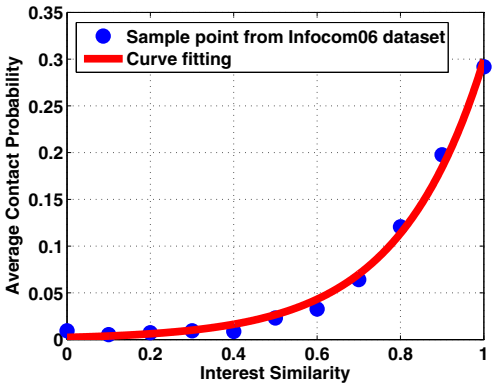


Fig. 2. The relationship between contact probability and interest similarity.

The total number of users M	200
The total number of contents F	200
Cache size of each user K	5
The parameter of Zipf distribution alpha α	0.5
The link parameter {d1,d2}	{4.84,1.55}

TABLE I. Baseline simulation parameters

average contact probability gets larger as the interest similarity increases. Through curve fitting of an exponential function in the form of (18), we have $d_1 = 4.839$ and $d_2 = 1.250$, with a 0.994 goodness of fit.

$$q_{m_1,m_2} = \exp\{d_1(d_{m_1,m_2} - d_2)\}, \qquad (18)$$

As can be seen, d_1 largely controls the sensitivity of interest similarity to the contact probability, and d_2 control the overall level of contact probability in a mobile social network. We term d_1 and d_2 as the *link parameters*.

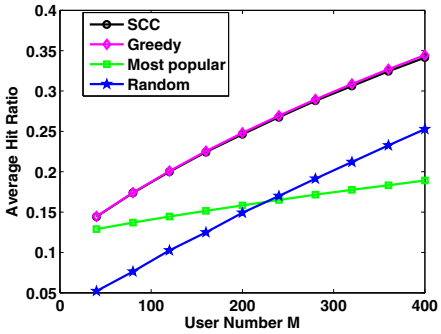
V. Evaluation

In this section, we investigate the performance of our proposed SCC and the other three existing algorithms. For all schemes, D2D link is used whenever possible (when two users are in contact) to facilitate cooperative caching. According to Section IV, we generate a hybrid setup where $\beta \in [0,1]$ portion of the users (with some rounding when necessary) share the same interest vector and the other $1-\beta$ portion of the users have independent randomly generated interest vector. In all simulations, we set baseline simulation parameters as in Table I. Then we vary one parameter while keeping the other parameters fixed. As follows, we first investigate the impact of virtual cache space, request probability distribution and interest similarity on system performance. Then we compare SCC with greedy caching to reveal the efficiency of algorithms.

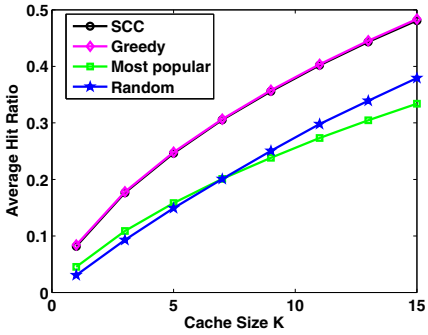
A. Comparison

We first present as a benchmark some widely used content placement scheme, namely, the random caching, the most popular caching, and the greedy caching schemes.

- *Random Caching*: Each user randomly caches K content files.
- *Most Popular Caching*: Each user caches the top K ranked content files according to their individual request probability.



(a) Hit ratio v.s User number



(b) Hit ratio v.s Cache size

Fig. 3. Average hit ratio as the virtual cache size space changes, for $\beta = 0.5$.

- *Greedy Caching*: Similar to the algorithm proposed in [4], the greedy caching starts with an empty set; at each step, it adds one element with the highest marginal value to the set until cache is full.

B. Evaluation result

1) The impact of virtual cache space: Virtual cache space is the set of individual cache which is associated with user number and individual cache size. Fig. 3 depicts the simulation results of hit ratio as the virtual cache space changes. As user number or cache size increases, more content copies can be cached and shared among users locally so that hit ratio grows. Our proposed SCC always have same performance with greedy scheme as virtual space varies. Besides, due to the cooperation among distributed caches, it is evident that our proposed SCC and greedy scheme perform significantly better than the random and most popular one for all user number and cache size. For user number of 160, our proposed SCC perform about 79.7% better than random caching and 48.0% better than most popular caching. In addition, random caching outperforms most popular caching when virtual cache space is large. This is due to the fact that the most popular scheme is selfish in nature, while other schemes take into consideration other users' need during content placement (passively so in certain sense for random caching scheme as being non-selfish).

Nowadays, with rapid development of mobile device and mobile users [3], larger virtual space will be provided to support caching at users' devices. Therefore, with the analysis above, we can conclude that caching at users' devices with SCC in MSN can efficiently cope with increasing users number and ensure hit ratio.

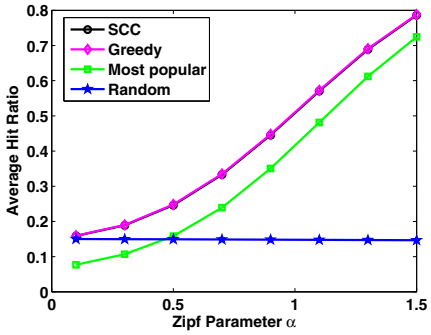


Fig. 4. Average hit ratio as the request probability distribution changes, for $\beta = 0.5$.

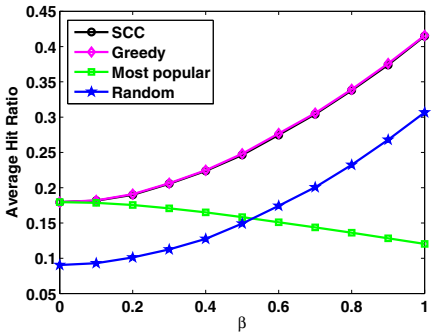


Fig. 5. Average hit ratio of as the interest similarity changes.

2) The impact of request probability distribution: Fig. 4 shows the evolution of different caching methods with different Zipf parameter α. From the equation (17), it shows that larger Zipf distribution parameter means steeper request probability distribution where majority of requests probably concentrate on the limited number of high rank popular contents. Similar with the analysis before, SCC achieve same hit ratio with greedy caching scheme. As the Zipf parameter α increases, the performance of all caching schemes improves except random caching. This is due to the fact that for random caching, each content is treated equally and has the same probability cached. For larger α, although more requests will focus on those high rank contents, more contents become less popular and there exists greater probability of caching those comparatively unpopular contents.

In reality, the dynamic of network load is mainly caused by the limited number of popular contents [16] (such as break event news or popular movies). In such situation, there is steep content request probability distribution and large α of Zipf distribution. Therefore, with the analysis above, caching at users' devices in MSN can be viewed as a promising way to handle network load dynamics.

3) The impact of interest similarity: Fig.5 shows the hit ratio when β changes. Larger β means more similar interest which means that users have larger probability of contacting and sharing with others. As β increases, the performance of all schemes improves except most popular caching. Because, for most popular caching where user caches contents according to their own interest, all users cache same contents and there is less chance of sharing. Besides, when users are equipped with

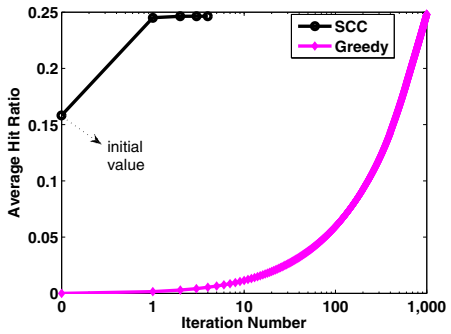


Fig. 6. The comparison of the converging speed between SCC and greedy caching schemes under baseline simulation parameters.

completely heterogeneous interest ($\beta = 0$), greedy, SCC and most popular caching have similar performance.

From the studies in [12], users with similar interest have large probability of gathering where congestion usually occurs with current network structure. Therefore, through the simulation results of hit ratio as β changes, we can consider caching at users' devices as an effective way to solve network congestion and our proposed SCC can guarantee the performance.

4) The comparison between SCC and Greedy caching: From Fig.3∼5, it's interesting to find that our proposed SCC caching scheme achieve same hit ratio with greedy one, no matter how virtual cache space/request probability distribution/interest similarity changes. However, SCC is considerably faster in convergence than the existing greedy algorithm, as can be seen from Fig. 6. Under baseline simulation parameter, SCC converge in 4 iterations while the greedy one converges in 1000 under baseline parameters. Although SCC achieve similar hit ratio with greedy caching scheme, the former's efficiency outperform the latter's by 99.6% in iteration process.

Currently, each individual device is equipped with limited computing capacity so that those high complexity caching schemes aren't suit for caching at users' devices especially in dynamic MSN. Our proposed caching scheme, SCC, is equipped with low complexity and high efficiency which can be executed at terminals for dynamic MSN, while guaranteeing hit ratio and maintaining much faster speed. Therefore, our proposed SCC scheme is an appropriate choice of cooperative caching at users' devices in MSN.

VI. Conclusion

In this paper, we design a cooperative caching scheme aimed at maximizing hit ratio, incorporating probabilistic modeling of user mobility and heterogeneous interests patterns from mobile social networks. The optimization problem is reformulated into a submodular function maximization and subgradient-based cooperative caching scheme with initial value of most popular caching (SCC) is proposed as solution. Through numerical investigations, it's demonstrated that caching at users' devices in MSN can effectively cope with the increasing user number, dynamic workload and network congestion. Furthermore, compared with the greedy caching scheme, SCC's efficiency outperforms the greedy one by 99.6% while guaranteeing performance and maintaining a much faster speed. Therefore, our proposed caching algorithm is an appropriate choice of caching at users' devices, especially for dynamic mobile social network and those individual devices with limited computing capacity.

Acknowledgment

This research is partially supported by the National Natural Science Foundation of China (NSFC) under grant 61571004, international cooperative project of the Ministry of Science and Technology under grant 2014DFE10160, the Ministry of Science and Technology (MOST) 863 Hi-Tech Program under grant 2014AA01A701, the key project of Science and Technology Commission of Shanghai Municipality (STCSM) under grant 15511103200, the EPSRC TOUCAN project under grant EP/L020009/1, the EU H2020 5G Wireless project under grant 641985, and the EU FP7 QUICK project under grant PIRSES-GA-2013-612652.

References

[1] C. Zhao, W. Zhang, Y. Yang, and S. Yao, "Treelet-Based Clustered Compressive Data Aggregation for Wireless Sensor Networks," IEEE Transactions on Vehicular Technology, vol. 64, pp. 4257-4267, 2015.

[2] H. Ahlehagh and S. Dey, "Video caching in radio access network: impact on delay and capacity," in *IEEE WCNC*, pp. 2276-2281, Apr. 2012.

[3] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," in *IEEE Communications Magazine*, vol. 52, pp. 131-139, Feb. 2014.

[4] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire,"Femtocaching: Wireless video content delivery through distributed caching helpers,"in *IEEE INFOCOM*, pp. 1107-1115, Mar. 2012.

[5] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," in *IEEE Communications Magazine*, vol. 52, pp. 82-89, Aug. 2014.

[6] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," in *IEEE Communications Magazine*, vol. 52, pp. 74-80, 2014.

[7] E. Jaho and I. Stavrakakis, "Joint interest-and locality-aware content dissemination in social networks," in *Sixth International Conference on Wireless On-Demand Network Systems and Services*, pp. 173-180, Feb. 2009.

[8] Y. Wang, J. Wu, and M. Xiao, "Hierarchical cooperative caching in mobile opportunistic social networks," in *IEEE GLOBECOM*, pp. 411-416, Dec. 2014.

[9] D. Wei, K. Zhu, and X. Wang, "Fairness-aware cooperative caching scheme for Mobile Social Networks," in *IEEE ICC*, pp. 2484-2489, Jun. 2014.

[10] G. Calinescu, C. Chekuri, M. Pl, and J. Vondrk, "Maximizing a submodular set function subject to a matroid constraint," in *Integer programming and combinatorial optimization*, pp. 182-196, 2007.

[11] R. Iyer, S. Jegelka, and J. Bilmes, "Fast Semidifferential-based Submodular Function Optimization," in *Proceedings of The 30th International Conference on Machine Learning*, pp. 855-863, 2013.

[12] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: positive and negative social effects," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 387-401, Feb. 2013.

[13] J. Iqbal and P. Giaccone, "Interest-based cooperative caching in multi-hop wireless networks," in *IEEE Globecom*, pp. 617-622, Dec. 2013.

[14] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, CRAWDAD trace cambridge/haggle/imote/infocom2006 (v. 2009-05-29). Downloaded from http://crawdad.cs.dartmouth.edu/ cambridge/haggle/imote/infocom2006, May. 2015.

[15] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *IEEE INFOCOM*, pp. 1368-1376, Mar. 2012.

[16] H. Zhang, G. Jiang, K. Yoshihira, and H. Chen, "Proactive workload management in hybrid cloud computing," in *Network and Service Management, IEEE Transactions on*, vol. 11, pp. 90-100, 2014.