Hybrid Human-Artificial Intelligence Enabled Edge Caching Based on Interest Evolution

Zhidu Li[1,2,3(✉)], Fuxiang Li[1,2,3], Dapeng Wu[1,2,3], Honggang Wang[4], and Ruyan Wang[1,2,3]

[1] School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China
lizd@cqupt.edu.cn
[2] Key Laboratory of Optical Communication and Networks, Chongqing, China
[3] Key Laboratory of Ubiquitous Sensing and Networking, Chongqing, China
[4] Electrical and Computer Engineering Department, University of Massachusetts Dartmouth, Dartmouth, USA

Abstract. How to cache appropriable contents for users from huge amount of candidates is a challenge in edge caching network. To address this challenge, this paper studies an edge caching scheme based on user interest, where an interest extraction and evolution network is developed. Specifically, the input features are first classified and embedding. The user interest is then mined and modeled according to the user historical behaviors with the gated recurrent unit network. Thereafter, the user interest evolution process is studied by analyzing the impact of the previous interests on the current interest through an attention mechanism. The group interest model is further studied by merging user interest evolution and social relationships among contents, based on which edge caching scheme is obtained. The effectiveness of the proposed scheme is finally validated by extensive experiments with a real-world dataset. The analysis in this paper sheds new light on edge content caching from user interest evolution perspective.

Keywords: Edge caching · User interest evolution · Group interest · Cache hit rate · User hit rate

This work was supported in part by the National Natural Science Foundation of China under grants 61901078, 61871062 and 61771082, and in part by Natural Science Foundation of Chongqing under grant cstc2020jcyj-zdxmX0024, and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under grant KJQN201900609, and in part by University Innovation Research Group of Chongqing under grant CXQT20017.

J. Xiong et al. (Eds.): MobiMedia 2021, LNICST 394, pp. 107–122, 2021.
https://doi.org/10.1007/978-3-030-89814-4_8

1 Introduction

With the widespread the rapid development of communication techniques, watching videos online has become one of the most popular entertainments [1–3]. However, due to the huge population and variety of contents, traditional cloud-based caching may not meet the service of quality for users. As a result, edge caching technique is developed to enable popular contents cached at an edge server close to users [4,5].

Due to limited caching capacity, how to cache contents efficiently is a fundamental issue of edge caching. As a consensus, user requests are highly related to user interest. Hence, an accurate prediction on user interest is able to guarantee an effective content caching decision, such that reducing the waiting time of content requests. User interest, however, is difficult to predict due to its time-varying property. For instance, a user is interested in the comedy movie at time t_1 while his/her interest tends towards the romance one at time t_2, as shown in Fig. 1. Consequently, dynamic user interest mining is critical in for edge caching.

In the literature, edge caching are usually based on content popularity with assumption of Zipf distribution [6]. In [7], an integrated content distribution framework was proposed cache contents based on contextual information. In [8], deep learning was applied to predict and cache popular contents in the edge server. In [9], content offloading and caching were jointly optimized by using reinforcement learning. Works [6–9] mainly designed caching schemes in terms of communication optimization, while user interest was not taken into account. Recently, some researches try to design caching policy based on recommendation system approach [10,11]. Such approach performs well in content feature characterization and user interest mining [12–16]. In [14], classical matrix factorization of the recommender system and convolutional neural network were combined to predict user interest, based on which contents are selected to cache at the edge server. In [15], the idea of soft cache was proposed to improve the cache hit rate. In detail, similar contents were recommended to a user before relaying the requested contents from the cloud if the user request was not meet at the edge server. To characterize the temporal features of content popularity, work [16] proposed a advanced long and short-term memory network. However, the above mentioned works only studied the caching performance from the edge server, while user hit rate was omitted. Additionally, how to employ the time-varying user interest to improve the edge caching performance is still an open issue.

Motivated by this, we study a hybrid human-artificial intelligence enabled edge caching scheme in a edge caching network. A cloud-edge-end collaboration framework is first constructed to improve the cache hit rate and user hit rate at the same time. An interest extraction and evolution network is then proposed the analyze the individual interest on the contents. Specifically, the gated recurrent unit (GRU) is employed to capture the user interest over a period of time. Additionally, attention mechanism is applied to analyze the impacts of user historical behaviors on the future behavior. Moreover, an intelligent based content prediction scheme and a socially-aware content prediction scheme are merged into a group interest model, based on which, the edge caching can be finally

decided. The effectiveness of the proposed caching scheme is further validated compared to four baseline schemes.

The rest of the paper is organized as follows. Section 2 introduces the system model as well as performance metrics. Section 3 proposes the IEEN model for user interest mining. In Sect. 4, content caching scheme is studied. In Sect. 5, extensive experiment results are presented and discussed. Finally, Sect. 6 concludes the paper.

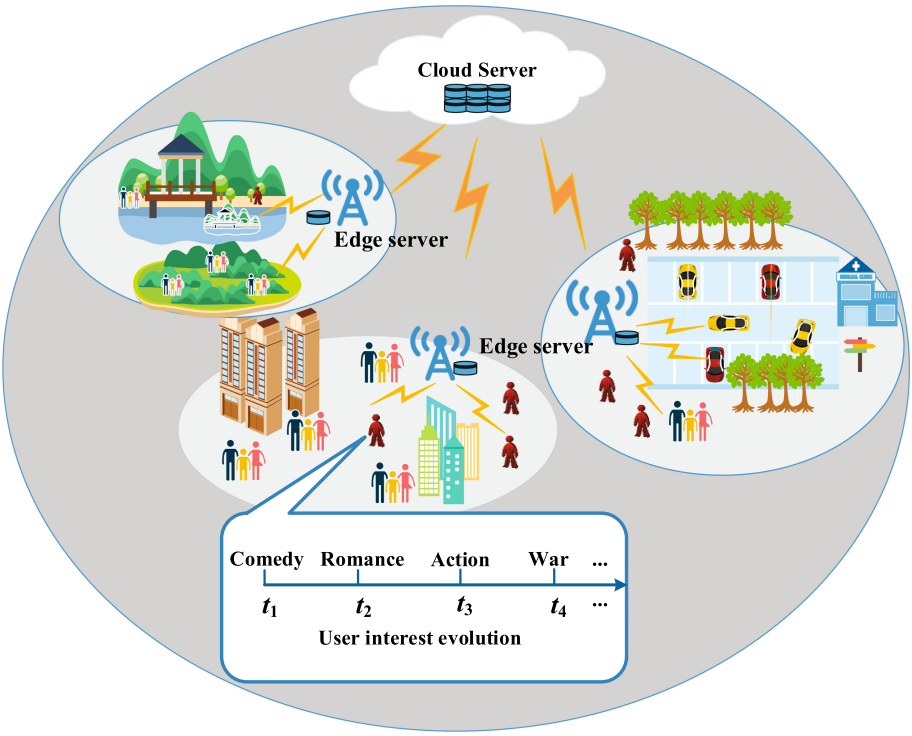

Fig. 1. Network model.

2 System Model

As depicted in Fig. 1, the considered network aims to provide services to users located in different types of areas wherein the users may have heterogeneous interests. A cloud server is deployed and assumed to store all of the contents that may be requested by the users. The contents stored in the cloud is denoted by set $\mathcal{F} = \{f_1, f_2, \cdots, f_C\}$, where f_c denotes the c-th content and C denotes the number of contents. Besides, the edge servers are deployed in each area to cache contents from the cloud server. In this regard, user requests can be responded more quickly if they are hit by the edge servers.

In this paper, we propose a cloud-edge-end collaboration policy to deal with such interest evolution problem and improve the edge caching performance. As

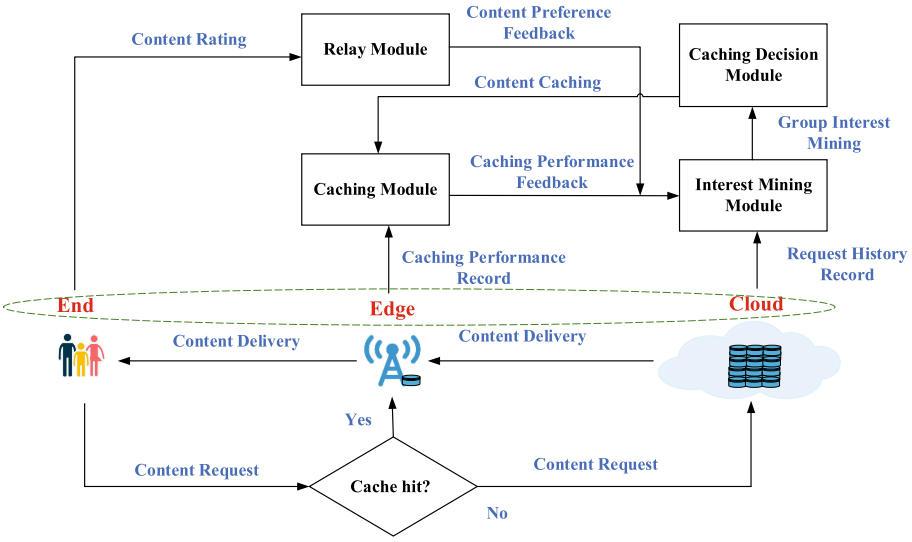

Fig. 2. Cloud-edge-end collaboration policy for content caching.

depicted in Fig. 2, the end users are responsible for sending to the edge servers their content requests that can represent their current interests. After the serving is completed, the users are encouraged to express their preference on the service content through rating. Each edge server is configured with a caching module and a relay module. The caching module is responsible for caching and delivering contents between the cloud server and the users. In addition, the caching module records the current caching performance and sends feedback to help the cloud server to adjust caching decision. The relay module is responsible for collecting user request history and content preference and forwarding them to the cloud server. Based on the feedbacks from the edge server, the cloud server mines the law of individual interest evolution and designs group interest model to make caching decision.

In practical, an edge caching decision duration should be larger than the reading or watching time of a content, since a user has a higher probability to stay within the service coverage of an edge sever. Consequently, a user may request multiple contents and a content may be requested by multiple users within an edge caching decision duration. Without loss of generality, we focus on an edge server network with caching capacity Φ. The users within the service coverage of such edge server is denoted by set $\mathcal{U} = \{U_1, U_2, \cdots, U_N\}$, where N denotes the number of users. Besides, we use binary variable $\alpha_{n,c}$ to indicate the user request. If content f_c is requested by user U_n, $\alpha_{n,c} = 1$, otherwise $\alpha_{n,c} = 0$. Similarly, binary variable $\beta_c = 1$ indicates content f_c is cached by the edge server, and otherwise $\beta_c = 0$. From the perspective of service provider, an effective caching scheme should make sure 1) as many cached contents as possible can be requested by users, and 2) as many user requests as possible can

be hit at the same time, such that a high profit can be acquired. In this sense, we introduce two metrics to evaluate the caching performance, which are cache hit rate (CHR) and user hit rate (UHR).

The CHR is defined as the proportion of the cached contents that are requested by users during an edge caching decision duration, there holds

$$\mathrm{CHR} = \frac{\sum\limits_{f_c \in \mathcal{F}} \min\{\sum\limits_{U_n \in \mathcal{U}} \alpha_{n,c}, 1\}\beta_c}{\Phi}. \tag{1}$$

The UHR is defined as the proportion of users whose requests can be hit by the edge server, there holds

$$\mathrm{UHR} = \frac{\sum\limits_{U_n \in \mathcal{U}} \min\{\sum\limits_{f_c \in \mathcal{F}} \alpha_{n,c}\beta_c, 1\}}{\sum\limits_{U_n \in \mathcal{U}} \min\{\sum\limits_{f_c \in \mathcal{F}} \alpha_{n,c}, 1\}}. \tag{2}$$

Therefore, we can formulate two optimization problems about edge caching scheme in terms of CHR and UHR respectively.

$$\begin{aligned} \mathbf{P1} \quad & \max_{\boldsymbol{\beta}} \ \mathrm{CHR} = \frac{\sum\limits_{f_c \in \mathcal{F}} \min\{\sum\limits_{U_n \in \mathcal{U}} \alpha_{n,c}, 1\}\beta_c}{\Phi} \\ \text{s.t.} \quad & \mathrm{C1}: \sum_{f_m \in \mathcal{M}} \beta_m \leq \Phi \end{aligned} \tag{3}$$

$$\begin{aligned} \mathbf{P2} \quad & \max_{\boldsymbol{\beta}} \ \mathrm{UHR} = \frac{\sum\limits_{U_n \in \mathcal{U}} \min\{\sum\limits_{f_c \in \mathcal{F}} \alpha_{n,c}\beta_c, 1\}}{\sum\limits_{U_n \in \mathcal{U}} \min\{\sum\limits_{f_c \in \mathcal{F}} \alpha_{n,c}, 1\}} \\ \text{s.t.} \quad & \mathrm{C1}: \sum_{f_m \in \mathcal{M}} \beta_m \leq \Phi \end{aligned} \tag{4}$$

where $\boldsymbol{\beta} = \{\beta_1, \beta_2, \cdots, \beta_C\}$ denotes the content caching decision. In problems P1 and P2, constraint C1 means that the number of contents selected to cache should not exceed the edge caching capacity. As mentioned before, user interest is time-varying and cannot be characterized by a model-based approach. Thus, the user requests $\boldsymbol{\alpha} = \{\alpha_{n,c} : 1 \leq n \leq N, 1 \leq c \leq C\}$ cannot be ascertain exactly in advance, which means maximizing CHR or UHR is infeasible. Moreover, it is unknown that whether there exits conflict between CHR and UHR. In this paper, data-based idea is resorted to design a hybrid human-artificial intelligence enabled content caching scheme where user interest evolution and group interest are both taken into account.

3 Individual Interest Evolution

In this section, an interest extraction and evolution network (IEEN) is designed to analyze the interest evolution process. As depicted in Fig. 3, our proposed

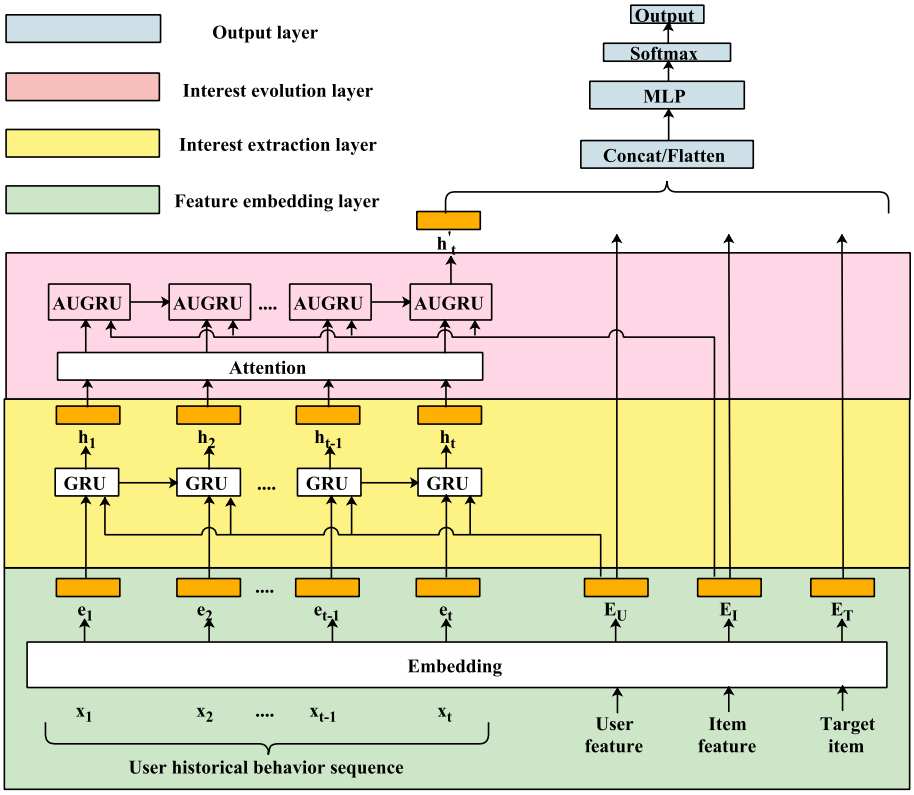

Fig. 3. Interest extraction and evolution network (IEEN).

IEEN model consists of four layers, i.e., feature embedding layer, interest extraction layer, interest evolution layer, and output layer. The feature embedding layer is responsible for converting the sparse high-dimensional input features into the dense low-dimensional ones, such that the features are more easily represented. The interest extraction layer is responsible for learning the behaviors of a user according to the input dense low-dimensional features. In particular, GRU is resorted to extract user interest from the historical behavior sequence preliminarily. The interest evolution layer is responsible for analyzing the relationship among the historical interests and then deducing the future interest for a user. In the output layer, user interest on a content is obtained through feature merging with a full-connective network.

3.1 Feature Embedding

In order to guarantee high prediction accuracy of user interest, the input features should be well represented by IEEN. In this paper, we choose four fields as the input of IEEN model, which are user behavior sequence, target item, item

feature, and user feature. The user behavior sequence includes a list of historical IDs of items browsed by the user, which can be expressed as:

$$\mathbf{X}_n = [x_1, x_2, x_3, ..., x_{t_0}, ..., x_t], \tag{5}$$

where $\mathbf{X}_n \in \mathbb{R}^{1\times t}$ represents the behavior sequence of the i-th user, x_{t_0} denotes the ID of the t_0-th item browsed by the user, and t denotes the length of the user behavior sequence.

Besides, target feature is defined as the items that are requested by the users, such as music, video and etc. Item feature usually includes item attributes, item category, description information, etc. User feature includes the user gender, age, occupation, residence information and etc. Among the item features and user features, item category, gender, age and etc. all belong to category features that can be expressed as sparse feature vectors with the one-hot encoding scheme. However, gradient update will be degraded if the sparse category features are directly sent to the interest extraction layer, which increases the model training complexity significantly. Hence, we apply feature embedding to convert the high-dimensional sparse vectors into low-dimensional dense vectors. Note that the user features, e.g., gender, age, and etc., are statistically stable within a large time scale, the model training time can consequently be reduced by pre-training the embedded user features. Moreover, feature embedding extracts the connection between different feature vectors, which enhances the model memory ability.

3.2 Interest Extraction

As the time-varying user behaviors can be represented by a sequence, we resort to the GRU model to extract the user interest from the user behavior sequence. Compared with recurrent neural network (RNN) and long short-term memory network (LSTM), GRU can retain more information of the long-term sequence. Additionally, the structure of the GRU model is simpler and the training speed is faster [17]. The GRU model consists of the update gate and reset gate. The update gate decides how much of the previous information in the user behavior sequence to remember and to be passed along to the future. The reset gate determines how much of the previous information should be omitted. The GRU model can be expressed as follows:

$$\mathbf{z_t} = \sigma\left(\mathbf{W}^z(\mathbf{e}_t + \mathbf{E}_U) + \mathbf{N}^z\mathbf{h}_{t-1} + \mathbf{b}^z\right), \tag{6}$$

$$\mathbf{r_t} = \sigma\left(\mathbf{W}^r(\mathbf{e}_t + \mathbf{E}_U) + \mathbf{N}^r\mathbf{h}_{t-1} + \mathbf{b}^{\mathrm{r}}\right), \tag{7}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}^h(\mathbf{e}_t + \mathbf{E}_U) + \mathbf{N}^h(\mathbf{r_t} \odot \mathbf{h}_{t-1}) + \mathbf{b}^h\right), \tag{8}$$

$$\mathbf{h}_t = (\mathbf{1} - \mathbf{z_t}) \odot \mathbf{h}_{t-1} + \mathbf{z_t} \odot \tilde{\mathbf{h}}_t, \tag{9}$$

where $\mathbf{z_t}$, $\mathbf{r_t}$, $\tilde{\mathbf{h}}_t$, $\mathbf{h}_t$ denotes the update gate, the reset gate, candidate hidden state vector, the hidden state vector of the current time step, respectively, σ is the sigmoid activation function, $\mathbf{W}^z, \mathbf{W}^r, \mathbf{W}^h$ and $\mathbf{N}^z, \mathbf{N}^r, \mathbf{N}^h$ are the training parameters, $\mathbf{e}(t)$ denotes the embedding vector of x_t, $\mathbf{E}_U$ denotes the embedding of user feature, $\mathbf{b}^z, \mathbf{b}^r, \mathbf{b}^h$ denote biases, $\odot$ denotes element-wise multiplication.

3.3 User Interest Evolution

As user interest is time-varying and highly related among different time, historical user interest should be considered while predicting the future user interest. Intuitively, different previous interests play different roles in the final prediction. In order to find out the impacts of different previous interests on the interest evolution process, an attention mechanism is introduced. The attention mechanism can automatically learn weights and extract important user interest features from a long user behavior sequence. The output of the attention network holds as [18]

$$\begin{aligned} \mathbf{a}(t) &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V} \\ &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{h}_t^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{h}_t \\ &= \text{softmax}\left(\frac{\mathbf{Q}\left\|\mathbf{h}_t\right\|}{\sqrt{d_k}}\right) \end{aligned} \tag{10}$$

where $\mathbf{Q} = [\mathbf{h}_1, \mathbf{h}_2, ...\mathbf{h}_{t_0}, ...\mathbf{h}_t]$ denotes the output sequence of the interest extraction layer, $\mathbf{h}_{t_0}$ denotes the output of t_0-th GRU in the interest extraction layer, $\mathbf{K} = \mathbf{V}$ denote the output of t-th GRU in the interest extraction layer, d_k represents the normalization parameter. From (10), the attention weight of use interest can be obtained. Note that in the long-term sequence, there may exist interest shift caused by other reasons, the attention network can ignore these small deviations and mines the main characteristics from the long-term sequence.

As user interest evolutes with time, we thus apply GRU with attention update gate(AUGRU) to learn the relationship between user interest of different time. The input of the GRU model includes the output attention weight $\mathbf{a}(t)$ and the item features $\mathbf{E}_I$. We use attention weight $\mathbf{a}(t)$ to update the gate of GRU.

$$\mathbf{z_t^{'}} = \mathbf{a}\,(\mathbf{t}) * \mathbf{z_t} \tag{11}$$

$$\mathbf{h_t^{'}} = \left(\mathbf{1} - \mathbf{z_t^{'}}\right) \odot \mathbf{h_{t-1}^{'}} + \mathbf{z_t^{'}} \odot \tilde{\mathbf{h}}_\mathbf{t}^{'} \tag{12}$$

The hidden state vector of the current time step $\mathbf{h}_t'$ can be trained and obtained similarly as the one in interest extraction layer.

3.4 Output Layer

In the output layer, the user preference of a given content can be analyzed by a full-connective network with the predicted future individual interest vector $\mathbf{h}_t'$, the user feature $\mathbf{E}_U$, the item features $\mathbf{E}_I$ and the target item information $\mathbf{E}_T$. The user preference on a content is formulated as a classification problem. Hence, cross-entropy loss function is employed for training the whole IEEN.

$$\text{Loss} = -\frac{1}{M}\sum_{(\mathbf{v},y)\in\mathbf{D}} (y \log \text{IEEN}(\mathbf{v}) + (1-y)\log(1-\text{IEEN}(\mathbf{v})), \tag{13}$$

where $\mathbf{D}$ denotes the training data set, M denotes the amount of training data, $\mathbf{v}$ denotes the input of IEEN including $\mathbf{X}_n$, $\mathbf{E}_U$, $\mathbf{E}_I$ and $\mathbf{E}_T$, IEEN$(\cdot)$ denotes the output of IEEN, $y \in \{0, 1\}$ denotes the label of content preference.

4 Hybrid Human-Artificial Intelligence Caching Scheme

For a real-world scenario, caching contents in terms of individual user interest directly may degrade the UHR, since a large amount of contents may be cached for a small amount of users that have high interest on lots of contents. As a result, edge caching should consider the common interest of each user. However, as common interest may not be the strongest interest of a user, caching contents in terms of common interest may lead to the CHR degradation if individual interest is omitted. In this section, we propose a hybrid human-artificial intelligence based group interest model which can guarantee high UHR and CHR at them same time.

Firstly, the obtained IEEN model is applied to predict the interest of each user on each content $\{p_{n,c} : 1 \leq n \leq N, 1 \leq c \leq C\}$. Then, interest evolutionary attention is resorted to quantify the group interest on each content, there holds

$$p_c^{\text{att}} = \text{softmax}(\frac{\sum_{t_0=1}^{t}\sum_{i=1}^{N} a_n(t_0) \odot p_{n,c}}{N}), \tag{14}$$

According to (14), we can sort the contents in a descend order in terms of p_c^{att}. The attention-based content set is denoted by $\mathcal{F}^{\text{att}}$.

In addition to the attention based approach that scores contents through user interest, the contents can also be scored based on the social relationship among contents. Specifically, association analysis method is applied to achieve the frequent itemsets. Let $\mathcal{X} \subseteq \mathcal{F}$ and $\mathcal{Y} \subseteq \mathcal{F}$ denote two different sets of the contents that are watched by users before. The support of $\mathcal{X}$ to $\mathcal{Y}$ can be represented as

$$\text{Support}(\mathcal{X} \rightarrow \mathcal{Y}) = \frac{||\mathcal{X} \cup \mathcal{Y}||}{N}, \tag{15}$$

where $||X \cup Y||$ denotes the number of users that watched the all the contents from set $\mathcal{X} \cup \mathcal{Y}$ before. By setting the support threshold, the frequent itemsets of items can be obtained as $\mathcal{Z}$. The prediction value of the new content f_c can be characterized with the confidence between $\mathcal{Z}$ and f_c, i.e.,

$$p_c^{\text{aa}} = \text{Confidence}(\mathcal{Z} \rightarrow f_c) = \frac{\text{Support}(\mathcal{Z} \cup f_c)}{\text{Support}(\mathcal{Z})}. \tag{16}$$

According to (16), we can sort the contents in a descend order in terms of p_c^{aa}. The content set based on association analysis is denoted by $\mathcal{F}^{\text{aa}}$.

By merging the content set obtained based on user interest and that based on social relationship of contents, the content caching scheme $\boldsymbol{\beta}$ can be obtained as

$$\beta_c = \begin{cases} 1, & \text{if } f_c \text{ belongs to the } m\text{-th } (m \leq \Phi) \text{ element of } (\mathcal{F}^{\text{att}} \cap \mathcal{F}^{\text{aa}}) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

5 Experiment Results

In this section, we carry out extensive experiments to validate the effectiveness of the proposed caching scheme. In particular, a well-known real-world dataset, i.e., MovieLens, is applied for experiments [19]. The dataset includes more than 1 million iteration records between more than 6,000 users and 3,000 movies. Additionally, the proposed caching scheme is compared with four caching schemes in terms of cache hit rate and user hit rate. The four baseline schemes are introduced as follows.

- Caching based on attention weight (IEEN-A): Individual interest is first predicted by IEEN, based on which contents are cached based on the attention weights. Note that in this scheme, social relationship among contents are not considered.
- Caching based on association analysis (AA): In this scheme, contents are cached based on frequent itemsets.
- Cache based on popularity: In this scheme, the most watched contents are selected to cache.
- Cache based on user interest: Individual interest values are first predicted by IEEN, and then the contents with the highest scores are selected to cache.

In the data processing stage, the category features, such as gender, age, occupation, etc., are one-hot coded. For continuous features, we normalize them to speedup the model convergence. In order to verify the accuracy of the obtained IEEN model, the contents of each user are sorted by timestamp, and the last 10 contents of each user is selected as the test set. The rest of the data is used as the training set.

From the dataset, the ratings of movies from different users are divided into 5 levels with values $\{1, 2, 3, 4, 5\}$. The user distribution at each rating level is depicted in Fig. 4. It is observed that about half of the movies are rated from 1 to 3, and the remaining half are rated from 4 to 5. Hence, we can model the data label with binary value. In specific, if the rating of a content from a user is 4 or 5, the data label is considered to be positive, i.e., the user is interested in such content. Otherwise, the data label is considered to be negative, i.e., the user is not interested in such content. Based on the dataset and corresponding labels, the user interest evolution prediction can be carried out by the proposed IEEN model. The prediction results of a random selected user is depicted as Fig. 5. It is verified that the predicted interest at different time coincides with the positive labels.

Figure 6 depicts the cache hit rate varying with edge caching capacity under the scenario with 1000 users. It is observed that the proposed scheme outperforms the other baseline schemes especially when edge capacity is small. Hence, it is

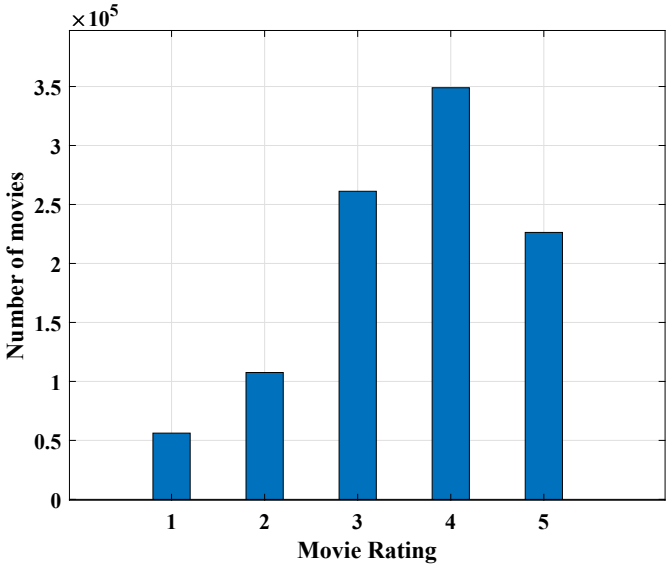

Fig. 4. Distribution of movie ratings in MovieLens dataset.

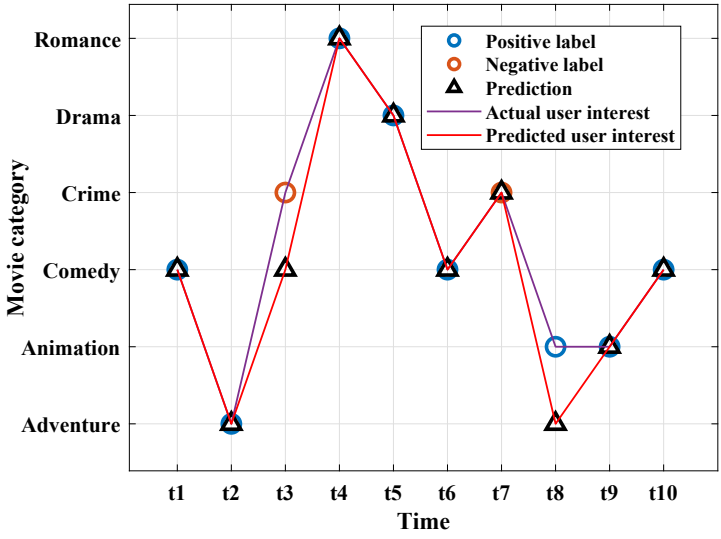

Fig. 5. User interest evolution process.

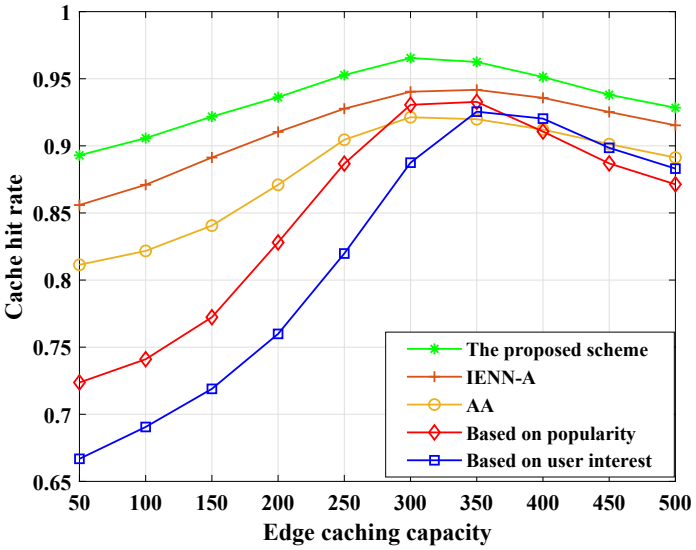

Fig. 6. Cache hit rate vs. edge caching capacity.

verified that merging user interest and social relationship of contents is able to improve the cache hit rate. Moreover, it is also found that there exists optimal caching capacity for each scheme, beyond which, more and more contents that are not interesting for users are cached. This phenomenon implies that caching capacity should be carefully designed from the profit view of point.

Figure 7 depicts the impact of cache capacity on user hit rate under the scenario with 1000 users. It is observed that the proposed scheme can guarantee more users' requests than the other baseline schemes. Additionally, user hit rate increases with the caching capacity when caching capacity is small. This is because a larger caching capacity can cache more contents that are interesting for some users but the interest values are smaller than that from other users. Additionally, it is found that when the caching capacity is large enough (e.g., $\Phi = 400$ for the proposed scheme), the user hit rate becomes invariant, which implies the system is statistically stable in this case.

Figure 8 depicts the impact of number of users on the cache hit rate under the scenario with caching capacity $\Phi = 300$. It is observed that the cache hit rate increases with the number of users. Besides, the proposed scheme achieves highest cache hit rate compared to the other four schemes especially when the number of users is small. In contrast, the scheme only based on individual interest suffers the worst cache hit rate. Such observations validate that group interest is useful in content caching and the proposed scheme is able to characterize the group interest accurately.

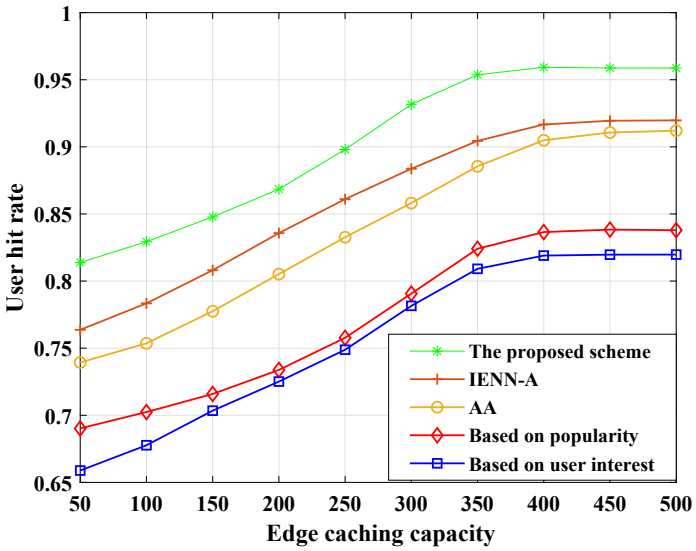

Fig. 7. User hit rate vs. edge caching capacity.

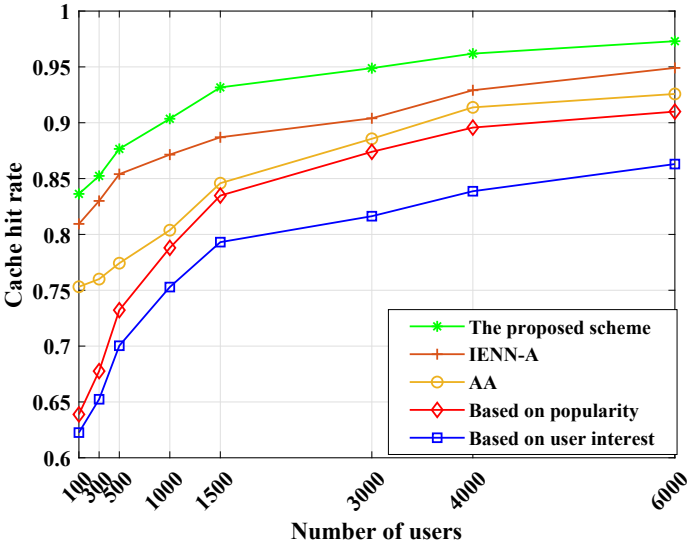

Fig. 8. Cache hit rate vs. the number of users.

Figure 9 depicts the user hit rate varying with the number of users under the scenario with caching capacity $\Phi = 300$. It is observed that as the number of users increases, the user hit rate first increases and then decreases. This is because when the number of users is small, user interest is hard to characterize. At this

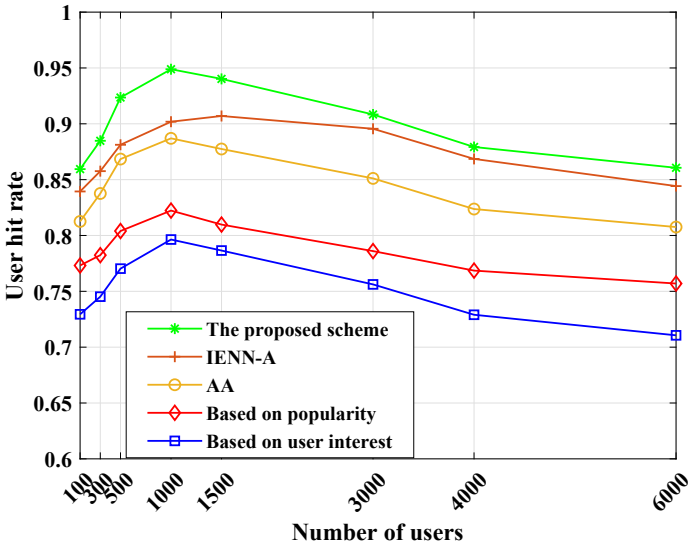

Fig. 9. User hit rate vs. the number of users.

case, increasing users is helpful to capture the common interest of users. However, as the number of users is large, increasing users may bring more interference on common interest capture, which degrades the user hit rate.

Figure 10 depicts the impact of historical behavior sequence length on the cache hit rate. The longer user historical behavior sequence are available, the

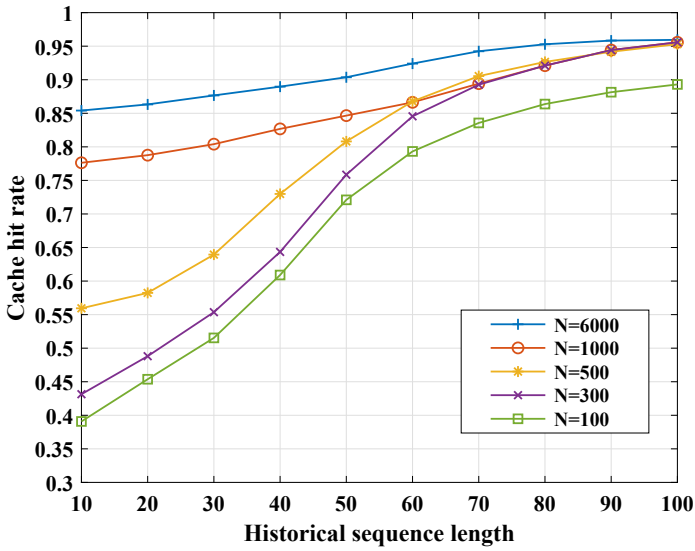

Fig. 10. Cache hit rate v.s. historical sequence length

higher cache hit rate can be obtained, since in this case the user interest evolution process are more dominant. Additionally, the scenario with more users can guarantee a higher cache hit rate. This is because increasing users can enriches features for model training.

6 Conclusions

In this paper, a hybrid human-artificial intelligence enabled edge caching scheme was studied in an edge caching network. The cache hit rate and user hit rate are first modeled. An interest extraction and evolution network was then constructed with consideration of user historical behaviors, content features and user features. Furthermore, a group interest model was proposed by merging weights from the intelligent attention approach and socially association analysis approach. Experiments revealed the impacts of edge caching capacity, the number of users as well as historical sequence length on the caching performance in terms of cache hit rate and user hit rate. Additionally, experiments validated that the proposed scheme performed better than the other baseline schemes with the help of a real-world dataset.

References

1. Sun, Y., Liu, J., Wang, J., Cao, Y., Kato, N.: When machine learning meets privacy in 6G: a survey. IEEE Commun. Surv. Tutorials **22**(4), 2694–2724 (2020)
2. Xiong, J., et al.: Enhancing privacy and availability for data clustering in intelligent electrical service of IoT. IEEE Internet Things J. **6**(2), 1530–1540 (2019)
3. Xiong, J., Chen, X., Yang, Q., Chen, L., Yao, Z.: A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing. IEEE Trans. Netw. Sci. Eng. **7**(4), 2347–2360 (2020)
4. Abuhadra, R., Hamdaoui, B.: Proactive in-network caching for mobile on-demand video streaming. In: IEEE International Conference on Communications (ICC), Kansas City, pp. 1–6 (2018)
5. Doan, T.V., Pajevic, L., Bajpai, V., Ott, J.: Tracing the path to youtube: a quantification of path lengths and latencies toward content caches. IEEE Commun. Mag. **57**(1), 80–86 (2019)
6. Jiang, W., Feng, G., Qin, S.: Optimal cooperative content caching and delivery policy for heterogeneous cellular networks. IEEE Trans. Mob. Comput. **16**(5), 1382–1393 (2017)
7. Xing, H., Song, W.: Collaborative content distribution in 5G mobile networks with edge caching. In: IEEE International Conference on Communications (ICC), Shanghai, China, pp. 1–6 (2019)
8. Liu, W., Zhang, J., Liang, Z., Peng, L., Cai, J.: Content popularity prediction and caching for ICN: a deep learning approach with SDN. IEEE Access **6**, 5075–5089 (2018)
9. Yang, Z., Liu, Y., Chen, Y., Tyson, G.: Deep reinforcement learning in cache-aided MEC networks. In: IEEE International Conference on Communications (ICC), Shanghai, China, pp. 1–6 (2019)

10. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: a survey and new perspectives. ACM Comput. Surv. (CSUR) **52**(1), 1–38 (2019)
11. Dara, S., Chowdary, C.R., Kumar, C.: A survey on group recommender systems. J. Intell. Inf. Syst. **54**(2), 271–295 (2019)
12. Wang, S., Cao, L., Wang, Y.: A survey on session-based recommender systems. arXiv preprint arXiv:1406.1078 (2019)
13. Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q., Orgun, M.: Sequential recommender systems challenges, progress and prospects. arXiv preprint arXiv:2001.04830 (2019)
14. Yin, Y., Chen, L., Xu, Y., Wan, J., Zhang, H., Mai, Z.: QoS prediction for service recommendation with deep feature learning in edge computing environment. Mob. Netw. Appl. 1–11 (2019)
15. Costantini, M., Spyropoulos, T., Giannakas, T., Sermpezis, P.: Approximation guarantees for the joint optimization of caching and recommendation. In: ICC 2020–2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, pp. 1–7 (2020)
16. Zhang, C., et al.: Toward edge-assisted video content intelligent caching with long short-term memory learning. IEEE Access **7**, 152832–152846 (2019)
17. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
18. Vaswani, A., et al.: Attention is all you need. Presented at the advances in neural information processing systems. (2017)
19. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. **5**(4), 1–19 (2015)