Predicting the Contents popularity with the help of Big Data and In-network Caching in Information Centric Wireless Networking

Kyi Thar, Choong Seon Hong
Department of Computer Science and Engineering, Kyung Hee University,
446-701, Republic of Korea.
Email:{kyithar, cshong}@khu.ac.kr

***Abstract*—Currently, the mobile data traffic is tremendously increasing and the solution is needed to handle this increasing demands. There are several solutions to solve this issue. Among them, the Information-Centric Networking (ICN) is one of the most promising solutions, where ICN reduces the network traffic by caching the contents temporarily. So, ICN-enabled Base Station(BS)s can provide these cached contents to the user, instead of retrieving from the original server. Also, ICN-enabled BSs aggregate the same content requests from different users at the Base Stations, instead of forwarding every request to the original server. Thus, the decision to cache the contents is important and it should be effectively reduced the data traffic. Even tough, there are several caching decision algorithms already existed, we can still improve the cache decision for large scale network with the help of big data (the information stored by the operators). Therefore, in this paper, we proposed a caching scheme which is working together with the big data platform to predict the popular contents and cache these contents at the BSs efficiently. We used Collaborative Filtering (CF) to fill up the sparse matrix with prediction values. Then we predict the popular contents list by using the Machine Learning (ML) technique with the consideration of others context information. Depending on the recommend list, BSs store the contents and send back the feedback to the controller to update the loss function.**

***Index Terms*—Content Centric Networking, Big Data, In-Network Caching, Collaborative Filtering.**

I. Introduction

The mobile data traffic growth is exponentially increasing, especially on video traffic. In order to reduce network data traffic, the Information-Centric Networking (ICN) [1] is proposed. The basic concept of the ICN is that the routers or nodes store the contents temporarily and provide these contents to the users, instead of retrieving from the original server. Among the ICN architecture, the Content Centric Networking [2] architecture is popular. Therefore, for the 5G network, we employed the ICN concept at the Radio Access Network, where Base Stations (BS)s and Small Cell Base Station (SBS)s are attached with cache space. Therefore, ICN-enabled BSs and SBSs can store the contents temporarily and provide these contents to their users, instead of retrieving from the original server. In order to improve the performance of the ICN caching, we used the benefit of Big Data technology [3] [4] to predict the contents popularity, generate the recommended list to cache the contents for each BS and SBS.

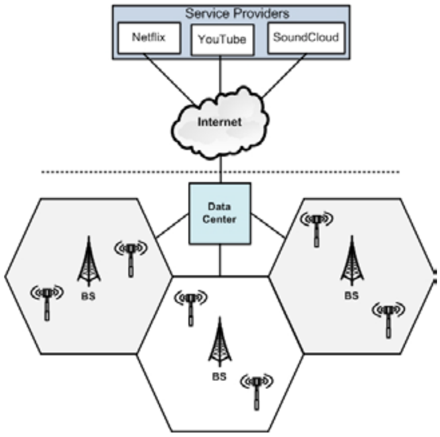

Fig. 1. System Model

Previously, the contents popularity is assumed as the Zipf distribution. Depending on the Zipf distribution, proposed a different kind of cache decision algorithm. For the current work, we consider predicting the content popularity and cache the contents depending on the popularity values by analyzing a large amount of user information. The challenges of this work are as follows. Matrix completion problem is NP-hard. It is difficult to solve overfitting and cold start problem, and popularity prediction mechanism with contextual information such as (event, the new movie or not).

II. System Model

The network model is shown in Fig. 1, where the content servers are located at the outside of the current Autonomous System (AS) and the controller is located at the Data Center. The complex calculation for predicting the content popularity and recommending the contents list to a store of each BS are done at the controller. The BSs and SBS are located inside the RAN, where BSs and SBs have attached with the cache space to temporary store the contents. All BSs and SBBs are connected with the core network/controller via the backhaul link. For the simplicity, we define the BSs and SBSs of the InP as BS$_j$, where $j = 1, ..., J$. There are total number of file F, where $F = 1, 2, .., f$. The cache capacity of the BS_j is denoted as $Z_j \in Z$, where Z is the total cache capacity of the whole network in Terabytes. The BS_j can not store the contents more than that it cache capacity, $\sum_{f=1}^{F} x_{jf}.s_f \leq Z_j, \forall j, f$, where x_{jf} is the binary variable and s_f is the size of the content f. The BS_j stores the content f when $x_{jf} = 1$, otherwise $x_{jf} = 0$. The users, those are connected with the BS_j are denoted as $u_j \in U$. a_{u_j} is the user association indicator, where $a_{u_j} = 1$ means that user u_j associates to BS_j, otherwise $a_{u_j} = 0$. Each user can only associates to only one BS, which is defined with $\sum_{j=1}^{J} a_{u_j} = 1, \forall u$.

The overview process of our proposed scheme is shown in Fig.2, which consists of five steps. 1) The first step is the information collecting stage or recommendation requesting stage. Every time t BSs and SBSs send their local information to the controller to get the recommendation list. By using these recommendation lists, BSs and SBSs make the cache decision. 2) In the second step, the controller constructs the sparse matrix. we used Collaborative filtering(CF) to estimate the missing values. 3) In the third step, we minimize the least square error of estimating the missing values by using the Alternating Least Square (ALS) method. Then, we get the non-sparse matrix with predicted/estimated values. 4) In the fourth step, controller filters and constructs the augmented popularity matrix with the contents popularity values from the non-sparse matrix. 5) In the fifth step, the controller generates an ordered recommendation list to store the contents for each BS and SBS.

III. Estimate the missing values

To estimate the missing values, we use CF, where CF systems can be categorized into two groups: memory-based and model-based methods. Memory-based methods simply memorize the rating matrix and issue recommendations based on the relationship between the queried user and item and the rest of the rating matrix. Model-based methods fit a parameterized model to the given rating matrix and then issue recommendations based on the fitted model. Memory-based methods store the rating matrix and estimate based on the relationship between user and item. Model-based methods construct the model with training data. Then predict the unseen ratings with that constructed model.

The base station-content matrix is denoted as $P = \{p_{ij}\}_{n_b x n_m}$, where p_{ij} is the popularity value of the content i at the BS_j, n_b is the number of BSs and n_m is the number of contents. Then, we estimate the missing values of p_{ij} in P with a low-rank approximation of the base station-content matrix P. This approach models both BSs and contents in a low dimensional feature space. Each BS and each content have a feature vector. Each popularity of a content of BS is modeled as the inner product of the corresponding BS and content vectors. Base Station and the content vector is B_{nxi}. Content popularity vector is C_{jxm}. Predicted popularity matrix is $\hat{P}_{nxm}$, where $\hat{p}_{ij} = b_i^T c_j$. Then we define the to minimize the root mean square error,

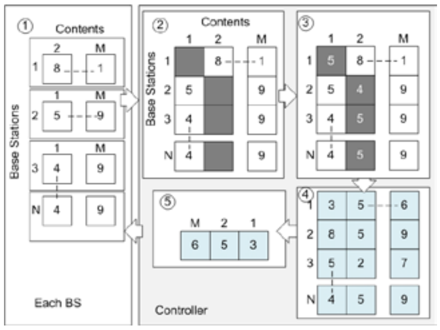

Fig. 2. Overview Process of proposed scheme

$$\min_{b^*, c^*} \sum_{ij \in K} (p_{ij} - \hat{p})^2 + \lambda(\|b_i\|^2 + \|c_i\|^2). \tag{1}$$

To solve the eq(1), we used ALS as in [5].

IV. Popularity Predicting

The controller predicts the content popularity depending on the context as in [6]. We consider, item fatigue, tiredness and temporal features, similarity or diversity features. From this step, we can get the ordered content popularity list. Then send the lists to each base station.

V. Conclusion

In this paper, we proposed a cache decision process with the help of Big Data, where do we use the Collaborative Filtering to estimate the missing popularity values and then we predict the content popularity by using context information. To estimate the missing popularity values, we applied Alternative Least Square and minimize the root square error. For the future work, we will run and test our proposed scheme at the SPARK environment.

References

[1] George Xylomenos, Christopher N Ververidis, Vasilios A Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V Katsaros, and George C Polyzos. A survey of information-centric networking research. *Communications Surveys & Tutorials, IEEE*, 16(2):1024–1049, 2014.

[2] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.

[3] Engin Zeydan, Ejder Baştuğ, Mehdi Bennis, Manhal Abdel Kader, Alper Karatepe, Ahmet Salih Er, and Mérouane Debbah. Big data caching for networking: Moving from cloud to edge. *arXiv preprint arXiv:1606.01581*, 2016.

[4] Ejder Baştuğ, Mehdi Bennis, Engin Zeydan, Manhal Abdel Kader, Ilyas Alper Karatepe, Ahmet Salih Er, and Mérouane Debbah. Big data meets telcos: A proactive caching perspective. *Journal of Communications and Networks*, 17(6):549–557, 2015.

[5] Dave Zachariah, Martin Sundin, Magnus Jansson, and Saikat Chatterjee. Alternating least-squares for low-rank matrix reconstruction. *IEEE Signal Processing Letters*, 19(4):231–234, 2012.

[6] Oren Sar Shalom, Noam Koenigstein, Ulrich Paquet, and Hastagiri P Vanchinathan. Beyond collaborative filtering: The list recommendation problem. In *Proceedings of the 25th International Conference on World Wide Web*, pages 63–72. International World Wide Web Conferences Steering Committee, 2016.