An Ensemble Learning based Mobile Recommender System for M2M service

Md. Golam Rabiul Alam, Chuan Pham, Tuan LeAnh, Choong Seon Hong
Department of Computer Engineering, Kyung Hee University
{robi, pchuan, latuan}@khu.ac.kr, cshong@khu.ac.kr

Abstract

Mobile recommender system offers personalized and context-sensitive recommendations. Due to the growing concern of IoT and M2M services, the recommender system for smart mobile devices is also drawing attentions as the contemporary research issue. Unlike the traditional recommender system, designing a mobile recommender system is complex because of the heterogeneity of data and required to consider the spatial and temporal co-relations. In this paper, Ensemble learning based context aware and personalized mobile recommender system is proposed for M2M content delivery service. The proposed Ensemble learning composed of collaborative filtering, k-NN and association rule based recommendation approaches. The prototype implementation and simulation results show lower error rate and higher user satisfactions in recommended contents.

1. Introduction

A recommender system is an information filtering system, which predicts the preferable contents, items or services for intended users. Therefore, the mobile recommender system offers recommendation services in ubiquitous environment. There is numerous recommendation options in M2M service delivery network [1] e.g. time and location based optimal route recommendation to vehicle drivers, spatiotemporal and interest based restaurant and hotel recommendation to travelers. The overall system for M2M service management architecture is presented in Fig, 1.

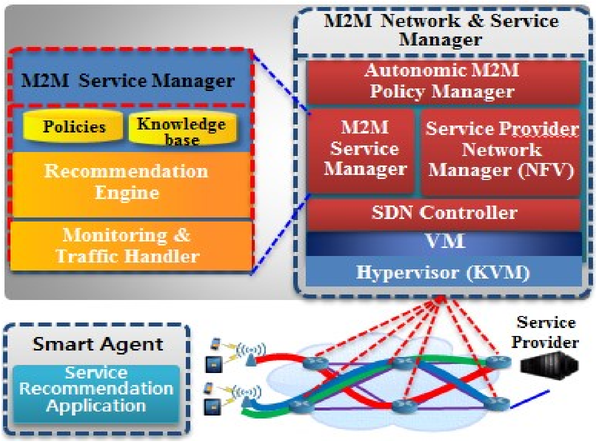

Figure 1. M2M network and service management framework

Smart-Agent enables user to subscribe to different personalized services and to customize choices of interest. Choices of interest mean environmental and usage information, according to which service-provider will recommend service to user.

Monitoring and Traffic Handler collect monitoring information and survey and real-time feedback from users. Service-usage information is collected by Monitoring System and is sent to Monitoring Handler. On the other hand, traffic handler works as an adapter to Service Providers Network Manager, which works to flow control in scalable software-defined network.

The service recommendation engine receives the real-time contextual data (e.g. content type) of smart agents from traffic monitoring unit. enables service-providers to infer service usage from user's. Based on the contextual data of monitored data traffic the ensemble algorithm based service recommendation engine recommends contents to the smart agent.

2. M2M service Recommender System

The proposed ensemble recommendation method is a hybrid recommendation policy, which combines the results of k-Nearest Neighbor (kNN) recommendation [2], Association Rule (AR) recommendation [3] and Collaborative Filtering (CF) recommendation [4] as shown in Fig. 2. The ensemble learning is designed through majority voting.

In general collaborative filtering based recommendation system the items are recommended by predicting the utility of items for a user based on the items previously rated by other like-minded users. To build such system we don't need to know the feature (i.e. x) values in advance to train the system. Both of the feature values (x) and learning parameter (θ, the pattern of COI) will be learned during the training of the system through optimizing equation (1). Where, $y^{(i,j)}$ is the rating by user *j* on item *i*; N_u and N_m is the number of user and items; λ is the regularization parameter.

$$\min_{\substack{x^{(1)},x^{(2)},\ldots,x^{(N_m)}\\ \theta^{(1)},\theta^{(2)},\ldots,\theta^{(N_u)}}} \frac{1}{2}\sum_{(i,j):r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{N_u}\sum_{k=1}^{n}\left(\theta_k^{(j)}\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{N_m}\sum_{k=1}^{n}\left(x_k^{(i)}\right)^2 \quad (1)$$

We found the optimized x and θ by applying gradient decent updates as in (2) and (3).

$$X_k^{(i)} := X_k^{(i)} - \alpha\left(\sum_{j:r(i,j)=1}\left(\left(\theta^{(j)}\right)^T X^{(i)} - y^{(i,j)}\right)\theta_k^{(j)} + \lambda X_k^{(i)}\right) \quad (2)$$

$$\theta_k^{(j)} := \theta_k^{(j)} - \alpha\left(\sum_{i:r(i,j)=1}\left(\left(\theta^{(j)}\right)^T X^{(i)} - y^{(i,j)}\right)X_k^{(i)} + \lambda\theta_k^{(j)}\right) \quad (3)$$

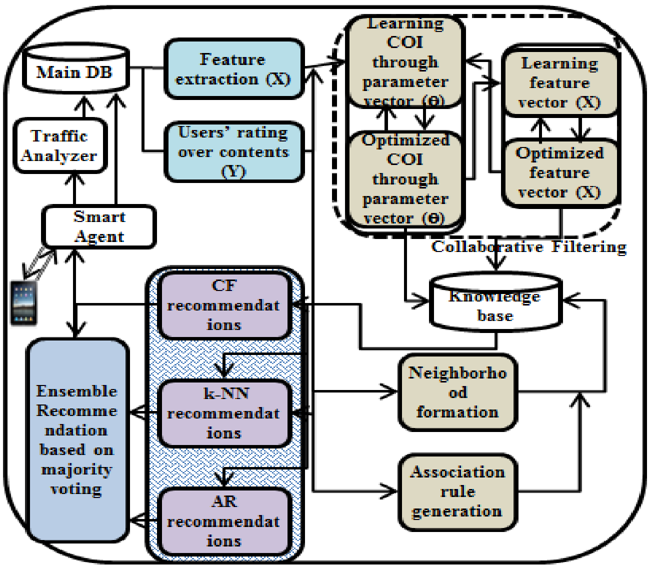

Fig. 2. M2M service Recommender

K-Nearest Neighbor (k-NN) recommender determines the similarity among the items according to their ratings. The similarity among the items is determined through Pearson's correlation. After finding k numbers of correlated items, it recommends feasible items according to the predicted ratings.

k-NN Neighborhood formation phase: The similarity of items *i* and *j* is computed as (4) using Pearson's correlation coefficient:

$$sim(i,j) = \frac{\sum_{u\in U}(r_{u,i}-\bar{r}_u)(r_{u,j}-\bar{r}_u)}{\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r}_u)^2}} \quad (4)$$

Here, $r_{u,i}$ is the rating of user *u* given to item *i*; and $r_{u,j}$ is the rating of user *u* given to item *j*; and $\bar{r}_u$ is the average ratings of user *u*.

k-NN Recommendation phase: After computing the similarity between items we select a set of *k* most similar items to the target item and generate a predicted value of user *u*'s rating using (5).

$$p(u,i) = \frac{\sum_{j\in J} r_{u,j}\times sim(i,j)}{\sum_{j\in J} sim(i,j)} \quad (5)$$

where j is the set of k similar items and $r_{u,j}$ is the rating of user *u* given to item *j*.

AR is the association rule based recommender: The items are recommended according to the usage pattern of different users. The frequent item set is determined based on the support (6) and confidence (7) values and using the popular *Apriori algorithm*.

$$Support(Item_X) = \frac{Cardinality\ (Item_X)}{|N_u|} \quad (6)$$

$$Confidence(Item_X \Rightarrow Item_y) = \frac{Support(Item_X U\ Item_Y)}{Support(Item_X)} \quad (7)$$

Ensemble collaborative filtering approach combines the recommendations of different legacy collaborative filtering approaches for better group decision making as well as of better recommender system. We apply *majority voting* to select the list of best recommended items following head to head consensus counting.

The pseudocode of ensemble recommendation system is presented in *Algorithm 1*.

Algorithm 1: Ensemble-recommendation ()

Get the set of items recommended by k-NN-recommender,
    P = k-NN_recommender()
Get the set of items recommended by AR-recommender,
    R = AR_recommender()
Get the set of items recommended by CF- recommender,
    T= CF_recommender()
$Q\{\, q_1, q_2, \cdots, q_p \,\} = P \cup R \cup T$
Initialize *recommend*
Do
For each s_i the member of different items set S i.e. $\{s_1, s_2, \cdots, s_M\}$
$C_{si}=0$
For each q_j of the element of set $Q\{\, q_1, q_2, \cdots, q_p \,\}$
    If $(s_i == q_j)$
    $C_{si}=C_{si}+1$
    End If
  End for
$|s_i| = C_{si}$
End for
$K = \arg\max_{s_i} |s_i|$
$I = I \cup K$
$S = S \setminus K$
recommend=*recommend*-1;
While (*recommend*)
Return I

---

4. Performance Evaluation

We developed a test bed to evaluate the performance of the proposed recommender system. In our test environment, the traffic handler captures the traffic from the contextual data of the smart agent and also the feedback from the agent.

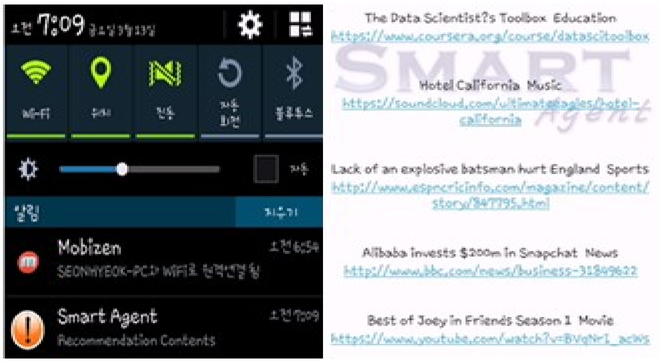

Fig. 3. Smart agent prototype with recommended contents

The traffic handler module tracks the traffic from the smart agent and stores the captured data to the database. Based on the user data and users current contextual data from the environment e.g. time, geo-location the ensemble learning algorithm sets the policy for the recommended contents which will be provided to the smart agent.

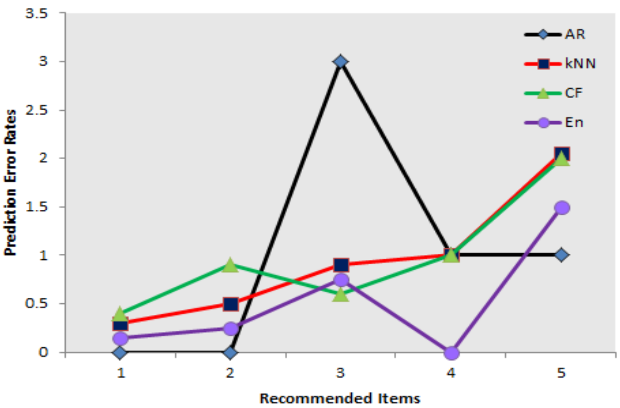

Fig. 4. Prediction error rates of different recommenders

The smart agent can also send the feedback through the smart client application which is used by the M2M service manager of cloud for generating the recommended content and service list as in Fig. 3.

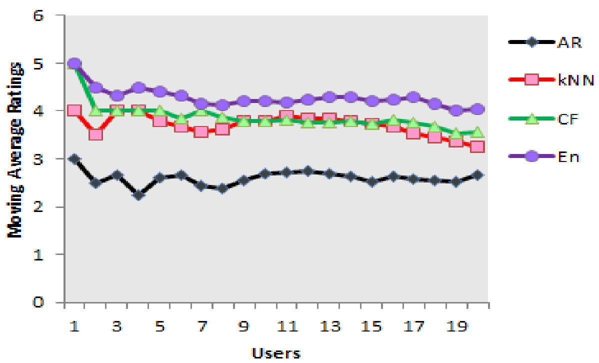

Fig. 5. Moving average user satisfaction ratings of different recommender systems

We also used MATLAB to study performance of the proposed Ensemble (En) recommender system in compare to state-of-the-art recommender system. Fig. 4 shows the prediction error rate of AR, kNN, CF and En are 29.41%, 27.94%, 28.82% and 15.58% respectively. We also studied user satisfaction of 20 users on recommended 5 contents by different recommender systems. We found higher user satisfaction on the proposed Ensemble recommender system as shown in Fig. 5.

3. Conclusion

Unlike the traditional recommender system the proposed mobile recommender system is personalized and context-aware. The smart agent application received several types of recommended content in a single platform based on user's interest and context. Although we tested it in M2M content delivery service perspective, the proposed generic recommender system is usable in some other M2M service recommendations. The proposed Ensemble mobile recommender system achieved lower prediction error and higher user satisfaction in the developed M2M network and service management framework.

Acknowledgement

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-(H8501-15-1015)) supervised by the IITP(Institute for Information & communications Technology Promotion). *Dr. CS Hong is the corresponding author.

References

[1] Surobhi, N. A., & Jamalipour, A. "A context-aware M2M-based middleware for service selection in mobile ad-hoc networks." *Parallel and Distributed Systems, IEEE Transactions on*, 25(12), pp. 3056-3065, 2014.

[2] Zhu, Q, Zhao L, Haohong W, Yimin Y, and Mei-Ling S. "Multimodal sparse linear integration for content-based item recommendation." *In Multimedia (ISM), 2013 IEEE International Symposium on*, pp. 187-194. IEEE, 2013.

[3] Lee, Seonah, Sungwon Kang, Sunghun Kim, and Matt Staats. "The Impact of View Histories on Edit Recommendations." *Software Engineering, IEEE Transactions on*, vol. 41, no. 3, pp:314-330, 2015

[4] Luo, X., Zhou, M., Xia, Y., & Zhu, Q. "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *Industrial Informatics, IEEE Transactions on*, 10(2), pp. 1273-1284, 2014.