LETTER

An Approach to Effective Recommendation Considering User Preference and Diversity Simultaneously

Sang-Chul LEE†, *Nonmember*, Sang-Wook KIM††a), *Member*, Sunju PARK†††, *and* Dong-Kyu CHAE††, *Nonmembers*

SUMMARY This paper addresses recommendation diversification. Existing diversification methods have difficulty in dealing with the tradeoff between accuracy and diversity. We point out the root of the problem in diversification methods and propose a novel method that can avoid the problem. Our method aims to find an optimal solution of the objective function that is carefully designed to consider user preference and the diversity among recommended items simultaneously. In addition, we propose an item clustering and a greedy approximation to achieve efficiency in recommendation.

key words: *diversification, e-commerce, recommender system*

1. Introduction

The *recommender systems* (*RS*) analyze each user's preference on items and provides her/him with a set of personalized items that s/he is likely to prefer most [7]. In most cases, however, the recommended items are similar to one another, which is called a *monotony* phenomenon [2], [4]. If all of the recommended items are from a core set that the user has purchased in the past, they may not be attractive to the user. Even if the items in the recommendation list are attractive to the user, he/she may choose only one in the list and ignore the rest. As a result, recommending the items only from the center of user preferences makes e-commerce sites waste the limited recommendation space that should be leveraged to stimulate the user's appetite to spend [5]. Therefore, it is important to take *diversity* of a recommendation into account.

A number of methods have been proposed for recommendation diversification (e.g., [1]–[4], [9]). To our knowledge, most methods employ the 2-step approach that considers user preference and diversity *independently*, as described in Fig. 1: (1) constituting a *candidate list* with regard to a user preference, and (2) making up a *recommendation list* by taking the diversity among the items into account. More specifically, the candidate list is found by selecting the top-m items according to their ratings predicted by any *collaborative filtering* (*CF*) method [8], [10]. Then, the recommendation list is derived by selecting from the candidate list a relatively small number of k ($m > k$)items which maximize the *diversity* of them. However, such a 2-step approach suffers from the difficulty in finding the optimal size (i.e., the number of included items) of a candidate list. If the size is small, most items in the candidate list may not be sufficiently diverse and thus the final recommendation list consists of items similar to one another. Enlarging the size of the candidate list may be a possible solution. However, this may result in the recommendation less relevant to a target user, since diversity is considered only when making up the final recommendation list. Moreover, the computation time would increase *exponentially* as the size of the candidate list grows, since it is an *NP-hard* problem to make up the recommendation list from the candidate list. In conclusion, the accuracy, diversity, and performance of the 2-step approach are highly governed by the size of the candidate list.

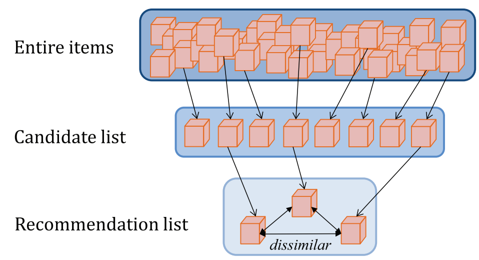

Fig. 1 A 2-step approach in diversification methods.

To avoid these problems, we propose a novel method to derive the final recommendation list where items are diverse as well as with high user preference. Rather than following the 2-step approach in previous methods, we find the items to be recommended with a *single step* to avoid choosing the size of the candidate list. To this end, we propose an objective function that measures the diversity and user preference of a given item set as a criterion for recommendation. Then, we formulate the recommendation as the problem of finding a set of k items that maximize the objective function. Indeed, our method does not need to decide the size of the candidate list, thereby avoiding the problem that the existing methods suffer from. Since finding the optimal solution is an NP-hard problem, we propose two strategies of (1) item clustering and (2) greedy-based approximation.

Manuscript received February 17, 2017.
Manuscript revised July 11, 2017.
Manuscript publicized September 28, 2017.
†The author is with Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea.
††The authors are with Department of Computer and Software, Hanyang University, Seoul, Korea.
†††The author is with School of Business, Yonsei University, Seoul, Korea.
a) E-mail: wook@hanyang.ac.kr
DOI: 10.1587/transinf.2017EDL8039

2. Proposed Method

2.1 Objective Function

Our objective function is designed to consider user preference and diversity *at the same time*. As the predictor of user preference, i.e., the rating of item I_i given by a target user, For predicting user preferences on unrated items, any CF algorithms can be applied here. We used the traditional user-based CF [6], the most popular method. In order to measure diversity, we compute the dissimilarities of additional information of items. Specifically, given two items I_i and I_j, we exploit their *external features* such as category, description, creators, names, and so on. For example, if I_i is the Star Wars movie, its set of external features A_{I_i} would be {SF, Action, George, Lucas, Star, Wars}, each of which is extracted from its genre, name of director, and title, respectively. Of course, we can add more the information into the set A_{I_i}, such as the names of actors and keywords of its synopsis. Using the *Jaccard coefficient*, we calculate the dissimilarity between I_i and I_j, $D(I_i, I_j)$, as shown in Eq. (1). The value of $D(I_i, I_j)$ ranges from 0 to 1: the higher the value, the more diverse I_i and I_j are.

$$D(I_i, I_j) = 1 - Jaccard(I_i, I_j) = 1 - \frac{\left|A_{I_i} \cap A_{I_j}\right|}{\left|A_{I_i} \cup A_{I_j}\right|} \qquad (1)$$

We define our objective function as shown in Eq. (2). Given a set R consisting of the pre-defined number of k items, the objective function $\Phi(R)$ is expressed by the following, where $P(I_i)$ represents the predicted preference normalized between 0 and 1, and diversity factor d, a tunable parameter (ranging from 0 to 1) that implies the degree of relative importance of diversity:

$$\Phi(R) = \frac{1-d}{|R|}\sum_{I_i \in R} P(I_i) + \frac{d}{|R| \times |R-1|}\sum_{I_i \in R}\sum_{I_j \in R} D(I_i, I_j) \qquad (2)$$

The first term in the objective function indicates the average user preference of items in R, and the second term does the average dissimilarity in R, i.e., the average dissimilarity of all possible pairs of the items in R. The two terms are linearly combined. The diversity factor d can adjust how diverse the final recommendation list should be.

Rather than using the weighted sum as in Eq. (2), we may define a different objective function, where the multiplication is used to combine the two factors as shown in Eq. (3):

$$\Phi(R) = \sum_{I_i \in R} P(I_i) \times \sum_{I_i \in R}\sum_{I_j \in R} D(I_i, I_j) \qquad (3)$$

2.2 Efficient Computation Strategies

It is practically infeasible to find an optimal solution of our objective functions, since (1) finding the optimal set R among n items is NP-hard which has a time complexity of $O(n!)$ and (2) the number of items, n, in e-commerce sites is very large in general. To achieve computational efficiency, we propose two additional strategies that can significantly reduce the number of possible item sets to be calculated.

2.2.1 Clustering Items

The main idea of this strategy is to substitute a bunch of similar items with an *item cluster*. In other words, our problem becomes to find k *clusters*, rather than k items. Then, we choose one item with the highest user preference from each cluster to make up the final recommendation list. This strategy not only reduces the number of comparisons, but also provides a more diverse recommendation list. Since n, the number of total items, is substituted with c, the total number of clusters ($c \ll n$), the number of comparisons with all possible solutions is reduced from ${}_{n}C_{k}$ to ${}_{c}C_{k}$. Moreover, the recommended items may be more diverse since similar items are likely to belong to the same cluster.

In order to group similar items, we employ a traditional distance-based clustering algorithm. Here, the well-known *k-medoid* [11] algorithm is used, but any distance-based clustering algorithms can be adopted. Once all items are clustered, we find a set of k clusters, which maximizes our objective function. At this step, (a) user preference on a cluster and (b) dissimilarity between two clusters should be measured. We define (a) as the average user preference on items included in the cluster and (b) as the average dissimilarity between all pairs of items from each cluster. Note that the attribute values of items do not change frequently, thereby not requiring to re-make clusters and to re-calculate dissimilarity between clusters repeatedly.

2.2.2 Approximation Algorithms

Item clustering significantly reduces the number of candidates to be compared. However, it is still NP-hard to find an optimal solution, with time complexity of $O(c!)$. Our second strategy for efficient computations is to employ *Stepwise Forward Selection* (*SFS*) and *Stepwise Backward Elimination* (*SBE*), the greedy-based approximation algorithms. They gradually build up the recommendation list that maximizes the objective function.

SFS approximates the optimal solution by inserting a cluster that maximizes the objective function at every step, eventually making up k clusters. At first, *SFS* compares every pair of two clusters and chooses one pair with the highest score of the objective function. Then, it gradually inserts clusters one by one while maximizing the objective function until k clusters are found. It takes $\frac{c(c-1)}{2} + c(k-2)$ times to find the solution, which has the time complexity of $O(c^2)$.

SBE reverses *SFS*. *SBE* first includes all clusters in the solution. Then, it approximates the optimal solution by gradually removing clusters one by one while maximizing the objective function at each step until k clusters remain. It takes $c - k$ iterations, each of which has the time complexity of $O(c^2)$. Thus, the total time complexity of *SBE* is $O(c^3)$.

3. Evaluation

3.1 Experimental Setup

For evaluation, we used the MovieLens 1M dataset. Note that we have proposed two distinct objective functions, which are based on weighted sum (denoted as *WS*) and multiplication (denoted as *M*), and two approximation algorithms, *SFS* and *SBE*. Therefore, we have four possible variations (denoted as *SFS_WS*, *SFS_M*, *SBE_WS*, and *SBE_M*, respectively) of the proposed method. We compared our algorithms with two existing diversification methods, denoted as *TD* (*topic diversification*) [3] and *DRCF* (*diversification and refinement in collaborative filtering*) [2], and an algorithm that does not consider diversity at all, proposed in [5], as a baseline.

We evaluated their top-k recommended items in terms of accuracy and diversity. We vary the value of k from 2 to 10 in an increment of 2. To measure diversity, we use *average dissimilarity* [2] among k items. To measure the accuracy, we adopt *precision*, *recall*, and *F-measure*. Among the items in our dataset, those rated as 4 or 5 are considered as ground truth (those liked by users). For a user u, precision $P_u@k$ and recall $R_u@k$ can be computed by $\frac{|Rel_u \cap Rec_u|}{|Rec_u|}$ and $\frac{|Rel_u \cap Rec_u|}{|Rel_u|}$, respectively, where Rec_u denotes a set of k items that each method recommends to u, and Rel_u denotes a set of items considered as ground truth. F-measure $F_u@k$ is computed by $\frac{2 \times P_u@k \times R_u@k}{P_u@k + R_u@k}$. All the measurements are averaged using 5-cross validation.

3.2 Experimental Results

Before comparing our methods with the existing methods, we first analyzed the accuracy of our methods with different settings of the two parameters: (1) the number of clusters c and (2) diversity factor d. These experiments provide us with the most appropriate values for parameters in our dataset.

Figures 2 (a)–(d) show the results of our methods with a varying number of clusters. In each graph, the x-axis represents the number of recommended items k; the y-axis does the accuracy of our methods. We only show the results of *F-measure* and omit those of the rest, since all the metrics show similar tendencies: our methods provide the highest accuracy when $c = 20$. Therefore, for the following experiments, we fix the number of clusters as 20.

Figures 3 (a)–(c) represent the results while varying d values which are used by *SFS_WS* and *SBE_WS*. The x-axis represents varying d, and the y-axis indicates the accuracy of our methods with $c = 20$. Here, we fix the number of the recommended items k as 10. Both methods show the highest accuracy when d is lower than 0.2: when d is higher than 0.2, the accuracy decreases with the increase of d. In other words, 0.2 provides our methods with the best tradeoff between accuracy and diversity. We fix d as 0.2 hereafter.

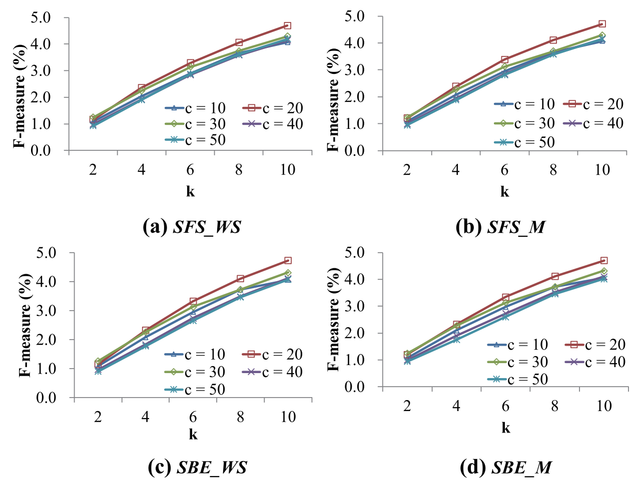

Fig. 2 Accuracy results according to the number of clusters.

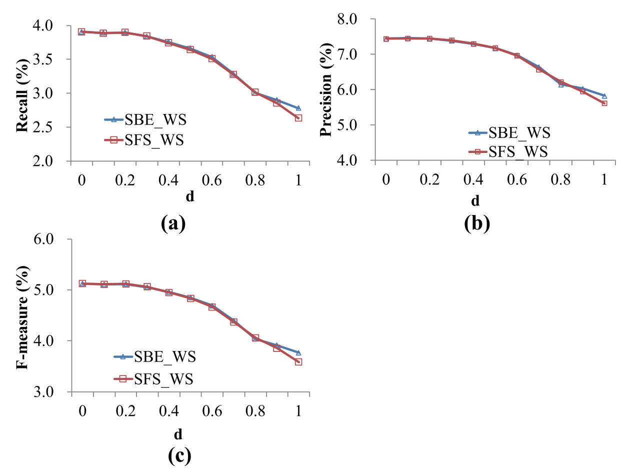

Fig. 3 Accuracy results according to the diversity factor.

With those parameter settings, Figs. 4 (a)–(c) show the results. The x-axis indicates varying k and the y-axis does the accuracy measures. Overall, our proposed methods outperform the existing methods. Specifically, our methods improve up to 81.1% recall, 139.7% precision, and 87.2% F-measure, respectively, compared to those of *DRCF*. This is because our methods consider both preference and diversity at once, while the existing methods consider them independently. We also note that our accuracy values are comparable to those of the popularly used recommender algorithms, e.g., *item-based CF*, *SVD*, and *SVD++*, on the MovieLens 1M dataset [12]. According to [12], *item-based CF*, *SVD*, and *SVD++* provide $P@10$ of about 9%, 8%, and 11.5%, respectively, and $R@10$ of 6%, 4.5%, and 7%, respectively. However, they do not consider the diversification of their top-k lists at all and thus have their recommended items tend to be similar to each other. This would lead to the failure of consistently stimulating the user's appetite to spend. In contrast, our approach to diversifying the top-k list could reduce such a risk by increasing the chance of introducing *a vari-*

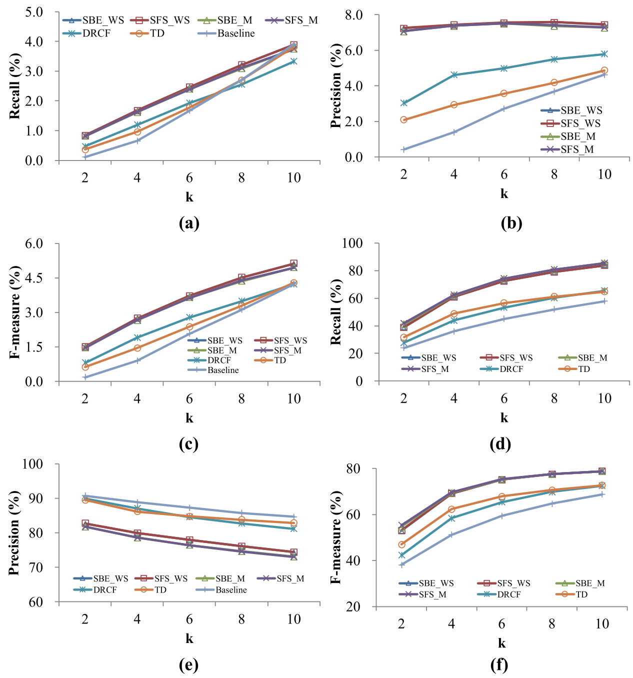

Fig. 4 Accuracy comparisons.

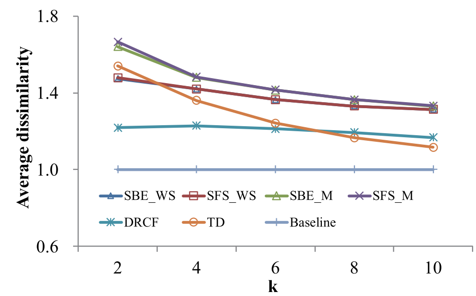

Fig. 5 Diversity comparisons.

ety of items that users are likely to be interested in. Therefore, our proposed method would be enough for practical use, providing higher diversification as well as comparable accuracy compared to other popular CF algorithms.

The previous experiments evaluate how exactly each recommendation method matches the *items* that a given user will purchase. In this experiment, we move our focus from matching items to matching categories (i.e., *genres*): we evaluate how exactly each method matches *categories* that a given user will be interested in. It is based on the simple intuition that, if a user has preferred many SF movies, recommending any SF movie may satisfy him. In order to evaluate the accuracy of matching categories, for a user u, we substituted the items in Rel_u and Rec_u with their corresponding genres, and then measured *precision*, *recall*, and *F-measure*.

Figures 4 (d)–(f) show the results. The x-axis in the graphs indicates varying k and the y-axis does the accuracy measures. We note that our methods provide *declined* precision up to 10.1%, as shown in Fig. 4 (e). This is because our methods recommend more diversified items than existing methods. However, our tendency of diversification eventually helps to recommend more satisfactory items to users, regarding that our improved recall and F-measure as shown in Figs. 4 (d) and 4 (f), respectively.

Finally, we compared the diversity of each recommendation method. The results are reported in Fig. 5, where the x-axis indicates the number of recommended items, k, and the y-axis does the ratio of the average dissimilarity of items recommended by a method X to that of the baseline method. Our proposed methods provide up to 66.5%, 36.6%, and 19.3% improved diversity compared to that of the baseline, *DRCF*, and *TD*, respectively. Our methods recommend more diverse items since they divide all the items into clusters with similar items while existing methods first find preferred items and then consider diversification among them. Moreover, depending on the business strategy, our methods can provide more diversified recommendation by increasing the diverse factor d.

4. Conclusions

Existing CF algorithms suffer from the monotony problem. Also, existing diversification methods commonly employ the 2-step approach that inherently limits diverse recommendation. We propose a novel diversification method that provides the final recommendation list in a single step where items are diverse as well as with high user preference. We define the objective function to measure the diversity and preference of a set of k items and then find the optimal solution. Since it is an NP-hard problem, we also propose two strategies for efficient recommendation: item clustering and greedy-based approximation. Through experiments, we observe that our method improves precision up to 81.1% and recall 139.7% compared to existing methods and provides up to 66.5% improved diversity. The results confirm our method successfully achieves both accuracy and diversity.

Acknowledgments

This work was supported by (1) the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT; Ministry of Science and ICT) (No. NRF-2017R1A2B3004581), (2) Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. NRF-2017M3C4A7083678), and (3) the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2017-2013-0-00881) supervised by the Institute for Information & communications Technology Promotion (IITP).

References

[1] R. Boim, T. Milo, and S. Novgorodov, "DiRec: Diversified recommendations for semantic-less collaborative filtering," Proc. 27th

IEEE Conf. on Data Engineering, pp.1312–1315, April 2011.
[2] R. Boim, T. Milo, and S. Novgorodov, "Diversification and refinement in collaborative filtering recommender," Proc. 20th ACM Int. Conf. on Information and Knowledge Management, pp.739–744, Oct. 2011.
[3] C.-N. Ziegler, S.M. McNee, J.A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," Proc. 14th int. Conf. on World Wide Web, pp.22–32, May 2005.
[4] T. Di Noia, V.C. Ostuni, J. Rosati, P. Tomeo, and E. Di Sciascio, "An analysis of users' propensity toward diversity in recommendations," Proc. 8th ACM Conf. on Recommender Systems, pp.285–288, Oct. 2014.
[5] F. Christoffel, B. Paudel, C. Newell, and A. Bernstein, "Blockbusters and Wallflowers: Accurate, diverse, and scalable recommendations with random walks," Proc. 9th ACM Conf. on Recommender Systems, pp.163–170, Sept. 2015.
[6] X. Su and T.M. Khoshgoftaar, "A survey of collaborative filtering techniques," Advances in Artificial Intelligence, vol.2009, Article ID 421425, Jan. 2009.
[7] W.-S. Hwang, J. Parc, S.-W. Kim, J. Lee, and D. Lee, ""Told you I didn't like it": Exploiting uninteresting items for effective collaborative filtering," Proc. 32nd IEEE Conf. on Data Engineering, pp.349–360, May 2016.
[8] W.-S. Hwang, H.-J. Lee, S.-W. Kim, Y. Won, and M.-S. Lee, "Efficient recommendation methods using category experts for a large dataset," Information Fusion, vol.28, pp.75–82, March 2016.
[9] X. Zhu, J. Guo, X. Cheng, P. Du, and H.-W. Shen, "A unified framework for recommending diverse and relevant queries," Proc. 20th ACM Conf. on World Wide Web, pp.37–46, April 2011.
[10] J. Ha, S.-H. Kwon, S.-W. Kim, C. Faloutsos, and S. Park, "Top-N recommendation through belief propagation," Proc. 21st ACM Conf. on Information and Knowledge Management, pp.2343–2346, Nov. 2012.
[11] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," Expert Systems with Applications, vol.36, no.2, pp.3336–3341, 2009
[12] J. Lee, D. Lee, Y.-C. Lee, W.-S. Hwang, and S.-W. Kim, "Improving the accuracy of top-*N* recommendation using a preference model," Information Sciences, vol.348, pp.290–304, June 2016.