디지털산업정보학회 논문지 제15권 제3호-2019년 9월
http://dx.doi.org/10.17662/ksdim.2019.15.3.129

A Situation-Based Recommendation System for Exploiting User's Mood

Younghyun Kim*·Lim, Woo Sub**·Jeong, Jae-Han**·Lee, Kyoung-Jun***

사용자의 기분을 고려하기 위한 상황 기반 추천 시스템

김 영 현·임 우 섭·정 재 한·이 경 전

<Abstract>

Recommendation systems help users by suggesting items such as products, services, and information. However, most research on recommendation systems has not considered people's moods although the appropriate contents recommended to people would be changed by people's moods. In this paper, we propose a situation-based recommendation system which exploits people's mood. The proposed scheme is based on the fact that the mood of a user is changed frequently by the surrounding environments such as time, weather, and anniversaries. The environments are defined as feature identifications, and the rating values on items are stored as feature identifications at a database. Then, people can be recommended diverse items according to their environments. Our proposed scheme has some advantages such as no problem of cold start, low processing overhead, and serendipitous recommendation. The proposed scheme can be also a good option as of assistance to other recommendation systems.

Key Words : Recommender System, Database, Mood, Serendipitous Recommendation

Ⅰ. Introduction

Recommendation systems (RS) help users, by suggesting items such as products, services, and information that best suit people's preferences [1]. Hence, people who utilize RS can easily make decisions what items to buy, what books to read, and what movies to watch. Over the last decade, various RS have been developed and used in diverse domains, i.e., the movie industry, the music industry, and so on [2]. Especially, online commerce industries such as Amazon.com and Netflix have successfully utilized to increase customer loyalty. For example, Amazon.com and Netflix have made and generated much of their sales by providing

* 한국소방산업기술원 기술연구소 (교신저자)
** 한국소방산업기술원 기술연구소
*** 경희대학교 경영대학

personalized items through RS [3-4].

Diverse RS such as personalized recommendations, content-based recommendations, and knowledge-based recommendations have been developed, but collaborative filtering (CF) is one of the most prominent RS. The key idea of CF is to calculate similarities between users (or items) based on a rating matrix and to predict preferences on items with which a user has not experienced. CF methods are generally classified into memory-based CF and model-based CF. In model-based CF, training datasets are used to develop a model for predicting user preferences. Also, different machine learning algorithms such as Bayesian networks, clustering, and rule-based, can be utilized to build the model. The main advantages of model-based CF are an improvement of prediction performance and tolerant of the data sparsity. However, it has some shortcomings such as an expensive cost for building a model [5]. On the other hand, memory-based CF does not build a specific model, but directly computes the similarity between users or items using the entire or a sample of rating matrix. Hence, memory based CF is easy to implement and effective to manage. However, it has also some drawbacks such as dependence on human ratings, performance decrease when data are sparse, and not to be able to recommend for new users and items [5].

CF is well-defined method, but it has some shortcomings which are vulnerable to data sparsity and cold-start problems [6]. Data sparsity refers to the problem with insufficient information about the ratings of users on items. If the amount of data is insufficient, predicted preference values become inaccurate. Also, new users or items cannot be reflected in the results of a CF process since CF is based on the ratings. Some researches have been conducted to solve these problems [7-8]. Again, the main research issues in RS are how to improve the prediction accuracy and how to provide serendipitous recommendations[1] for people. First, in order to increase the prediction accuracy, new similarity models have been proposed in [10-11]. They are based on PIP (*Proximity-Impact-Popularity*) and *Jaccard* similarity schemes, respectively, and enhanced performance has been observed. On the other hand, [12] suggested a typicality-based collaborative filtering recommendation method named by TyCo considering typicality degrees. Several studies on serendipitous recommendation systems have been conducted in [13-15].

As mentioned above, most research on RS has not considered people's moods although the appropriate contents recommended to people would be changed by people's moods. For example, on a sunny day at noon, a user wants to hang out with his/her friends at a park or a theme park. However, with a change in the weather, such as a rainy day, a user does not want any outdoor activities. Moreover, on a sunny day, a user would be more energetic, in contrast to depressed feeling of a user when the weather is gloomy. Hence, in this paper, we exploit the mood of a user for recommendations. To this end, we propose a

1) Most people have eager for something useful they have not experienced. This is usually called by serendipity, and it means that recommended items are unpredictable, unexpected, or surprising to users in recommender systems research [9].

situation-based recommender (SBR) system which based on the surrounding environments of a user, such as a time, a weather, a location, and so on. Main contribution of this paper are as follows. First, to the best of our knowledge, this is the first work on recommendation systems exploiting people's moods. Also, our proposed scheme can tackle the cold start problem. Finally, the SBR system can be easily adopted to other recommendation systems because of its low processing overhead.

The remainder of this paper is organized as follows. We present the system model after summarizing related works in Section Ⅱ and Section Ⅲ, respectively. We then, describe the situation-based recommendation (SBR) system and discuss the proposed scheme qualitatively. Finally, we conclude our work.

Ⅱ. Related Works

Recently, extensive works on recommender systems have been conducted in the literature including machine learning based systems and deep learning based systems[2].

As mentioned before, machine learning based recommender systems are divided into collaborative filtering techniques, content-based filtering systems, and hybrid filtering systems [18]. Collaborative filtering (CF) techniques recommend items to a particular user based on rating/opinions of the other users. Content-based (CB) approaches recommend an item which is similar in characteristic to items that other users have already used in the past. Hybrid filtering schemes achieve by a combination of two or more recommendation systems in order to get better performance over collaborative filtering and content-based filtering. However, these filtering schemes are heavily dependent on extensive user or item profile information that makes a cold start problem, data sparsity problem, a scalability problem, and so on [18]. To address the cold start problem, a centrality-based social network analysis method has been proposed in [19]. The proposed method considers social network information of users. To this end, a social relation value is newly proposed for new users. And the authors showed that the proposed scheme is more suitable for new users in the IoT environment.

Association rules also being able to solve above problems are independent of any personal user model and do not require a complex system of ratings. An association rule normally consists of a set of antecedent items that lead to a consequent item with a certain confidence [20]. Based on a set of observed items selected by a user, the association rule produces a set of items ranked by confidence of their being observed next. A conventional association rule and a pairwise association rule have been implemented and compared on a large data set of real dietary recalls in [20-21].

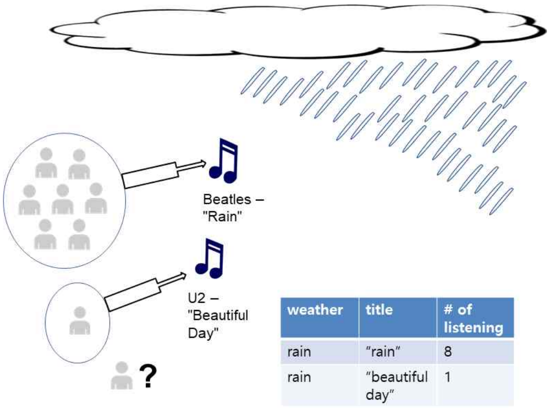
As deep learning's revolutionary advances in speech recognition, image analysis, and natural language, deep learning technology has also been applied into recommender system [22-23]. Deep learning based recommender systems are divided to two categories, i.e., *recommendation with neural*

2) Recent research trends of machine learning and deep learning can be found in [16-17].

building blocks, and *recommendation with deep hybrid models* [23]. The first one is to utilize one deep learning model. On the other hand, the last one utilizes more than one deep learning technique. There are many possible combinations of deep learning techniques, such as recurrent neural network (RNN) + convolutional neural network (CNN), an autoencoder (AE) + CNN, RNN + AE, and so on. Compared with traditional recommender systems, deep learning based recommender systems can automatically learn the latent features of user and item, and improve the accuracy of recommendation. However, the main tasks of the deep learning based recommender systems are how to organize and build the massive multi source heterogeneous data. Deep learning based association rules have been also proposed in the literature [24].

Ⅲ. System Model

The key idea of the situation-based recommendation (SBR) system is described in Figure 1. In this figure, a database (DB) maintains the number of times being accessed two songs when raining. From Figure 1, it can be observed that "Rain" and "Beautiful Day" have been listened 8 times and 1 time, respectively on a rainy day. Hence, we can make a deduction that many people enjoy listening "Rain" during the rain. Finally, the SBR system recommends a new user (or new users) "Rain" which is suited for the situation and verified by the 'wisdom of the crowd.'



<Figure 1> Example of the proposed scheme

Similarly, the DB structure for the proposed scheme is defined as Table 1.

<Table 1> DB structure

Feature ID	Item ID	Ratings (r)
d_1	i_1	$r^d_{1,1}$
d_1	i_2	$r^d_{1,2}$
d_1	i_8	$r^d_{1,8}$
⋮	⋮	⋮
d_a	i_{m-4}	$r^d_{a,m-4}$
d_a	i_m	$r^d_{a,m}$
t_1	i_1	$r^t_{1,1}$
t_2	i_3	$r^t_{2,3}$
t_{10}	i_1	$r^t_{10,1}$
⋮	⋮	⋮
t_b	i_m	$r^t_{b,m}$
w_1	i_3	$r^w_{1,3}$
w_2	i_{21}	$r^w_{2,21}$
⋮	⋮	⋮
w_c	i_m	$r^w_{c,m}$
⋮	⋮	⋮

The DB consists of three degrees, i.e., *feature* *identification (ID), item ID, and ratings*. The ratings can be explicit or implicit values. If the ratings are stored as explicit values, they should be normalized to check whether the corresponding contents are preferred or not. Assume that 5-point scale, i.e., from 1 point to 5 point, is used for the ratings. Then 1 point means the worst user experience (UX), but the cumulative rating value will be increased if many people access the same content and give 1 point out of 5. Hence, it is difficult to make a decision whether a content is a valuable or not, after reaching a steady state. To mitigate this problem, we can simply convert 1, 2, 3, 4, 5 into −2, −1, 0, 1, 2. Then, a high rating value is regarded as a valuable content since the low preference, i.e., −2 point or −1 point, reduces the cumulative ratings. On the other hand, when the implicit ratings are applied, high/small value of ratings are considered as strong/little level of interest, respectively.

In Table 1, *d, t,* and *w* represent day, time, and weather, respectively. Also, *a, b,* and *c* are the granularities of *d, t,* and *w*, respectively. If we categorize *w* as *cold, hot, rain,* and *snow,* then *c* is 4 and the corresponding feature IDs become w_1, w_2, w_3 and w_4, respectively. Feature IDs are easily expandable like *season* or *anniversary*. This is dependent on the characteristics of dataset or recommendation system. Also, feature IDs can be divided or merged by the policy of recommendation systems. For example, suppose that d_1 and d_2 mean Saturday and Sunday, respectively. If both Saturday and Sunday are wanted to consider, we have to just add d_1 and d_2 rows in a rating matrix. The rating matrix will be described in the next section. In this paper, we assume that the rating value r is a content access frequency on feature IDs. For example, the values of $r^d_{1,1}$, $r^t_{1,1}$, and $r^w_{1,1}$ become 10, respectively, if people have accessed a content i_1 10 times on d_1, t_1, and w_1.

Ⅳ. Situation-based Recommendation System

The SBR system is described in Figure 2. First, *a, b, c,* and *m* are initiated according to the DB. As mentioned before, *a, b,* and *c* are the granularities of day, time, and weather, respectively. *m* is the number of total items. And then, the rating matrix *R* is generated and initiated from the DB. *R* can be loaded with one variable or several variables by feature IDs. In the case of the former, *R (a+b+c* by *m)* is stored in memory as Eq. (1).

$$R=\begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,m} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{a,1} & r_{a,2} & \cdots & r_{a,m} \\ r_{a+1,1} & r_{a+1,2} & \cdots & r_{a+1,m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{a+b,1} & r_{a+b,2} & \cdots & r_{a+b,m} \\ r_{a+b+1,1} & r_{a+b+1,2} & \cdots & r_{a+b+1,m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{a+b+c,1} & r_{a+b+c,2} & \cdots & r_{a+b+c,m} \end{pmatrix} \qquad (1)$$

When a request message with user identification *u* arrives at the system, the list of contents which *u* has accessed is stored in *tmp* (line 5 in Figure 2). The purpose of this process is to provide *u* the long tail contents by excluding the items within *tmp*.

After that, normalized day, time, and weather based on the location of u are assigned to d, t, and w, respectively (lines 6-8). For example, if t_6 represents from 3pm to 6pm, and the current time of u is 5pm, then 5pm is assigned to t_6. The rest variables, i.e., d and w, are assigned by the similar way.

After normalizing day, time, and weather information of u, dth, $(a + t)$th, and $(a + b + w)$th rows are stored in $r_d[m]$, $r_t[m]$, and $r_w[m]$, respectively (lines 9-11). Each row means the situation information of u, i.e., day, time, and weather, when u is recommended some items. And $r_d[m]$, $r_t[m]$, and $r_w[m]$ are the rating values with respect to the d day, t time, and w weather, respectively. Top-N items are given by *CalTopN* with $r_d[m]$, $r_t[m]$, and $r_w[m]$ (line 12). Diverse methods such as arithmetic mean, geometric mean, harmonic mean, etc., can be utilized for the *CalTopN* function.

Finally, like lines 14-21 in Figure 2, $I_u[N]$ items are compared with *tmp*. For example, if $I_u[0]$ item is not included in *tmp*, the process gets out of the while loop and $I_u[0]$ item will be recommended to u (see lines 15-16). However if all items in $I_u[N]$ belong to *tmp*, the best item for u, i.e., $I_u[0]$ is recommended. However, this can be changed by the policy of system.

1 Initiate a, b, c;
2 Initiate m;
3 $R \leftarrow a + b + c$ by m matrix;
4 Receive a request message with u;
5 $tmp \leftarrow$ the list of contents which u has accessed;
6 $d \leftarrow$ normalized current day for u;
7 $t \leftarrow$ normalized current time for u;
8 $w \leftarrow$ normalized current weather for u;
9 $r_d[m] \leftarrow R[d]$;
10 $r_t[m] \leftarrow R[a + t]$;
11 $r_w[m] \leftarrow R[a + b + w]$;
12 $I_u[N] \leftarrow$ CalTopN($r_d[m], r_t[m], r_w[m]$);
13 $a \leftarrow 0$;
14 **while** $a < N$ **do**
15   **if** $I_u[a] \notin tmp$ **then**
16   break;
17   **else if** $a == N - 1$ **then**
18         $a \leftarrow 0$;
19   break;
20   **else** $a \leftarrow a + 1$;
21 **end**
22 Recommend $I_u[a]$ to u;

<Figure 2> Situation-based recommendation algorithm

Ⅴ. Discussion

In this section, we qualitatively evaluate the proposed recommendation scheme.

First of all, our proposed scheme can solve the cold start problem, especially for new users. Most widely known recommender systems including collaborative filtering (CF) and content-based (CB) schemes have the cold start problem for new users. In these schemes, it is difficult for new users to recommend specific contents since CF and CB schemes require the past access histories or preferences. However, our proposed scheme needs not personal access histories[3] but the 'wisdom of the crowd' to recommend items. And people will feel useful since the most popular items are recommended according to their environments. The SBR scheme may have the *starvation* problem of new items because the rating values of existing items have been cumulated and increased. Some

3) The SBR system does not consider personal access histories, and thus it cannot recommend personalized items. However, it is noted that the main purpose of this work is to exploit the surrounding (or social) environments such as time, anniversary, weather, and so on.

methods can be utilized to solve this problem. The first one is to define an upper limit and drop the rating value having the maximum. TCP (Transmission Control Protocol) *Tahoe* or *Reno* is an example of the first solution [25]. The last one is to give new items a high priority for a period of predefined time like *aging (scheduling)* used in operating systems (OS) [26].

The SBR system has the low processing overhead. The time complexity of the SBR system is $O(n_f \cdot m)$, where n_f is the number of features and m is the number of items. For example, $n_f = 3$ in Table 1 since the features are defined as *d*, *t*, and *w*. Thus, the SBR system can be easily extendable without the consideration of processing overhead because the processing time increases linearly by n_f and m Because of this property, the SBR system can be freely utilized to solve the cold start problem or exploit individual environments in existing recommendation systems. Similar to the low time complexity, the DB construction cost is also low.

The SBR system makes people feel *serendipity* since the SBR system does not consider individual preferences and is only based on the 'wisdom of the crowd' unlike general CFs, Hence, people get more chances to be recommended diverse items without taking their preferences. Clustering schemes are good options for improving *serendipity*.

VI. Conclusions

In this paper, we proposed a situation-based recommendation (SBR) system which considers people's mood. To this end, we exploit the surrounding environments of a person. In the SBR system, a database is built by feature identifications, i.e., day, weather, time, and so on. Then, the SBR system recommends specific contents that many people love to enjoy according to the occasion. It has some advantages such as no cold start problem, low time complexity, an extensibility, and an easy implementation. The SBR system can itself work well, but it can be also a good option as of assistance to other recommendation systems. In our future work, we will apply the SBR system to mobile application and analyze the performance of the SBR scheme.

참고문헌

[1] T. Mahmood and F. Ricci, "Improving Recommender Systems with Adaptive Conversational Strategies," in Proc. ACM Conference on Hypertext and Hypermedia, Torino, Italy, 2009, pp.73-82.

[2] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, 2005, pp.734-749.

[3] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, Vol. 7, No. 1, 2006, pp.76-80.

[4] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE Computer*, Vol. 42, No. 8, 2009,

pp.30-37.

[5] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques, *Hindawi Advances in Artificial Intelligence*," Vol. 2009, No. 421425, 2009, pp.1-19.

[6] G. Guo, J. Zhang, and D. Thalman, "Merging Trust in Collaborative Filtering to Alleviate Data Sparsity and Cold Start," *ELSEVIER Knowledge-Based System*, Vol. 57, 2013, pp.57-68.

[7] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A Collaborative Filtering Approach to Mitigate the New User Cold Start Problem," *ELSEVIER Knowledge-Based System*, Vol. 26, 2012, pp.225-238.

[8] H. Sobhanam and A. K. Mariappan, "A Hybrid Approach to Solve Cold Start Problem in Recommender Systems using Association Rules and Clustering Technique," *International Journal of Computer Applications*, Vol. 74, No. 4, 2013, pp.17-23.

[9] H. Yamaba, M. Tanoue, K. Takatsuka, N. Okazaki, and S. Tomita, "On a Serendipity-Oriented Recommender System Based on Folksonomy," *Springer Artificial Life and Robotics*, Vol. 18, No. 1-2, 2013, pp.89-94.

[10] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A New User Similarity Model to Improve the Accuracy of Collaborative Filtering," *ELSEVIER Knowledge-Based System*, Vol. 56, 2014, pp.155-166.

[11] B. H. Huang and B. R. Dai, "A Weighted Distance Similarity Model to Improve the Accuracy of Collaborative Recommender System," in *Proc. IEEE International Conference on Mobile Data Management (MDM)*, Pennsylvania, USA, 2015, pp.104-109.

[12] Y. Cai, H.-F. Leung, Q. Li, H. Min, J. Tang, and J. Li, "Typicality-Based Collaborative Filtering Recommendation," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 3, 2014, pp.766-779.

[13] Q. Lu, T. Chen, W. Zhang, D. Yang, and Y. Yu, "Serendipitous Personalized Ranking for Top-N Recommendation," in *Proc. IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, Macau, China, 2012, pp.258-265.

[14] K. Oku and F. Hattori, "Fusion-based Recommender System for Improving Serendipity," in *Proc. Novelty and Diversity in Recommender Systems (DiveRS 2011) (Conjunction with ACM RecSys 2011)*, Chicago, USA, 2011, pp.1-19.

[15] P. Adamopoulos and A. Tuzhilin, "On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected," *ACM Transactions on Intelligent Systems and Technology*, Vol. 5, No. 4, 2014, pp.1-32.

[16] C.-S. Kim, N. Kim, and K.-Y. Kwahk, "Research Trends Analysis of Machine Learning and Deep Learning: Focused on the Topic Modeling," *Journal of Korea Society of Digital Industry and Information Management,* Vol. 15, No. 2, 2019, pp.19-28.

[17] S. Lim and M. Lee, "A Computer Aided Diagnosis Algorithm for Classification of Malignant Melanoma Based on Deep Learning," *Journal of Korea Society of Digital Industry and Information Management,* Vol. 14, No. 4, 2018, pp.69-77.

[18] P. Kumar and R. S. Thakur, "Recommendation System Techniques and Related Issues: A Survey," *Springer International Journal of Information Technology*, Vol. 10, No. 4, 2018, pp.495-501.

[19] J. Kwon and S. Kim, "Recommendation Method Considering New User in Internet of Things Environment," *Journal of Korea Society of Digital*

Industry and Information Management, Vol. 13, No. 1, 2017, pp.23-35.

[20] T. Osadchiy, I. Poliakov, P. Olivier, M. Rowland, and E. Foster, "Recommender System Based on Pairwise Association Rules," *Elsevier Expert Systems With Applications,* Vol. 115, 2019, pp.535-542.

[21] W. DuMouchel and D. Pregibon, "Empirical Bayes Screening for Multi-Item Associations," in *Proc. ACM International Conference on Knowledge Discovery and Data Mining (KDD),* San Francisco, USA, 2001, pp.67-76.

[22] R. Mu, "A Survey of Recommender Systems Based on Deep Learning," *IEEE Access,* Vol. 6, 2018, pp.69009-69022.

[23] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning Based Recommender Systems: A Survey and New Perspectives," *ACM Computing Surveys,* Vol. 52, No. 1, 2019, pp.5:1-5:38.

[24] E. Cho, S. Min, S. Kim, B. Kim, "Development of Extracting System for Meaning Subject Related Social Topic using Deep Learning," *Journal of Korea Society of Digital Industry and Information Management,* Vol. 14, No. 4, 2018, pp.34-45.

[25] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach (6th Edition), Pearson, 2012.

[26] A. Silberschatz, P. B Galvin, and G. Gagne, Operating System Concepts (8th Edition), Wiley, 2008.

■ 저자소개 ■

김 영 현
Younghyun Kim

2018년 12월~현재
한국소방산업기술원 기술연구소
2013년 2월 고려대학교 전자전기컴퓨터공학과 (공학박사)
2005년 2월 숭실대학교 컴퓨터공학과(공학사)

관심분야 : 인공지능 소방 IT
E-mail : ykim@kfi.or.kr

임 우 섭
Woo Sub Lim

2008년 4월~현재
한국소방산업기술원 기술연구소
2005년 2월 부경대학교 안전공학과(공학박사)
1999년 2월 부경대학교 안전공학과(공학석사)
1996년 2월 부경대학교 화학공학과(공학사)

관심분야 : 화재안전
E-mail : presafety@kfi.or.kr

정 재 한
Jae-Han Jeong

2003년 12월~현재
한국소방산업기술원 기술연구소
1993년 2월 서울대학교 금속공학과(공학박사)
1989년 2월 서울대학교 금속공학과(공학석사)
1987년 2월 서울대학교 금속공학과(공학사)

관심분야 : 소방R&D기획
E-mail : cjh@kfi.or.kr

이 경 전
Kyoung-Jun Lee

2003년 3월~현재
경희대학교 경영대학 교수
1995년 2월 KAIST 경영과학(공학박사)
1992년 2월 KAIST 경영과학(공학석사)
1990년 2월 KAIST 경영과학(공학사)

관심분야 : Business method, Business model
E-mail : klee@khu.ac.kr

논문접수일 : 2019년 7월 17일
수 정 일 : 2019년 8월 27일
게재확정일 : 2019년 8월 28일