

# A Cascade-hybrid Recommendation Algorithm based on Collaborative Deep Learning Technique for Accuracy Improvement and Low Latency

Hyun-ho Lee<sup>†</sup>, Won-jin Lee<sup>\*\*</sup>, Jae-dong Lee<sup>\*\*\*</sup>

## ABSTRACT

During the 4th Industrial Revolution, service platforms utilizing diverse contents are emerging, and research on recommended systems that can be customized to users to provide quality service is being conducted. hybrid recommendation systems that provide high accuracy recommendations are being researched in various domains, and various filtering techniques, machine learning, and deep learning are being applied to recommended systems. However, in a recommended service environment where data must be analyzed and processed real time, the accuracy of the recommendation is important, but the computational speed is also very important. Due to high level of model complexity, a hybrid recommendation system or a Deep Learning-based recommendation system takes a long time to calculate. In this paper, a Cascade-hybrid recommended algorithm is proposed that can reduce the computational time while maintaining the accuracy of the recommendation. The proposed algorithm was designed to reduce the complexity of the model and minimize the computational speed while processing sequentially, rather than using existing weights or using a hybrid recommendation technique handled in parallel. Therefore, through the algorithms in this paper, contents can be analyzed and recommended effectively and real time through services such as SNS environments or shared economy platforms.

**Key words:** Hybrid recommendation algorithm, Collaborative filtering, Deep learning

## 1. INTRODUCTION

In the era of the 4th Industrial Revolution, where the digital revolution is taking place, various attempts are made to provide various contents in the digital environment. There are a variety of techniques, ranging from simple techniques for sorting and filtering data to providing data to users in a desired form, to analyzing users and content using artificial intelligence and deep learning. A system that provides a proper analysis of the information

that users want is called a recommendation system and is being developed based on various filtering techniques and machine learning techniques. Examples of recommended systems include Collaborative Filtering(CF), Content-Based Filtering(CBF), and Rule-Based Filtering(RBF). CF recommends content through analysis of similar users or similar items.[1][2][3] If a user B is similar to user A, then the method of recommending the contents of the two users' mix to each other is used. CBF is simpler, and users will find and recommend con-

---

\* Corresponding Author : Jae-dong Lee, Address: (16890) 152, Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do, TEL : +82-31-8005-2384, FAX : +82-31-8021-7422, E-mail : letsdoit@dankook.ac.kr

Receipt date : Nov. 28, 2019, Approval date : Dec. 23, 2019

<sup>†</sup> Department of Computer Engineering, Dankook University (E-mail : leehh4016@naver.com)

---

<sup>\*\*</sup> Research Institute of Information and Culture Technology, Dankook University (E-mail : god7300@dankook.ac.kr)

<sup>\*\*\*</sup> Department of Software, Dankook University

\* This research was supported by the MSIT(Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program(2019-0-01577) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)

tent similar to the content C used. RBF is a way for developers to prepopulate systems with rules about what content matches the user's age and gender, and to recommend content according to those rules.

It is common to build a recommended system by applying the appropriate filtering techniques according to the profile structure of the service system or the service model. However, each filtering technique has a problem. One major problem with collaborative filtering is the cold start problem.[4] The cold start problem is also known as an initial user problem, which is not recommended because initial time users do not have any historical data, which results in the same problem with content-based collaborative filtering. That is, finding similar users requires finding users with similar historical information and having them recommend different content to each other, other than crossing from historical information, but without historical information. The problem is that content-based recommendations are likely to be inaccurate. Content-based recommendations are methods of recommending content that are similar to content used by users, which only determine similarity of content and cannot reflect the user's actual preferences or circumstances.

Hybrid filtering is something that emerged to compensate for each disadvantage of filtering.[5] Recommended system that uses more than one filtering to compensate for each of the disadvantages of filtering. A variety of studies have been conducted using a hybrid system with the aim of solving problems not recommended by early users and improving the accuracy of recommendations. However, another problem is that the algorithms and models are more complex and the computational speed is reduced in order to increase the accuracy of recommendations. In particular, the complexity of the models has also increased dramatically as AI and Deep Learning technologies, the major technologies of the 4th industrial revolution, have

been applied to the recommended systems. Generally, the problem of delayed computing speed in the recommendation system is solved by calculating the contents or user analysis in advance as a pretreatment process and making recommendations using the results analyzed. However, this solution is not a fundamental solution. When users need to analyze their situations and preferences in real time and recommend content that fits their current state, the usual method cannot resolve delays in computational speed. As hardware performance improves over time, you may expect to improve your computing power. However, positive improvements were expected through improvements in software models and algorithms. Therefore, this paper proposes sequential recommended algorithms. To reduce the computational speed of the recommendation system that predicts users' next behavior based on deep learning, this paper proposed a Collaborative Deep Learning that can reduce the complexity of the model while maintaining the accuracy of the recommendation by applying various filtering techniques that perform simple operations on deep learning. In addition, the cold start problem was solved by allowing early users to accumulate more than a certain amount of history through rule-based filtering and receive a high level of recommendation based on deep learning. The proposed algorithm can expect the accuracy and low computational speed of the recommendation and consists of sequential processes that do not cause the inherent cold start problems of the recommendation system.

## 2. RELATED WORK

### 2.1 Recommendation based on filtering techniques

Most of the filtering-based recommended systems work with algorithms that filter and sort data through similarity calculations to recommend optimal content to users.[6] Typical algorithms include

collaborative filtering, content-based filtering, rule-based filtering, and location-based filtering. Collaboration filtering is divided into user-based and contents-based.

However, there are problems with the single use of filtering techniques. For collaborative filtering, there is a cold start problem that occurs because early users or items do not have a history. Because content-based filtering is recommended considering only the attributes of items, it is difficult to consider users' preferences or situations and thus has a problem of low accuracy. Because rule-based filtering is difficult to respond to various situations or phenomena, and location-based filtering is only applicable to limited services.

## 2.2 Hybrid Recommendation Algorithm and Deep learning

Hybrid filtering is a concept that emerged to compensate for problems arising from recommended systems using a single existing filtering system.[7] Mixing filtering improves recommendation performance by combining two or more filtering or various techniques to address the problems of chronic filtering, such as cold start problems arising from collaborative filtering, low-recommended accuracy from content-based filtering, user preference and status, and rule-based filtering limitations and limited recommended coverage issues.

Shih, Ya-Yueh, and Duen-Ren Liu[8] proposed a hybrid recommendation system based on collaborative filtering via valuable content information. Wang, Xinxi, and Ye Wang[9] proposed music recommendation using deep learning for improving content-based filtering. Lucas, Joel P., et al.[10] proposed a hybrid recommendation approach for a tourism system. In this paper, a performance comparison test of the recommended and proposed systems based on hybrid filtering and deep learning, and the characteristics of the algorithms used for each comparison was conducted and were

Table 1. The comparison of recommendation algorithm

Existed research	Main method
Shih, Ya-Yueh, and Duen-Ren Liu[8]	Hybrid filtering
Wang, Xinxi, and Ye Wang[9]	Deep learning
Lucas, Joel P., et al.[10]	Hybrid filtering
Proposed algorithm	Hybrid filtering + Deep Learning

shown in Table 1.

Because these methods perform an analysis for recommendation at a given time interval, only the same recommendations are given for a given period of time. The advantages of this thesis are that three techniques are performed sequentially and that low model complexity can be expected, allowing users to reflect feedback in real time and receive new recommendations for each recommendation. In particular, these advantages can be strong in social networking sites or on shared-economic platforms that require new recommendation results in real time.

Deep Learning is defined as a set of machine learning algorithms that attempt to abstract at a high level through a combination of nonlinear conversion techniques, and can be said to be a field of machine learning that teaches computers how to think in a large frame.

If Deep Learning is applied to a recommendation system, accuracy of recommendation can be improved significantly and various information such as user preferences and properties of contents can be analyzed and recommended. These Deep Learning-based recommendation systems can be considered suitable in environments where users and content can be preanalyzed and recommended over a given period of time. For example, the introduction of new users and content may be appropriate in the recommendation of contents such as movies, books, and music, which are limited. However, Deep Learning is not suitable in a recom-

mended service environment that analyzes and provides users in real time, such as SNS or shared economy platforms, where new content is generated in real time, due to the high complexity of the model and long computing times.

### 3. THE PROPOSED CASCADE-HYBRID RECOMMENDATION ALGORITHM

#### 3.1 The Processing of Algorithm

The proposed Cascade-hybrid Recommendation Algorithm(ChRA) is performed in the process shown in Fig. 1.

First, build a profile for recommendation. The profile consists of user history-based data set H, user information data set U, and content data set C. Based on the next deployed profile, the user to be recommended is determined whether or not the user is an initial user. Because there is no history for early users, it is impossible to analyze the Collaborative Deep Learning(CDL) until sufficient experience is accumulated by moving to a rule-based filtering step. Users with sufficient experience will move to the CDL stage. Based on the history of the user, Deep Learning analysis is performed to predict the following behavior, which is

used in this paper by using the Recurrent Neural Network(RNN) algorithm. RNN algorithm is a type of deep learning that is an artificial neural network predictor with the most basic yet powerful predictability. The following actions, as predicted by RNN, become a content, and to generate a recommendation list similar to that content, through content-based filtering, create a recommendation list. Then, through CF techniques, the recommendation list is rearranged once again. Such an architecture allows RNNs to be less complex in their models to compensate for incorrectly predicted outcomes through collaborative filtering. This means that the lower complexity ensures lower computational speed while maintaining the accuracy of recommendations. Afterwards, the forecast results are corrected once again through rule-based filtering, which is provided to the user as the final recommendation result. Users will use or purchase the content provided, and the history will follow the feedback process applied to the profile, and the profile will be updated to apply to the next recommendation. The details of each process are described in the following sections.

#### 3.2 Create Profiles

The creation of a profile is performed to construct data for recommendations and also to rebuild data by reflecting feedback on each recommendation. The three profiles, H, U and C, are constructed as shown in Table 2.

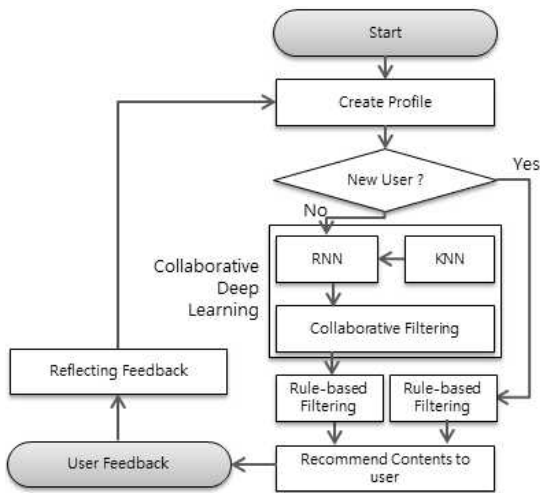


Fig. 1. The Process of ChRA based on Collaborative Deep Learning.

Table 2. The comparison of recommendation algorithm

Profile	Attribution	Type
H	UID	CHAR
	$h_n$	FLOAT
U	UID	CHAR
	Gender	INT
	Age	INT
	Location	LINE STRING
	Etc.	-
C	CID	CHAR
	$Prop_n$	CHAR or FLOAT

User history-based profile  $H$  includes a history of content usage  $h_n$  that users accumulate while using the service. The UID is designated as the value that separates the user, and it stores the user's content history information. The content history information is the value of normalizing the content ID between 0 and 1 to facilitate the recommended operation.

User information  $U$  includes the user's demographic information. Analyze similar users through Collaborative filtering using  $U$  demographic information.  $U$  includes information such as gender, age, etc. to distinguish users.

Content information  $C$  can consist of various at-

tributes, such as CID that can distinguish content and standards, and  $Prop_n$  such as types, colors, and so on, that can represent the properties of the content.

### 3.2 New User

The process of identifying new users is done to solve the cold start problem. There are many ways to solve the cold start problem. Netflix, a movie recommended service, also collects basic historical information by requiring users to enter 20 movies of their preference before they actually use the service. Because the algorithmic process was constructed sequentially in this thesis, it was designed

---

#### Algorithm 1. Collaborative Deep Learning based on RNN

---

Input:

User History dataset:  $H$

Output:

Predicted next action of  $U_n$ :  $\hat{h}_n$

Recommendation List:  $RNN\_Rec\_List_n$

```

01: def sigmoid(x); // Compute sigmoid nonlinearity
02: def sigmoid_output(output); //Convert output of sigmoid function to its derivative
03:
05: train_data = H; // Training Dataset Generation
06: // set input variables
07: alpha = 0;
08: input_dim = 0;
09: hidden_dim = 0;
10: output_dim = 0;
11:
12: initialize_nn_weight(); // initialize neural network weights(input, hidden, output)
13: training_model(train_data) {
14:     for i in range(10000) :
15:         forward_propagation();
16:         backpropagation();
17:     chra_model = 0;
18: }
19:
20: predict_user_action(chra_model) {
21:     for n in  $U_n$ :
22:         collab_filt(train_data);
23:         user_sim( $U_i, U_j$ );
24:         knn_grouping_user();
22: }
23:
24: make_rec_list();
25:

```

---

to run rule-based filtering only until enough historical information was gathered beyond the section that caused the cold start problem.

### 3.4 Collaborative Deep Learning

The Collaborative Deep Learning proposed in this paper is the process of creating a primary recommendation by combining content-based filtering, collaboration filtering, and deep learning techniques. Content-based filtering is a pretreatment process in this procedure. This is the final recommendation through RNN(Recurrent Neural Network), which is one of Deep Learning algorithms, to predict users' next behaviors. The following behavior ultimately results in one content, and similar content can be viewed as a single recommendation list. Therefore, similar content is grouped through KNN-clustering of content-based content in order to find groups similar to those predicted through RNN. The RNN algorithm to generate the primary recommendation list is computed as shown in algorithm 1, while KNN-cluster is computed.

In Algorithm 1, user history dataset  $H$  set to input data and predicted user's next action and  $RNN\_Rec\_List_n$  set to output. First, sigmoid have to define in first step. Then, training dataset should be generated and set some input variables such as alpha, input dimensions(input\_dim), hidden dimensions(hidden dim), and output dimensions (output\_dim). Next, all weight value should be initialize and start to make training model "chra\_model". In RNN, basically generate forward propagation and back-propagation to find error and correct error. Predicted user's next action is generated in line 20 using "chra\_model" and group similar user using KNN-clustering. KNN-clustering formula using Manhattan distance functions is shown in formula (1).

$$\sum_{i=1}^k |x_i - y_i| \tag{1}$$

Finally, recommendation list  $RNN\_Rec\_List_n$  is made on based on user's similarity generated on KNN-clustering and training model "chra\_model",

in line 24..

### 3.5 Rule-based Filtering

The primary recommendation is rearranged once again by rule-based filtering. This process is carried out for two purposes. The first is to recommend content to early users without additional historical information to solve the cold start problem. Early users ignore the preceding process and perform rule-based filtering first until sufficient experience is gained. Second, the recommended accuracy is improved by rearranging the primary recommendation list. The first recommendation is recommended based on the user's history, but the recommendation list will be sorted once again because users with less regularity in their history or a high-exceptional history of the model may have a higher recommended accuracy by rule-based filtering. This is not a significant change when the primary recommendation has the correct recommendation, but it is a process that can correct the major error in the primary recommendation. Rule-based filtering is calculated as shown in algorithm 2 below.

In Algorithm 2, rules for recommendation and recommendation list  $RNN\_Rec\_List_n$  set to input, and recommendation list  $RBF\_Rec\_List_n$  set to

---

#### Algorithm 2. Rule-based filtering

---

Input:

Rules for Recommendation  
 Recommendation List:  $RNN\_Rec\_List_n$

Output:

Recommendation List:  $RBF\_Rec\_List_n$

```

01: rec_rules = Rules for Recommendation;
02:
03:           filtering_rnn_based_on_rules
    ( $RNN\_Rec\_List_n$ ){
04:   for i in range(1000):
05:     reflect(rec_rules);
06: }
07:
08:  $RBF\_Rec\_List_n$  = filtering_rnn_based_on_rules
    ( $RNN\_Rec\_List_n$ );
09:
    
```

---

Table 3. The Experiment Environment

Element	Performance
OS	Windows 10
CPU	Intel(R) Core(TM) i7-7700K CPU @ 4.20 GHz 4.20 GHz
Memory	16.00 GB
HDD	256 GB SDD
Analysis tool	R-3.5.1 and R studio
Dataset	Recommendation curation service for vitalizing sports for all by the Korean Ministry of Culture, Sports and Tourism

output. First, rules for recommendation set to “rec\_rules” in line 1. Then re-filtering using  $RNN\_Rec\_List_n$  in previous collaborative deep learning step and rules for rule-based filtering.

### 3.6 Reflecting Feedback

When the user receives the final recommendation results and uses the contents, the information will be reflected in the profile as a new historical information. The new experience reflected in real time is immediately applied to the following recommendations and can be given a new recommendation list. Because of the characteristics of algorithms that are constructed sequentially, this method is possible due to the low computational speed even when new recommended operations are performed on each recommendation.

## 4. EVALUATION

The experiment was conducted to verify the recommended accuracy and performance of the proposed algorithm. The experimental environment is shown in Table 3. It is recommended to note that for computational speed, the performance of the computer is very important, and that the experimental results may vary depending on the experimental environment.

The experiment was conducted at R, a powerful data analysis tool.[11] R provides packages for variety of analysis algorithms such as collaborative filtering, content-based filtering, and deep learning.

[12] Table 4 shows the packages and libraries used in the experiment and their uses.

The data used in the experiment were used by 300 users(U) except initial user, 2,000 data on sports matching history(H) between two team, and 1,000 data on each game’s specific results and team information for content(C), gathered from the recommendation curation service for vitalizing daily sports conducted by the Korean Ministry of Culture, Sports and Tourism. Part of the data is shown in Fig. 2, and parts that could violate privacy were mosaic treated.

For model generation, the data were separated from the learning data by a ratio of 8:2. The experimental data is constructed with user competition history information H, user information U and competition details C explained in chapter 3.2.

The experiment was carried out in two experiments. The first is an experiment to derive parameters set in the model showing optimal accuracy and computational speed from the proposed algorithm. For Deep Learning, there are parameters, such as the number of redundant nodes, that can determine the complexity of the model when creat-

Table 4. The R Package for Experiments

R Library - method	Function
class - KNN	knn(train, test, cl, k)
recommenderlab - CBF	recommender(x, “CBF”) predict(x, y, type=“ratings”)
rnn - RNN	predict_rnn(model, x, hidden)
metrics - MAE	mae(actual, predicted)

ing the model. The first experiment resulted in a conclusion showing optimal accuracy and computational speed and a performance evaluation of the algorithms proposed in this paper, compared with the hybrid filtering and Deep Learning-based recommendation system proposed in this and other papers.

MAE, the most common evaluation method, was used to derive the accuracy of the recommendation. In the creation of a model through the tutorial, the model is created with results that are the actual correct answer to the data in advance. MAE is the method of calculating the average error of the difference between the actual correct value and the value predicted by the model.[13] MAE computations can produce the average error number for the recommendation, which results in a lower value if the recommendation is correct and a higher result if the recommendation is not related. In addition, the accuracy of the recommendation can be derived by converting the results into probabilities, since error values for each recommendation are derived. MAE's calculation formula is as shown in formula (2). The system.time() function, which is provided by R, is used to measure the operation speed.

$$\frac{1}{n} \sum_1^n |d_i - \hat{d}_i| \tag{2}$$

In formula (2),  $d_i$  is the actual rating,  $\hat{d}_i$  is the predicted rating, and n is the number of items.

#### 4.1 Experiment 1: Deducting Optimized Parameter in ChRA

First, the proposed algorithm was tested to derive algorithm parameters showing optimal recommended accuracy and time of operation. The parameter to be derived through this experiment is "hidden\_dim". This is the size of the hidden layer that will be storing carry bit. The more hidden layers, the more accurate the model, but at the same time causing higher latency. In this experiment, parameter value of "hidden\_dim" derived that guarantees a low delay rate while maintaining the accuracy of the model.

Theoretically, a large number of hidden layers increases the accuracy of the model, but also increases processing latency. The proposed ChRA should reduce processing latency while ensuring the accuracy of the model by applying existing filtering techniques. In this experiment, RNN with only deep learning techniques and ChRA are given the same number of hidden layers, and the performance is verified by comparing processing latency of the two algorithms. Next, check how many hidden layers are designated for the ChRA to ensure the best model accuracy and processing latency, and use the corresponding parameters in Experiment 2.

The experiment gradually increases the hidden layer from the minimum number of 2 to 15 to evaluate each model accuracy. The number of back-

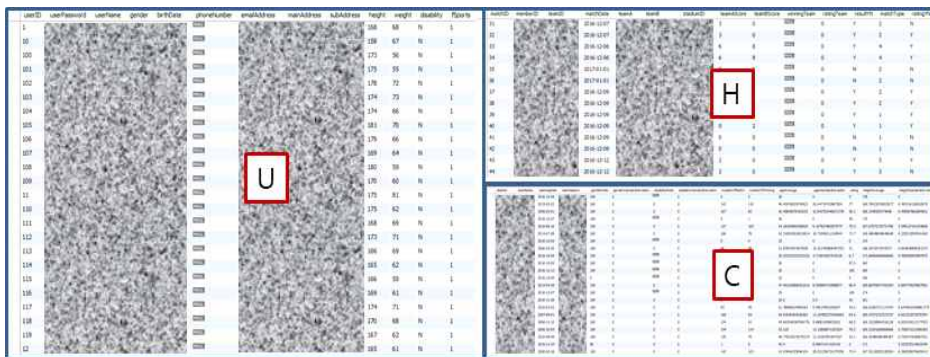


Fig. 2. The part of data for experiments (personal privacy information were mosaic treated).



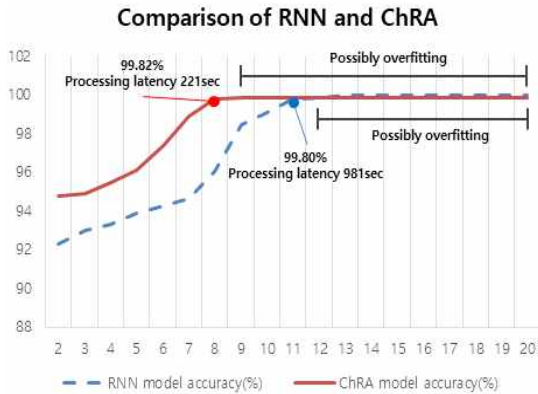


Fig. 3. The comparison of RNN and ChRA results from Experiment 1.

propagation and forward propagation repeated in model generation is limited to 100 times. At the same time, the time it takes to create a model is measured, and the two figures are charted to find points that can optimize the two figures. The accuracy of the model was calculated using MAPE, and the delay time was calculated using the R default library.[12][14] The results of applying this experiment to RNN and the ChRA are as shown below Table 5 and Fig. 3.

The results of the experiment showed high accuracy 99.8% when the number of hidden layers was set to 11 for RNN and forward propagation and backpropagation was performed 100 times to derive the model, and it was found to be overfitted after 12. Overfitting refers to the fact that the model complexity has become too high to increase the accuracy of the model rapidly, resulting in worse results when analyzed with new data.[15] For the proposed ChRA, the number of hidden layers was set to eight and the model was derived, showing the highest accuracy 99.82% and being overfitted after nine. The processing latency was measured at 981 seconds when the hidden layer was set to 11 in RNN, and the ChRA measured at 211 seconds when the hidden layer was set to 8.

#### 4.2 Experiment 2: Performance Comparison

Second, the proposed algorithm and performance comparison were conducted with existing hybrid filtering and deep learning-based recommendation systems. Each algorithm has the characteristics shown in Table 1. The proposed algorithm used the accuracy and computational time derived from

Table 5. Experiment 1. Deducting Optimized Parameter(number of hidden layer) in ChRA

Number of hidden layer	RNN		ChRA	
	model accuracy(%)	processing latency(sec)	model accuracy(%)	processing latency(sec)
2	92.34	32	94.78	36
3	93.02	37	94.90	48
4	93.30	49	95.50	60
5	93.90	84	96.15	91
6	94.30	99	97.38	115
7	94.63	155	98.90	170
8	96.05	199	99.82	211
9	98.50	231	99.99(possibly overfitting below)	239
10	99.12	448	99.99	456
11	99.80	981	99.99	991
12	99.99(possibly overfitting below)	1228	99.99	1232
13	99.99	1687	99.99	1695
14	99.99	2846	99.99	2851
15	99.99	4318	99.99	4335

Table 6. The comparison of recommendation algorithm

Recommendation Algorithms	MAE results
Shih, Ya-Yueh, and Duen-Ren Liu[8]	5.32
Wang, Xinxi, and Ye Wang[9]	3.22
Lucas, Joel P., et al.[10]	3.8
Proposed ChRA	2.02

the value of the optimal parameters derived from the first experiment, and the results of the experiment are shown in Table 6.

In order to determine the accuracy of the model, the average MAE derived from the model was used, and the average of the MAE figures produced 100 times each was derived. The experimental results showed that Shih, Ya-Yueh, and Duen-Ren Liu[8] showed an MAE of 5.32, Wang, Xinxi, and Ye Wang[9] with 3.22, Lucas, Joel P., et al.[10] with 3.8 and the proposed ChRA with 2.02.

## 5. CONCLUSION

In this paper, a Cascade-Hybrid recommended algorithm was proposed that is suitable for SNS environments or shared economic platform environments where recommendation results should be updated in real time and user feedback should be reflected. By placing and mixing algorithms sequentially, the proposed algorithm reduces the computational complexity of the model, while maintaining high recommended accuracy.

Cascade-Hybrid recommending algorithm generates the first recommendation list through Collaborative Deep Learning that combines collaboration filtering, content-based filtering, and deep learning. Later, through rule-based filtering, the recommended results are adjusted once again to avoid errors and improve the recommended accuracy, and early users only use rule-based filtering steps until sufficient experience is accumulated. After recommending content to users, the content used by users can be immediately applied as feed-

back, update the profile, and provide new recommendations in real time on the next recommendation.

Two experiments were conducted to assess the performance of the proposed algorithm. First, an Experiment 1 was conducted to derive the optimal parameters to minimize the complexity of the model while maintaining the recommended accuracy when “hidden\_dim” set to 8. The results of the experiment showed impressive results with 99.45% recommended accuracy and 211sec processing latency. Although ChRA applied three less hidden layers than when it used deep learning algorithm alone, it ensured accuracy of recommendation and was able to see that processing latency decreased by 443.9%(about 4 times) from 981 seconds to 221 seconds.

Next, a comparative Experiment 2 was conducted with a recommended system based on hybrid filtering and deep learning, which was previously studied to assess the performance of the proposed algorithm. The experimental results showed that Shih, Ya-Yueh, and Duen-Ren Liu[8] showed an MAE of 5.32, Wang, Xinxi, and Ye Wang[9] with 3.22, Lucas, Joel P., et al.[10] with 3.8 and the proposed ChRA with 2.02. That is, the proposed ChRA had the lowest error rate and performed better than any other hybrid recommendation algorithm or deep learning algorithm.

Future research will further investigate where the proposed algorithm needs to be calculated in real time to further enhance its performance and how it can be improved through preprocessing to better utilize deep learning technology. In addition, the dynamic profile proposed in the previous paper[16] be used to improve the performance of the ChRA.

## REFERENCE

- [ 1 ] J.L. Herlocker, J.A. Konstan, A.I. Borchers, and J. Riedl, “An Algorithmic Framework for Performaing Collaborative Filtering,” *Pro-*

- ceedings of the 22nd Annual International Association for Computing Machinery Special interest Group on Information Retrieval Conference on Research and Development in Information Retrieval*, pp. 230-237, 1999.
- [ 2 ] V.M. Robin and M.V. Someren, "Using Content-based Filtering for Recommendation," *Proceedings of the Machine Learning in the New Information Age: MLnet/ ECML2000 Workshop*, pp. 47-56, 2000.
- [ 3 ] A. Nguyen, N. Denos, and C. Berrut, "Improving New User Recommendations with Rule-based Induction on Cold User Data," *Proceedings of the 2007 ACM Conference on Recommender Systems*, pp. 121-128, 2007.
- [ 4 ] X.N. Lam, T. Vu, T.D. Le, and A.D. Duong, "Addressing Cold Start Problem in Recommendation Systems," *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication. ACM*, pp. 208-211, 2008.
- [ 5 ] M. Nilashi, O.B. Ibrahim, and N. Ithnin, "Hybrid Recommendation Approaches for Multi-criteria Collaborative Filtering," *Expert Systems with Applications*, Vol. 41, Issue 8, pp. 3879-3900, 2014.
- [ 6 ] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook, Recommender Systems Handbook*, Springer, Boston, 2011.
- [ 7 ] A. Albadvi and M. Shahbazi, "A Hybrid Recommendation Technique based on Product Category Attributes," *Expert Systems with Applications*, Vol. 36, No. 9, pp. 11480-11488, 2009.
- [ 8 ] Y.Y. Shih and D.R. Liu, "Hybrid Recommendation Approaches: Collaborative filtering via Valuable Content Information," *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, IEEE*, pp. 217b-217b, 2005.
- [ 9 ] X. Wang and Y. Wang, "Improving Content-based and Hybrid Music Recommendation using Deep Learning," *Proceedings of the 22nd ACM International Conference on Multimedia. ACM*, 2014.
- [10] J.P. Lucas, N. Luz, M.N. Moreno, R. anacleto, A.A. Figueiredo, and C. Martins, "A Hybrid Recommendation Approach for a Tourism System," *Expert Systems with Applications*, Vol. 40, No. 9, pp. 3532-3550. 2013.
- [11] R.D. Peng, *R Programming for Data Science*, Leanpub Publishers, British Columbia, Canada, 2015.
- [12] W.N. Venables, D.M. Smith, and the R Core Team, *An Introduction to R*, Springer, New York, 1999.
- [13] Description of MAE, [https://en.wikipedia.org/wiki/Mean\\_absolute\\_error](https://en.wikipedia.org/wiki/Mean_absolute_error) (accessed May 20, 2018).
- [14] P.M. Swamidass, *Encyclopedia of Production and Manufacturing Management*, Springer Science and Business Media, New York, 2000.
- [15] D.M. Hawkins, "The Problem of Over-Fitting," *Journal of Chemical Information and Computer Sciences*, Vol. 1, No. 44, pp. 1-12, 2004.
- [16] H.H. Lee, W.J. Lee, and J.D. Lee, "An Intelligent Recommendation Service System for Offering Halal Food (IRSH) based on Dynamic Profiles," *Journal of Korea Multimedia Society*, Vol. 22, No. 2, pp. 260-270, 2019.



Hyunho Lee

2011 Bachelor in Tianjin Nankai University Law Course  
2014 Bachelor in Computer Engineering Department, Hoseo University  
Present : Master's and Doctor's Integrated Program in Computer

Science of Dankook University Graduate School  
Study Fields : Recommendation, Customized System, Profile



Jaedong Lee

1985 Bachelor in Inha University's Electronic Calculations  
1991 Master of Computer Science, Cleveland State University Graduate School  
1996 Ph.D. in Electronic

computing & Computer science, Kent State University  
2002 Director of Dankook University Information and Communication Institute  
2008 Director of Dankook University CT Laboratory  
Present : A Professor in Department of Software, Dankook University  
Study Fields : Distributed System, Mobile Platform, Convergence Contents Recommender



Wonjin Lee

2002 Bachelor in Computer Science Department, Kyungil University  
2004 Master of Computer Engineering, Kyungpook National University  
2009 Ph.D. in

Telecommunications Engineering, Kumo University of Technology  
Present : An Assistant Professor of Institute of Information and Culture Technology of Dankook University  
Study Fields : Convergence Contents Recommender, Computer Security