Regular paper

Journal of the KIECS. pp. 125-134, vol. 16, no. 1, Feb. 28. 2021, t. 105, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2021.16.1.125

잠재요인 모델 기반 영화 추천 시스템

Chen Ma* · 김강철**

Movie Recommendation System based on Latent Factor Model

Chen Ma* · Kang-Chul Kim**

요 약

영화 산업의 빠른 발전으로 영화의 제작 수가 급격하게 증가하고 있으며, 영화 추천 시스템은 관객들의 과거 행동이나 영화 후기에 기반하여 관객들의 선호도를 예측하여 영화의 선택에 도움을 주고 있다. 본 논문은 평점의 평균과 편향의 보정을 이용하여 잠재요인 모델에 기반한 영화 추천 시스템을 제안한다. 특이값 분해 방법이 평점 매트릭스 분해에 사용되고, 통계 경사 하강법이 최소자승 손실 함수의 파라미터 최적합에 사용된다. 그리고 평균 제곱근 오차를 사용하여 제안한 시스템 성능을 평가한다. Surprise 패키지를 이용하여 제안한 시스템을 구현 하였으며, 모의실험 결과는 평균 제곱근 오차가 0.671이며, 다른 논문에서 방법에 비하여 좋은 성능을 가진다는 것을 확인하였다.

ABSTRACT

With the rapid development of the film industry, the number of films is significantly increasing and movie recommendation system can help user to predict the preferences of users based on their past behavior or feedback. This paper proposes a movie recommendation system based on the latent factor model with the adjustment of mean and bias in rating. Singular value decomposition is used to decompose the rating matrix and stochastic gradient descent is used to optimize the parameters for least-square loss function. And root mean square error is used to evaluate the performance of the proposed system. We implement the proposed system with Surprise package. The simulation results shows that root mean square error is 0.671 and the proposed system has good performance compared to other papers.

Key words

Movie Recommendation System, Improved Latent Factor Model, Average Rating, User Bias, Movie Bias, Prediction Rating Range

영화 추천시스템, 향상된 잠재요인모델, 평균 평점, 소비자 편향, 영화 편향, 예측 평점 구간

Ⅰ. Introduction

A large number of movies makes it difficult for users to select their favorite movies quickly, but this problem can be solved with the development of the recommendation system dramatically. The recommendation system used in the movie industry can make personalized recommendations based on

* 전남대학교 대학원 컴퓨터공학과 석사과정 (1095643660@qq.com)

** 교신저자 : 전남대학교 전기전자컴퓨터공학부

・접 수 일 : 2020. 11. 12
・수정완료일 : 2020. 12. 30
・게재확정일 : 2021. 02. 17

・Received : Nov. 12. 2020, Revised : Dec. 30, 2020, Accepted : Feb. 17, 2021

・Corresponding Author : Kang-Chul Kim
School of Electricity, Electronic communication and Computer Engineering, Chonnam National University,
Email : kkc@jnu.ac.kr

the user's past behavior or personal information. Fig. 1 shows the movie recommendation system manly serves three components: user consumer, movie resource provider, and platform. User consumers can watch and rate movie works, and movie resource providers hope to make profits by providing movies for users. The platform is the core component of the movie website. The platform needs to collect necessary user information, recommend movies to users, and provide movies. As an essential part of the platform, the movie recommendation system can not only save time for users but also sell movie products quickly for merchants.

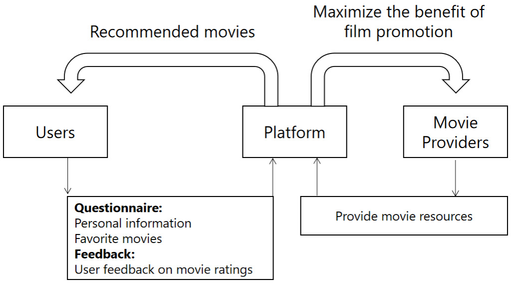


그림. 1 영화 추천 시스템의 역할
Fig. 1 The Function of recommendation system

Fig. 2 shows the classification of many recommendation methods such as content-based, association rule-based, collaborative filtering, hybrid recommendations[1][2][3] and other deep learning methods[4]. Collaborative filtering is the most commonly used filtering technique in the recommendation system, and there are multiple ways to find similar users or movies. Then the recommendation system predicts the ratings based on similar users or similar films. The collaborative filtering algorithm can be mainly divided into three types such as neighborhood-based, model-based, and hybrid methods. Neighborhood-based collaborative filtering are based on measuring the similarities between different users or items to make recommendations and mainly classified into two types, such as item-based and user-based collaborative filtering. User-based collaborative filtering is a technique used to predict the items that target user might like on the basis of ratings given to that item by the other users who have similar preference with target user. Item-based collaborative filtering focuses on the similarities between items that a user rates. Model-based collaborative filtering technique fits a model based on the previous rating of users and makes recommendations based on predictions by the fitted model. The typical model-based collaborative filtering is the latent factor model, whose main idea is to comprise an alternative method by transforming both movies and users to the same latent factor space.

This paper focuses on sparse data challenge faced by the movie recommendation system, and the rating predicted by the latent factor model requires to add overall average rating and biases to complete further necessary improvements. Besides, the ratings predicted by the proposed model should be in the range of 0-5, which can get more accurate results.

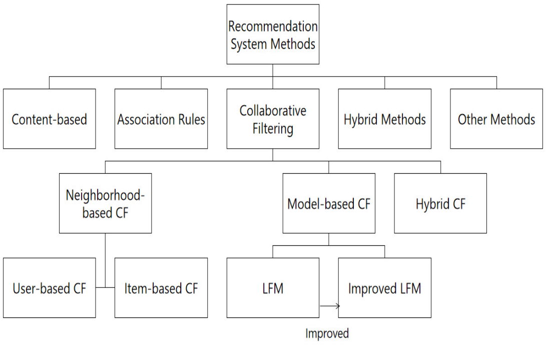


그림. 2 추천 시스템의 방법론적 분류
Fig. 2 MethodoloGical Classification Of The Recommendation System

This paper is organized as follows. In the following section II, the related works are described to show current methods for the recommendation system . In section Ⅲ, the proposed system and methods are shown and described in detail. Results and analyses are shown in section Ⅳ, and conclusions are described in section Ⅴ.

Ⅱ. Related works

Since the early 1990s, personalized recommendation systems get attention and are widely used in the online platform. For example, YouTube has provided personalized recommendations based on user behavior and has become the most popular online video community in the world[5]. In 2006, Netflix held a machine learning and data mining recommendation system competition. This competition greatly promoted the development of the recommendation system[6].

The recommendation system still faces many challenges such as cold start, sparse data, changing data, user preferences, privacy protection, etc[5]. Cold start can be classified into three categories such as new users, new items and new platform. It is difficult to recommend items to new users when the recommendation system does not have any information related to their past behaviors or it might be possible that they have not rated any item yet, so their taste are unknown to the system. Sparse data leads to increased difficulty in recommendation, and customers need a real-time recommendation when data and user preference change. Privacy is one of the most important issues for users. In order to give the most accurate recommendation to the user, the system must use the most privacy information of user, including age, sex, hobbies, the past watch history, the address, etc.

The advantages and disadvantages of various recommendation system methods coexist. Content-based filtering[7] doesn't need any data about other users since the recommendations are specific to this user. Association rule-based[8] recommendation system is easy to implement, but unifying product names is a big challenge. For example, sneakers and gym shoes have the different name but same meanings, and the recommendation system does not know that they are the same item. Memory-based collaborative filtering[9] is easy implementation, but a large amount of memory and CPU time are required. In 1994, a team from the University of Minnesota launched the first automated recommendation system using collaborative filtering as the leading technology recommendation system[6]. Amazon published an item-based joint filtering paper in 2003[10]. Model-based collaborative filtering[11] does not require so much memory and CPU time and has better sparsity and scalability. But the disadvantage is that the cost is too high for improved performance of recommendation.

Simple Python Recommendation System Engine(Surprise) is a famous package in python, which is used to build and analyze recommendation system. Surprise has two functions such as grid search and cross-validation. Gridsearch is a technique that attempts to compute the best values for a given parameters. This is an exhaustive search performed on specific parameter values of the model. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The recommendation system runs a cross-validation procedure for the given grid search parameters to avoid overfitting.

There is always the problem of data sparseness in the movie recommendation system, but the latent factor model has become popular algorithm in recent years because it can obtain personalized recommendation results with higher accuracy even in the huge data sparseness. In this paper, the latent factor model adds overall average rating, user bias, movie bias and rating prediction range adjustment. Besides, Root Mean Square Error(RMSE) is calculated because it is the most commonly used indicators to measure the accuracy of variables.

Ⅲ. The proposed system configuration and methodology

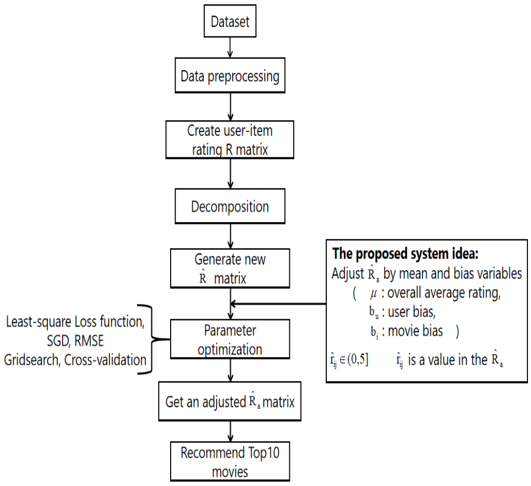

Dataset
Data preprocessing
Create user-item rating R matrix
Decomposition
Generate new $\hat{R}$ matrix
Least-square Loss function, SGD, RMSE Gridsearch, Cross-validation
Parameter optimization
Get an adjusted $\hat{R}_a$ matrix
Recommend Top10 movies
The proposed system idea:
Adjust $\hat{R}_a$ by mean and bias variables
(μ : overall average rating,
b_u : user bias,
b_i : movie bias)
$\hat{r}_{ij} \in (0,5]$
$\hat{r}_{ij}$ is a value in the $\hat{R}_a$

그림. 3 추천 시스템 흐름도
Fig. 3 Recommendation system flow chart

Fig. 3 shows the recommendation system flow chart in this paper. MovieLens[12] is a web-based research recommendation system that debuted in Fall 1997 and MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. An extension of the Movielens 100K dataset contains 100000 ratings for 943 users, 1682 movies. It includes many attributes, such as user id, movie id, rating, title, genres. The data sets used in the proposed system are mainly made up of two data files in MovieLens, one describes the rating data of the users and movies, and the other describes the detailed information about the movies.

In the data preprocessing, the proposed model merges two data sets and changes the 0 and 1 values in the movie data to genres. Rating distribution trend graph, the most popular 10 movies and genres, the favorite movies and genres of recommended users can be obtained after data processing.

The next step is to generate a user-item rating R matrix for the latent factor model. The latent factor model[13][14][15] is a model-based collaborative filtering and latent-factor based approaches relate users and items in a low-dimensional latent feature space K $\in$ R^d , where R is user-item rating matrix, d is the dimension of the latent feature space and K describes the hidden characteristics of movies and users.

In statistics, the least-square loss function in equation(1) is used for parameter estimation by calculate the difference between the estimated value and the true value. Equation 1 shows residual sum of squares plus a penalized term to prevent overfitting:

$$L = \sum_{r_{ui} \in R_{train}} (r_{ui} - \widehat{r_{ui}})^2 + \lambda (b_i^2 + b_u^2 + \| p_u \|^2 + \| q_i \|^2) \qquad (1)$$

where λ is a constant and named shrinkage penalty, or regularization coefficient. r_{ui} represents the value of u to i in the original rating matrix, and $\widehat{r_{ui}}$ represents the value of u to i in the predicted rating matrix. In addition, b_i denotes movie i's bias, b_u denotes user's bias.

The optimization of the least-square loss function is achieved by using stochastic gradient descent(SGD) algorithm. The algorithm updates the parameters by taking steps in the negative direction of gradient of the objective function, which directly leads towards a local minimum in equation(2) to (5).

$$\frac{\partial L}{\partial p_{uk}} = -2(r_{ui} - \sum_{k=1}^{K} p_{u,k} q_{k,i}) q_{k,i} + 2\lambda p_{u,k} \qquad (2)$$

$$\frac{\partial L}{\partial q_{k,i}} = -2(r_{ui} - \sum_{k=1}^{K} p_{u,k} q_{k,i}) p_{u,k} + 2\lambda q_{k,i} \qquad (3)$$

$$p_{u,k} = p_{u,k} + \alpha((r_{ui} - \sum_{k=1}^{K} p_{u,k} q_{k,i}) q_{k,i} - \lambda p_{u,k}) \tag{4}$$

$$q_{k,i} = q_{k,i} + \alpha((r_{ui} - \sum_{k=1}^{K} p_{u,k} q_{k,i}) p_{u,k} - \lambda q_{k,i}) \tag{5}$$

where α is the step size (also called the learning rate) in the process of SGD.

Gridsearch and cross-validation are used in parameter optimization part. The proposed recommendation system runs a cross-validation procedure for a given algorithm, reporting accuracy measures and RMSE is used to report accuracy.

The proposed model obtains the final adjusted rating $\widehat{R_a}$ matrix after traversing all data training. The last step is to make personalized recommendations for users of 10 movies based on the predicted rating matrix.

The RMSE is one of the most commonly used method to measure the recommendation system, which is used to calculate the closeness between the estimated value and the original observed value. As the number of training increases, the RMSE becomes smaller and smaller, and it is meaningless when RMSE is close to 0, indicating that the predicted score value is not different from the actual score value. Further, the larger the RMSE value, the more the predicted score differs greatly from the original score.

$$RMSE = \sqrt{\frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} (\overline{\mathrm{R}_{\mathrm{i}}} - \mathrm{R}_{\mathrm{i}})^2} \tag{6}$$

In equation (6), $\overline{R_i}$ denotes the predicted rating and R_i is the actual rating of the movie. Furthermore, N is the number of all data sets.

Ⅳ. Simulation results and analysis

Table 1 shows that U.data file, which is the user rating file of the movie, including user id, movie id, rating, and timestamp. Table 2 shows the u.item file, which contains 24 columns of information about the movie, from left to right each column means movie id, movie title, release date, video release date, IMDb URL and the last 19 fields which are the movie genres. The genres are represented by 19 features, which represent unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western. The movie genre is represented by 1 and 0, for example, 0010000010000001000 means that the movie belongs to adventure|drama|romance genres.

표 1. U.data 파일 모맷
Table 1. u.data file format

user	item	rating	timestamp
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

표 2. U.item 파일 포맷
Table 2. u.item file format

mid	mtitle	0	1	2	3-21
1	Toy Story(1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	......
2	GoldenEye (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?GoldenEye%20(...	......
3	Four Rooms (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Four%20Rooms%...	......
4	Get Shorty (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Copycat%20(1995)	......
5	Copycat (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Copycat%20(1995)	......

After the model merges the above two files with the same meaning and the model converts 0 and 1 values into genres, five useful contents for model analysis are shown in Table 3, such as user, mid, rating, mtitle, genres. For example, movie No.242 whose genre is comedy is named kolya, and user 196 likes that movie with 3 star.

표 3. 조합된 MovieLens 데이터
Table 3. Combined MovieLens data

user	mid	rating	mtitle	genres
196	242	3	Kolya(1996)	Comedy
186	302	3	L.A. Confidential(1997)	Crime, Film-Noir, Mystery, Thriller
22	377	1	Heavyweights(1994)	Children's, Comedy
244	51	2	Legends of the Fall(1994)	Drama, Romance, War, Western
166	346	1	Jackie Brown(1997)	Crime, Drama

Fig. 4 shows most movies are rated between 3 and 5 stars.

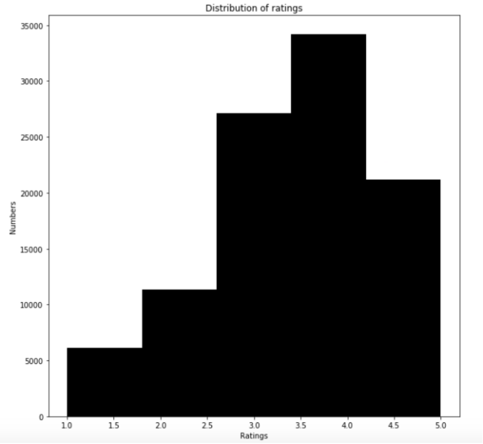
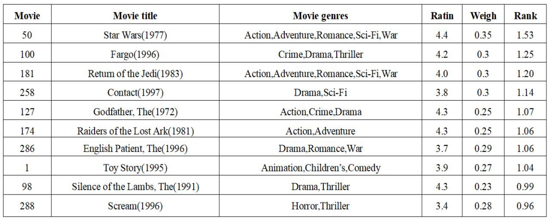


그림. 4 평점 분포
Fig. 4 Distribution of ratings

Table 4 is a list of the top 10 popular movies, and Fig. 5 shows the types of popular movies in a word cloud. The favorite ranking of movies depends on the average rating of each movie multiplied by the weight. The larger the text size is, the more users like the type of movies, such as drama, comedy, action, romance, adventure and thriller that are popular among the crowd.

표 4. 상위 10개 영화
Table 4. Top 10 movies



Movie	Movie title	Movie genres	Ratin	Weigh	Rank
50	Star Wars(1977)	Action,Adventure,Romance,Sci-Fi,War	4.4	0.35	1.53
100	Fargo(1996)	Crime,Drama,Thriller	4.2	0.3	1.25
181	Return of the Jedi(1983)	Action,Adventure,Romance,Sci-Fi,War	4.0	0.3	1.20
258	Contact(1997)	Drama,Sci-Fi	3.8	0.3	1.14
127	Godfather, The(1972)	Action,Crime,Drama	4.3	0.25	1.07
174	Raiders of the Lost Ark(1981)	Action,Adventure	4.3	0.25	1.06
286	English Patient, The(1996)	Drama,Romance,War	3.7	0.29	1.06
1	Toy Story(1995)	Animation,Children's,Comedy	3.9	0.27	1.04
98	Silence of the Lambs, The(1991)	Drama,Thriller	4.3	0.23	0.99
288	Scream(1996)	Horror,Thriller	3.4	0.28	0.96

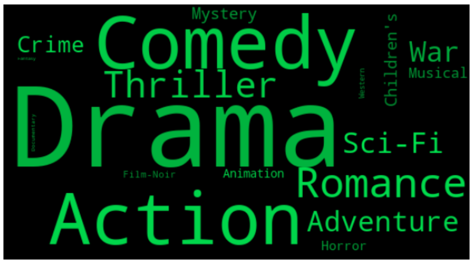
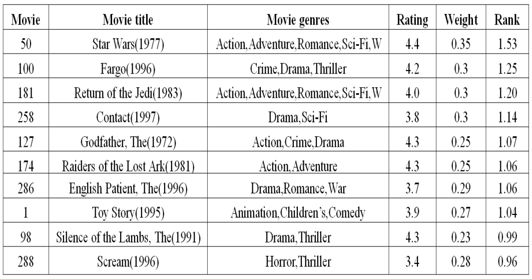


그림. 5 인기 영화 장르의 워드클라우드
Fig. 5 Word cloud for popular movie genres

Table 5 shows only 10 favorite movies of user 1 with rating5 from the dataset, and Fig. 6 is the word cloud of genres that user 1 likes.

표 5. 사용자 1의 선호 영화
Table 5. Favorite movies for user 1



Movie	Movie title	Movie genres	Rating	Weight	Rank
50	Star Wars(1977)	Action,Adventure,Romance,Sci-Fi,W	4.4	0.35	1.53
100	Fargo(1996)	Crime,Drama,Thriller	4.2	0.3	1.25
181	Return of the Jedi(1983)	Action,Adventure,Romance,Sci-Fi,W	4.0	0.3	1.20
258	Contact(1997)	Drama,Sci-Fi	3.8	0.3	1.14
127	Godfather, The(1972)	Action,Crime,Drama	4.3	0.25	1.07
174	Raiders of the Lost Ark(1981)	Action,Adventure	4.3	0.25	1.06
286	English Patient, The(1996)	Drama,Romance,War	3.7	0.29	1.06
1	Toy Story(1995)	Animation,Children's,Comedy	3.9	0.27	1.04
98	Silence of the Lambs, The(1991)	Drama,Thriller	4.3	0.23	0.99
288	Scream(1996)	Horror,Thriller	3.4	0.28	0.96

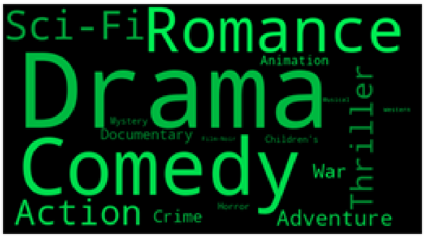


그림. 6 사용자 1의 선호 영화 장르의 워드클라우드
Fig. 6 Word cloud for user 1 favorite movies genres

For the proposed model, parameters are experimented with K from 10, 20, 30, 40, 50, and for epochs, 400, 500, 600 are explored. For learning rate, we explored three different choices of [0.001, 0.005, 0.01] and [0.05, 0.1, 0.3, 0.5] are explored in shrinkage penalty. GridSearchCV can try all parameter combinations and report the best measurement parameters. In selecting the tuning parameters, 5-fold cross-validation is performed, and chooses RMSE as the measurement method. Table 6 shows the simulation results have the best RMSE 0.671 for learning rate = 0.001, shrinkage penalty = 0.1, epochs = 500 and K = 50, and the RMSE of the proposed algorithm is slightly lower than other papers[16][17].

표 6. 최적화 모델의 파라미터 값
Table 6. Parameters of the optimal model

K	Epochs	Learning rate	Shrinkage penalty	biased	Predicted rating adjustment	RMSE
50	500	0.001	0.1	True	True	0.671

Table 7 shows a list of 10 movies recommended to user 1 using the proposed model.

표 7. 사용자1에 대한 추천 영화
Table 7. Recommended movies to user 1

Recommendations for user 1:	
Ranking	Movie Title
1	Close Shave, A (1995)
2	Pather Panchali (1955)
3	Secrets & Lies (1996)
4	L.A. Confidential (1997)
5	Ran (1985)
6	Faust (1994)
7	Lawrence of Arabia (1962)
8	City of Lost Children, The (1995)
9	Night on Earth (1991)
10	Two or Three Things I Know About Her (1966)

Ⅴ. Conclusions

This paper proposed the improved latent factor model with adjustment of overall average rating, user bias, movie bias and prediction rating range. The datasets in MovieLens are merged to make a rating matrix. The least-squares loss function and SGD are used to reduce the predicted error, and Surprise package is used to determine the optimal parameters. The simulation results show that RMSE is 0.671 and the latent factor model method adjustment of mean and bias variables is effective.

References

[1] B. Patel, P. Desai, and U. Panchal, "Methods of recommender system: A review," *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, 2017, pp. 1-4.

[2] S. Sharma, A. Sharma, Y. Sharma, and M. Bhatia, "Recommender system using hybrid approach," *2016 International Conference on Computing, Communication and Automation (ICCCA)*, Noida, India, 2016, pp. 219-223.

[3] K. Shah, A. Salunke, S. Dongare, and K. Antala, "Recommender systems: An overview of different approaches to recommendations," *2017*

International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2017, pp. 1-4.

[4] L. Chen and C. Kim, "Design of E-Commerce Service on The Web Based on Data Mining", *J. of the Korea Institute of Electronic Communication Science,* vol. 15, no. 04, Aug. 2020, pp. 703-708.

[5] S. Jain, A. Grover, P. Thakur, and S. Choudhary, "Trends, problems and solutions of recommender system," *International Conference on Computing, Communication & Automation,* Noida, India, 2015, pp. 955-958.

[6] W. Liu, B. Wang, and D. Wang, "Improved Latent Factor Model in Movie Recommendation System," *2018 International Conference on Intelligent Autonomous Systems (ICoIAS),* Singapore, Singapore, 2018, pp. 101-104.

[7] R. Manjula and A. Chilambuchelvan, "Content Based Filtering Techniques in Recommendation System using user preferences," *Int. J. Innov. Eng. Technol.,* vol. 7, no. 4, 2016, pp. 149-154.

[8] H. Ceong and C. Park, "Enzyme Metabolite Analysis Using Assoication Rules Mining", *J. of the Korea Institute of Electronic Communication Sciences,* vol. 11, no. 10, 2016, pp. 969-982.

[9] S. Gong, H. Ye, and H. Tan, "Combining Memory-Based and Model-Based Collaborative Filtering in Recommender System," *2009 Pacific-Asia Conference on Circuits, Communications and Systems,* Chengdu, China, 2009, pp. 690-693.

[10] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *in IEEE Internet Computing,* vol. 7, no. 1, 2003, pp. 76-80.

[11] S. Kim and D. Cho and S. Bracha, "Design and Implementation of Hashtag Recommendation System Based on Image Label Extraction using Deep Learning", *J. of the Korea Institute of Electronic Communication Science,* vol. 15, no. 04, Aug. 2020, pp. 709-716.

[12] Z. Jun-Yao, Z. Zi-Qian, S. Ji-Yun, and C. Jie-Hao, "Solutions to cold-start problems for latent factor models," *2017 17th International Symposium on Communications and Information Technologies (ISCIT),* Cairns, Australia, 2017, pp. 1-5.

[13] Z. Zhang, Y. Xiao, W. Zhu, X. Jiao, K. Zhu and H. Deng, "A context-aware recommendation system based on latent factor model," *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD),* Guilin, China, 2017, pp. 1-6.

[14] J. Zeng, "Latent Factor Models for Recommender Systems and Market Segmentation Through Clustering", Master Thesis, The Ohio State University, 2017.

[15] H. Nguyen and T. Dinh, "A Modified Regularized Non-Negative Matrix Factorization for MovieLens," *2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future,* Ho Chi Minh City, Vietnam, 2012, pp. 1-5.

[16] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," *In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM,* New York, U.S.A, 2008. pp. *426 -434.*

[17] M. Khoshneshin and W. N. Street, "Collaborative filtering via euclidean embeddin g", *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems,* Barcelona Spain, Sep. 2010, pp. 87-94.

저자 소개

Chen Ma

2017년. Beijing Information Science and Technology University, Applied Statistics
2021. 현재 전남대학교 대학원 컴퓨터공학과 재학
※ 관심분야 : Big data, Recommendation System

김강철(Kangchul kim)

1981년 서강대학교 전자공학과 학사
1983년 서강대학교 전자공학과 석사
1996년 경상대학교 전자공학과 박사
현재 전남대학교 전기전자통신컴퓨터공학부 교수
※ 관심분야 : 임베디드시스템, NoC, IoT
Pattern Recognition