applied sciences

MDPI

Article

Movie Recommendation through Multiple Bias Analysis †

Tae-Gyu Hwang and Sung Kwon Kim *

School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, Korea; tghwang@alg.cse.cau.ac.kr

* Correspondence: skkim@cau.ac.kr; Tel.: +82-02-825-6997

† This paper is an extended version of our paper published in "Multi-Label Bias-Based Predictor".

Abstract: A recommender system (RS) refers to an agent that recommends items that are suitable for users, and it is implemented through collaborative filtering (CF). CF has a limitation in improving the accuracy of recommendations based on matrix factorization (MF). Therefore, a new method is required for analyzing preference patterns, which could not be derived by existing studies. This study aimed at solving the existing problems through *bias analysis*. By analyzing users' and items' biases of user preferences, the bias-based predictor (BBP) was developed and shown to outperform memory-based CF. In this paper, in order to enhance BBP, multiple bias analysis (MBA) was proposed to efficiently reflect the decision-making in real world. The experimental results using movie data revealed that MBA enhanced BBP accuracy, and that the hybrid models outperformed MF and SVD++. Based on this result, MBA is expected to improve performance when used as a system in related studies and provide useful knowledge in any areas that need features that can represent users.

Keywords: rating prediction; collaborative filtering; movie recommendation; *bias analysis*

check for updates

Citation: Hwang T.-G.; Kim S.K. Movie Recommendation through Multiple Bias Analysis. *Appl. Sci.* **2021**, *11*, 2817. http://doi.org/10.3390/app11062817

Academic Editor: Álvaro Tejeda Lorente

Received: 26 February 2021
Accepted: 18 March 2021
Published: 22 March 2021

1. Introduction

1.1. Research Flow of Recommendation System

Search engines have developed, thanks to the spread of the Internet, and the online transaction market has developed into a digital content market, thanks to the spread of smart devices. Consequently, more costs are now required for users to find information. Information retrieval (IR) [1] is a field to search for documents that are suitable for users' information needs, and recommender systems (RSs) are an application field that is derived from IR. The RS development has gone through a process, like that of IR, since RS developed and applied IR.

An RS is an agent that recommends items that are suitable for users, and collaborative filtering (CF) is typically used. Regarding the analysis method, CF is divided into memory-based CF (User-based CF (UBCF) [2,3]), Item-based CF (IBCF) [3–6]), model-based CF (factorization model [7–10]), content-based CF [11], and context-aware CF [12], and there is a hybrid model that synergizes the foregoing CF models to complement their advantages and disadvantages [13,14].

The main issues of CF include cold start [15,16], scalability [2,6,17], and accuracy. The cold start refers to a problem in which no recommendation is possible due to no record of decision-making for new users or new items—meaning that no items can be recommended to new users, and new items cannot be recommended to any users. Content-based CF is a representative model for solving cold-start problems, and it enables recommendations through metadata analysis, even without a record of decision-making.

Scalability is a complex problem occurring in the real world and it entails difficulty in applying a small-scale research model to worksite operations. This problem is mitigated by simplifying or parallel processing the CF algorithm. In general, studies [2,6] have been conducted for executing CF algorithms in a distributed/parallel processing environment, such as Hadoop [18] or Spark [19]. However, because the technical difficulty for parallelizing the algorithm is high, it is difficult to apply sophisticated or the latest CF. Apache Mahout [20] is an open source application programming interface (API) that is represented by CF in a distributed environment.

Accuracy means how accurately the rated scores are predicted, and the matrix factorization (MF) model [8] has been evaluated as the most excellent so far. Furthermore, models aiming at extreme accuracy consider the viewpoints of ranking [10], zero-order [21], diversity [22,23], etc., together. The core common feature of studies related to the foregoing is the enhanced accuracy obtained by deepening the existing analysis. Additionally, CF accuracy at a small adjustment level reflecting optimization according to domains was shown from NetFlix prize 09 [8] to neural collaborative filtering (NCF) [24]. Representative causes of the performance limit include the lack of data and the obsolescence of analysis algorithms.

Deep learning is attracting attention for solving the existing performance limitations in the fields of speech recognition, image processing, and natural language processing, and recent study trends are also applying it to RS [24–26]. RS combined with deep learning is divided into text and rating analyses, and NCF is a representative study case. Rating analysis builds a model using the latent factors that are decomposed through factorization models (MF, Singular value decomposition (SVD) [9], SVD++ [8]) as inputs of neural networks (NNs), and NCF belongs to the foregoing. Text analysis means analyzing a user's review data, and the core is building it using natural language processing (NLP), like Word2Vec or BERT (Bidirectional Encoder Representations from Transformers) [27]. Even recently, RSs using deep learning showed good performance results [25,26], and RSs can be evaluated as a method for solving the two elements specified as the causes of limitations in the existing performance. However, because rating data are a pattern suitable for regression, there is a limitation in applying deep learning, and deep learning algorithms, such as NetFlix prize 09', will be limited in performance [28].

1.2. Research Motivation

Users' ratings (decision-making) of movies are determined by the effects of various elements, such as the differences in the degree to which users give high or low scores on average, users' tastes, cinematic quality, popularity, etc. The traditional CF principle is finding correlations from the mathematical patterns appearing in ratings and searching for similar movies. The attributes of users and items are critical because individual subjectivity is reflected in movie selection. Because users have different tastes for age, gender, occupation, genre, director, and actor, the probability of selecting the same movies differs according to users. Elements, such as a mania for a certain genre or series, fans of the director or actor, for example, are reflected in movie selection.

Bias means learning to a side, b for axis movement in $y = ax + b$, distortion [29], custom [30], taste [31], and individual subjectivity [32,33]. Because bias exists in all data in reality, the bias is computed in data mining and machine learning for model optimization. For example, there are newspapers that write newspaper articles leaned to one side, and sample groups participating in a questionnaire survey or the processes of reducing the sample groups are affected by the bias. Additionally, for content (newspaper articles, music, movies, videos, webtoons, etc.), people's bias is reflected in their content rating or review. Alternatively, people's bias is reflected in other users' ratings or review evaluations. Although positive/negative views of bias conflict with each other, bias is among the facts in which users' preferences are clearly expressed.

Bias-Based Predictor (BBP) [32] is a method that reflects the foregoing to analyze bias. Appropriate bias means the average learning computed from fair criteria (the mediator). BBP proposed a method for finding moderators and outperformed memory-based CF by analyzing user and item biases. However, because decision-making, in reality, reflects more complex relationships than BBP, BBP should be expanded to analyze more relationships [33].

Multiple Bias Analysis (MBA) analyzed users' preference patterns centering on bias to find new clues, undiscovered by existing CFs, and tried to understand users' decision-making processes using metadata. The bias can be rationalized to the relevant user himself/herself, but not to other users. For example, evaluations of newspaper articles (comments) and postings on portal sites (blogs) may be sympathized by users who are like the relevant user by chance, but, in other cases, they are an act of forcing others. In particular, because the nature of bias is clearly distinguished in social and political issues, bias cannot be justified to others. Furthermore, the issue of dataset fairness should be considered together. The propensity of a certain portal site can be affected by the users constituting the portal site, and the model that is learned from this data can produce unfair results for certain users. Additionally, in contrast to the intention, a distorted result may occur while reducing the data size. MBA used BBP as a preceding study to consider these elements. BBP searches for pinpoint scores to alleviate this problem.

MBA for analyzing more relationships than BBP using metadata and a hybrid model that combines MBA and MF are proposed. The core of MBA is to find a feature that can represent users, and feature bias was analyzed by two ways; sympathized with or independent of other users. The experimental results using movie data showed 9.03%, 8.20%, 7.23%, 4.68%, and 0.97% higher accuracy of MBA than UBCF, IBCF, BBP, MF, and SVD++, respectively.

Section 2 reviews MF and BBP, Section 3 introduces MBA, Section 4 presents the experimental results, and Section 5 deals with conclusions.

2. Related Works

2.1. MF: Matrix Factorization

MF [8,10] is a sort of model-based CF. It became known through 'NetFlix Prize 09' and it has been considered the best model to date. Although SVD [9,10] is used because MF can decompose only square matrices, the two terms MF and SVD are used interchangeably.

When a rating matrix $\mathbb{R}^{m\times n}$ has been given for m users and n items, the core of learning is decomposing the original matrix $\mathbb{R}$ into several f-dimensional matrices (MF: $p_u, q_i \in \mathbb{R}^f$, SVD: $U, \Sigma, V \in \mathbb{R}^f$) and ensuring that the decomposed matrices are to the original matrix. Equation (1) is the rating prediction, Equation (2) is the objective function, and the model is learned through Equations (3)–(5). p_u is the users' latent factor (users' vector), q_i is the items' latent factor (items' vector), λ is regularization, and γ is the learning rate.

$$P^{MF}(u,i) = q_i^T p_u \tag{1}$$

$$\min_{p,q} \sum_{(u,i)\in\tau} \Big(r(u,i) - P^{MF}(u,i)\Big)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{2}$$

$$\epsilon = P^{MF}(u,i) - r(u,i) \tag{3}$$

$$q_i = q_i + \gamma(\epsilon \cdot p_u - \lambda \cdot q_i) \tag{4}$$

$$p_u = p_u + \gamma(\epsilon \cdot q_i - \lambda \cdot p_u) \tag{5}$$

SVD++ [8,10] refers to a model in which bias and temporal score [31] were added to MF [9]. Equation (6) is the prediction of rated scores, Equation (7) is the objective function, and the model is learned through Equations (8)–(13).

$$P^{SVD++}(u,i) = \mu + b_u + b_i + q_i^T \Big(p_u + |I_u|^{-0.5} \sum_{j\in I_u} y_i\Big) \tag{6}$$

$$\min_{p,q} \sum_{(u,i)\in\tau} \Big(r(u,i) - P^{SVD++}(u,i)\Big)^2 + \lambda(b_u^2 + b_i^2 + \|q_i\|^2 + \|p_u\|^2 + \|y_j\|^2) \tag{7}$$

$$\epsilon = P^{SVD++}(u,i) - r(u,i) \tag{8}$$

$$q_i = q_i + \gamma\{\epsilon \cdot (p_u + |I_u|^{-0.5} \sum_{j \in I_u} y_j) - \lambda \cdot q_i\} \tag{9}$$

$$p_u = p_u + \gamma(\epsilon \cdot q_i - \lambda \cdot p_u) \tag{10}$$

$$b_u = b_u + \gamma(\epsilon - \lambda \cdot b_u) \tag{11}$$

$$b_i = b_i + \gamma(\epsilon - \lambda \cdot b_i) \tag{12}$$

$$\forall j \in R(u) : y_j = y_j + \gamma(\epsilon \cdot |I_u|^{-0.5} \cdot q_i - \lambda \cdot y_j) \tag{13}$$

MF and SVD++ are implemented and distributed in many Open APIs. In this study, the experiments on MF and SVD++ were conducted using MyMediaLite [34].

2.2. BBP: Bias-Based Predictor

BBP [32] analyzes preferences based on the viewpoint of bias, assuming that bias and preferences are closely related to each other. The core of BBP is finding a pinpoint score to act as a fair moderator and analyzing user and item biases through the pinpoint score. The pinpoint score is expressed as s, and the model is learned, as shown in Algorithm 1 using Equation (21). Equation (14) shows the rating prediction of BBP.

$$P_{wt}(s,u,i) = s + b_{wt}(s,u) + b_{wt}(s,i) \tag{14}$$

P refers to the prediction score, wt refers to the weight type, u refers to the target user, i refers to the target item, s refers to the pinpoint score, and b refers to the bias. $P_{wt}(s,u,i)$ refers to the predicted score of item i of user u computed through s and wt, $b_{wt}(s,u)$ refers to the bias score of user u computed through s and wt, and $b_{wt}(s,i)$ refers to the bias score of item i computed through s and wt. Three weight types are used: weight based on rating frequency (RF), on amplified RF (ARF), and on logarithmic RF (LRF), as defined in Equations (15)–(20), below.

The difference between the pinpoint score and average rated score is used as the bias, and the reliability of the average rated score is reflected using the rating frequency as a weight. RF, ARF, and LRF were used as the types of weights of the rating frequency and they were indicated by substituting the wt digit into the equation. Equations (15)–(20) show the bias scores using the rating frequency weights.

RF was used to directly reflect the weight according to the rating frequency. The weight value is computed in a range of 0–1, depending on the rating frequency. The larger the size of the dataset, the higher the accuracy when compared to LRF.

$$b_{RF}(s,u) = \frac{|I(u)|}{|I(u)|+1} \times (\bar{r}(u) - s) \tag{15}$$

$$b_{RF}(s,i) = \frac{|U(i)|}{|U(i)|+1} \times (\bar{r}(i) - s) \tag{16}$$

$b_{RF}(s,u)$ is the bias score of user u that is computed through s and RF, $I(u)$ is the set of items rated by user u, and $\bar{r}(u)$ is the average of the rated scores by user u. $b_{RF}(s,i)$ is the bias score of item i computed through s and RF, $U(i)$ is the set of users who rated item i, and $\bar{r}(i)$ is the average of the rated scores of item i.

ARF was used to amplify and reflect the rating frequency weight of RF. The weight value is computed as -1 to 1. Because RF accuracy affects ARF, ARF has a problem that the deviations of accuracy among users are bigger than RF—it shows more accurate results for users for whom it predicts scores well while showing more inaccurate results for users for whom it predicts scores poorly.

$$b_{ARF}(s,u) = \left(2 \times \frac{|I(u)|}{|I(u)|+1} - 1\right) \times (\bar{r}(u) - s) \tag{17}$$

$$b_{ARF}(s,i) = \left(2 \times \frac{|U(i)|}{|U(i)|+1} - 1\right) \times (\bar{r}(i) - s) \tag{18}$$

LRF was used to reflect the weight of the rating frequency of RF in log form. The core reflects the log of the value normalized by the maximum rating frequency, and the weight value is computed in a range of 0–1. The accuracy is generally high, and, the smaller the size of the dataset (MovieLens 1M or less), the better the results shown.

$$b_{LRF}(s,u) = \frac{\log|I(u)|}{\log|RFMax(User)|} \times (\bar{r}(u) - s) \tag{19}$$

$$b_{LRF}(s,i) = \frac{\log|U(i)|}{\log|RFMax(Item)|} \times (\bar{r}(i) - s) \tag{20}$$

$RFMax(User)$ refers to the rating frequency of user v with $\max_{v\in User}|I(v)|$ and maximum rating frequency. Likewise, $RFMax(Item)$ refers to the rating frequency of item j with $\max_{j\in Item}|U(j)|$ and maximum rating frequency.

The pinpoint score is computed while using Equation (21) as the objective function.

$$\epsilon_{wt}(s_x) = \sqrt{\frac{1}{|\tau|}\sum_{(u,i)\in\tau}\Big(P_{wt}(s_x,u,i) - r(u,i)\Big)^2} \\ + (s_x-\mu)^2\Big(\sum_u b_{wt}(s_x,u)^2 + \sum_i b_{wt}(s_x,i)^2\Big) \tag{21}$$

τ is the set of all rated scores included in the training set, and μ is the average rated score of training set τ. The core of Equation (21) is inducing the search of the selected pinpoint score s_x adjacent to μ and Equation (21) was designed to add the penalty score in order to avoid overfitting to root mean square error (RMSE).

Algorithm 1 Bias-Based Predictor's (BBP's) recursive learning of pinpoint score s.

Require: divide number n, training-set τ
Ensure: selected pinpoint score s

```
$s = minBound = maxBound = 0$;
$range =$ costFunc($\mu$);
$minBound = \mu - range/2, maxBound = \mu + range/2$;
$s =$ RecursiveLearning($\mu, range, minBound, maxBound, n$);

function RecursiveLearning($ps, range, minBound, maxBound, n$)
    $interval = range/(n-1)$;
    $arrPS =$ new $Array(n)$;
    $arrCost =$ new $Array(n)$;
    for $x = 0$ to $n$ do
        $arrPS[x] = ps + (interval * x)$;
        $arrCost[x] =$ costFunc($arrPS[x]$);
    end for

    if hasBetterCandidate($ps, range$,$arrPS, arrCost$) then
        nextBoundary(&$ps$, &$minBound$, &$maxBound$, $arrPS, arrCost$);
        $range = maxBound - minBound$;
        $s =$ RecursiveLearning($ps, range, minBound, maxBound, n$);
    else
        $idx = \arg\min_x arrCost[x]$
        $s = arrPS[idx]$;
    end if

    return $s$
end function
```

As with Algorithm 1, BBP is to find pinpoint score s_x through recursive search. First, the initial search range is set to $\mu \pm (\epsilon_{wt}(\mu)/2)$ (line 3) and an values of search (line 4) is entered. Thereafter, the search range is divided into n ranges and the $\epsilon_{wt}(s_x)$ for the evenly spaced reference points (s_x) is computed (lines 7–12). Here, if there is an $\epsilon_{wt}(s_x)$ smaller than that at the previous time point $\epsilon_{wt}(\mu)$, two reference points are selected (line 15), the two reference points are selected as a new search range (line 16), the search range is divided into n ranges, and the foregoing process is recursively repeated (line 6–18). The recursive search is repeated until $\epsilon_{wt}(s_x)$ is minimized (lines 7–18), and $\epsilon_{wt}(s_x)$ are simultaneously computed through parallelization (line 12). The recursive search showed an accuracy that was close to that of gradient descent (GD) and it was optimized faster than GD through parallelization.

Lines 1–4 takes $O(|\tau|)$ time to compute μ. In *RecursiveLearning()*, each call of *costFunc()* in line 12 takes $O(|\tau|)$ time and, thus, lines 10–13 requires $O(n \cdot |\tau|)$ time. *RecursiveLearning()* repeats line 18 as many times until the condition of line 15 is satisfied. Therefore, the complexity of *RecursiveLearning()* is $O(n \cdot |\tau| \cdot No.of Recursions)$. The computation of *costFunc()* presented in line 12 can be parallelized by applying parallel tasks to reduce the execution time of algorithm.

3. MBA: Multiple Bias Analysis

Users' and movies' attributes affect users' decision-making for movies. However, MBA could not but differ from BBP, because MBA limits users' attributes to age, gender, and occupation, and movies' attributes to actors, directors, and genres, since MBA is built with only the information that is provided by the experimental dataset.

When a user selects a movie, variables, such as whether he/she is a fan of the actors or the director, his/her preference for a certain movie genre, empathy according to the user's gender or occupation, etc., are involved in the selection [33]. Because BBP only analyzes user and item biases, it cannot analyze user decision-making regarding multiple biases. The core of MBA is that it expanded BBP to analyze user decision-making regarding multiple biases.

3.1. Vanilla Model

From Figure 1, MBA was expanded to reflect the generalized feature (GF) and personalized feature (PF) on BBP, and Equation (22) shows the rating prediction of the Vanilla model.

$$P_{wt}^{VM}(s,u,i) = s + b_{wt}(s,u) + b_{wt}(s,i) + GF(s,u,i) + PF(s,u,i) \tag{22}$$

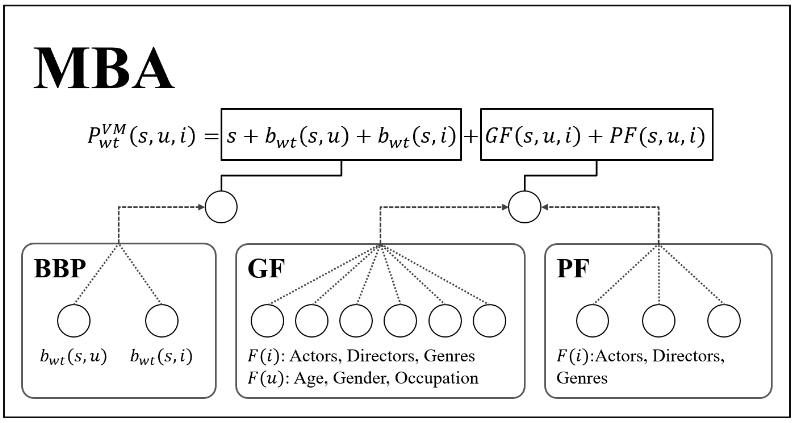
MBA
$P_{wt}^{VM}(s,u,i) = s + b_{wt}(s,u) + b_{wt}(s,i) + GF(s,u,i) + PF(s,u,i)$
BBP
$b_{wt}(s,u)$
$b_{wt}(s,i)$
GF
$F(i)$: Actors, Directors, Genres
$F(u)$: Age, Gender, Occupation
PF
$F(i)$:Actors, Directors, Genres

Figure 1. Summary of Multiple Bias Analysis (MBA).

The wt and s that are used in Equation (22) are the weight type and pinpoint score introduced in Section 2.2. MBA optimizes s using Algorithm 1 in the preprocessing stage before computing GF and PF. GF means a bias (global bias) sympathized with other users, and PF means a bias (local bias) independent of other users. Regarding Jim Carrey, GF refers to the public's tendency toward Jim Carrey, and PF refers to the personal tendency toward Jim Carrey.

GF represents the tendency of a feature that is sympathized with other users. The GF-Score was computed, as shown in Equation (23), and it was designed to reflect the distance ω between the feature bias $b(s,f)$ and user u computed from all users.

$$\begin{aligned} GF(s,u,i) &= \frac{1}{|UF(u)|}\sum_{f\in UF(u)} \omega(u,f)\cdot b(s,f) \\ &\quad + \frac{1}{|IF(i)|}\sum_{TF\subset IF(i)} \frac{1}{|TF|}\sum_{f\in TF} \omega(i,f)\cdot b(s,f) \\ &= \frac{1}{|UF(u)|}\sum_{f\in UF(u)} \omega(u,f)\cdot \left(\frac{\sum_{v,j\in R(f)} r(v,j)}{|R(f)|} - s\right) \\ &\quad + \frac{1}{|IF(i)|}\sum_{TF\subset IF(i)} \frac{1}{|TF|}\sum_{f\in TF} \omega(i,f)\cdot \left(\frac{\sum_{v,j\in R(f)} r(v,j)}{|R(f)|} - s\right) \end{aligned} \tag{23}$$

$UF(u)$ is a feature set for the user u's age, gender, and occupation, and $IF(i)$ is a feature set for the actor, director, and genre of item i. Here, because the elements of $IF(i)$ are a set (actor, director, genre), it was designed to add the average of the target feature TF. $R(f)$ is the set of rated scores belonging to feature f, $r(v,j)$ is the rated score for item j of user v, $\omega(u,f)$ is the weight of feature f of user u, and $\omega(i,f)$ is the weight of feature f of item i. The range of the values of the feature weight ω is -1 to 1, and the value that is computed through Equation (26) is used in Equations (28) and (29). In GF, ω is optimized to compute how close the feature f was to the popular bias.

If user u is a *30-year-old male student*, *UF(u)={Male, 30(Group-3), Student}* will be used. The first term of GF can be disjoined and written, as follows:

$$\begin{aligned} \frac{1}{|UF(u)|}\sum_{f\in UF(u)} \omega(u,f)\cdot b(s,f) =& \omega(u, AgeGroup(u))\cdot b(s, AgeGroup(u)) \\ &+\omega(u, Gender(u))\cdot b(s, Gender(u)) \\ &+\omega(u, Occupation(u))\cdot b(s, Occupation(u))) \end{aligned}$$

If item i is the movie *The Truman Show*, *IF(i)={(Jim Carrey, $\cdots$), Peter Weir, (Comedy, $\cdots$)}* will be used. The second term of GF can be disjoined and written, as follows:

$$\begin{aligned} \frac{1}{|IF(i)|}\sum_{TF\subset IF(i)} \frac{1}{|TF|}\sum_{f\in TF} \omega(i,f)\cdot b(s,f) = & \frac{1}{|Actors(i)|}\sum_{f\in Actors(i)} \omega(i,f)\cdot b(s,f) \\ &+\frac{1}{|Director(i)|}\sum_{f\in Director(i)} \omega(i,f)\cdot b(s,f) \\ &+\frac{1}{|Genres(i)|}\sum_{f\in Genres(i)} \omega(i,f)\cdot b(s,f) \end{aligned}$$

When feature f is Jim Carrey, the $b(s,f)$ that is used in GF-Score is the bias (global bias) of Jim Carrey regarding all users, and $\omega(u,f)$ is the distance between user u and $b(s,f)$ (or importance).

PF represents a personalized tendency. The PF-Score was designed to be computed, as shown in Equation (24), and it reflected the reliability ω for feature bias $b(s,f)$ and $b(s,f)$ computed only for itself. Here, because PF is computed only for itself, all $UF(u)$s become $\bar{R}(u) - s$. Therefore, it is computed using only $IF(i)$.

$$\begin{aligned} PF(s,u,i) =& \frac{1}{|IF(i)|} \sum_{TF \subset IF(i)} \frac{1}{|TF|} \sum_{f \in TF} \omega(u,f) \cdot b(u,f) \\ =& \frac{1}{|IF(i)|} \sum_{TF \subset IF(i)} \frac{1}{|TF|} \sum_{f \in TF} \omega(u,f) \cdot \left(\frac{\sum_{j \in R(u,f)} r(u,j)}{|R(u,f)|} - s \right) \end{aligned} \tag{24}$$

The range of ω values used in PF is -1 to 1, and the values that are computed through Equation (27) are used for learning of Equation (29). In PF, ω is optimized to compute how certain his/her own feature bias was. Because PF is only computed using his/her own ratings, as shown in Equation (24), individual subjectivity is computed. However, because the rating frequency is below that of GF computed from the ratings of other users, PF is less reliable than GF. Therefore, if the value of $\omega(u,f)$ of PF is close to 1, the feature bias can represent himself/herself and, otherwise, the feature bias is unreliable.

MBA learns by reflecting error values to ω as much as an arbitrary number of iterations ($Iter$), learns using all of the training set τ, and computes ϵ while using Equation (25).

$$\epsilon(s,u,i) = P^{VM}_{wt}(s,u,i) - r(u,i) \tag{25}$$

The $r(u,i)$ used in Equation (25) is an element of τ, and $P^{VM}_{wt}(s,u,i)$ is a predicted value. Equation (25) is computed to decrease the ω of MBA when $\epsilon(s,u,i)$ is below 0, and increase the ω of MBA when $\epsilon(s,u,i)$ exceeds 0. $\epsilon(s,u,i)$ is used in Equations (26) and (27), $\epsilon(s,u,i,GF)$, and $\epsilon(s,u,i,PF)$ are computed, reflecting the ratios of GF and PF.

$$\epsilon(s,u,i,GF) = \lambda \cdot \epsilon(s,u,i) \cdot \frac{|GF(s,u,i)|}{|GF(s,u,i)| + |PF(s,u,i)|} \tag{26}$$

$$\epsilon(s,u,i,PF) = \lambda \cdot \epsilon(s,u,i) \cdot \frac{|PF(s,u,i)|}{|GF(s,u,i)| + |PF(s,u,i)|} \tag{27}$$

$$\omega(u,f) = \tanh(\omega(u,f) - \epsilon(u,i,TF(f))) \tag{28}$$

$$\omega(i,f) = \tanh(\omega(i,f) - \epsilon(u,i,TF(f))) \tag{29}$$

The λ used in Equations (26) and (27) is the learning rate, and $|GF(s,u,i)|$ and $|PF(s,u,i)|$ are the absolute values of score. The value computed with Equation (26) learns about the ω used in GF through Equations (28) and (29), and the value computed with Equation (27) learns about the ω in PF through Equation (29). Because PF is computed using only $IF(i)$, only Equation (29) is used to learn the ω. The $TF(f)$ that is used in Equations (28) and (29) means the output of the upper sub-feature to which feature f belongs. If feature f is an action, the output of $TF(f)$ will become Genre. MBA learns to optimize ω through Equations (25)–(29) and, thereafter, predicts the rated scores using Equation (22).

3.2. Heuristics Approach

In MBA, the ω used in GF and PF showed deviations in accuracy regarding the initial values. The heuristics approach deals with the content that is related to the feature weight ω used in GF and PF to enhance MBA accuracy.

From Figure 2, the ω to which the heuristics approach is applied is divided into feature weight fw that is used in MBA and weight type bwt used in BBP.

- **Case A**: deature weight fw is initialized by reflecting the concept of the weight type of BBP introduced in Section 2.2.

Case A is initializing the feature weight ω used in GF and PF by reflecting the concept of the weight type of BBP and observing MBA accuracy. For $fw(RF)$, the initial value of the $\omega(u,f)$ of GF is computed, as follows:

$$\omega(u,f) = \frac{|R(f)|}{|R(f)|+1}$$

$R(f)$ is the set of rated scores that belong to feature f in the training set τ. Because the number of weight types of BBP is 3 in total (RF, ARF, and LRF), the experimental results for Case A have the number of cases of 9 ($9 = 3 \times 3 = |bwt| \times |fw|$).

- **Case B**: feature weight fw is initialized by unifying it into an arbitrary value.

In Case B, the feature weight $\omega(u,f)$ used for GF and PF was changed by 0.1 per time from -1 to 1 to observe changes in accuracy. The experimental results for Case B have the number of cases of 63 ($63 = 3 \times 21 = |bwt| \times |fw|$).

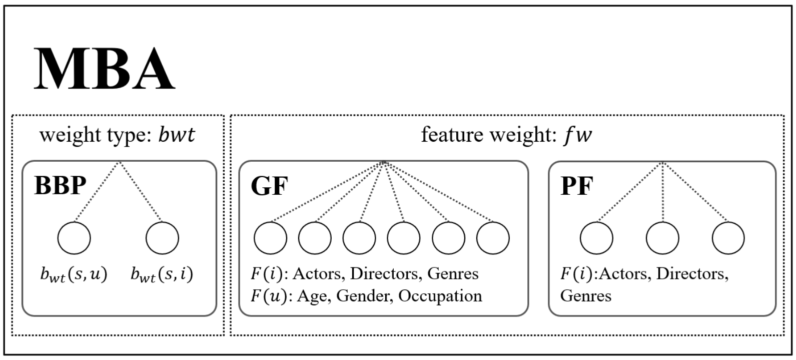
MBA
weight type: *bwt*
feature weight: *fw*
BBP
$b_{wt}(s,u)$
$b_{wt}(s,i)$
GF
$F(i)$: Actors, Directors, Genres
$F(u)$: Age, Gender, Occupation
PF
$F(i)$:Actors, Directors, Genres

Figure 2. Description of MBA's heuristics approach.

3.3. Hybrid Model

The Vanilla model is combined with MF to propose a hybrid model to enhance the prediction accuracy by complementing the shortcomings of the two models. The hybrid model is made by performing preprocessing of MF (Section 2.1) and Vanilla model (Section 3.1) to optimize them, respectively, and then combining the two models and the hybrid model is optimized thereafter. Equation (30) is the rating prediction of the hybrid model, Equation (31) is the objective function of the hybrid model, and Equations (32) and (33) are the learning of the hybrid model. Equation (32) is computed to reduce the bias score that is used for prediction in Equation (33) if the predicted value exceeds the actual value, and increase the bias value that is used for prediction if the actual value exceeds the predicted value. The λ in Equation (32) is the learning rate, and the f in Equation (33) is the attribute regarding u and i.

$$P_{wt}^{HM}(s,u,i) = s + b_{wt}(s,u) + b_{wt}(s,i) + GF(s,u,i) + PF(s,u,i) + q_i^T p_u \tag{30}$$

$$\min_{p,q,b(s,*)} \sum_{(u,i)\in\tau} \left(r(u,i) - P_{wt}^{HM}(s,u,i)\right)^2 \tag{31}$$

$$\epsilon = \lambda \cdot \left(P_{wt}^{HM}(s,u,i) - r(u,i)\right) \tag{32}$$

$$b(s,f) = b(s,f) - \epsilon \cdot b(s,f) \tag{33}$$

4. Experiment and Results

4.1. Dataset

MovieLens 100K dataset [35] was used as rating data to evaluate accuracy. However, since there is no information on actors and directors in the MovieLens dataset, as shown in Table 1, the information on actors and directors was supplemented using the HetRec2011 dataset [36]. In the process of creating the experimental dataset, items that could not be referenced from HetRec2011 were removed, and a total of 89 items and 3792 ratings that belonged to them were removed.

Table 1. Metadata in datasets.

		MovieLens 100K	HetRec 2011
User	Age	O	X
	Gender	O	X
	Occupation	O	X
Item	Actors	X	O
	Directors	X	O
	Genres	O	O

The experimental dataset used in our experiment is shown in Table 2. The ratings were aligned based on each user's timestamp, and were then composed into the five-fold cross-validation. For performance comparison, the average of the results of the evaluation scale computed from the five-set (five-fold cross-validation) was used. For the age groups of the users in Table 2, the users are divided into seven age groups as in MovieLens 1M.

Table 2. Dataset used in experiments.

MovieLens 100K (+HetRec 2011)		
	No. of users	943
	No. of items	1593
	No. of ratings	96,271
Users profile	No. of age groups	7
	No. of genders	2
	No. of occupations	21
Items metadata	No. of actors	14,856
	No. of directors	671
	No. of genres	19
5-fold cross-validation (training-set: 80%, testset: 20%)		

4.2. Evaluation Metrics

For the evaluation scale, the root mean square error (RMSE) was used, as shown in Equation (34), and smaller values mean higher accuracy. In Equation (34), P_{wt} refers to the predicted score that is computed through the model, and r refers to the rated score of the testset.

$$RMSE_{wt}(s) = \sqrt{\frac{\sum_{(u,i) \in Testset}(P_{wt}(s,u,i) - r(u,i))^2}{|Testset|}} \quad (34)$$

4.3. Results

MBA proposed the Vanilla model (Section 3.1) to improve BBP accuracy, tried the heuristics approach (Section 3.2) to improve the performance of Section 3.1, and attempted to supplement the shortcomings of the existing model through the hybrid model (Section 3.3). Therefore, the results of the MBA experiment were analyzed separately for the Vanilla model in Section 4.3.1, for the heuristic approach in Section 4.3.2, for the hybrid model in Section 4.3.3, and all of the experimental results were integrated in Section 4.3.4 to compare the accuracy.

4.3.1. Vanilla Model

MBA reconstructed the dataset to use metadata (Section 4.1). In this process, an experiment was conducted on the existing CF when considering changes in the environment of the dataset. Table 3 compares the previous study and the Vanilla model accuracies, and the optimized settings are also specified. RMSE results for different values of n and of $Iter$ are shown in Figures A1 and A2 of Appendix A.

In the performance comparison, the existing method was shown as $BBP(wt)$, and the proposed method was shown as $MBA(wt)$, and wt means Weight Type (Section 2.2). For instance, $BBP(RF)$ is a model using Equations (15) and (16), and $MBA(LRF)$ is a model using Equations (19) and (20). n is the value that is used to search the reference point, $Iter$ is the number of iterations for learning of ω, and λ is the learning rate used in Equations (26) and (27).

Table 3. Root mean square error (RMSE) comparison of the methods used in the experiment. RMSE results of five-fold cross-validation of every method can be viewed in more detail in Table A1 of Appendix A.

Method	RMSE	std	Model Environment
UBCF	1.054902	0.063437	Cosine similarity
IBCF	1.045365	0.049861	Pearson correlation coefficient
BBP(RF)	1.044754	0.053992	$n = 13$
BBP(ARF)	1.032625	0.044879	$n = 45$
BBP(LRF)	1.034218	0.049582	$n = 3$
MBA(RF)	1.058177	0.060937	$n = 3, Iter = 250, \lambda = 0.001$
MBA(ARF)	1.038827	0.041348	$n = 7, Iter = 500, \lambda = 0.001$
MBA(LRF)	1.028426	0.048566	$n = 11, Iter = 10, \lambda = 0.001$

In RF, the smaller the data size, the lower the pinpoint score optimization performance, because MBA computes multiple biases through pinpoint scores, inducing more errors. However, $MBA(ARF)$ and $MBA(LRF)$ outperformed UBCF and IBCF, and $MBA(LRF)$ showed the best results. Through these results, it is observed that the Vanilla model performance is determined by the pinpoint score. Because the RF performance is good when MovieLens exceeds 10M [32], RF should be reevaluated through experiments using a large dataset. Additionaly, in the process of analyzing the Vanilla model, variations in performance were observed, depending on the initial value of ω (GF, PF). This problem occurred in all Vanilla models, and the heuristics approach was conducted, as follows, to alleviate the problem presented.

4.3.2. Heuristics Approach

In order to improve MBA performance, experiments were conducted using the method presented in Section 3.2. Figure 3 shows the experimental results for Case A and Figure 4 for Case B. $bwt(RF)$ means that the weight type wt of BBP is used as RF, and $fw(RF)$ means that the feature weight fw was initialized into RF.

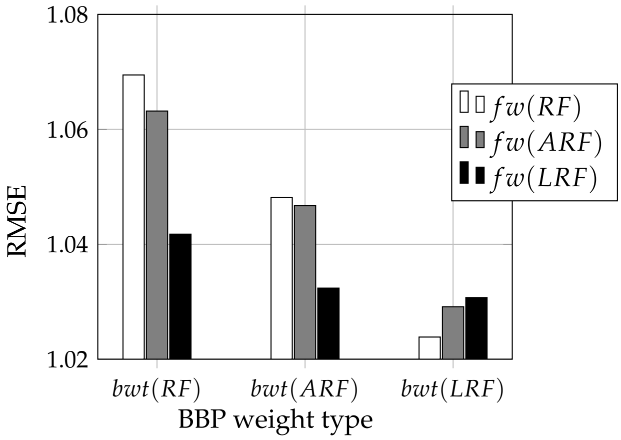

Figure 3. Comparison of MBA's RMSE for Case A.

It is observed that Case A performance is the best when $bwt(LRF)$ was conducted with $fw(RF)$ (Figure 3), and that the performance variations of the $bwt(LRF)$ series is the least.

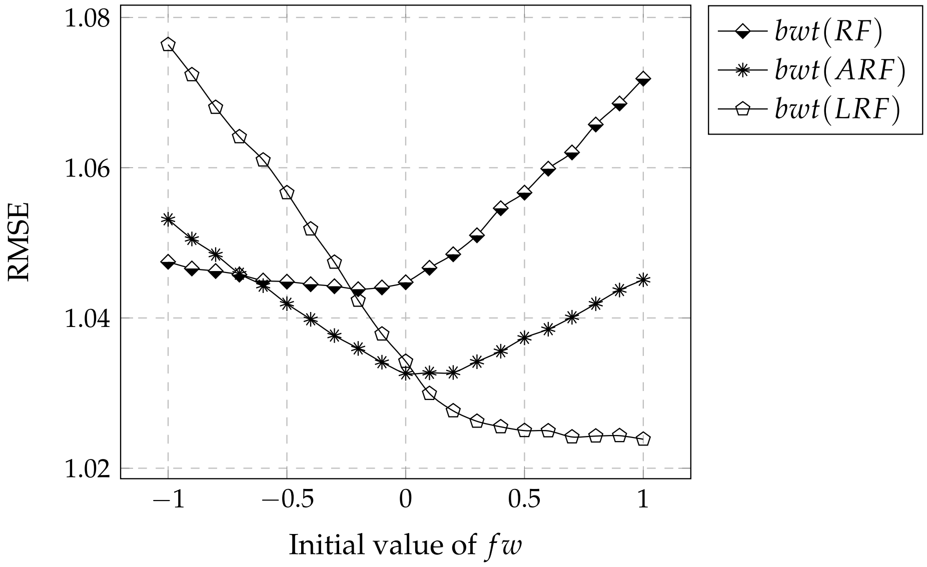

Figure 4. Comparison of MBA's RMSE for Case B.

Case B showed the results where the RMSE of $bwt(LRF)$ was the best (Figure 4), and RMSE converged when fw was 0.7 or higher. Additionally, Cases A and B showed better RMSE when compared to the Vanilla model. Based on the foregoing, the accuracy may be enhanced by improving the Vanilla model learning policy. Table 4 summarizes the heuristic approach results (-H) and compares the RMSE with the Vanilla model.

Table 4. Comparison of results (Heuristics approach).

Method	RMSE	Parameters		
		Iterations ($Iter$)	Divide-n	Learning Rate (λ)
MBA(RF)	1.058177	250	3	0.001
MBA(RF)-H	1.041759	800	13	0.001
MBA(ARF)	1.038827	500	7	0.001
MBA(ARF)-H	1.032367	200	45	0.001
MBA(LRF)	1.028426	10	11	0.001
MBA(LRF)-H	1.023850	10	3	0.001

From Table 4, it is seen that RMSE is improved through the heuristics approach. Additionally, when considering parameter n and $Iter$ used for learning, it is seen that the cost used for learning is lower.

4.3.3. Hybrid Model

MF and SVD++ also showed excellent performance in the dataset that was used in the experiment. MF and SVD++ were tested using the MyMediaLite [34] (Available: http://www.mymedialite.net/ (accessed on 20 March 2021)), and the hybrid model and RMSE are compared in Table 5. The hybrid model results outperformed Tables 3 and 4 and showed the best performance result when HybridSVD++(LRF) showed in Table 5. MF and SVD++ both showed results in which RMSE was improved when combined with MBA.

Table 5. Comparison of results (Hybrid model).

Method	RMSE	Hybrid Parameters		Factorization
		Iter	λ	Model Parameters
MF	1.006741	N/A	N/A	$\lambda = 0.01, \gamma = 0.005,$ $f = 10, Iter = 40$
HybridMF(RF)	1.023260	200	0.001	
HybridMF(ARF)	1.015874	200	0.001	
HybridMF(LRF)	0.996805	150	0.001	
SVD++	0.968985	N/A	N/A	MF: $\lambda = 0.01, \gamma = 0.005,$ $f = 10, Iter = 40$ Bias: $\lambda = 0.5, \gamma = 0.35,$
HybridSVD++(RF)	1.009454	200	0.001	
HybridSVD++(ARF)	1.001016	250	0.001	
HybridSVD++(LRF)	0.959599	150	0.001	

4.3.4. Results Summary

Figure 5 shows the results of integration, summarization, and comparison of the RMSE results of Sections 4.3.1–4.3.3. In Figure 5, the x-axis represents the weight type, and the y-axis represents the RMSE. Here, UBCF, IBCF, MF, and SVD++, which are unrelated to the weight type wt, were specified with a line graph, and BBP and MBA that were affected by wt were specified with a bar graph.

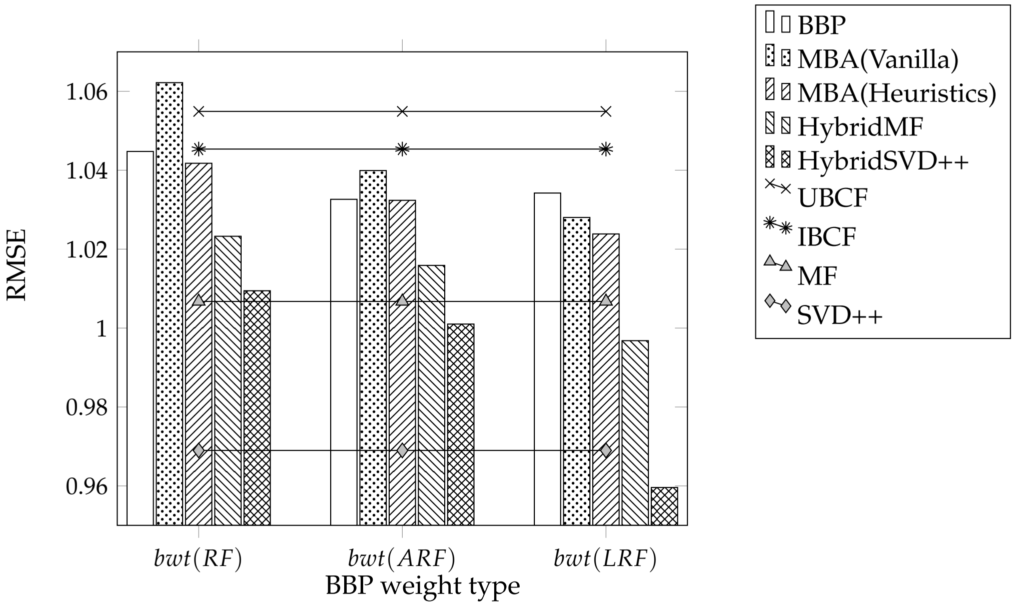

Figure 5. Comparison of RMSE.

Through Figure 5, an improvement in the overall performance of MBA is seen when the weight type was set to LRF. When RMSE is compared based on LRF, it is seen that the Vanilla model surpasses BBP, the heuristics approach surpasses the vanilla model, the HybridMF outperforms MF, and the HybridSVD++ outperforms SVD++. SVD++ is evaluated to be the best among model-based CFs due to rigorous verification for a long time by related researchers. Because MBA's HybridSVD++ (LRF) outperformed SVD++, there was a new preference pattern that was previously unconsidered in the bias, and MBA analyzed it in order to identify why its accuracy exceeded that of SVD++.

5. Conclusions

The performance limit of recommendation systems follows analysis algorithm and data problems. The analysis algorithm problem is "What algorithm fits the given data?"

If "an arbitrary dataset χ showed 91% performance when classified by support vector machine (SVM) and 85% when classified by k-nearest neighbor(k-NN)", SVM can be said to be good. Here, SVM and k-NN become the analysis viewpoints, and a hybrid model that adds viewpoints can be used to improve the performance. The data problem is a missing-value problem. Because rating is the result of decision-making regarding a user's subjectivity, preference, situation, and environment, consistency cannot be guaranteed. However, rating data are uncertain correct answer sheets, because information about the process whereby the user makes the decision is lacking. Therefore, the accuracy of recommendation systems should have converged on the current level because a limit to understanding decision-making exists.

When considering the process for a user to select a movie, actors, director, and genres can be important variables. If the user likes a certain genre, he/she is highly likely to prefer an actor thta is close to the action or to be a fan of the actor. For example, Arnold Schwarzenegger, Sylvester Stallone, and Keanu Reeves belong to action. Furthermore, for a director that is represented by a certain genre, actors related to the director can be considered. MBA's research motive is to observe this pattern through the rating frequency and the degree of tendency of ratings, and this was defined as a bias analysis. MBA is a model to analyze the foregoing, and through experiments, BBP accuracy, which is the basis of MBA, has been improved, and it was shown the result of Table A2 that the hybrid model outperforms MF and SVD++. Based on this result, it can be argued that the bias is reflected in users' decision-making for movies.

MBA was designed to learn the feature weight ω that was used in the user's GF and PF. When the ω values used for GF and PF were observed, it was identified that they could be used as the user's unique characteristic. Because HybridSVD++ (LRF) outperformed SVD++, which is limited to the foregoing, it can be a clue to explain the user's decision-making process using the ω values of GF and PF. Furthermore, using the ω values of GF and PF, it is possible to design a recommendation system that describes the reason for recommendation simultaneously with the recommendation. For example, when MBA has recommended the movie "Iron Man", how crucial "Actor-Robert Downey Jr." and "Genre-SF" were can be computed numerically. Reflecting this concept, it is possible to perform a data description that explains the reason for recommendation simultaneously with the recommendation.

Thus far, bias has been considered at the graph-axis movement level, such as $y = ax + b$ and negative variables. Although using bias requires more observation and verification, given the influence of bias that is involved in decision-making in music, food, movies, books, news, etc., bias is an element that is more important than the graph-axis movement. Additionally, recently, studies on fairness models, noting potential biases in data and analysis models, are emerging, and the importance of studies that are related to fairness is increasing. Fairness is a critical issue in bias analysis. While checking the limitations of the vanilla model through the heuristics approach, the MBA showed results with small variations in accuracy when a pinpoint score appropriate as a mediator was selected. When synthesizing the results and summarizing future research plans to improve MBA performance, the MBA accuracy is expected to improve through studies on fairness and by enhancing the Vanilla model learning method.

Author Contributions: Conceptualization, T.-G.H. and S.K.K.; Data curation, T.-G.H. and S.K.K.; Formal analysis, T.-G.H. and S.K.K.; Funding acquisition, S.K.K.; Investigation, T.-G.H.; Methodology, T.-G.H.; Resources, S.K.K.; Software, T.-G.H.; Supervision, S.K.K.; Validation, T.-G.H. and S.K.K.; Visualization, T.-G.H.; Writing—original draft, T.-G.H.; Writing—review & editing, S.K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2019R1F1A1059952).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: MovieLens 100K, https://grouplens.org/datasets/movielens/100k/ (accesed on 20 March 2021); HetRec 2011, http://files.grouplens.org/datasets/hetrec2011/hetrec2011-movielens-2k-v2.zip (accesed on 20 March 2021); MBA experimental dataset, https://drive.google.com/file/d/14bOmNhtDD6KZ_zSgl4D4fghGkg82raT4/view?usp=sharing (accesed on 20 March 2021).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Results

Table A1. For each method, 5-fold cross-validation was performed (1–5) and RMSE values are shown and their averages and variances are also shown.

Method	RMSE						
	Validation Set						
	1	2	3	4	5	μ	σ
UBCF	1.14409	0.99254	1.02768	1.01318	1.09700	1.05490	0.06343
IBCF	1.11224	1.00425	1.03284	1.99669	1.08080	1.04536	0.04986
MF	1.11380	0.95553	0.98895	0.95996	1.01544	1.00674	0.05812
SVD++	1.04787	0.92311	0.96036	0.93895	0.97461	0.96898	0.04345
BBP(RF)	1.12591	0.99084	1.02516	1.01131	1.07054	1.04475	0.05399
BBP(ARF)	1.10368	0.98568	1.02043	1.00957	1.04378	1.03263	0.04488
BBP(LRF)	1.11752	0.98933	1.02408	1.00637	1.03487	1.03444	0.04958
MBA(RF)	1.15449	1.00475	1.03260	1.01856	1.08049	1.05818	0.06094
MBA(ARF)	1.10513	0.99629	1.03009	1.01642	1.04620	1.03883	0.04135
MBA(LRF)	1.10926	0.98242	1.01818	1.00241	1.02987	1.02843	0.04857

Table A2. HybridSVD++ with three weight types, RF, ARF and LRF were compared with other methods, and HybridSVD++(LRF) showed best reduction in RMSE. (↑ better , ↓ worse).

Method		HybridSVD++		
		RF	ARF	LRF
	UBCF	↑+4.31 (%)	↑+5.11 (%)	↑+9.03 (%)
	IBCF	↑+3.44 (%)	↑+4.24 (%)	↑+8.20 (%)
BBP	RF	↑+3.38 (%)	↑+4.19 (%)	↑+8.15 (%)
	ARF	↑+2.24 (%)	↑+3.06 (%)	↑+7.07 (%)
	LRF	↑+2.41 (%)	↑+3.23 (%)	↑+7.23 (%)
MBA (Vanilla)	RF	↑+4.60 (%)	↑+5.40 (%)	↑+9.32 (%)
	ARF	↑+2.83 (%)	↑+3.64 (%)	↑+7.63 (%)
	LRF	↑+1.84 (%)	↑+2.67 (%)	↑+6.69 (%)
	MF	↓−0.27 (%)	↑+0.57 (%)	↑+4.68 (%)
	SVD++	↓−4.18 (%)	↓−3.31 (%)	↑+0.97 (%)

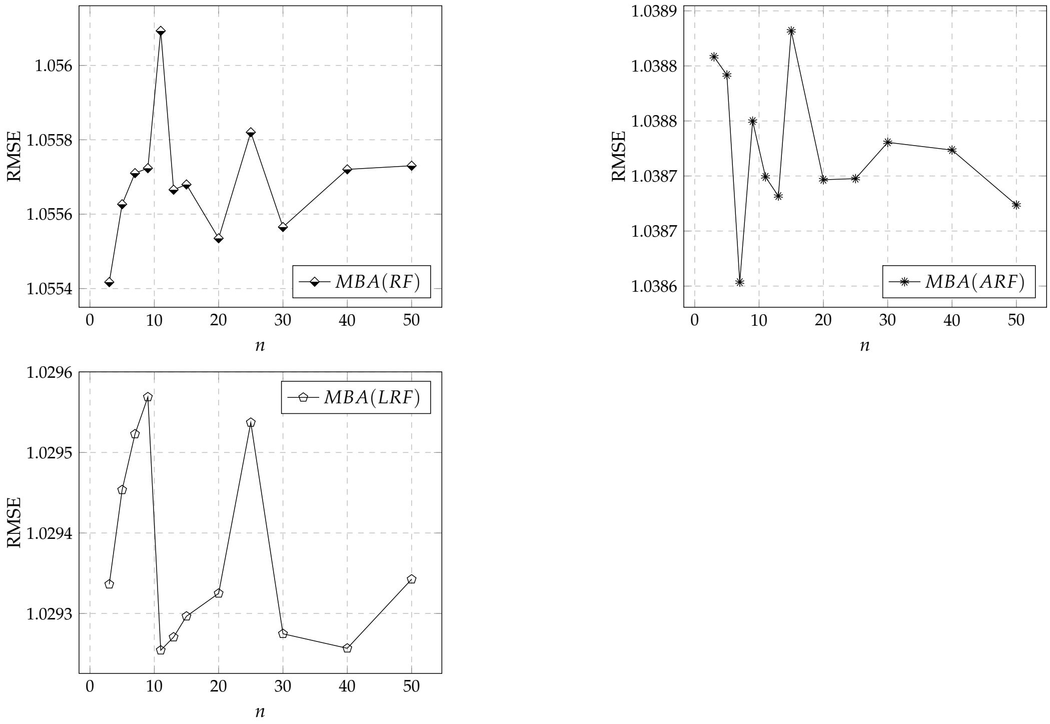

Figure A1. Comparison of RMSE in MBA(Vanilla) by n. MBA(RF) is best when $n = 3$, MBA(ARF) is best when $n = 7$ and MBA(LRF) is best when $n = 11$.

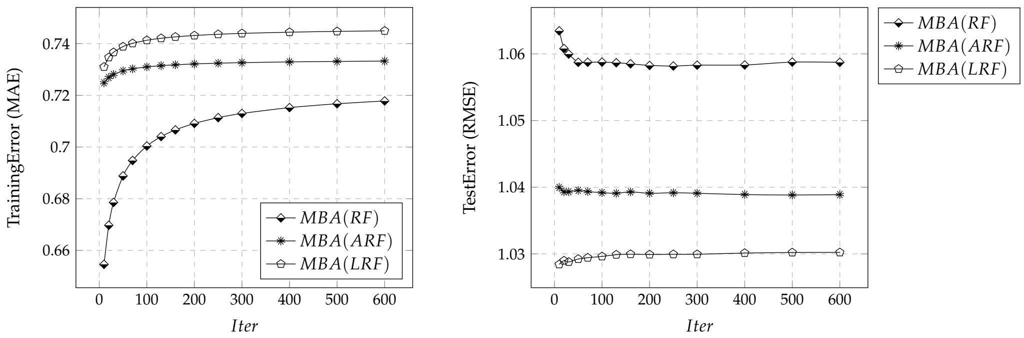

Figure A2. Comparison of RMSE in MBA(Vanilla) by $Iter$. The experiment was done after fixing n as: MBA(RF) with $n = 3$, MBA(ARF) with $n = 7$ and MBA(LRF) with $n = 11$, as found in Figure A1.

References

1. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; Volume 39.
2. Zhao, Z.D.; Shang, M.S. User-based collaborative-filtering recommendation algorithms on hadoop. In Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining, Phuket, Thailand, 9–10 January 2010; pp. 478–481.
3. Hwang, T.G.; Park, C.S.; Hong, J.H.; Kim, S.K. An algorithm for movie classification and recommendation using genre correlation. *Multimed. Tools Appl.* **2016**, *75*, 12843–12858. [CrossRef]
4. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; pp. 285–295.
5. Ding, Y.; Li, X. Time weight collaborative filtering. In Proceedings of the 14th ACM International Conference on Information and Knowledge management, Bremen, Germany, 31 October–5 November 2005; pp. 485–492.
6. Jiang, J.; Lu, J.; Zhang, G.; Long, G. Scaling-up item-based collaborative filtering recommendation algorithm based on hadoop. In Proceedings of the 2011 IEEE World Congress on Services, Washington, DC, USA, 4–9 July 2011; pp. 490–497.
7. Zhou, Y.; Wilkinson, D.; Schreiber, R.; Pan, R. Large-scale parallel collaborative filtering for the netflix prize. In Proceedings of the International Conference on Algorithmic Applications in Management, Shanghai, China, 23–25 June 2008; pp. 337–348.
8. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]

9. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Incremental singular value decomposition algorithms for highly scalable recommender systems. In Proceedings of the Fifth International Conference on Computer and Information Science, Dhaka, Bangladesh, 27–28 December 2002; Volume 1, pp. 27–28.
10. Liu, D.; Ye, X. A matrix factorization based dynamic granularity recommendation with three-way decisions. *Knowl. Based Syst.* **2020**, *191*, 105243. [CrossRef]
11. Pazzani, M.J.; Billsus, D. Content-based recommendation systems. In *The Adaptive Web*; Springer: Berlin/Heisenberg, Germany, 2007; pp. 325–341.
12. Adomavicius, G.; Tuzhilin, A., Context-aware recommender systems. In *Recommender Systems Handbook*; Springer: Berlin/Heisenberg, Germany, 2011; pp. 217–253.
13. Son, J.; Kim, S.B.; Kim, H.; Cho, S. Review and analysis of recommender systems. *J. Korean Inst. Ind. Eng.* **2015**, *41*, 185–208. (In Korean) [CrossRef]
14. Park, D.H.; Kim, H.K.; Choi, I.Y.; Kim, J.K. A literature review and classification of recommender systems research. *Expert Syst. Appl.* **2012**, *39*, 10059–10072. [CrossRef]
15. Bobadilla, J.; Ortega, F.; Hernando, A.; Bernal, J. A collaborative filtering approach to mitigate the new user cold start problem. *Knowl. Based Syst.* **2012**, *26*, 225–238. [CrossRef]
16. Wei, J.; He, J.; Chen, K.; Zhou, Y.; Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* **2017**, *69*, 29–39. [CrossRef]
17. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. Scalable collaborative filtering approaches for large recommender systems. *J. Mach. Learn. Res.* **2009**, *10*, 623–656.
18. Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters. *Commun. ACM* **2008**, *51*, 107–113. [CrossRef]
19. Meng, X.; Bradley, J.; Yavuz, B.; Sparks, E.; Venkataraman, S.; Liu, D.; Freeman, J.; Tsai, D.; Amde, M.; Owen, S. Mllib: Machine learning in apache spark. *J. Mach. Learn. Res.* **2016**, *17*, 1235–1241.
20. Anil, R.; Capan, G.; Drost-Fromm, I.; Dunning, T.; Friedman, E.; Grant, T.; Quinn, S.; Ranjan, P.; Schelter, S.; Yılmazel, O. Apache Mahout: Machine Learning on Distributed Dataflow Systems. *J. Mach. Learn. Res.* **2020**, *21*, 1–6.
21. Breese, J.S.; Heckerman, D.; Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv* **2013**, arXiv:1301.7363.
22. Zhang, M.; Hurley, N. Avoiding monotony: improving the diversity of recommendation lists. In Proceedings of the 2008 ACM Conference on Recommender Systems, Lausanne, Switzerland, 23–25 October 2008; pp. 123–130.
23. Zhou, T.; Kuscsik, Z.; Liu, J.G.; Medo, M.; Wakeling, J.R.; Zhang, Y.C. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proc. Natl. Acad. Sci. USA* **2010**, *107*, 4511–4515. [CrossRef] [PubMed]
24. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, T.S. Neural collaborative filtering. In Proceedings of the 26th International Conference on world Wide Web, Perth, Australia, 3–4 May 2017; pp. 173–182.
25. Bobadilla, J.; Alonso, S.; Hernando, A. Deep learning architecture for collaborative filtering recommender systems. *Appl. Sci.* **2020**, *10*, 2441. [CrossRef]
26. Li, S.; Kawale, J.; Fu, Y. Deep collaborative filtering via marginalized denoising auto-encoder. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; pp. 811–820.
27. Wang, T.; Fu, Y. Item-based Collaborative Filtering with BERT. In Proceedings of The 3rd Workshop on e-Commerce and NLP, Seattle, WA, USA, 9–10 July 2020; pp. 54–58.
28. Rendle, S.; Krichene, W.; Zhang, L.; Anderson, J. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In Proceedings of the Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, 22–26 September, 2020; pp. 240–248. [CrossRef]
29. Abdollahpouri, H.; Mansoury, M.; Burke, R.; Mobasher, B. The impact of popularity bias on fairness and calibration in recommendation. *arXiv* **2019**, arXiv:1910.05755.
30. Sreepada, R.S.; Patra, B.K.; Chakrabarty, A.; Chandak, S. Revisiting tendency based collaborative filtering for personalized recommendations. In Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, Goa, India, 11–13 January 2018; pp. 230–239.
31. Koren, Y. Collaborative filtering with temporal dynamics. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 447–456.
32. Hwang, T.G.; Kim, S.K. Bias-Based Predictor to Improve the Recommendation Performance of the Rating Frequency Weight-based Baseline Predictor. *J. KIISE* **2017**, *44*, 486–495. (In Korean) [CrossRef]
33. Hwang, T.G.; Kim, S.K. Multi-Label Bias-Based Predictor. In Proceedings of the 2019 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea, 28–30 January 2019; pp. 1–5.
34. Gantner, Z.; Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. MyMediaLite: A free recommender system library. In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 305–308.
35. Harper, F.M.; Konstan, J.A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* **2015**, *5*, 1–19. [CrossRef]
36. Cantador, I.; Brusilovsky, P.; Kuflik, T. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). In Proceedings of the Fifth ACM Conference on Recommender Systems, Chicago, IL, USA, 23–27 October 2011; pp. 387–388.

Short Biography of Authors

Tae-Gyu Hwang is currently a Ph.D. degree course student in Computer Science and Engineering from Chung-Ang University, Seoul, Korea. His areas of research interest are data mining and recommender system.

Sung Kwon Kim received the B.S. degree from Seoul National University, Seoul, Korea, and the M.S. degree from Korea Advanced Institute of Science and Technology (KAIST), Korea, and the Ph.D. degree from University of Washington, Seattle, USA. He is currently a professor at School of Computer Science and Engineering, Chung-Ang University, Seoul, Korea. His areas of research interest are information security and computational geometry and recommender system.