# Hybrid Attribute and Personality based Recommender System for Book Recommendation

'Adli Ihsan Hariadi
*School of Computing*
*Telkom University*
Bandung, Indonesia
adliih@student.telkomuniversity.ac.id

Dade Nurjanah
*School of Computing*
*Telkom University*
Bandung, Indonesia
dadenurjanah@telkomuniversity.ac.id

***Abstract*— In recent years, with the rapid increases of books, finding relevant books has been a problem. For that, people might need their peers' opinion to complete this task. The problem is that relevant books can be gained only if there are other users or peers have same interests with them. Otherwise, they will never get relevant books. Recommender systems can be a solution for that problem. They work on finding relevant items based on other users' experience. Although research on recommender system increases, there is still not much research that considers user personality in recommender systems, even though personal preferences are really important these days. This paper discusses our research on a hybrid-based method that combines attribute-based and user personality-based methods for book recommender system. The attribute-based method has been implemented previously. In our research, we have implemented the MSV-MSL (Most Similar Visited Material to the Most Similar Learner) method, since it is the best method among hybrid attribute-based methods. The personality factor is used to find the similarity between users when creating neighborhood relationships. The method is tested using Book-crossing and Amazon Review on book category datasets. Our experiment shows that the combined method that considers user personality gives a better result than those without user personality on Book-crossing dataset. In contrary, it resulted in a lower performance on Amazon Review dataset. It can be concluded that user personality consideration can give a better result in a certain condition depending on the dataset itself and the usage proportion.**

***Keywords—recommender system; hybrid method; user personality; attribute based; book***

## I. Introduction

With the rapid increases of book, how can people find relevant books for themselves? People might need their friends' opinion to complete this task[1]. However, whether their friends have same interests with them is questionable. If so, they will get relevant books they need, otherwise they will not. This is where recommender systems become useful. Recommender systems can recommend relevant books by analyzing the historical data of the users[2, 3]. The advantages of recommender systems are obvious. Even Amazon finds that recommender systems are useful because 35% of their earning are boosted by recommender systems[4, 5].

In general, recommender system methods are grouped into three: content-based, collaborative filtering, and hybrid-based (combining content-based and collaborative filtering). The main problem of content-based methods is that they may result in overspecialized recommendations, which means that they tend to recommend over similar items from already rated items [3, 6]. This problem can be solved by implementing collaborative filtering, but it suffers from cold start conditions [6]. Then, hybrid methods are seen as a solution to reduce content-based and collaborative filtering problems.

Previous research has tried to improve recommender system performance, such as attribute-based methods, which fully utilize e-learning materials' attributes to improve the accuracy of recommendation [7, 8]. Another attempt to improve the recommender system performance is by taking user personality into account in recommendation. This method has been implemented in a restaurant selection case, for which the dataset were gathered from social media [9]. The method resulted in an improved accuracy of resulted prediction.

This paper discusses our research on combining attribute- and personality-based methods. The remaining part of this paper is organized as follows. Section two, summary of related work, discusses current studies on recommender systems. Section three discusses our proposed method, while section four discusses experiments and results. Finally, section five presents the conclusions.

## II. Related Work

This part discusses recommender system methods, including content-based, collaborative filtering, and hybrid-based methods. Furthermore, it discusses two similarity identification methods which are based on interpersonal ratings and interests.

### *A. Content-based Method*

Content-based methods give recommendations of similar items from good-rated items[7]. In this paper, we combine attribute-based and content-based methods to calculate the similarity between book $p$ and user $i$ that can be denoted as $sim(L_i, M_p)$ using equation below[7]:

$$sim(L_i, M_p) = \frac{\sum_{j=1}^{T} w_{ij} \,.\, \sum_{k=1}^{K} s_{ikj} . m_k(M_p, L_{ij})}{T.K} \tag{1}$$

$$w_{ij} = e^{-\lambda(t(L_{ij}) - T)} \tag{2}$$

Where $L_i$ is user $i$ and $M_p$ is book $p$, $w_{ij}$ is the weighting function of rated book $j$ by user $i$ that we got from equation (2) and needs to be normalized as $\sum_{j=1}^{T} w_{ij} = 1$. Furthermore,

978-1-5386-1449-5/17/$31.00 ©2017 IEEE

$\lambda$ is an adjustable parameter with its optimum value is 0.3 [7], $s_{ikj}$ is the value of $k$-th attribute of book $j$ by user $i$ from personal preference matrix, and $t(L_{ij})$ is the order access of rated book $j$ by user $i$ based on his/her history. Books with lowest number of ratings mean that they are recently rated. Furthermore, $T$ is the number of rated books, $K$ is the number of attributes used, and $m_k(M_p, L_{ij})$ is a matching function of $k$-th attribute of book $p$ and rated book $j$ by user $i$. If the value of $k$-th attribute between two book is similar, this function should return 1, otherwise it gives 0.

### B. Collaborative Filtering

While content-based method is used to calculate the similarity between item and user, collaborative filtering is used to calculate the similarity between users based on ratings they have given to items [6]. Same as content-based, attribute-based collaborative filtering aims to find the similarity between two users, denoted as $sim(L_i, L_j)$, using equation (3) [7]:

$$sim(L_i, L_j) = \frac{\sum_{h=1}^{T_i} \sum_{f=1}^{T_j} w_{ih}.w_{jf}.\sum_{k=1}^{K} s_{ikh}.s_{jkf}.m_k(L_{ih}, L_{jf})}{T_i.T_j.K} \quad (3)$$

Where $T_i$ and $T_j$ is the number of rated items from user $i$ and $j$ resprectively. This similarity score can also be used to predict ratings of unrated items. The rating prediction of book $p$ by user $a$, denoted as $P(L_a, i)$, can be calculated using equation (4) [7]:

$$P(L_a, i) = \overline{R_{L_a}} + \frac{\sum_{j \in N(L_a)} sim(L_a, L_j) \times (R_{L_j}(i) - \overline{R_{L_j}})}{\sum_{j \in N(L_a)} sim(L_a, L_j)} \quad (4)$$

### C. Hybrid Attribute-based Recommendation Method

A hybrid attribute-based recommendation method that gives the best performance is Most Similar Visited Material to the most Similar Learner (MSV-MSL)[7]. MSV-MSL calculates the recommendation scores of rated books from neighbors using the similarity scores between a target book and its neighbors and between the active user and his/her neighbors. The score of book $p$, denoted as $Score_p$, can be calculated using equation (5) [7]:

$$Score_p = \sum_{L_i \in N(L_a)} Sim(L_i, p).Sim(L_a, L_i) \quad (5)$$

Where $Sim(L_i, p)$ represents the suitability of book $p$ for user $i$; it is calculated using formula (1). $Sim(L_a, L_i)$ is a similarity score between the active user $L_a$ and user $i$ that can be calculated using formula (3). Furthermore, $L_i$ is user $i$ which is one of the the active user's neighbors.

### D. Interpersonal Interest Similarity

Interpersonal interest similarity is a user personality factor that can be mined to give a better recommendation, it can show how much a user likes each category[9]. If a user often rates books in a certain category, then the interest score for that category will be higher. Interpersonal interest similarity between two users, denoted as $W_{u,v}$, can be calculated using equation (6) [9]:

$$W_{u,v} = \frac{D_u \cdot D_v}{|D_u| \times |D_u|} \quad (6)$$

Where $D_u$ and $D_v$ are vectors of naïve rated topic distribution of user $u$ and $v$ respectively. Naïve topic distribution is calculated by counting the rated books in each category then dived all the value by the number of rated books.

### E. Interpersonal Rating Behaviour Similarity

Interpersonal rating behavior similarity shows how similar the rating patterns are from two users[9]. It is denoted as $E_{u,v}$, and calculated using following equations:

$$E_{u,v} = \frac{1}{E(U_u, U_v)} \quad (7)$$

$$E(U_u, U_v) = -\sum_{c'=1}^{n} p(d_{c'}) \log_2 p(d_{c'}) \quad (8)$$

$$d_{c'} = |K_{u,v}^{c'}| \times |R_{u,c'} - R_{v,c'}| \quad (9)$$

Where $n$ is the list of categories, $p(d_{c'})$ is the error frequency of $d_{c'}$, while $d_{c'}$ denotes rating difference between two users in $c'$ category. Furthermore, $|K_{u,v}^{c'}|$ is a matching function between user $u$ and $v$ in $c'$ category, which will return 1 if both users have rated at least 1 item in $c'$ category, otherwise it will return 0. $R_{u,c'}$ and $R_{v,c'}$ are the average ratings of user $u$ and $v$ (respectively) in $c'$ category. The value of $E_{u,v}$ needs to be normalized as $\sum_V E_{u,v}^* = 1$.

## III. Proposed Method

From the previous related work, a hybrid attribute-method has been applied in e-learning material dataset which is the most similar category type from our dataset category, i.e book category. Hence, the hybrid attribute-based method is our base method and the personality-based method is an add-on to the based method. The attribute-based method we use is the MSV-MSL, since it is the best method in hybrid attribute-based recommender system. The personality factor is used for calculating the similarity between users when creating a neighborhood which will be explained in further details below.

### A. Creating User Profile

To create a user profile, we use personal preference matrix as shown in figure 1.

| | $r_{i1}$ | $r_{i2}$ | .. | $r_{iT}$ |
|---|---|---|---|---|
| $A_1$ | $s_{i11}$ | $s_{i12}$ | .. | $s_{i1T}$ |
| $A_2$ | $s_{i21}$ | $s_{i22}$ | .. | $s_{i2T}$ |
| .. | .. | .. | .. | .. |

| | | | | |
|---|---|---|---|---|
| $A_K$ | $s_{iK1}$ | $s_{iK2}$ | .. | $s_{iKT}$ |

Fig. 1. Personal Preference Matrix[7]

The value of $s_{ikj}$, which is the value of $k$-th attribute of book $j$ by user $i$, can be calculated using equation (10):

$$s_{ikj} = AW_k r_{ij} \tag{10}$$

Where $A_k$ is the the $k$-attribute, $T$ is the number of rated books, $K$ is the number of used attributes (see Fig. 1), $r_{ij}$ is the rating of book $j$ by user $i$, $AW_k$ is the appropriate weight of each $k$-attribute value. To be noted that since $AW_k$ is the weighting value, the total value of $AW_k$ must be 1, $\sum_{k=1}^{K} AW_k = 1$. Weighting is done by counting each attribute that has been rated by users and divided by the number of rated books. Attributes used for creating user profiles depend on the requirements and objective of the recommender. It could be, for example, title, author, publication year, publisher, price, and categories.

### B. *Creating a Neighborhood*

To create a neighborhood, we first need to compute the similarity between the active user and other users. The similarity is not only computed using collaborative filtering, but also using two user personality factors, which are interest and rating behavior similarities. Since we want to utilize both user personality factors evenly, we sum them up, then divide it by 2, as of the following equation to create a user personality similarity:

$$Sim_p(U_u, U_v) = \frac{W_{u,v} + E_{u,v}}{2} \tag{11}$$

In the formula, $Sim_p(U_u, U_v)$ denotes the user personality similarity between user $u$ and $v$. $E_{u,v}$ is rating behaviour similarity between user $u$ and user $v$ that can be calculated using (7). $W_{u,v}$ is the interest similarity between user $u$ and user $v$ that is calculated using (6). Note that $W_{u,v}$ is a cosine similarity between two naïve topic distributions that represent the interest of both users. A weight vector can also represent the interest each user towards categories/attributes. We can use weight vectors to replace the naïve topic distributions in (6).

We use a personality similarity along with the attribute-based user similarity resulted from collaborative filtering. We define a new formula to calculate the similarity between two users that utilizes personality and attribute based collaborative filtering according to the following equation:

$$Sim(U_u, U_v) = Sim_{cf}(U_u, U_v)\,.Sim_p(U_u, U_v) \tag{12}$$

$$Sim_{cf}(U_u, U_v) = \frac{\sum_{h=1}^{T}\sum_{f=1}^{T} w_{ih}.w_{jf}.\sum_{k=1}^{K} s_{ikh}.s_{jkf}.m_k(L_{ih}, L_{jj}}{T^2.K} \tag{13}$$

The optimal number of neighborhoods is 18[7].

### C. *Calculating the Book Score*

To calculate the book scores, we use a MSV-MSL method with the score of book $b$ from user $u$, denoted as $Score_b$, which is computed using the following equation:

$$Score_b = \sum_{U_v \in N(U_u)} Sim(U_v, b).Sim(U_u, U_v) \tag{14}$$

Where $U_v$ denotes user $v$ that also a of user $u$. $Sim(U_v, b)$ is the similarity between user $v$ and book $b$ based on content-based as of (1). $Sim(U_u, U_v)$ is the similarity between user $u$ and user $v$ based on collaborative filtering and user personality as shown in formula (14).

### D. *Predicting Rating*

To predict rating, we use formula (4) as a principal equation, then we replace the similarity value using the one that also utilize user personality as of (13), so we can get a new equation as follows:

$$P(U_u, b) = \overline{R_{U_u}} + \frac{\sum_{v \in N(U_u)} sim(U_u, U_v) \times (R_{U_v}(b) - \overline{R_{U_v}})}{\sum_{v \in N(U_u)} sim(U_u, U_v)} \tag{15}$$

Where $P(U_u, b)$ denotes the prediction rating of book $b$ by user $u$, $\overline{R_{U_u}}$ and $\overline{R_{U_v}}$ denotes the average rating of user $u$ and user $v$. Furthermore, $sim(U_u, U_v)$ denotes the similarity between two users as of (14) and $R_{U_v}(b)$ is the rating of book $b$ by user $u$.

### E. *Performance Measurement*

Since F-1 Measure [7, 10] is often used to evaluate the performance of recommender systems, we use this metric in our experiments. F-1 Measure is a classification metric that measures the harmonic mean of precision and recall. For calculating F-1 Measure, we need to classify the recommendation results into 4 categories, including True-Positive, False-Positive, False-Negative, and True Negative that can be seen in Table 1.

TABLE I. CATEGORIZATION OF RECOMMENDATION RESULTS

| | **Recommended** | **Not Recommended** |
|---|---|---|
| **Used** | True-Positive (TP) | False-Negative (FN) |
| **Not-Used** | False-Positive (FP) | True-Negative (TN) |

The next step is calculating the F-1 Measure score using equation 16:

$$F - 1\,Measure = 2\frac{Precission\,.Recall}{Precision + Recall} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

## IV. Experiment and Results

The experiment tested the following proposed method combination: MSV-MSL that utilizes both rating patterns and personality, MSV-MSL that utilizes only rating patterns, and MSV-MSL that utilizes only user personality. The only difference between those combinations lies on how we calculate the similarity between users. In MSV-MSL that utilizes both rating patterns and personality, we use a full equation, while in MSV-MSL that utilizes either rating patterns or personality, we used user personality according to equation (12), without $Sim_p(U_u, U_v)$ and $Sim_{cf}(U_u, U_v)$.

We used Book Crossing and Amazon review datasets. Book-crossing has been used for testing the performance of recommender system and can be downloaded at Institut für Informatik website[1]. The Amazon review dataset can be downloaded online[2]. Book-crossing dataset is the only dataset used for book-based recommender system experiment in the first experiment. Both datasets were used in the second and third experiments to measure the influence of user personality in hybrid attribute-based method.

Experiments done once, since we did not use any randomization in splitting data. Hence, the result should be same although we repeat each experiment. The splitting method used in the previous experiment (Hybrid Attribute-based Recommender System) removed the last 5 ratings of each user and assigned them as the recommendation targets and the remaining ratings were used to create a user model[7]. So, we did the same splitting since the base recommender system method that we use is Hybrid Attribute-based Recommender System.

### A. *Dataset*

Book-crossing dataset consist of 3 fields including users, books, and book-ratings[11] and user data have 3 fields including User-ID, Location, and Age. On the other hand, Books data have 8 fields, ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, and Image-URL-L. Books-ratings consist of User-ID, ISBN, and Rating in range 0-10, where 0 denotes implicit ratings and 1-10 denotes explicit ratings. Implicit ratings mean the book purchases are only considered as an expression of appreciation[11].

Before using Book-crossing dataset, we applied a pre-processing consisting of the following steps:

1) Remove implicit (zero) ratings. This experiment only deals with explicit ratings since implicit ratings tends to add noise to the collected information[11].
2) Remove users never rating and books never rated. The MSV-MSL method gives scores only to books that have been rated by neighbors, so that unrated books will never be scored.
3) Delete all users who have rated less than 15 books. The recommender needs 10 ratings to make a user model and 5 ratings to be used as the data target[7].
4) Delete the image url field in books data.

The Amazon review dataset consists of two files: review and books metadata[13]. Books metadata have ASIN, a unique product identifier by Amazon that consists of books id, book price, book title, and list of categories which cover the books. Reviews have reviewer ID, ASIN, reviewer's name, overall (rating), and timestamps. Since this dataset does not have attributes as many as Book-crossing dataset, we use the list of categories to make custom attributes of the books. Made-up attributes are created by making a list of all possible categories in the dataset and it will be filled by 1 if a book has a category for that index or 0 otherwise. For example, if a list is of all categories is [Book, Horror, Comedy, Romance] and a certain book has Book and Comedy category, thus made-up attributes for that book are 1, 0, 1, and 0.

### B. *Performance of Hybrid Attribute and Personality based Recommender System*

To fully understand hybrid attribute- and personality-based recommender systems, we use MSV-MSL with a certain variation on finding user similarity, i.e. MSV-MSL that utilizes both rating patterns and personality, MSV-MSL that utilizes only rating patterns, and MSV-MSL that utilizes only user personality. Table 2 shows the experiment result of those variations in Book-crossing dataset.

TABLE II. Experiment Result Of Book-crossing Dataset

| Method | F-1 Measure |
|---|---|
| MSV-MSL based on rating patterns and personality | 0.026389 |
| MSV-MSL based on rating patterns only | 0.014352 |
| MSV-MSL based on personalityonly | 0.035648 |

From this experiment, we can see that MSV-MSL which utilizes only user personality results in the best performance. However, this result is based on Book-crossing dataset only. Table 3 shows the result of the proposed method implementation for Amazon Review dataset. We did not implement the MSV-MSL based on only user personality, because we did not have resources to run the program.

TABLE III. Experiment Result Of Amazon Review Dataset

| Method | F-1 Measure |
|---|---|
| MSV-MSL based on rating patterns and personality | 0.000099 |
| MSV-MSL based on rating patterns only | 0.000397 |

In Book-crossing dataset, user personality can increase the F-1 Measure which is a good result. However, in Amazon Review dataset, user personality decrease the F-1 Measure which is a bad result. Those opposite results shows that user personality cannot give a proper improvement to a hybrid attribute-based recommender system.

The result shows that hybrid attribute- and personality-based recommender system works better on Book-crossing dataset than on Amazon Review dataset. This happened because the model created from Book-crossing consists of real

[1]http://www2.informatik.uni-freiburg.de/~cziegler/BX/

[2] http://jmcauley.ucsd.edu/data/amazon/

attributes, such as title, author, and year of publication. In contrast, the model created from Amazon Review dataset consists of unreal attribute, but also made-up attributes which is built using list of categories. This result is not surprising, since recommender systems always give different errors depending on the algorithm and even data [14].

## V. Conclusion And Future Works

From the experimental result analysis, we can conclude that the combination of hybrid attribute- and personality-based recommender system affect the recommender performance only when it is applied to datasets containing real attributes. Even though the results depend on the data contained in datasets, the proposed method is still open for improvement. Future research can study an exact specification of dataset that can result in a better performance.

Immediate future work will involve a broader domain of datasets such as movies, fashion, literature, jokes, paper, et cetera, in order to find the most suitable data for the proposed method. Another part that can be more explored is the portion of user-personality inclusion in recommender systems. In the future, we plan to fully use of user-personality method, including the training process to see if it is potential to give improvements.

## References

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, 2011, pp. 1–35.

[2] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10059–10072, Sep. 2012.

[3] D. H. Park, H. K. Kim, J. K. Kim, I. Y. Choi, and J. K. Kim, "A review and classification of recommender systems research," *School of Management, KyungHee University, Seoul, Korea, IPEDR*, vol. 5, 2011.

[4] M. Marshall, "Aggregate knowledge raises $5M from Kleiner, on a roll | VentureBeat | News | by Matt Marshall," 05-Mar-2016. [Online]. Available: http://venturebeat.com/2006/12/10/aggregate-knowledge-raises-5m-from-kleiner-on-a-roll/. [Accessed: 05-Mar-2016].

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

[6] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[7] M. Salehi and I. N. Kmalabadi, "A hybrid attribute–based recommender system for e–learning material recommendation," *IERI Procedia*, vol. 2, pp. 565–570, 2012.

[8] M. Salehi, M. Pourzaferani, and S. A. Razavi, "Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model," *Egyptian Informatics Journal*, vol. 14, no. 1, pp. 67–78, Mar. 2013.

[9] G. Zhao, X. Qian, and H. Feng, "Personalized recommendation by exploring social users' behaviors," in *MultiMedia Modeling*, C. Gurrin, F. Hopfgartner, W. Hurst, H. Johansen, H. Lee, and N. O'Connor, Eds. Springer International Publishing, 2014, pp. 181–191.

[10] I. Avazpour, T. Pitakrat, L. Grunske, and J. Grundy, "Dimensions and metrics for evaluating recommendation systems," in *Recommendation Systems in Software Engineering*, M. P. Robillard, W. Maalej, R. J. Walker, and T. Zimmermann, Eds. Springer Berlin Heidelberg, 2014, pp. 245–273.

[11] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th International Conference on World Wide Web*, New York, NY, USA, 2005, pp. 22–32.

[12] P. Cremonesi, F. Garzottto, and R. Turrin, "User effort vs. accuracy in rating-based elicitation," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, New York, NY, USA, 2012, pp. 27–34.

[13] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," 2015, pp. 43–52.

[14] M. Ekstrand and J. Riedl, "When recommenders fail: predicting recommender failure for algorithm selection and combination," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, New York, NY, USA, 2012, pp. 233–236.