An Efficient Scheme to Combine The User Demographics and Item Attribute for Solving Data Sparsity and Cold-start Problems

Noor Ifada [1], Mochammad Kautsar Sophan [2], Irvan Syachrudin [3], Selgy Zahranida Sugiharto [4]
Informatics Department
University of Trunojoyo Madura
Bangkalan, Indonesia
[1]noor.ifada@trunojoyo.ac.id, [2]kautsar@trunojoyo.ac.id, [3]irvansyachrudin97@gmail.com, [4]selgyarafaa@gmail.com

***Abstract*—This paper investigates several schemes to combine the user demographic information and item attribute data that respectively beneficial to solve the data sparsity and cold-start problems in recommendation systems. We propose four schemes that are varied based on how the combination of the two data can be constructed. To test and evaluate the concept, we implement the schemes on a probabilistic-attribute method adapted to suit our attribute model. Compared to the benchmark methods, experiment results show that our approach is superior in solving the data sparsity and cold-start problems. In general, the scheme that combines the item attribute data with a partial user demographic information performs better than the other variations of the combined-attribute scheme. This finding confirms that combining both the user demographic information, though not all of them, and the item attribute can efficiently solve the data sparsity and cold-start problems.**

***Keywords*—cold-start, combined-attribute scheme, data sparsity, item attribute, recommendation system, user demographic**

I. INTRODUCTION

Recommendation System (RS) is a system that provides a list of top-*N* item recommendation to its users based on the users past rating history [1, 2]. RS is expected to learn its users' preferences efficiently and, therefore, the generated list of recommendations suit to the users' interest and expectation.

Collaborative Filtering (CF) is commonly used and studied as a learning model in RS [1]. CF-based RS generates the recommendations based on a user's previous history and also other users that are considering to have a similar preference. In other words, CF builds its model based on the known user-item relationships such that the similarities, of either users or items, are efficiently captured and effected the recommendation process [2].

Data sparsity and cold-start are common problems in CF-based RS [2-5]. Data sparsity is the condition in which most of the user-item associations are unidentified [2, 3, 6, 7]. Whereas the cold-start is the condition in which a user has none or very few rating histories [2-4, 8]. An efficient alternative solution of the data sparsity problem is by adding the attribute data in the learning model [7, 9-11]. On the other hand, the employment of user demographic information is beneficial in alleviating the cold-start problem [4]. We conjecture that solving both the data sparsity and cold-start problems by taking into account both the user demographic information and item attribute data.

This paper investigates and compares several combine-attribute schemes that can alleviate the data sparsity and cold-start problem by taking into account both the user demographic information and item attribute data in the learning model. We propose four schemes that are varied based on how the combination of the two data can be achieved. We implement the proposed schemes on a probabilistic-attribute method [12], which employs an attribute model, to test and evaluate our idea. However, we need to adapt such that it suits to our schemes. The original probabilistic method builds its attribute model only based on the rating matrix and item attribute data, while we add the user demographic information. Therefore, we need to add the process of building a combined-attribute matrix before building the attribute model such that the method can integrate the user demographic information. For the empirical analysis, we use datasets that fulfill both the sparsity and cold-start requirements.

The contribution of this paper is a follows: (1) introducing comprehensive combined-attribute schemes to merge the user demographic information and item attribute data, (2) adopting a probabilistic-attribute method for implementing various combine-attribute schemes, and (3) showing how the proposed schemes are able to address the data sparsity and cold-start problems.

The remaining of this paper is organized as follows. Section II reviews the related works. Section III presents the notations used in the paper, while Section IV details the proposed combined-attribute schemes. Section V describes the probabilistic-attribute method used to implement proposed schemes. Finally, Section VI and VII respectively present the empirical analysis and the conclusion of this paper.

II. Related Work

CF is a well-known learning model in RS that generates the list of item recommendations to a target user by considering both his/her individual fondness as well as other users of similar rating preferences based on the known user-item relationships [1]. On the other hand, CF is also known to suffer from the data sparsity and cold-start problems [2, 3, 5], since most of the user-item relationships are unknown and sometimes the target user has none or very limited numbers of rating data.

Employing item attribute data has shown to alleviate the sparsity problem and, thus, improve the recommendation performance in CF. This approach can be implemented by combining: the items similarities and their matching attributes [13], the rating data and item attribute similarities [9], and the items' similarities to their matching attributes and time-weight scores [10]. On the other hand, employing user demographic information is advantageous in alleviating the cold-start problem [14].

For the above reasons, this paper proposes to combine the advantage of both the item attribute data and user demographic information, i.e., to solve the data sparsity and cold-start problems, introducing the four combined-attribute schemes. We demonstrate how different schemes form different combined-attribute matrix representation. To study the impact of the proposed schemes to the recommendation performances, we adapt a probabilistic-attribute method [12] that employs an attribute-based approach.

III. Preliminaries

Define $U = \{u_1, u_2, u_3, \dots, u_c\}$ and $I = \{i_1, i_2, i_3, \dots, i_d\}$ as the set of c users and d items. The rating matrix $R \in \mathbb{Z}^{c \times d}$ represents the relationship between users and items, thus $r_{u,i}$ holds the rating value given by user u to item i. The set of items who have been rated by user u is labelled as I_u.

Define also $\Delta = \{\delta_1, \delta_2, \delta_3, \dots, \delta_l\}$ and $\Theta = \{\theta_1, \theta_2, \theta_3, \dots, \theta_m\}$ as the set of the l user demographic and m item attribute information. The $Q \in \mathbb{Z}^{c \times l}$ and $V \in \mathbb{Z}^{d \times m}$ matrices respectively represent the binary connection between users and items with their demographics or attributes. In this case, $q_{u,\delta}$ and $v_{i,\theta}$ are set to 1 if user u or item i has demographic or attribute, and 0 otherwise.

The toy examples in Fig. 1(a) show that there are two users $U = \{u_1, u_2\}$ and three items $I = \{i_1, i_2, i_3\}$, in which $I_1 = \{1\}$ and $I_2 = \{2,3\}$. Meanwhile, Fig. 1(b) and (c) respectively show that there are three user demographics $\Delta = \{\delta_1, \delta_2, \delta_3\}$, and two item attributes $\Theta = \{\theta_1, \theta_2\}$. The rating, user demographic, and item attribute matrices are then respectively built as $R \in \mathbb{Z}^{2\times3}$, $Q \in \mathbb{Z}^{2\times3}$, and $V \in \mathbb{Z}^{3\times2}$.

IV. The Proposed Combined-Attribute Schemes

An efficient alternative solution of the data sparsity problem is by adding the attribute data in the learning model [7, 9-11]. On the other hand, the employment of user demographic information is beneficial in alleviating the cold-start problem [4]. We conjecture that solving both the data sparsity and cold-start problems by taking into account both the user demographic information and item attribute data.

$$\begin{array}{c} \\ \boldsymbol{u_1} \\ \boldsymbol{u_2} \end{array}\!\!\begin{array}{c} \begin{array}{ccc} \boldsymbol{i_1} & \boldsymbol{i_2} & \boldsymbol{i_3} \end{array} \\ \begin{bmatrix} 5 & 0 & 0 \\ 0 & 4 & 3 \end{bmatrix} \end{array} \quad \begin{array}{c} \\ \boldsymbol{u_1} \\ \boldsymbol{u_2} \end{array}\!\!\begin{array}{c} \begin{array}{ccc} \boldsymbol{\delta_1} & \boldsymbol{\delta_2} & \boldsymbol{\delta_3} \end{array} \\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} \\ \boldsymbol{i_1} \\ \boldsymbol{i_2} \\ \boldsymbol{i_3} \end{array}\!\!\begin{array}{c} \begin{array}{cc} \boldsymbol{\theta_1} & \boldsymbol{\theta_2} \end{array} \\ \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

(a) (b) (c)

Fig. 1. Toy examples: (a) Rating matrix $R \in \mathbb{Z}^{2\times3}$, (b) User demographic matrix $Q \in \mathbb{Z}^{3\times3}$, and (c) Item attribute matrix $V \in \mathbb{Z}^{3\times2}$

TABLE I. The Proposed Combined-attribute Schemes

Combined-Attribute Scheme	Join Data		Combined-Attribute Matrix	
	Formulation	Example	Size	Example
All-demographic	$Q \bowtie R \bowtie V$	$\begin{bmatrix} \boldsymbol{u} & \boldsymbol{i} & \boldsymbol{r} & \boldsymbol{\delta_1} & \boldsymbol{\delta_2} & \boldsymbol{\delta_3} & \boldsymbol{\theta_1} & \boldsymbol{\theta_2} \\ 1 & 1 & 5 & 1 & 0 & 1 & 1 & 1 \\ 2 & 2 & 4 & 0 & 1 & 1 & 0 & 1 \\ 2 & 3 & 3 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$	$W \in \mathbb{Z}^{d\times z}$ where $z = m + l$	$\begin{array}{c} \\ \boldsymbol{i_1} \\ \boldsymbol{i_2} \\ \boldsymbol{i_3} \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} \boldsymbol{w_1} & \boldsymbol{w_2} & \boldsymbol{w_3} & \boldsymbol{w_4} & \boldsymbol{w_5} \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$
User-demographic	$Q \bowtie R$	$\begin{bmatrix} \boldsymbol{u} & \boldsymbol{i} & \boldsymbol{r} & \boldsymbol{\delta_1} & \boldsymbol{\delta_2} & \boldsymbol{\delta_3} \\ 1 & 1 & 5 & 1 & 0 & 1 \\ 2 & 2 & 4 & 0 & 1 & 1 \\ 2 & 3 & 3 & 0 & 1 & 1 \end{bmatrix}$	$W \in \mathbb{Z}^{d\times z}$ where $z = l$	$\begin{array}{c} \\ \boldsymbol{i_1} \\ \boldsymbol{i_2} \\ \boldsymbol{i_3} \end{array}\!\!\begin{array}{c} \begin{array}{ccc} \boldsymbol{w_1} & \boldsymbol{w_3} & \boldsymbol{w_2} \end{array} \\ \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$
No-demographic	$R \bowtie V$	$\begin{bmatrix} \boldsymbol{u} & \boldsymbol{i} & \boldsymbol{r} & \boldsymbol{\theta_1} & \boldsymbol{\theta_2} \\ 1 & 1 & 5 & 1 & 1 \\ 2 & 2 & 4 & 0 & 1 \\ 2 & 3 & 3 & 1 & 1 \end{bmatrix}$	$W \in \mathbb{Z}^{d\times z}$ where $z = m$	$\begin{array}{c} \\ \boldsymbol{i_1} \\ \boldsymbol{i_2} \\ \boldsymbol{i_3} \end{array}\!\!\begin{array}{c} \begin{array}{cc} \boldsymbol{w_1} & \boldsymbol{w_2} \end{array} \\ \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \end{array}$
Partial-demographic	$\tilde{Q} \bowtie R \bowtie V$	$\begin{bmatrix} \boldsymbol{u} & \boldsymbol{i} & \boldsymbol{r} & \boldsymbol{\delta_1} & \boldsymbol{\delta_2} & \boldsymbol{\theta_1} & \boldsymbol{\theta_2} \\ 1 & 1 & 5 & 1 & 0 & 1 & 1 \\ 2 & 2 & 4 & 0 & 1 & 0 & 1 \\ 2 & 3 & 3 & 0 & 1 & 1 & 1 \end{bmatrix}$	$W \in \mathbb{Z}^{d\times z}$ where $z < m + l$	$\begin{array}{c} \\ \boldsymbol{i_1} \\ \boldsymbol{i_2} \\ \boldsymbol{i_3} \end{array}\!\!\begin{array}{c} \begin{array}{cccc} \boldsymbol{w_1} & \boldsymbol{w_2} & \boldsymbol{w_3} & \boldsymbol{w_4} \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$

In this paper, we propose four combined-attribute schemes to efficiently merge the two data: (a) *all-demographic*, (b) *user-demographic*, (c) *no-demographic*, and (d) *partial-demographic*. The results are stored as the combined-attribute matrix $W \in \mathbb{Z}^{d \times z}$ where the value of z depends on which scheme is implemented.

A. All-demographic Scheme

The *all-demographic* scheme builds the combined-attribute matrix by joining the user demographic Q with the rating R and item attribute V. The join $Q \bowtie R \bowtie V$ results into attribute matrix $W \in \mathbb{Z}^{d \times z}$ where $z = m + l$. Table I shows the join data of $R \in \mathbb{Z}^{\mathbf{2 \times 3}}$, $Q \in \mathbb{Z}^{\mathbf{2 \times 3}}$, and $V \in \mathbb{Z}^{3 \times 2}$ results into $W \in \mathbb{Z}^{3 \times 5}$.

B. User-demographic Scheme

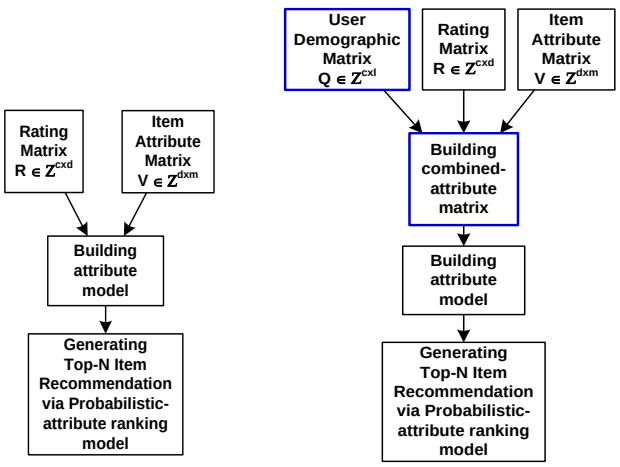
The *user-demographic* scheme builds the combined-attribute matrix from the user demographic Q and rating R. The join $Q \bowtie R$ results into attribute matrix $W \in \mathbb{Z}^{d \times z}$ where $z = l$. Table I shows the join data of $R \in \mathbb{Z}^{2 \times 3}$ and $Q \in \mathbb{Z}^{2 \times 3}$ results into $W \in \mathbb{Z}^{3 \times 3}$.

C. No-demographic Scheme

The *no-demographic* scheme builds the combined-attribute matrix by disregarding the user demographic Q and using only the rating R and item attribute V. The join $R \bowtie V$ results into attribute matrix $W \in \mathbb{Z}^{d \times z}$ where $z = m$. Table I shows the join data of $R \in \mathbb{Z}^{\mathbf{2 \times 3}}$ and $V \in \mathbb{Z}^{3 \times 2}$ results into $W \in \mathbb{Z}^{3 \times 2}$.

D. Partial-demographic Scheme

The *partial-demographic* scheme builds the combined-attribute matrix by joining the partial user demographics $\tilde{Q} \subseteq Q$ with the rating R and item attribute V matrices. The join $\tilde{Q} \bowtie R \bowtie \tilde{V}$ results into attribute matrix $W \in \mathbb{Z}^{d \times z}$ where $z < m + l$. Table I shows the join data of $R \in \mathbb{Z}^{\mathbf{2 \times 3}}$, $\tilde{Q} \in \mathbb{Z}^{\mathbf{2 \times 2}}$, and $V \in \mathbb{Z}^{3 \times 2}$ results into $W \in \mathbb{Z}^{3 \times 4}$.



(a) (b)

Fig. 2. The comparison of the framework of the probabilistic-attribute methods: (a) original version and (b) adapted version used in this paper

V. Probabilistic-Attribute Method

The probabilistic-attribute method [12] is a method that implements the combination of attribute model and probabilistic ranking approach for generating a list of recommendations. This method has been shown as an effective approach to solve the data sparsity issue.

In this paper, we use and adapt this method to implement our proposed combined-attribute schemes. The adaptation is required as the original probabilistic method builds its attribute model only based on the rating matrix and item attribute data, whereas our approach needs to also take into account the user demographic information. Therefore, we need to add the process of building a combined-attribute matrix before building the attribute model. Fig. 2 shows the comparison of the framework between the original (Fig. 2(a)) and the adapted version (Fig. 2(b)) of probabilistic-attribute methods.

A. Building Combined-Attribute matrix

The combined-attribute matrix $W \in \mathbb{Z}^{d \times z}$ is built by implementing the proposed combined-attribute schemes, details in the previous section.

B. Building Attribute Model

The attribute model is built within the four stages of constructing: (1) user-attribute frequency matrix $A \in \mathbb{Z}^{c \times z}$, that lists the attribute usage of each user on items; (2) attribute-item frequency matrix $B \in \mathbb{Z}^{z \times d}$ that lists the popularity of each attribute amongst users; (3) the neighborhood model J that is built based on the users' similarities $S \in \mathbb{Z}^{c \times c}$ calculated using the cosine function; and (4) the attribute model K that is built based on the fondness of each user towards each attribute $F \in \mathbb{Z}^{c \times z}$. Fig. 3 shows the algorithm of building the attribute model.

C. Generating Top-N Recommendation using Probabilistic-attribute ranking model

Generating the top-N recommendation for a target user u via probabilistic-attribute ranking model is conducted by probabilistically rank the list of items that the user has not rated conditional to his/her attribute model. This procedure consists of the three stages of the process of: (1) getting the list of items that the target user u has not selected, i.e., $\tilde{I}_u = I - I_u$ where $I_u \cap \tilde{I}_u = \emptyset$; (2) calculating the probability of user u for an item i given the attribute model K_u using the Naïve Bayes [15]; and (3) generating the top-N recommendation $TopN_u$ based on the probability values. Fig. 4 shows the algorithm of generating top-N recommendation using the probabilistic-attribute ranking model.

D. Complexity Analysis

The complexity of the learning process, i.e. building the attribute and probabilistic-attribute ranking models, in the probabilistic-attribute method is $O\left(y\left(c^2 dxz + c\left|\tilde{I}_u\right|\right)\right)$ where x and y are respectively the sizes of the neighborhood and attribute models. Note that z depends on which combined-attribute scheme is implemented.

Algorithm : **Building Attribute Model**
Input : Rating data $R \in \mathbb{Z}^{c \times d}$, User demographic $Q \in \mathbb{Z}^{c \times l}$, Item attribute $V \in \mathbb{Z}^{d \times m}$, Combined-attribute matrix $W \in \mathbb{Z}^{d \times z}$, Size of neighborhood model x, Size of attribute model y
Output : Attribute model K

1. Construct the user-attribute frequency matrix $A \in \mathbb{Z}^{c \times z}$:
   Initialization:
   $A \leftarrow zeros(c, z)$
   For each user u and attribute h:
   For each item i:
   if $r_{u,i}$ <> 0:
   $a_{u,h} \leftarrow a_{u,h} + w_{i,h}$
2. Construct the attribute-item frequency matrix $B \in \mathbb{Z}^{z \times d}$:
   Initialization:
   $B \leftarrow zeros(z, d)$
   For each attribute h and item i:
   if $w_{i,h}$ <> 0
   For each user u:
   if $r_{u,i}$ <> 0:
   $b_{h,i} \leftarrow b_{h,i} + 1$
3. Construct the neighbourhood model J:
   For each user u and user v:
   $s_{u,v} \leftarrow$ **cosine**(u, v, A, B)
   For each user u and user v:
   if $s_{u,v}$ is within top-x in $s_{u,*}$:
   $J_u \leftarrow J_u \cup \{v\}$
4. Construct the attribute model K:
   For each user u and attribute h:
   $f_{u,h} \leftarrow$ **fondness**(J_u, A, S)
   For each user u and attribute h:
   if $f_{u,h}$ is within top-y in $f_{u,*}$:
   $K_u \leftarrow K_u \cup \{h\}$

Fig. 3. Algorithm of building attribute model

Algorithm : **Generating top-N recommendation using Probabilistic-attribute Ranking Model**
Input : Rating data $R \in \mathbb{Z}^{c \times d}$, User-attribute frequency matrix $A \in \mathbb{Z}^{c \times z}$, Attribute-item $B \in \mathbb{Z}^{z \times d}$, Target user u, List of items I, List of user's items I_u, Attribute model K, number of recommendation N
Output : Top-N recommendation $TopN_u$

1. Get the list of items that the target user u has not selected:
   $\tilde{I}_u = I - I_u$ where $I_u \cap \tilde{I}_u = \emptyset$
2. Calculate the probability of target user u to select item $\tilde{I}_u$ given K_u:
   For each $i \in \tilde{I}_u$:
   $p_{u,i} \leftarrow$ **NaïveBayes**(u, i, K_u, R, A, B)
3. Generate the top-N item recommendation for target user u $TopN_u$:
   For each $i \in \tilde{I}_u$:
   if $p_{u,i}$ is within top-N in $p_{u,*}$
   $TopN_u \leftarrow TopN_u \cup \{i\}$

Fig. 4. Algorithm of generating top-N recommendation using probabilistic-attribute ranking model

VI. Empirical Analysis

Experiments are conducted to investigate the proposed combined-attribute schemes performances in solving the data sparsity and cold-start.

A. Dataset

This paper uses the MovieLens rating dataset (https://grouplens.org/datasets/movielens/) that has both the user demographics and item attribute information, detailed in Table II. We implement the 5-fold cross-validation method evaluation approach such that each fold has two sets: (1) training D_{train} set used to build the model, and (2) test set D_{test} used to evaluate the recommendation performance.

Next, we have to make sure that the MovieLens dataset suits the focus of this study, i.e., solving the data sparsity and cold-start problems. Table II shows that the sparsity of the dataset is 93.6953%, and therefore, it fulfills the data sparsity problem requirement. However, Table II also mentions that each user in the dataset has at least rated 20 items, i.e., a sufficient number for not categorizing the dataset as a cold-start dataset. This fact indicates that we need to filter further the dataset such that it satisfies the cold-start problem requirement, i.e., a target user that the target user of D_{test} has none or only very few number of rating in D_{train}. The following three variations of the MovieLens dataset are generated and used in the experiments:

- ***ML0***: The MovieLens dataset is filtered such a target user in D_{test} has no rating history in D_{train}. This dataset represents the condition in which the severe cold-start problem occurs, i.e., each target user u has $|I_u| = 0$.
- ***ML5***: The MovieLens dataset is filtered such a target user in D_{test} only has five number of ratings in D_{train}. This dataset represents the condition in which the moderate cold-start problem occurs, i.e., each target user u has $|I_u| = 5$.
- ***ML10***: The MovieLens dataset is filtered such a target user in D_{test} has at least ten number of ratings in D_{train}. This dataset represents the no cold-start problem condition, i.e., each target user u has $|I_u| \geq 10$.

TABLE II. THE MOVIELENS DATASET

Type of data		Number of data
Total user (c)		943
Total item (d)		1682
Total rating ($r_{*,*} == 1$)		100000
Percentage of density ($\frac{r_{*,*}==1}{c \cdot d}$)		6.3047
Percentage of sparsity ($100\% - $ density)		93.6953
Rating per user		≥ 20
User demographics (Δ)	Gender	2
	Age	7 (groups)
	Occupation	21
Item Attribute (Θ)	Genre	18

B. Methods

We develop six probabilistic-attribute methods from the four schemes proposed in this paper, as listed in Table III. We also benchmark the performance of the above methods with the standard recommendation methods:

- ***UB*** [16]: the user-based method that does not take into account the user demographics and item attribute information. *UB* generates the list of top-*N* item recommendation by employing the similarities between users.
- ***IB*** [6]: the item-based method that does not take into account the user demographics and item attribute information. *IB* generates the list of top-*N* item recommendation by employing the similarities between items.

C. Evaluation Criteria

We use the NDCG (Normalized Discounted Cumulative Gain) metric to measure the quality of recommendation performance. The NDCG score of a top-*N* recommendation for a target user u is formulated as:

$$NDCG_u(N) := \frac{DCG_u(N)}{IDCG(N)} \quad (1)$$

where

$$DCG_u(N) := \sum_{n=1}^{N} \frac{1}{\log_2(1+n)} \cdot \mathbb{I}(Top_u(n) \in GT_u) \quad (2)$$

$$IDCG(N) := \sum_{n=1}^{N} \frac{1}{\log_2(1+n)} \quad (3)$$

Note that $\mathbb{I}(\cdot)$ results 1 when the condition is fulfilled, and 0 if otherwise.

D. Results and Discussion

Fig. 5, Fig. 6, and Fig. 7 show the comparison of performance at top-1, 5, 10, 15, and 20 on respectively the *ML0*, *ML5*, and *ML10* datasets. We discuss two observations based on the results.

First, the *UB* and *IB* as the benchmarking methods perform very poorly compared to our methods that implement the proposed combined-attribute scheme. This observation confirms that our methods are superior in solving data sparsity and cold-start problems.

TABLE III. VARIATIONS OF THE PROBABILISTIC-ATTRIBUTE METHOD BASED ON THE PROPOSED COMBINED-ATTRIBUTE SCHEMES

Method	Implemented Scheme
PAD	*all-demographic*
PUD	*user-demographic*
PND	*no-demographic*
PPD_g	*partial-demographic*, using the gender demographic information
PPD_a	*partial-demographic*, using the age demographic information
PPD_o	*partial-demographic*, using the occupation demographic information

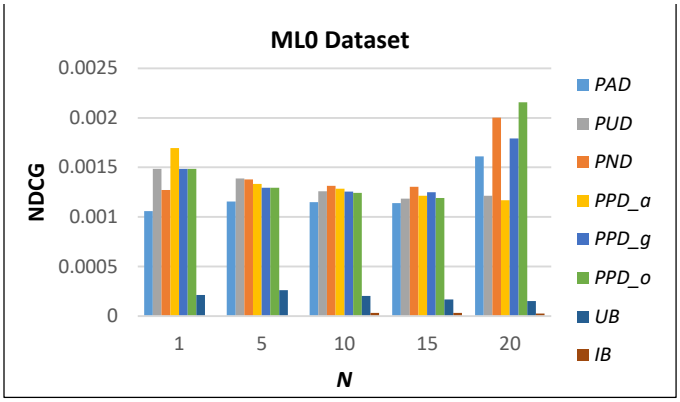


Fig. 5. The performance comparison on ML0 dataset

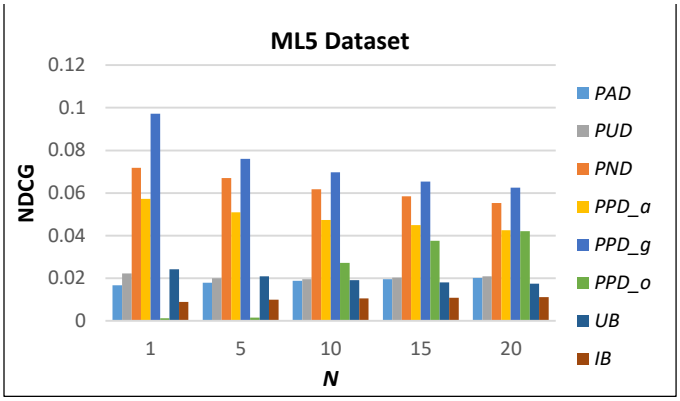


Fig. 6. The performance comparison on ML5 dataset

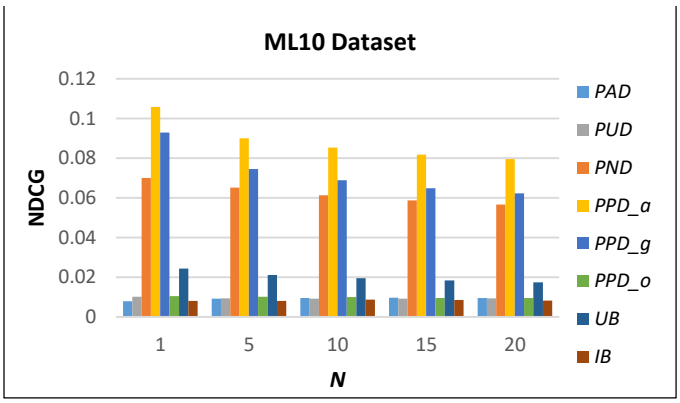


Fig. 7. The performance comparison on ML10 dataset

Second, the performance of each combined-attribute scheme varies on the different datasets:

- The *all-demographic* and *user-demographic* schemes achieve good performances only on the *ML0* dataset, i.e., the severe cold-start dataset. Therefore, we do not recommend to use those two schemes on any moderate cold-start or normal dataset.
- The *no-demographic* scheme achieves sensible performance on the *ML5* and *ML10* dataset. Its best achievement is when it is implemented on the severe cold-start dataset, even better than the *all-demographic* and *user-demographic* schemes. Note that implementing

the *no-demographic* scheme is equivalent to implementing the original probabilistic-attribute method.

- In general, the *partial-demographic* scheme performs better than the other three variations of the combined-attribute scheme on all datasets. This observation confirms that combining both the user demographic information, though not all of them, and the item attribute can efficiently solve the data sparsity and cold-start problems. However, the performance of the method would depend on the choice of which demographic information used.

VII. CONCLUSION

This paper solves the data sparsity and cold-start problems of recommendation system by proposing four combined-attribute schemes to be implemented on a probabilistic-attribute method. We develop six variations of methods out of the proposed schemes. For the benchmarking purpose, we also compare the performance of the methods with two standard recommendation methods. Our series of experiments show that our methods are superior in solving the data sparsity and cold-start problems, compared to the benchmark methods. In general, the *partial-demographic* scheme performs better than the other three variations of the combined-attribute scheme on all datasets. This observation confirms that combining both the user demographic information, though not all of them, and the item attribute can efficiently solve the data sparsity and cold-start problems. However, the performance of the method would depend on the choice of which demographic information used.

In the future, we plan to investigate the possibility of using a robust-based approach, such as the Dempster-shafer or entropic approaches [17], in the probabilistic-attribute method that implement our proposed combined-attribute schemes.

VIII. ACKNOWLEDGEMENT

This study is sponsored by the Ministry of Research, Technology and Higher Education (Indonesia), grant scheme: “Penelitian Dasar”, financial year: 2019.

REFERENCES

[1] J. A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience," *User Modeling and User-Adapted Interaction*, vol. 22, pp. 101-123, 2012.

[2] C. C. Aggarwal, *Recommender Systems: The Textbook*. Switzerland: Springer International Publishing, 2016.

[3] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowledge-Based Systems*, vol. 57, pp. 57-68, 2014.

[4] S. Loh, F. Lorenzi, R. Granada, D. Lichtnow, L. K. Wives, and J. P. de Oliveira, "Identifying Similar Users by their Scientific Publications to Reduce Cold Start in Recommender Systems," in Proceeding of *5th International Conference on Web Information Systems and Technologies (WEBIST 2009)*, pp. 593-600, 2009.

[5] N. Ifada and R. Nayak, "How Relevant is the Irrelevant Data: Leveraging the Tagging Data for a Learning-to-Rank Model," in Proceeding of *The 19th ACM International Conference on Web Search and Data Mining*, San Francisco, California, US, pp. 23-32, 2016.

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms," in Proceeding of *The 10th International Conference on World Wide Web*, Hong Kong, pp. 285-295, 2001.

[7] S. J. Gong, "Employing user attribute and item attribute to enhance the collaborative filtering recommendation," *Journal of Software*, vol. 4, pp. 883-890, 2009.

[8] N. Ifada and R. Nayak, "An Efficient Tagging Data Interpretation and Representation Scheme for Item Recommendation," in Proceeding of *The 12th Australasian Data Mining Conference*, Brisbane, Australia, pp. 205-215, 2014.

[9] N. Ifada, E. H. Prasetyo, and Mula’ab, "Employing sparsity removal approach and Fuzzy C-Means clustering technique on a movie recommendation system," in Proceeding of *The 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, Surabaya, Indonesia, pp. 157-162, 2018.

[10] Q. Chen, W. Li, and J. Liu, "Collaborative Filtering Algorithm Based on Item Attribute and Time Weight," in Proceeding of *The 2016 International Conference on Automatic Control and Information Engineering*, Hong Kong, pp. 12-15, 2016.

[11] X. Luo, Y. Ouyang, and Z. Xiong, "Improving neighborhood based Collaborative Filtering via integrated folksonomy information," *Pattern Recognition Letters*, vol. 33, pp. 263-270, 2012.

[12] N. Ifada, I. Syachrudin, M. K. Sophan, and S. Wahyuni, "Enhancing the Performance of Library Book Recommendation System by Employing the Probabilistic-Keyword Model on a Collaborative Filtering Approach," in Proceeding of *4th International Conference on Computer Science and Computational Intelligence (ICCSCI)*, Yogyakarta, Indonesia, pp. 345–352, 2019.

[13] P. Pirasteh, J. J. Jung, and D. Hwang, "Item-based collaborative filtering with attribute correlation: a case study on movie recommendation," in Proceeding of *Asian Conference on Intelligent Information and Database Systems*, Yogyakarta, Indonesia, pp. 245-252, 2014.

[14] S. K. Tiwari and S. K. Shrivastava, "An approach for recommender system by combining collaborative filtering with user demographics and items genres," *International Journal of Computer Applications*, vol. 128, pp. 16-24, 2015.

[15] L. D. Baker and A. K. McCallum, "Distributional Clustering of Words for Text Classification," in Proceeding of *The 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, pp. 96-103, 1998.

[16] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative Filtering Recommender Systems," *Human–Computer Interaction*, vol. 4, pp. 81-173, 2010.

[17] F. A. Palmieri and D. Ciuonzo, "Objective priors from maximum entropy in data classification," *Information Fusion*, vol. 14, pp. 186-198, 2013.