

## Article

# Modeling and Applying Implicit Dormant Features for Recommendation via Clustering and Deep Factorization

Alpamis Kutlimuratov<sup>1</sup>, Akmalbek Bobomirzaevich Abdusalomov<sup>1</sup> , Rashid Oteniyazov<sup>2</sup>, Sanjar Mirzakhalilov<sup>3</sup> and Taeg Keun Whangbo<sup>1,\*</sup>

<sup>1</sup> Department of Computer Engineering, Gachon University, Sujeong-gu, Seongnam-si 461-701, Korea

<sup>2</sup> Department of Telecommunication Engineering, Nukus Branch of Tashkent University of Information Technologies Named after Muhammad Al-Khwarizmi, Nukus 230100, Uzbekistan

<sup>3</sup> Department of Information-Computer Technologies and Programming, Tashkent University of Information Technologies Named after Muhammad Al-Khwarizmi, Tashkent 100200, Uzbekistan

\* Correspondence: tkwhangbo@gachon.ac.kr

**Abstract:** E-commerce systems experience poor quality of performance when the number of records in the customer database increases due to the gradual growth of customers and products. Applying implicit hidden features into the recommender system (RS) plays an important role in enhancing its performance due to the original dataset's sparseness. In particular, we can comprehend the relationship between products and customers by analyzing the hierarchically expressed hidden implicit features of them. Furthermore, the effectiveness of rating prediction and system customization increases when the customer-added tag information is combined with hierarchically structured hidden implicit features. For these reasons, we concentrate on early grouping of comparable customers using the clustering technique as a first step, and then, we further enhance the efficacy of recommendations by obtaining implicit hidden features and combining them via customer's tag information, which regularizes the deep-factorization procedure. The idea behind the proposed method was to cluster customers early via a customer rating matrix and deeply factorize a basic WNMF (weighted non-negative matrix factorization) model to generate customers preference's hierarchically structured hidden implicit features and product characteristics in each cluster, which reveals a deep relationship between them and regularizes the prediction procedure via an auxiliary parameter (tag information). The testimonies and empirical findings supported the viability of the proposed approach. Especially, MAE of the rating prediction was 0.8011 with 60% training dataset size, while the error rate was equal to 0.7965 with 80% training dataset size. Moreover, MAE rates were 0.8781 and 0.9046 in new 50 and 100 customer cold-start scenarios, respectively. The proposed model outperformed other baseline models that independently employed the major properties of customers, products, or tags in the prediction process.

**Keywords:** recommendation system; clustering-based recommendation system; heterogeneous information; weighted nonnegative matrix factorization; implicit features; tag information; deep factorization



**Citation:** Kutlimuratov, A.; Abdusalomov, A.B.; Oteniyazov, R.; Mirzakhalilov, S.; Whangbo, T.K. Modeling and Applying Implicit Dormant Features for Recommendation via Clustering and Deep Factorization. *Sensors* **2022**, *22*, 8224. <https://doi.org/10.3390/s22218224>

Academic Editor: Nikos Fotiou

Received: 9 September 2022

Accepted: 24 October 2022

Published: 27 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Currently, information overload has become an issue because of the advancement of Internet technology and the influx of data from all domains. Numerous well-known websites and e-commerce platforms utilize a variety of practical and efficient recommender systems (RS) to address this issue, enhance their level of customer care, and attract and maintain regular customers. For instance, TikTok and Instagram social networks, Netflix movie recommendations, the AppStore and Play Market marketplace, YouTube online videos, etc. Thus, customers can obtain more relevant content due to the support of the recommendation algorithms in speeding up searches. Recommendation systems are created based on the collected data; therefore, their deployment and architecture are impacted by

the diversity of data. Content-based filtering (CBF) and collaborative filtering (CF) [1–3] are two conventional methods for developing recommendation systems. When providing recommendations, CF-based approaches [4,5] exploit the products that customers have rated to anticipate unrated objects. These predictions are then automated by obtaining customer perceptions from the intended audience. All types of recommender system methods and approaches must prevent and handle [6] the cold-start problem and data sparsity, which are ongoing problems in the recommender system research field that can affect CF-based recommender systems. The rating matrices are unable to generate predictions due to data sparsity, which is brought about by less customer activity with products in a customer–product rating matrix. As a result, just 5 to 20 percent of matrices are contained with ratings. Additionally, the cold-start problem [7] occurs when there is inadequate knowledge of new customers and/or products to provide appropriate suggestions. CBF approaches create suggestions by examining customer–product interaction data that are available, which generally necessitates gathering explicit data [8,9]. For instance, content-based movie suggestions consider a film’s features that correspond to the customer’s previous preferences. Finding a link between movies and customers is crucial. On the other hand, researchers and developers attempt to design a recommendation system as a hybrid technique to improve recommendation accuracy [10–12] based on the combination of CF and CBF approaches to mitigate their individual limitations. In general, the aforementioned approaches to build an RS lead mostly to the cold-start problem, which has an impact on prediction accuracy when brought on by a lack of knowledge about new customers or objects. In addition, scalability and data sparsity issues arise in recommendation systems when the number of customers or products increases exponentially quickly; therefore, a recommendation technique should be quick and effective for big datasets. Therefore, clustering approaches aid in handling the sparsity problem more effectively and in reducing the processing time required for recommendation [13,14]. For example, several businesses, including Artsy, Netflix, and Pandora Internet Radio [15], have created unique clustering-based recommendation systems Art Genome Project, Micro-Genres of Movies, and Music Genome Project, respectively. In addition, many research works [16–19] have already been carried out on clustering and learning representative features of users in terms of similarity, which are important for modeling recommender system. Without regard to whether the length of the music list consumed is short or not, ref. [20] created a music clustering model to extract the interest points for a music recommendation system without having to predetermine the number of clusters. Bharti et al. [21] developed a model to deliver the best and fastest recommendations by maintaining and clustering current users and items of the system. Triyanna et al. [22] also proposed a recommendation model that integrates clustering technique and user behavior score-based similarity to reduce model computation complexity. To avoid the data sparsity problem, the research [23] presented a general framework to cluster users with respect to their tastes when the registers stored about the interactions between users and products are extremely scarce. Liu et al. [24] presented a clustering-based recommendation model that explores knowledge transfer and further aids the inferences about user interests.

Eventually, by leveraging tags and hierarchically organized hidden implicit features through early clustering and deep factorization, we attempted to solve the aforementioned problems in our study, thereby enhancing the performance of recommender systems as a whole. Generally, modern approaches for recommending videos mainly rely on ratings, textual data from the video such as labels (i.e., tags, reviews), or generating features from genre categories.

Specifically, by giving it to the prediction algorithm as a value, tag information, which consists of brief expression or words provided to movies by a customer who represents their affiliations or behavior, makes predictions easier. To enhance outcomes and address concerns with data sparsity and product cold start, engineers have already commented the advantages of providing recommendations through tags [25–27]. In contrast, hidden information that is organized hierarchically makes it easier to purposefully reveal information

about specific products or customers, such as product categories on e-commerce websites (e.g., AliExpress, Coupang, and Wish) or genre categories of movies on popular services (e.g., IMDb and Netflix) [28].

Movies and customers of actual, useful recommendation systems may display certain hierarchical structures. A customer (female) in Figure 1 may, for instance, choose movies from the drama main category, or more precisely, she might choose movies from the romantic drama subcategory. Similar to that, the product (the Amazfit GTS 2 smart watch) may be classified as belonging to the subcategory “smart watches” under the general heading “electronics”. An object is categorized into the relevant lower-level categories or nodes in a progressive manner. They will probably receive comparable ratings owing to the likelihood that products at the same hierarchical level would have similar features. Customers of the same level in the hierarchy are equivalently more likely to have similar tastes, which makes it more probable that they would evaluate specific goods likewise [29]. For this reason, when it comes to large dataset, we took advantage of early clustering customer–product interactions and integrated simultaneously tag information and acquired hierarchically structured hidden information of products and customers for prediction process to mitigate the above-mentioned issues and improve overall RS performance. We investigated the hierarchical structures of customers and products for recommender systems in part due to the importance of hierarchically organized hidden information and their limited availability. For the purpose of developing mathematical model, the study was focused on obtaining products and customers’ hierarchical structures for generating recommendations. Additionally, it was researched how to combine customers’ tag annotation through mathematically obtained customers and products’ hierarchical structures to create a structured model that serves as the foundation for a recommender system. To the best of our knowledge, the customers and products’ hierarchical structured implicit features and tag information have never been used in conjunction based on early clustered customer rating matrix and deep factorization although extensive research has been done to show how two characteristics may be used individually in recommender systems.

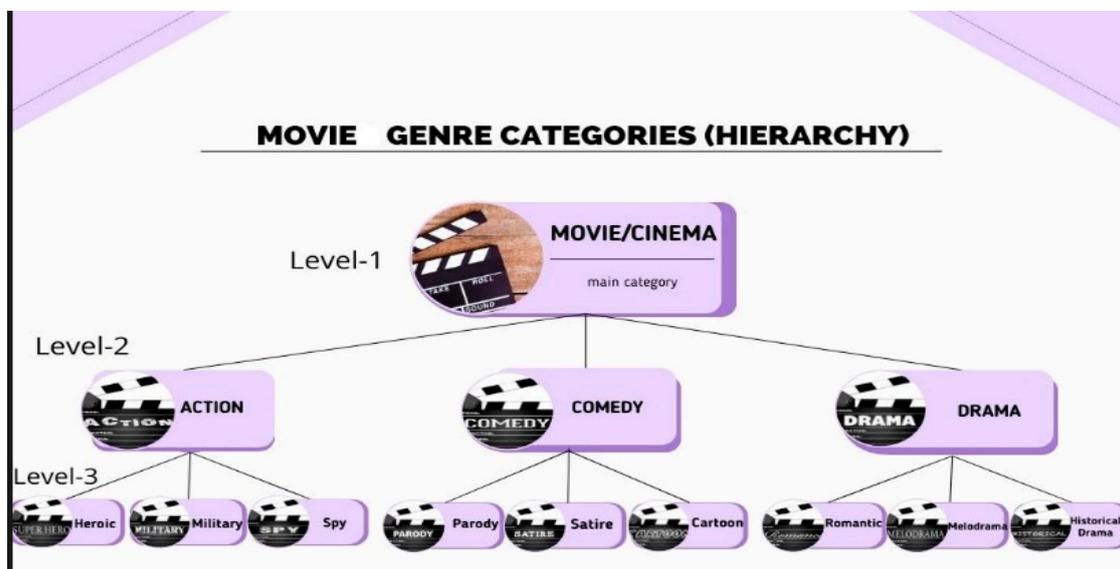


Figure 1. Movie categories.

In this article, a novel approach that employs clustering customers and deep factorization on customers and products procedure was proposed. Particularly, clustering technique is utilized to create customer groups with similar rating score history on products. After creating customer groups, the deep-factorization technique was applied to obtain hierarchically organized hidden implicit features of customers and products of one group, whereas the features used to predict ratings within group and tag information

were combined synchronously as additional parameter to regularize the deep-factorization process. Clustering the customers as one group in early stage and deeply factorizing the customer–product interaction matrix in that group to produce hierarchical relationships of customers and products through regularizing the factorization process via tag information to predict ratings were the guiding theory behind the suggested approach.

Our primary contributions via the suggested approach were as follows:

- Create the smoothed dense rating matrix using early clustering;
- Obtain hierarchically structured implicit features of customers and products;
- Mathematically model the synchronous impact of hierarchically structured implicit features and tag information for recommendation;
- Regularize via the auxiliary parameter based on tag information;
- Minimize product cold start and data sparsity difficulties;
- Increase the overall performance of recommendation when a dataset is large.

The rest of this paper is formatted as follows. In Section 2, we discuss several studies on producing hierarchical features, accurate MF techniques, and clustering- and tag-based recommender systems. In Sections 3 and 4, we go over the suggested approach in great depth and demonstrate its correctness through tests and comparisons against other methods. The results and scope of the future research are presented in Section 5. Finally, the referenced materials are cited, many of which are more contemporary works.

## 2. Literature Review

### 2.1. Clustering-Based Recommender Systems

Several strategies, largely based on clustering techniques, have been developed to avoid substantial job-specific feature engineering because of the dramatically increased size of the datasets. There are many research works and examples of pure advanced clustering methods [30–33] to cluster a dataset. Yunfan Li et al. [30] recommend a one-stage online clustering method that directly generates positive and negative instance pairs using data augmentation and afterwards projects the pairs in a feature space. The row and column spaces are used to perform the instance- and cluster-level contrastive learning, accordingly, by maximizing the similarities of positive pairings and reducing those of negative ones. Peng et al. [32] also developed a novel subspace deep clustering method to manage real data that do not have the linear subspace structure. Especially, in order to gradually map input data points into nonlinear latent spaces, the clustering methods learns a series of explicit transformations while maintaining the local and global subspace structure. Clustering techniques are applied as a first step to enhance the performance of recommender systems when customers suffer from information overload. In particular, based on customer evaluations generated by customers who are similar to target consumers, CF is a method that forecasts which products should be offered to target customers. Accordingly, we anticipate an improvement in forecasting precision owing to the early clustering of individuals with comparable characteristics. Therefore, there are many studies [34–37] related to the dependability of recommendations, variety and regularity, as well as the data sparsity on customer-preference matrices and shifts in customer personal tastes over period, which may help to solve recommendation systems. The authors of [37] presented a novel collaborative-filtering method that relies on clustering customer preferences to eliminate the effects of data scarcity. Customer groups were first created to differ between clients who had distinct tastes. Subsequently, based on the tastes of an active customer, a list of the nearest neighbors from the pertinent customer group (or groups) is then produced. The aim of [38] was to lower the cost of finding the closest neighbor using the k-means approach to cluster customers and potential projects. Moreover, the sparseness of the rating matrix of past customers and the cold start of new customers [7] restrict the practical usefulness of CF models. In other words, to address data heterogeneity and sparsity, [39] provided a combined filtering technique based on bi-clustering and information entropy.

It specifically uses bi-clustering to identify the dense modules of a rating matrix, followed by an information entropy metric to assess how similar a new customer is to

the dense modules. As previously demonstrated, clustering can be used as a preventive measure before recommending products.

### *2.2. Recommender Systems Based on Tag and Hierarchically Organized Data*

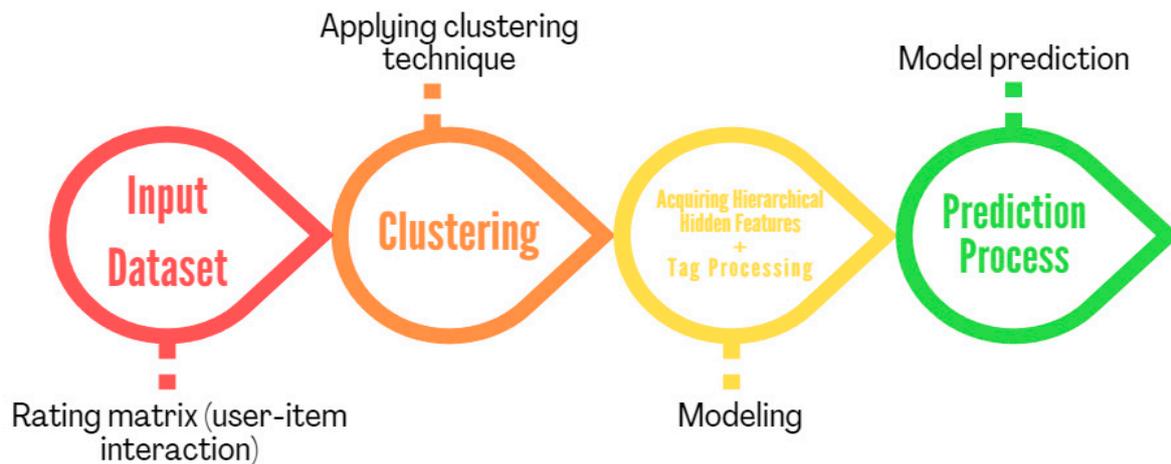
Recent research has taken advantage of tags and hierarchically organized features as additional characteristics to overcome concerns with data scarcity and cold starts in recommendation engines [25,40–42]. CF RS models are frequently used to predict ratings connected with customer’s previous experiences; however, they disregard expensive dormant features that avoid cold starts and sparse data problems, which in turn degrade performance. Because of this, supplemental features have been incorporated into the recommendation process by many studies [43–45]. A rich knowledge architecture, i.e., hierarchy with relationships, is frequently maintained through supplementary features. To increase recommendation accuracy and overcome the cold-start issue, Yang et al. [40] suggested an MF-based framework incorporating recursive regularization that examines the effects of hierarchically arranged features in customer–product interactions. In an attempt to discover more trustworthy neighbors, Lu et al. [42] created a framework that uses hierarchical relationships depending on the preferences of potential customers. The hierarchical product space rank (HIR) technique uses the product space’s inherent hierarchical structure to reduce data sparsity, which might impair the effectiveness of predictions [43]. Before providing recommendations, the majority of contemporary recommender systems comb through implicit and explicit features as relevant data, such as social information, photos, textual information, and ratings about products and customer qualities. Consequently, we can conclude that investigating tag data is crucial in recommendation systems because the data not only summarize the properties of products but also aid in determining customer preferences. As an illustration, to determine consumers’ preferred meal components and features [25], food suggestions are created using a model trained on a dataset of customer preferences obtained from tags and ratings provided in product forms. In their general solution, Karen et al. [27] suggested breaking down 3D correlations into three 2D correlations and modifying the CF algorithms to account for tags. In addition, Gilberto Borrego et al. [46] proposed a classification technique to recommend tags from topics in chat/message using NLP methods. Moreover, the research in [47] provided a semantic tagging strategy that makes use of Wikipedia’s knowledge to methodically identify content for social software engineering while also semantically grounding the tagging process. Despite the availability of advanced clustering methods [30–33], we aimed to show contribution of clustering technique with basic k-means algorithm to improve additionally the effectiveness of hierarchically organized hidden implicit features of customers and products in building a recommendation model.

Therefore, our proposed methodology is based on the early clustering of customer–product interactions and simultaneously integrating tag and hierarchically structured information into the rating-prediction process. In summary, existing MF models that use hierarchical and tag information individually deliver satisfactory results despite their complexity. However, to the best of our knowledge, there is no available advantageous study that seamlessly incorporates hierarchical and tag information simultaneously by early clustering customer–product interactions to improve the overall performance of the proposed recommender model. In summation, considering their complexity, current MF models that utilize tag information and hierarchically structured dormant implicit features separately produce good results. To the best of our knowledge, no useful study has yet been published that successfully combines the two data by early-clustering customer–product interactions to enhance the overall performance of the suggested recommender model.

### **3. The Proposed Approach**

This section illustrates our proposed methodology that clusters early customer–product interactions and predicts rating scores by acquiring hierarchical structured hidden features of products and customers simultaneously with a mathematically modeled combination of

customers tag annotation. In particular, a foundational model that serves as the foundation for generating dormant features is detailed after the clustering approach employed in this model is introduced. The specifics of the model's elements that mathematically represent the hidden, hierarchically organized dormant characteristics of products and consumers while also integrating tag data to produce an optimization issue are then discussed. Finally, a productive algorithm is provided for addressing this problem. The Figure 2 shows steps of the modeling process to reach the productive algorithm.

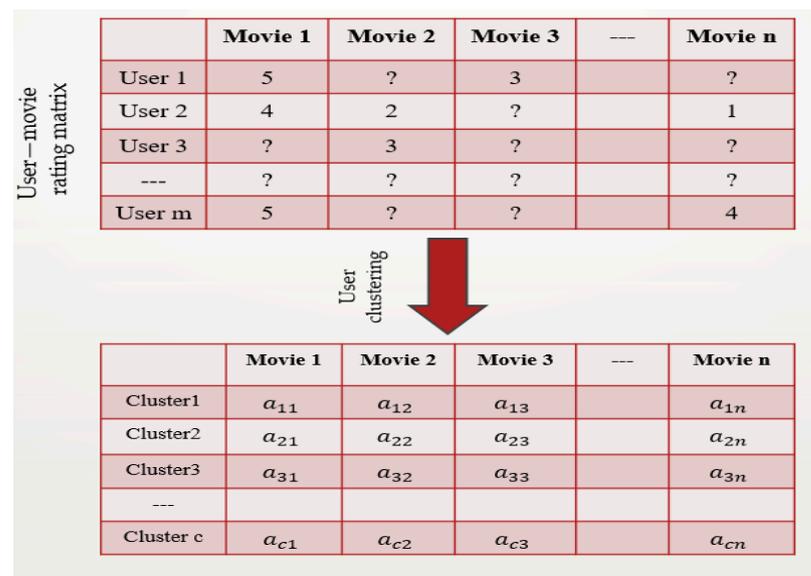


**Figure 2.** Modeling process.

The details of each component of the modeling process are provided in the following section.

### 3.1. Early Clustering

The time-consuming adjacent collaborative-filtering inquiry of the prospective customers in the whole customer domain results in the incapacity to guarantee the real-time need of recommender systems when customers and goods in e-commerce websites increasingly rise. Additionally, when the customer database's record count increases, it loses quality owing to its poor design. The main factor contributing to the low quality was the sparseness of the original dataset. This research offers a customized recommendation technique that uses an early customer-clustering method to address the issues of scalability and sparsity in building recommendation systems. In this study, we concentrate on grouping comparable customers using k-means clustering as a first step, and then, we further enhance the efficacy of recommendations by gathering the hidden attributes of customers and things. Customers are grouped into clusters based on a customer–product rating matrix. The closest neighbors of the target customer may be identified and used to smooth the prediction as needed based on the similarity of the target customer and cluster centers. Customer-clustering techniques determine groups of customers who seem to have common ratings. Predictions for a target customer can be generated after the clusters have been formed by averaging the feedback from other customers in that cluster. Each customer is portrayed using certain clustering approaches as having varying degrees of membership in various clusters. Next, the weighted average of the predictions for each cluster is calculated. However, the performance can be quite good once customer clustering is finished, as the size of the group that has to be evaluated is significantly lower [48]. The concept employs a customer-clustering algorithm to partition customers of the collaborative-filtering system into neighborhoods, as shown in Figure 3. Depending on the similarity criterion, the clustering algorithm may produce divisions of a specific size or specified number of partitions of variable sizes.



**Figure 3.** Customer clustering.

A detailed Algorithm 1 for the early customer-clustering technique is presented as follows:

---

**Algorithm 1:** The early customer-clustering technique

---

**Input:** User-item rating matrix, clustering number  $k$

**Output:** The smoothed dense user-item matrix

**Start:**

Select user set  $u = \{u_1, u_2, u_3 \dots, u_m\}$ ;

Select item set  $i = \{i_1, i_2, i_3 \dots, i_n\}$ ;

Select the top  $k$  rating users as the clustering  $cu = \{cu_1, cu_2, cu_3 \dots, cu_m\}$ ;

The clustering center is null as  $c = \{c_1, c_2, c_3 \dots, c_k\}$ ;

do

for each user  $u_i \in u$

for each cluster center  $cu_i \in cu$

calculate the similarity  $(u_i, cu_i)$ ;

end for

$sim(u_i, cu_m) = \max\{sim(u_i, cu_1), sim(u_i, cu_2) \dots, sim(u_i, cu_k)\}$ ;

$c_m = c_m \cup u_i$

end for

for each cluster  $c_i \in c$

for each user  $u_j \in u$

$cu_i = average(c_i, u_j)$ ;

end for

end for

while (c is not change)

**End**

---

Data sparsity is one of the difficulties associated with RS. In the customer-product rating dataset, we explicitly utilized customer clusters to which we applied our prediction technique for each individual cluster. By calculating the customer-clustering algorithm, we obtained dense customers who interacted with specific products. Therefore, the original sparse customer-product rating matrix then became a dense customer-product matrix in each cluster.

### 3.2. Founding Model

By grouping comparable customers using a customer-clustering algorithm, we obtained several customer clusters. Then, we applied our key idea to obtain hierarchically structured hidden features of customers in each cluster and related products. The core principle is a basic weighted nonnegative matrix factorization (WNMF) model, which is efficient and simple to use in recommender systems with huge and sparse datasets. Two nonnegative matrices  $\mathbf{P}$  and  $\mathbf{Q}$  with sizes  $n \times r$  and  $r \times m$  were created via the WNMF, which factorizes deeply clustered customer–product rating matrix.

$$\mathbf{R}' \approx \mathbf{P}\mathbf{Q} = \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{bmatrix} [q_1 q_2 \dots q_m] \quad (1)$$

The rating score assigned by  $p_i$  to  $q_j$  is then derived as  $\mathbf{R}'(i, j) = \mathbf{P}(i, :)\mathbf{Q}(:, j)$ .  $\mathbf{P}$  and  $\mathbf{Q}$  were estimated by resolving the following optimization issue:

$$\min_{\mathbf{P}, \mathbf{Q}} \|\mathbf{W} \odot (\mathbf{R} - \mathbf{P}\mathbf{Q})\|_{\mathbb{F}}^2 + \lambda(\|\mathbf{P}\|_{\mathbb{F}}^2 + \|\mathbf{Q}\|_{\mathbb{F}}^2) \quad (2)$$

where  $W$  is the hyperparameter that balances the aid of  $\mathbf{R}'(i, j)$  in the learning process such that  $W(i, j) = 1$  for  $\mathbf{R}'(i, j) > 0$ ; else,  $W(i, j) = 0$ .  $\odot$  is the Hadamard element-wise multiplication operator,  $\lambda$  is the regularization parameter applied to alleviate the overfitting and intricacy under learning, and  $\|\mathbf{P}\|_{\mathbb{F}}^2$  and  $\|\mathbf{Q}\|_{\mathbb{F}}^2$  are the Frobenius norms of the respective matrices [27].

### 3.3. Generating the Implicit Dormant Features

Customers and products have hidden and hierarchically organized implicit features. Figure 1 depicts a hierarchical structure for organizing film genres as one example. If we illustrate films as categorized, it is highly probable that films in detailed genres have more in common with one another than films in subgenres. Therefore, a film within the same specific genre as one with a high customer rating score must be appropriate to suggest. The overall performance of recommendations could be even more strengthened by synchronously acquiring the supplementary features included in hierarchical customer and product structures. The fundamental WNMF model is deeply factorized in order to obtain the hierarchically organized hidden implicit features of customers and products. Moreover, they can be mathematically modelled for the rating-prediction process based on the following theory.

The theory here is that:

- Products with similar features within the same hierarchical level are more likely to be given identical ratings.
- Customers within the same hierarchy level are more likely to have similar tastes, which makes it probable that they would score particular products identically.
- Thus, in this subsection, the way of generating hierarchically structured hidden implicit features of customers and products is represented with the WNMF. Finding useful information from the characteristics of highly linked customers and products in their interaction, which serves as the foundation for the prediction process, is one of the biggest problems of recommendation systems. However, these characteristics are commonly depicted in a hierarchy, i.e., a multilevel structure, as a nested tree of nodes (for instance, film genres or customer profession). Film genres and product categories on e-commerce websites are straightforward illustrations of a hierarchical structure. For instance, the film *The Godfather* (a product) may be categorized by moving through

the nodes of the hierarchical tree as shown in Figure 4: main category → subcategory, which appears as Crime → Gangster.

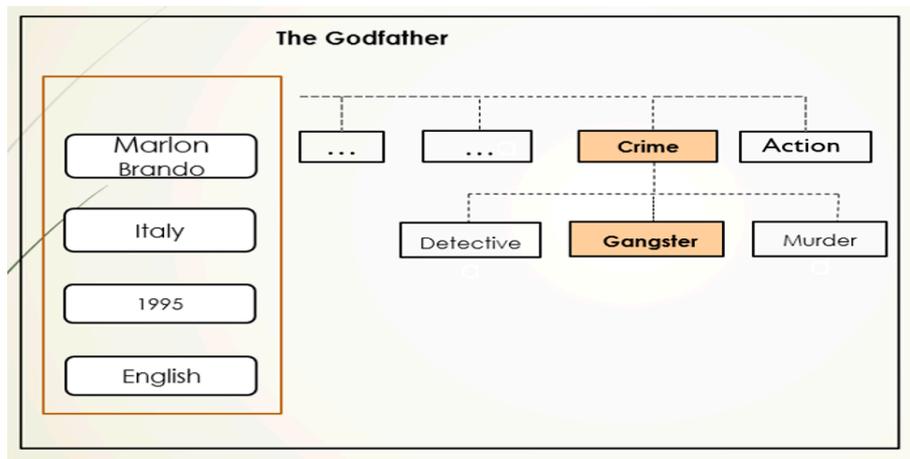


Figure 4. Movie genre categories and other flat features.

In a similar manner, LG OLED 4K TV (a product) can be classified in a hierarchical structure as Home appliance > TV/Video appliances > TV (primary category → subcategory → explicit subcategory), as shown in Figure 5.

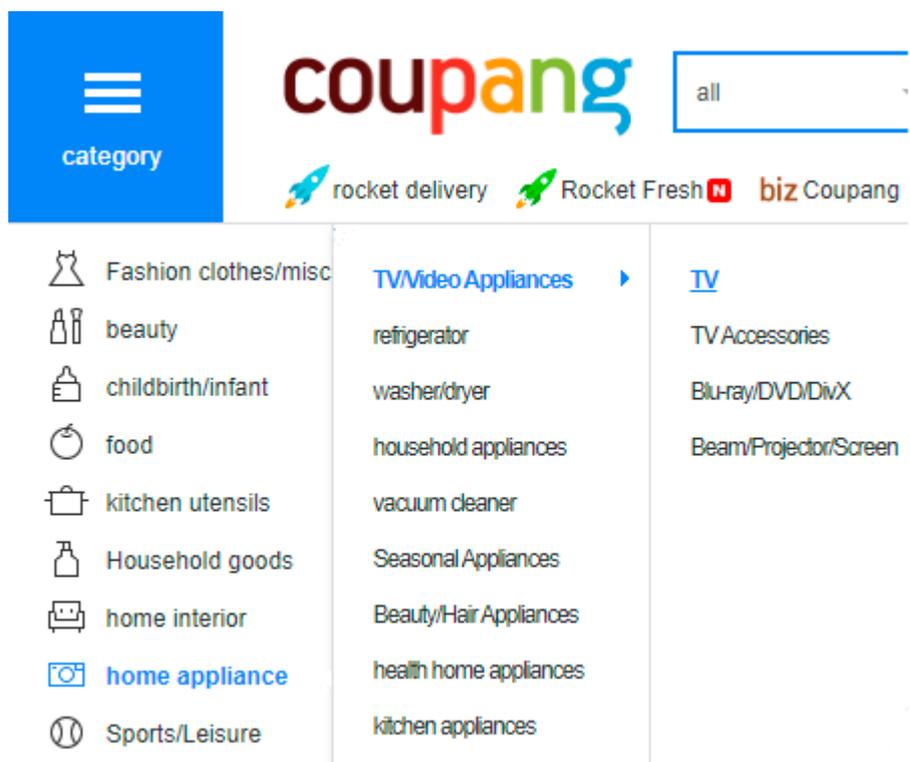


Figure 5. Coupang product categories.

Customer preferences follow a similar pattern. For example, a customer who prefers to score crime films may like the gangster in subgenre above others, and customers who regularly score products that have similar qualities when browsing a product catalog may be expressing coincidental preferences. In order to extract implicit hidden hierarchical features of customers and customers and then anticipate rating scores, the WNMF primary model described in Section 3.2 was utilized. The customer–product rating matrix in each

cluster was broken down into two nonnegative matrices,  $\mathbf{P}$  and  $\mathbf{Q}$ , which indicate customer preferences and product features, accordingly, and are stated as flat features.  $\mathbf{P}$  and  $\mathbf{Q}$  are nonnegative; thus, we factored them using a nonnegative matrix to understand the related hierarchically organized features, which allowed us predict the rating scores provided by Formula (1). In order to identify the latent projections of  $n$  customers and  $m$  products in an  $r$ -dimensional latent category,  $\mathbf{P}$  and  $\mathbf{Q}$  were retrieved so that  $\mathbf{P} \in \mathbb{R}^{n \times r}$  and  $\mathbf{Q} \in \mathbb{R}^{r \times m}$  were created (space). Because  $\mathbf{P}$  and  $\mathbf{Q}$  are nonnegative, they could be additionally factorized to mimic the hierarchical structure. Consequently, in a certain implementation,  $\mathbf{P}$  is factorized into two matrices,  $\mathbf{P}_1 \in \mathbb{R}^{n \times n_1}$  and  $\tilde{\mathbf{P}}_2 \in \mathbb{R}^{n_1 \times r}$ , as follows:

$$\mathbf{P} \approx \mathbf{P}_1 \tilde{\mathbf{P}}_2 \quad (3)$$

where  $n$  is the quantity of customers,  $r$  is the quantity of latent categories (space) in the main (first) hierarchically organized layer, and  $n_1$  is the quantity of subcategories in the next (second) hierarchically organized layer. Thus,  $\mathbf{P}_1 \in \mathbb{R}^{n \times n_1}$  depicts the association between  $n$  customers and  $n_1$  subcategories.  $\tilde{\mathbf{P}}_2$  stands for the second hierarchically organized layer of the customers' hierarchical structure, which was determined by relating the quantity of latent categories (space) in the main (first) hierarchically organized layer to the quantity of latent subcategories in the hierarchically organized layer. Formula (4) provides customer's third hierarchically organized layer, and then,  $\tilde{\mathbf{P}}_2$  is additionally factorized as  $\mathbf{P}_2 \in \mathbb{R}^{n_1 \times n_2}$  and  $\tilde{\mathbf{P}}_3 \in \mathbb{R}^{n_2 \times r}$ :

$$\mathbf{P} \approx \mathbf{P}_1 \mathbf{P}_2 \tilde{\mathbf{P}}_3 \quad (4)$$

where  $n_2$  is the quantity of subcategories in the third hierarchically organized layer. Thus, deep factorization on  $\mathbf{P}$  is used to determine the customer's  $x$ -th hierarchically organized layer.  $\mathbf{P}_x$  is carried out by factorizing  $\tilde{\mathbf{P}}_{x-1}$ , the latent category relationship matrix of the  $(x-1)$ th layer of the hierarchical structure, into nonnegative matrices as follows:

$$\mathbf{P} \approx \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_{x-1} \mathbf{P}_x \quad (5)$$

where  $\mathbf{P}_i \geq 0$  for  $i \in \{1, 2, \dots, x\}$ ,  $\mathbf{P}_1$  is an  $n \times n_1$  matrix such that  $\mathbf{P}_i$  is an  $n_{i-1} \times n_i$  matrix, and  $\mathbf{P}_x$  is  $n_{x-1} \times r$  matrix.

For  $\mathbf{Q}$ , the aforementioned factorization procedure (Figure 6) was replicated to acquire hierarchically structured implicit features of the products. For that, the association of  $m$  products with  $r$ -dimensional latent categories (space) is depicted as  $\mathbf{Q} \in \mathbb{R}^{r \times m}$ , which is additionally factorized into  $\mathbf{Q}_1 \in \mathbb{R}^{m_1 \times m}$  and  $\tilde{\mathbf{Q}}_2 \in \mathbb{R}^{r \times m_1}$  to characterize products' the hierarchically organized layer in the hierarchy as follows:

$$\mathbf{Q} \approx \tilde{\mathbf{Q}}_2 \mathbf{Q}_1 \quad (6)$$

where  $m_1$  is the quantity of sub-categories in the second hierarchically organized layer, and  $\mathbf{Q}_1 \in \mathbb{R}^{m_1 \times m}$  is the association of  $m$  products to the  $m_1$  latent subcategories. The latent category association of the nonnegative matrix  $\tilde{\mathbf{Q}}_2 \in \mathbb{R}^{r \times m_1}$  of the second hierarchically organized layer is defined as the affiliation between  $r$ -dimensional latent categories (space) in the first hierarchically organized layer and  $m_1$  latent subcategories in the second hierarchically organized layer. Formula (7) provides the third hierarchically organized layer of products, where  $\tilde{\mathbf{Q}}_2$  is also factorized as  $\mathbf{Q}_2 \in \mathbb{R}^{m_2 \times m_1}$  and  $\tilde{\mathbf{Q}}_3 \in \mathbb{R}^{r \times m_2}$ , where  $m_2$  is the number of subcategories in the third hierarchically organized layer:

$$\mathbf{Q} \approx \tilde{\mathbf{Q}}_3 \mathbf{Q}_2 \mathbf{Q}_1 \quad (7)$$

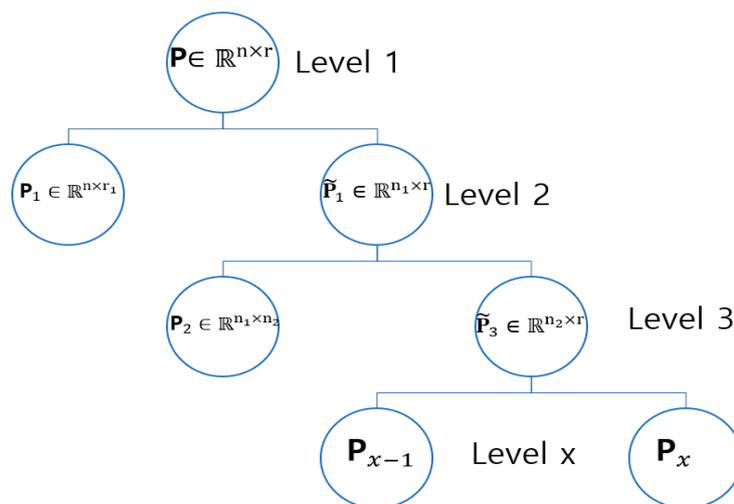


Figure 6. Generating customer’s implicit dormant features.

As shown in Figure 7, carrying out the deep-factorization process with  $Q$  assures the products’  $y$ -th hierarchically organized layer,  $Q_y$  which is accomplished by factorizing  $\tilde{Q}_{y-1}$ , in the  $(y - 1)$ th layer of the hierarchy, as follows:

$$Q \approx Q_y Q_{y-1} \dots Q_2 Q_1 \tag{8}$$

where  $Q_j \geq 0$  for  $j \in \{1, 2, \dots, y\}$ ,  $Q_1$  is an  $m_1 \times m$  matrix such that  $Q_j$  is an  $m_j \times m_{j-1}$  matrix, and  $Q_y$  is an  $r \times m_{y-1}$  matrix.

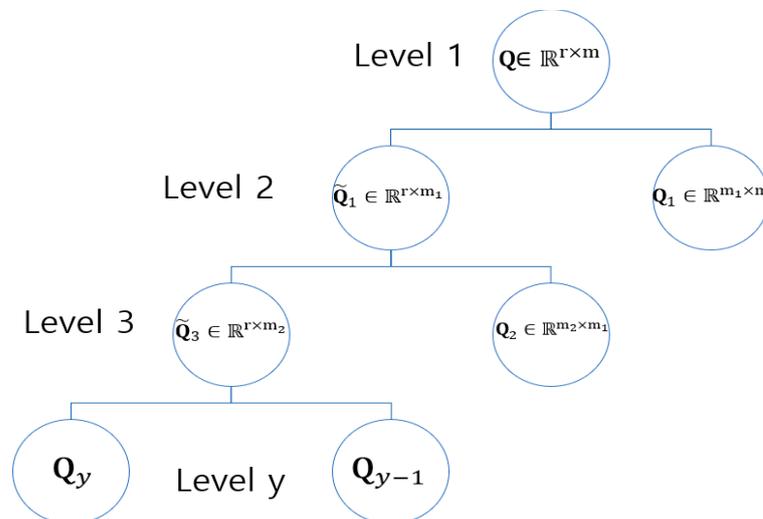


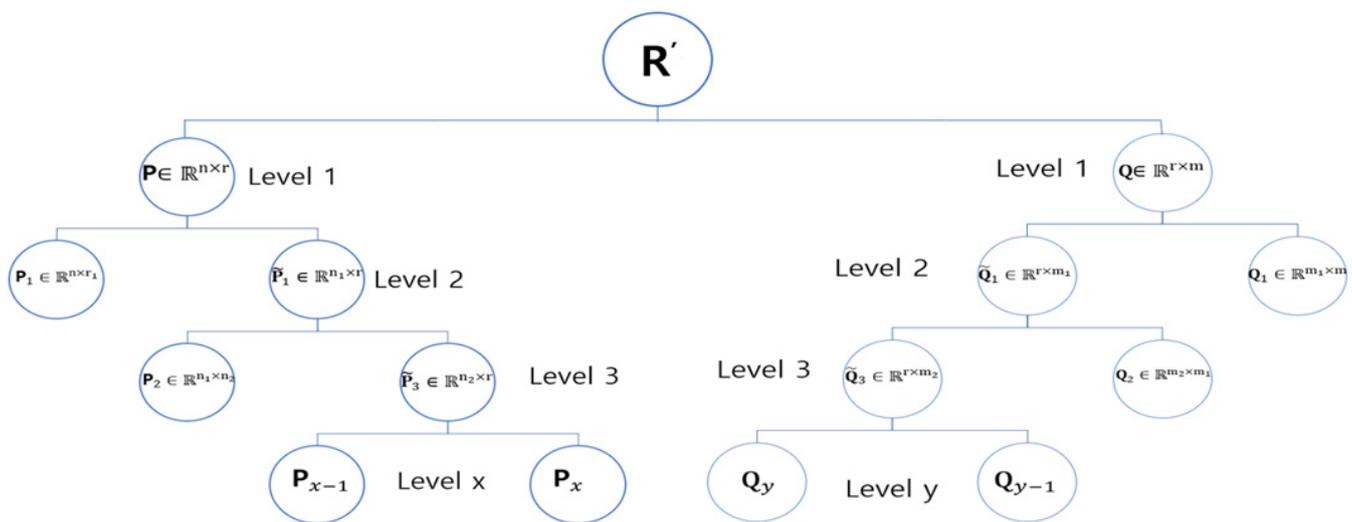
Figure 7. Generating product’s implicit dormant features.

Conclusively, to create a systematic model that depicts the products and customers’ hierarchically organized layers, the following optimization issue must be solved:

$$\min_{P_1, \dots, P_x, Q_1, \dots, Q_y} \|W \odot (R - P_1 \dots P_x Q_y \dots Q_1)\|_F^2 + \lambda \left( \sum_{i=1}^x \|P_i\|_F^2 + \sum_{j=1}^y \|Q_j\|_F^2 \right) \tag{9}$$

where  $P_i \geq 0$  for  $i \in \{1, 2, \dots, x\}$ , and  $Q_j \geq 0$  for  $j \in \{1, 2, \dots, y\}$ .

Figure 8 illustrates the rating-prediction approach that generates the products and customers’ hierarchically organized layers.



**Figure 8.** The design of generating products and customers' implicit dormant features.

### 3.4. Integrating Customers' Tag Annotation

While customer ratings are considered as the main data source for a rating-prediction process, the customers' characteristics or products' properties are not considered in most research works. In this case, tags offer valuable auxiliary information for recommender systems because they represent customer preferences or product characteristics. In addition, tag information plays a crucial role in recommendation systems; obviously, it is natural customer-generated resource text data that express customers' interests in various ways towards products. Customers who post similar tags are likely to have similar interests; therefore, they are likely to give similar ratings to products. The auxiliary information provided by tags leads to the advancement of recommendation systems to the next level. Tag information is a word or a short phrase for products given by customers. Thus, customers' preferences for products may be indirectly expressed by the tags, and this tag information could offer valuable information for the movie prediction process. Therefore, in order to infer a correlation between the supplemental information requested from WNMF and tag constant repetition in products [5], tag information was specifically integrated into our suggested technique. For instance, a customer's "organized crime" tag applied to the movie *The Godfather* (product) may also be applied to other items with comparable features, which is represented in the degree of repetition.

As illustrated in Figure 9, customer A often uses tags Mafia and Gangster, whereas customer B uses tags Crime and Mafia; hence, both customers may like movies A and B. Here, the intersection of the tagging history between customers is the Mafia tag. Hence, they had a similar tagging history. Therefore, a similar tagging history may indicate a similar customer's personal interest in products and/or similarities between products. Additionally, tags can be seen as product descriptions, which may help define a product's character or nature. The purpose of using tag information is to find similarities between products based on the tag information as illustrated in Figure 10 and then use product similarity as an additional parameter to organize the factorization process. The idea behind incorporating tag information is to use product similarities based on tag information to regularize the factorization process of the proposed prediction model. Therefore, the matrix factorization process of the weighted nonnegative matrix factorization model is regularized based on tag information. For clarity, we formed two product-specific latent feature vectors that are as similar as possible if the two products have similar tag information.

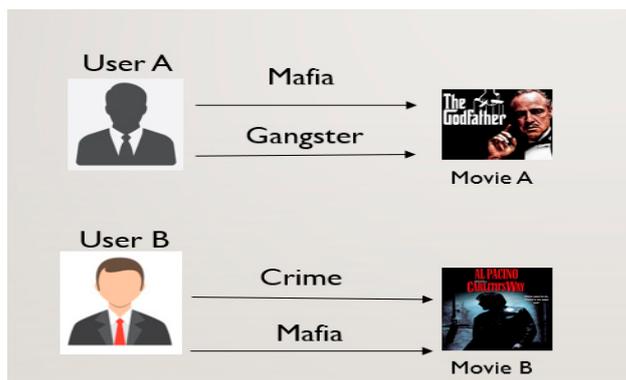


Figure 9. Tag annotation.

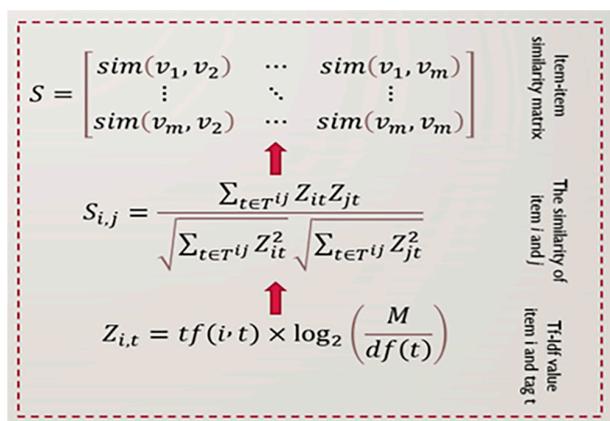


Figure 10. Tag processing part.

Thus, in order to finish our rating-prediction approach, tag information is utilized to regularize the deep-factorization process of a fundamental WNMf model. In essence, we want to create two similar-natured, product-specific latent feature vectors from our fundamental WMNF model’s factorization process. These vectors would comprise products with comparable tag information. Each tag information matrix  $T$  with components  $T_{it}$  for product  $i$  and tag  $t$  is a  $tf * idf$  value [49].

$$T_{it} = tf(i, t) * \log_2\left(\frac{m}{df(t)}\right) \tag{10}$$

where  $tf(i, t)$  is the normalized frequency of  $t$  occurring in  $i$ ,  $df(t)$  is the quantity of products containing  $t$ , and  $m$  is the total quantity of products. Thus, the similarity between products  $i$  and  $j$  is estimated using the cosine similarity formula given as follows:

$$S_{i,j} = \frac{\sum_{t \in T^{ij}} T_{it} T_{jt}}{\sqrt{\sum_{t \in T^{ij}} T_{it}^2} \sqrt{\sum_{t \in T^{ij}} T_{jt}^2}} \tag{11}$$

where  $T^{ij}$  is the index of tags occurring in both products  $i$  and  $j$ . The two product-specific latent feature vectors that are most similar are then obtained by affixing a product similarity regularization criterion function to the WNMf model as follows:

$$\begin{aligned} \frac{\beta}{2} \sum_{i=1}^N \sum_{j=1}^N S_{i,j} \|q_i - q_j\|_F^2 &= \frac{\beta}{2} \sum_{i=1}^N \sum_{j=1}^N \left[ S_{i,j} \sum_{r'=1}^r (q_{r' i} - q_{r' j})^2 \right] \\ &= \frac{\beta}{2} \sum_{r'=1}^r Q_{r'^s} L Q_{r'^s}^T = \frac{\beta}{2} \text{tr}(Q L Q^T) \end{aligned} \tag{12}$$

where  $S_{i,j}$  defines the similarity between  $i$  and  $j$ ;  $q_1, q_2, \dots, q_m$  are latent characteristic vectors that populate  $\mathbf{Q}$ ;  $r$  is the dimension of each product in the vector; i.e.,  $q_{r' i}$  and  $q_{r' j}$  are the values of vector products  $i$  and  $j$  of the  $r'$ th dimension;  $\mathbf{L}$  defines the Laplacian matrix given by  $\mathbf{L} = \mathbf{D} - \mathbf{S}$  for a diagonal matrix  $\mathbf{D}$  such that  $\mathbf{D}_{ij} = \sum_j S_{ij}$ .  $\text{tr}(\cdot)$  is a trace of the matrix;  $\beta$  is an extra regularization parameter that controls the balance of the tag information [50].

Mixing Formulas (9) and (12) utilized for the rating-prediction process and the corresponding objective function is minimized optimally.

$$\underset{\mathbf{P}_1, \dots, \mathbf{P}_x, \mathbf{Q}_1, \dots, \mathbf{Q}_y}{\text{Min}} \quad \|\mathbf{W} \odot (\mathbf{R} - \mathbf{P}_1 \dots \mathbf{P}_x \mathbf{Q}_y \dots \mathbf{Q}_1)\|_{\text{F}}^2 + \lambda \left( \sum_{i=1}^x \|\mathbf{P}_i\|_{\text{F}}^2 + \sum_{j=1}^y \|\mathbf{Q}_j\|_{\text{F}}^2 \right) + \frac{\beta}{2} \text{tr}(\mathbf{Q} \mathbf{L} \mathbf{Q}^T) \quad (13)$$

where  $\mathbf{P}_i \geq 0$  for  $i \in \{1, 2, \dots, x\}$ , and  $\mathbf{Q}_j \geq 0$  for  $j \in \{1, 2, \dots, y\}$ .

### 3.5. Optimization

Any algorithm that determines the minimum or maximum of a function must first determine the best method for performing the rating procedure. Numerous studies have employed various optimization strategies and uncertainty simulation techniques in recent years to address optimization issues involving unknown factors. Because of the non-convexity of the objective function, optimization problems are inherently challenging tasks. The superiority of any approach that can be used in a recommendation system is also a result of the problem being solved. Thus, the switching operation [51] is utilized as our optimization technique. In particular, all variables are updated reciprocally in the abovementioned objective function, leading to the function becoming convex.

#### 3.5.1. Updating $\mathbf{P}_i$

When  $\mathbf{P}_i$  is updated, terms distinct to  $\mathbf{P}_i$  are eliminated by fixing the remaining variables, and the last objective function is declared as

$$\underset{\mathbf{P}_i \geq 0}{\text{min}} \|\mathbf{W} \odot (\mathbf{R} - \mathbf{A}_i \mathbf{P}_i \mathbf{H}_i)\|_{\text{F}}^2 + \lambda \|\mathbf{P}_i\|_{\text{F}}^2 \quad (14)$$

where  $\mathbf{A}_i$  and  $\mathbf{H}_i$  for  $1 \leq i \leq x$  are determined as:

$$\mathbf{A}_i = \begin{cases} \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_{x-1} & \text{if } i \neq 1 \\ \mathbf{I} & \text{if } i = 1 \end{cases} \quad (15)$$

$$\mathbf{H}_i = \begin{cases} \mathbf{P}_{i+1} \dots \mathbf{P}_x \mathbf{Q}_y \dots \mathbf{Q}_1 & \text{if } i \neq x \\ \mathbf{Q}_y \dots \mathbf{Q}_1 & \text{if } i = x \end{cases} \quad (16)$$

The Lagrangian function in Formula (14) is:

$$\mathbf{L}(\mathbf{P}_i) = \|\mathbf{W} \odot (\mathbf{R} - \mathbf{A}_i \mathbf{P}_i \mathbf{H}_i)\|_{\text{F}}^2 + \lambda \|\mathbf{P}_i\|_{\text{F}}^2 - \text{Tr}(\mathbf{M}^T \mathbf{P}_i) \quad (17)$$

where  $\mathbf{M}$  indicates the Lagrangian multiplier. The derivative of  $\mathbf{L}(\mathbf{P}_i)$  with respect to  $\mathbf{P}_i$  is then given by

$$\frac{\partial \mathbf{L}(\mathbf{P}_i)}{\partial \mathbf{P}_i} = 2\mathbf{A}_i^T \|\mathbf{W} \odot (\mathbf{A}_i \mathbf{P}_i \mathbf{H}_i - \mathbf{R})\|_{\text{F}}^T + 2\lambda \mathbf{P}_i - \mathbf{M} \quad (18)$$

By utilizing the Karush–Kuhn–Tucker complementary requirement [52,53] that is equal to 0,  $\mathbf{M}(s, t) \mathbf{P}_i(s, t) = 0$ , we derive

$$\left[ \mathbf{A}_i^T [\mathbf{W} \odot (\mathbf{A}_i \mathbf{P}_i \mathbf{Q} - \mathbf{R})] \mathbf{H}_i^T + \lambda \mathbf{P}_i \right] (s, t) \mathbf{P}_i(s, t) = 0 \quad (19)$$

Lastly, the updated rule of  $\mathbf{P}_i$  is estimated utilizing

$$\mathbf{P}_i(s, t) \leftarrow \mathbf{P}_i(s, t) \sqrt{\frac{\left[\mathbf{A}_i^T(\mathbf{W} \odot \mathbf{R})\mathbf{H}_i^T\right](s, t)}{\left[\mathbf{A}_i^T(\mathbf{W} \odot (\mathbf{A}_i\mathbf{P}_i\mathbf{H}_i))\mathbf{H}_i^T + \lambda\mathbf{P}_i\right](s, t)}} \quad (20)$$

### 3.5.2. Updating $\mathbf{Q}_i$

Likewise, for  $\mathbf{Q}_i$ , the distinct terms are initially eliminated by fixing the remaining variables, and the last objective function is declared as

$$\min_{\mathbf{Q}_i \geq 0} \|\mathbf{W} \odot (\mathbf{R} - \mathbf{B}_i\mathbf{Q}_i\mathbf{K}_i)\|_F^2 + \lambda\|\mathbf{Q}_i\|_F^2 + \frac{\beta}{2}\text{tr}(\mathbf{Q}\mathbf{L}\mathbf{Q}^T) \quad (21)$$

where  $\mathbf{B}_i$  and  $\mathbf{K}_i$  for  $1 \leq i \leq x$  are determined as:

$$\mathbf{B}_i = \begin{cases} \mathbf{P}_1 \dots \mathbf{P}_x \mathbf{Q}_y \dots \mathbf{Q}_{y+1} & \text{if } i \neq y \\ \mathbf{P}_1 \dots \mathbf{P}_x & \text{if } i = y \end{cases} \quad (22)$$

$$\mathbf{K}_i = \begin{cases} \mathbf{Q}_{y-1} \dots \mathbf{Q}_1 & \text{if } i \neq 1 \\ \mathbf{I} & \text{if } i = 1 \end{cases} \quad (23)$$

We could then estimate the updated rule for  $\mathbf{Q}_i$  in the same way as  $\mathbf{P}_i$ :

$$\mathbf{Q}_i(s, t) \leftarrow \mathbf{Q}_i(s, t) \sqrt{\frac{\left[\mathbf{B}_i^T(\mathbf{W} \odot \mathbf{R})\mathbf{K}_i^T + \frac{\beta}{2}\text{tr}(\mathbf{Q}\mathbf{L}\mathbf{Q}^T)\right](s, t)}{\left[\mathbf{B}_i^T(\mathbf{W} \odot (\mathbf{B}_i\mathbf{Q}_i\mathbf{K}_i))\mathbf{K}_i^T + \lambda\mathbf{Q}_i + \frac{\beta}{2}\text{tr}(\mathbf{Q}\mathbf{L}\mathbf{Q}^T)\right](s, t)}} \quad (24)$$

The approximation of the components in the suggested approach is expected to be revealed through optimization using the aforesaid updating strategies for  $\mathbf{P}_i$  and  $\mathbf{Q}_i$ . In order to derive a preliminary estimation of the matrices  $\mathbf{P}_i$  and  $\mathbf{Q}_i$ , every hierarchically organized layer is pretrained. The customer-product rating matrix in each cluster is factorized into  $\tilde{\mathbf{P}}_i \tilde{\mathbf{Q}}_i$  by calculating Formula (2).  $\tilde{\mathbf{P}}_i$  and  $\tilde{\mathbf{Q}}_i$  are then additionally factorized into  $\tilde{\mathbf{P}}_i \approx \mathbf{P}_1 \tilde{\mathbf{P}}_2$  and  $\tilde{\mathbf{Q}}_i \approx \tilde{\mathbf{Q}}_2 \mathbf{Q}_1$ , respectively. The deep-factorization process is maintained until the  $p$ th customer and  $q$ th product hierarchically organized layers are acquired. The fine-tuning is accomplished by updating  $\mathbf{P}_i$  and  $\mathbf{Q}_i$  utilizing Formulas (20) and (24) accordingly: The initial movement covers updating  $\mathbf{Q}_i$  in order and then  $\mathbf{P}_i$  in sequence. Lastly, the proposed prediction rating matrix is equal to  $\mathbf{R}' = \mathbf{P}_1 \dots \mathbf{P}_x \mathbf{Q}_y \dots \mathbf{Q}_1$ .

### 3.6. Convergence Analysis

The suggested approach's convergence was examined using the following methodology. The aide function in [54] was utilized to demonstrate the approach's convergence.

**Definition 1.** The aide function [54] is determined as  $G(h, h')$  for  $F(h)$  if the following criteria are met.

$$G(h, h') \geq F(h), \quad G(h, h) = F(h) \quad (25)$$

**Assumption 1.** If  $G$  [54] is an aide function for  $F$ , then  $F$  is nonincreasing under the update.

$$h^{(t+1)} = \text{arh min} G(h, h^{(t)}) \quad (26)$$

**Proof.**

$$F(\mathbf{h}^{t+1}) \leq G(\mathbf{h}^{t+1}, \mathbf{h}^{(t)}) \leq G(\mathbf{h}^{(t)}, \mathbf{h}^{(t)}) \leq G(\mathbf{h}^{(t)}) \quad (27)$$

□

**Assumption 2.** For any matrix  $\mathbf{A} \in \mathbb{R}_+^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}_+^{k \times k}$ ,  $\mathbf{S} \in \mathbb{R}_+^{k \times k}$ , and  $\mathbf{S}' \in \mathbb{R}_+^{k \times k}$ , where  $\mathbf{A}$  and  $\mathbf{B}$  are symmetric [55,56], the following inequality holds:

$$\sum_{s=1}^n \sum_{t=1}^k \frac{(AS'B)(s,t)S^2(s,t)}{S'(s,t)} \geq \text{Tr}(S^T ASB) \quad (28)$$

By introducing quadratic terms and eliminating terms that are distracting to  $\mathbf{P}_i$ , the objective function in Formula (14) may be expressed as follows:

$$J(\mathbf{P}_i) = \text{Tr}(-2\mathbf{A}_i^T(\mathbf{W} \odot \mathbf{R})\mathbf{H}_i^T \mathbf{P}_i^T) + \text{Tr}(\mathbf{A}_i^T(\mathbf{W} \odot (\mathbf{A}_i^T \mathbf{P}_i \mathbf{H}_i))\mathbf{H}_i^T \mathbf{P}_i^T) + \text{Tr}(\lambda \mathbf{P}_i \mathbf{P}_i^T) \quad (29)$$

**Theorem 1.**

$$\begin{aligned} G(\mathbf{P}, \mathbf{P}') &= -2 \sum_{s,t} (\mathbf{A}_i^T(\mathbf{W} \odot \mathbf{R})\mathbf{H}_i^T)(s,t) \mathbf{P}_i(s,t) \left(1 + \log \frac{\mathbf{P}_i(s,t)}{\mathbf{P}'_i(s,t)}\right) \\ &+ \sum_{s,t} \frac{(\mathbf{A}_i^T(\mathbf{W} \odot (\mathbf{A}_i^T \mathbf{P}_i \mathbf{H}_i))\mathbf{H}_i^T)(s,t) \mathbf{P}_i^2(s,t)}{\mathbf{P}'_i(s,t)} + \text{Tr}(\lambda \mathbf{P}_i \mathbf{P}_i^T) \end{aligned} \quad (30)$$

The above function is an aide function for  $J(\mathbf{P}_i)$ . Moreover, it is a convex function in  $(\mathbf{P}_i)$ , and its global minimum is

$$\mathbf{P}_i(s,t) \leftarrow \mathbf{P}_i(s,t) \sqrt{\frac{[\mathbf{A}_i^T(\mathbf{W} \odot \mathbf{R})\mathbf{H}_i^T](s,t)}{[\mathbf{A}_i^T(\mathbf{W} \odot (\mathbf{A}_i \mathbf{P}_i \mathbf{H}_i))\mathbf{H}_i^T + \lambda \mathbf{P}_i](s,t)}} \quad (31)$$

**Proof.** The confirmation is identical to that of [55]; thus, the details are skipped. □

**Theorem 2.** Updating  $\mathbf{P}_i$  using Formula (20) monotonically decreases the value of the objective in Formula (13).

**Proof.** With Assumption 1 and Theorem 1, we have:

$$J(\mathbf{P}_i^{(0)}) = G(\mathbf{P}_i^{(0)}, \mathbf{P}_i^{(0)}) \geq G(\mathbf{P}_i^{(1)}, \mathbf{P}_i^{(0)}) \geq J(\mathbf{P}_i^{(1)}) \quad (32)$$

□

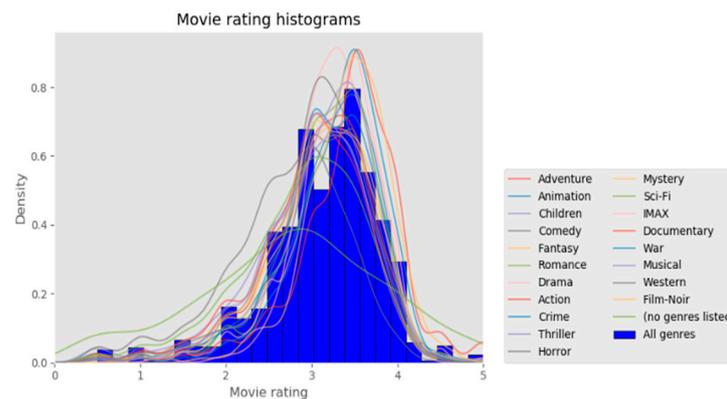
Particularly,  $J(\mathbf{P}_i)$  reduces monotonically. Analogously, the update rule for  $\mathbf{Q}_i$  monotonically reduces the value of the objective in Formula (13). We can demonstrate that the optimization technique of the suggested approach converges since the value of the objective in Formula (13) is at best edged by "0".

## 4. Model Evaluation

### 4.1. Data Preparation

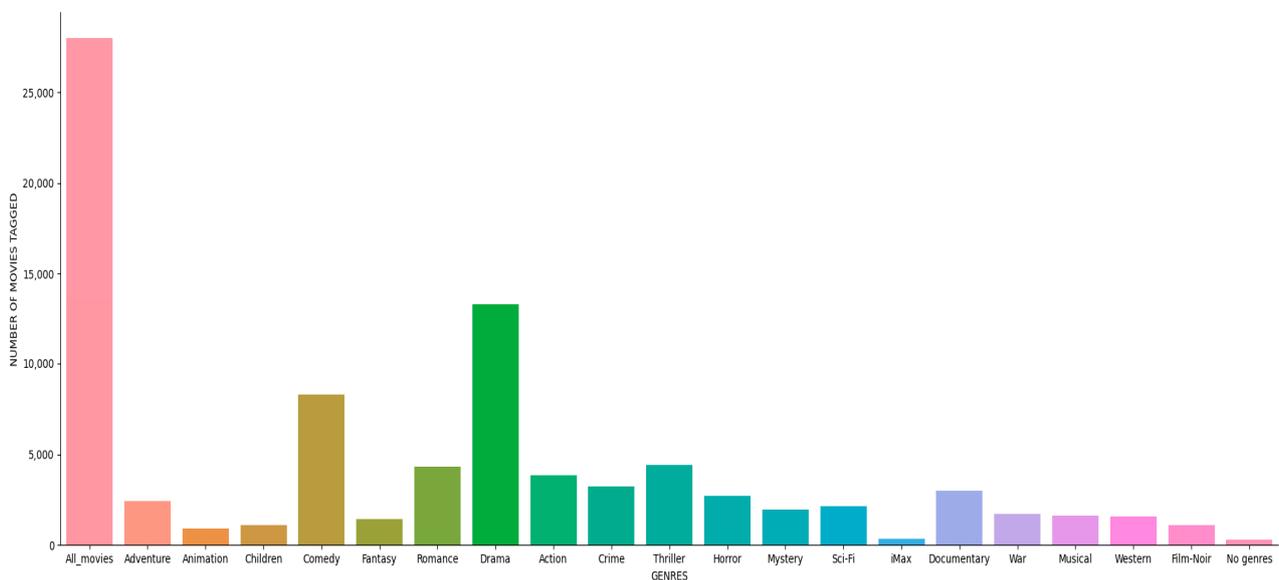
The design of recommendation systems is based on the kind of information acquired, and therefore, the variety of information affects how they are developed and are organized. Finding accurate and insightful data is thus the main goal of developing a recommender system. For recommendation systems, numerous datasets are obtainable, each of which includes different kinds of data. In this study, we developed our suggested recommendation algorithm and assessed its efficacy using the MovieLens 20M dataset. Customers were

selected randomly for the dataset. At least 20 films were rated by each of the selected customers. The MovieLens online movie recommendation service's 138,493 users assigned 20,000,263 ratings (Figure 11) and 465,564 tags to 27,278 films in the MovieLens 20M dataset. Each client gave a movie a rating between 1 and 5, with 5 being the greatest and 1 being the worst. For inclusion, the customers were chosen at random. At least 20 movies were rated by each chosen customer. MAE (mean absolute error) and precision/recall were chosen as measurement metrics to evaluate the proposed approach's prediction accuracy, top N performance, and user cold-start problem.



**Figure 11.** Ratings distribution in MovieLens 20M.

In the dataset, all film genres had a similar tendency (right-skewed log-normal distribution), with the possible exception of horror films, which had a minor leftward skew (poorer ratings). Figure 12 shows the distribution of the tagged movies by genre.



**Figure 12.** Number of tagged movies in each genre in MovieLens 20M.

#### 4.2. Model Parameters

In this paper, the proposed approach tries to learn the customer–product interaction with the main optimal parameters in the given Table 1. The model was learned and showed the best results with the following parameters. The number of users and movies in the 1st hierarchical level ranged from {50, 100, 150, 200, 250} and {100, 200, 300, 400, 500}, accordingly. The value of the parameter  $r$  (number of movie genres) was “20”. The values of levels  $x$  and  $y$  in the hierarchy were similar, which effected to the model performance with optimal value “2”. To balance the deep-factorization procedure, the tag-based auxiliary

regularization parameter revealed its strength on the performance, reaching a lowest error between 0.9 and 2.3. Thus, the optimal degree of auxiliary regularization parameter was taken with “1.7” value.

**Table 1.** Model parameters.

Parameter	Description	Value
r	Number of movie genres	20
$n_1$	Number of users in the 1st hierarchical level	{50, 100, 150, 200, 250}
$m_1$	Number of movies in the 1st hierarchical level	{100, 200, 300, 400, 500}
x	Optimal user’s hierarchical level	2
y	Optimal movie’s hierarchical level	2
$\beta$	Tag-based auxiliary regularization parameter	1.7

#### 4.3. Experimental Conclusions

The proposed approach was evaluated via the best recommendation system indicators, including the rating-prediction error, extent of mitigating the user cold-start problem, and top-N performance results. It is important to note that the all experiments conducted to confirm and compare its advantage over other chosen baseline recommendation models, where the outcomes are illustrated in Tables 2–4.

**Table 2.** The model prediction error with comparisons.

Training Dataset Size (%)	MAE					
	MF	WNMF	F-ALS	BOW-TRSDL	Proposed	
					With Cluster	Deep WNMF
60	0.8859	0.8797	0.8562	0.8363	0.8011	0.8281
80	0.8438	0.8662	0.8315	0.8177	0.7965	0.8101

**Table 3.** Customer cold-start performance.

Cold Start	MAE					
	MF	WNMF	F-ALS	BOW-TRSDL	Proposed	
					With Cluster	Deep WNMF
New 50 users	0.8946	0.8902	0.8954	0.8884	0.8781	0.8908
New 100 users	0.9383	0.9465	0.9472	0.9131	0.9046	0.9165

**Table 4.** Top-10 performance comparisons.

Top-10	Methods					
	MF	WNMF	F-ALS	BOW-TRSDL	Proposed	
					With Cluster	Deep WNMF
Prec@10	0.3247	0.2694	0.2984	0.3392	0.3405	0.3313
Recall@10	0.2053	0.1375	0.1851	0.2113	0.2371	0.2229

#### 4.3.1. The Model Prediction Error

1. MF—matrix factorization: modeled by Koren et al. [5]; to reduce the difference between the anticipated and actual ratings, this approach factorizes a rating matrix and then acquires the resulting product and customer latent feature vectors.
2. WNMF—weighted nonnegative matrix factorization: the method is the basis of the suggested approach as a founding method to generate implicit dormant features. The WNMF tries to factorize a weighted rating matrix into two nonnegative matrices to reduce the difference between the anticipated and actual ratings.
3. F-ALS—fast alternating least squares matrix factorization: in order to decrease runtime and increase model efficiency than simple MF, the approach aims to create a model with more latent components to learn rating matrix.
4. BOW-TRSDL: the method attempts to develop product and customer's profiles with benefits of bag-of-words (BOW) as the first step. Afterwards, DNN (deep neural networks) is utilized to retrieve the customers and products' latent features, and then, these features are used to predict ratings.

#### 4.3.2. User Cold-Start Decision

The cold-start dilemma explains situations where a recommendation system is unable to provide pertinent suggestions since there are no ratings yet. This problem could make collaborative-filtering RS less effective. Cold-start issues are especially common with collaborative-filtering algorithms in particular. For this reason, the suggested solution can reduce the problem of user cold-start if user tag annotation and hierarchically organized implicit features are both available for usage. Furthermore, early-clustering customers provide a user group who have more similar interests on specific products. In addition to containing a description of the products, tags also provide user sentiment. Therefore, new customers without any preferences for any products can receive recommendations from it. The characteristic "profession" is uniquely hierarchically organized, and there is a connection between the layers. Customers in the same hierarchical tier are expected to have comparable traits, and consequently, it is likely that they will rate products similarly. As a result, the connections between consumers at various levels of the hierarchical structure produce rating forecasts and perhaps serve as additional implicit characteristics. A new user of the system is specifically positioned in a database based on their occupation. New customer locations in the industry create new connections between returning customers and a new client, and these connections will provide more data to forecast ratings for new clients. This was put to the test by treating 50 and 100 randomly chosen customers of the 80% training dataset as labeled new customers by ignoring their ratings. The proposed approach, which makes early-grouping customers and generates customers and products' hierarchically organized implicit dormant features by integrating customer's tag annotation to the prediction procedure in the state of the customer cold-start problem, exceeded competitive models and confirmed its comparative outcomes (Table 3) with the carried-out tests. It is noticeable that the proposed approach surpassed its counterparts and succeeded in alleviating the cold-start issue in both 100 and 200 new customer cold-start situations.

#### 4.3.3. Top-N Performance

The proposed approach also succeeded at the top N recommendations test in addition to delivering outstanding MAE scores on rating prediction. The films that perfectly matched the customer's interests were found through experiments on the top N recommendations test. These films were determined by having hierarchically organized hidden implicit dormant features and customers' tag annotation to the films. The higher-rated films were listed as top 10 recommendations for each customer, utilizing 80% training dataset to assess the proposed approach's top N performance. The most popular 20M dataset was utilized to compare the proposed approach to other benchmark cutting-edge approaches, as shown in Table 4. In the comparison scenario, the proposed approach's top 10 performance was satisfactory and succeeded due to early clustering the sparse large dataset and applying the

dormant features into recommendations. Specifically, the top 10 performance was superior among the methods with a precision of 0.3405 and a recall of 0.2371 by requiring expensive operations for the initialization and fine-tuning processes. On the other hand, WNMF had the lowest performance and obtained 0.2694 precision and 0.1375 recall values, whereas these values for F-ALS were 0.2984 and 0.1851, accordingly. Moreover, MF achieved 0.3247 precision and 0.2053 recall top 10 performance results. However, the BOW-TRSDL model had close results in both precision and recall to the proposed model, with 0.3402 precision and 0.2113 recall results. Based on the experiment, it could be confirmed that the proposed method still worked successfully, and its superiority was clearly verified.

## 5. Conclusions and Future Scope

Although the evolution of customized recommender systems has progressed to a significant degree, there are still outstanding difficulties in the recommendation system field that need to be resolved, including data sparsity, cold starts, and enhancing recommender systems performance. This research suggests a unique rating-prediction approach that uses dormant implicit information and is based on deep factorization and early clustering. This model addresses the prediction performance, customer cold-start, and data sparsity difficulties in developing an efficient recommender system. First, a thorough examination of the underlying concepts, advantages, categories, and current issues of recommendation systems was conducted. Subsequently, a variety of relevant studies on clustering techniques, deep-factorization models, tag data, and hierarchically organized features were reviewed and assessed to lay the groundwork for the research project under consideration.

Employing implicit customer and product information plays a key role in improving the RS standard for online companies. In particular, gathering hierarchically organized hidden characteristics of persons and products enables one to overcome RS constraints and has been demonstrated to be essential in several research studies. Further, intentionally obtained tag data provide value to RS's hierarchically organized hidden features and aid in improving the prediction model's learning process by capturing the essence of customer-product interactions. The proposed approach in this study attempts to acquire hierarchically structured hidden implicit dormant features of customers and products and combine them via customers' tag annotations. This regularizes the matrix factorization process of a fundamental weighted nonnegative matrix factorization (WNMF) model. The concept behind the proposed method is to regularize the process by utilizing customers' tag annotations as a supplementary parameter to extract hidden hierarchical aspects of customer preferences and product attributes that indicate a deep link between them. The experimental results demonstrated a significant improvement in the rating-prediction process and product cold-start problem mitigation over previous MF systems when hierarchical features and tag information were combined.

Only in the case of products forming tag information with the hierarchical information of customers and products was the entire process of our suggested model for completion of rating predictions. Owing to their non-negativity, the customer preference and product characteristic matrices underwent deep factorization to produce hidden-level hierarchically organized features. To complete our prediction model, a straightforward matrix factorization process for the WNMF model was regularized using tag information. During the experimental testimony phase, we found that the efficiency of the proposed model initially improved and subsequently degraded when the values of the dimensions varied.

The advantage gained through this integration is that the designed model overcomes the data sparsity and user cold-start problems by early clustering customers based on the customer rating matrix. Furthermore, hierarchically structured hidden features are obtained and integrated with tag information for the prediction process in each customer cluster. Additionally, the experiments on the proposed model were conducted on the established MovieLens 20M dataset and proved that the proposed model is effective in improving the accuracy of top N recommendations with resistance to rating sparsity and cold-start problems when compared to the state-of-the-art CF-based recommendation mod-

els. Especially, the MAE of the rating prediction was 0.8011 with 60% training dataset size, while the error rate was equal to 0.7965 with 80% training dataset size. Moreover, MAE rates were 0.8781 and 0.9046 in new 50 and 100 customer cold-start scenarios, respectively. In terms of top 10 recommendations, precision and recall were 0.3405 and 0.2371. This indicates that our proposed model is effective in improving the accuracy of rating predictions and top N performance and alleviating the customer cold-start problem.

In reality, the adaptability and variety of the underlying idea, which make contributions to a number of topics, are seen as its greatest strengths. The world of recommender systems is not necessarily the only one to which these contributions apply. It is now obvious how crucial and successful recommender systems are for modern Internet enterprises, and the suggested algorithm has room for development.

The following are some directions for further research:

- To design a recommender system that is understandable and comprehensible using implicit hidden characteristics;
- To use metaheuristic techniques to enhance performance metrics [57];
- To handle the “grey sheep” issue, which occurs when a customer cannot be matched with any other customer group, and the system is unable to produce helpful recommendations [58];
- To provide dynamic predictions with the least amount of complexity;
- To develop an emotion-based movie recommendation model [59,60];
- To integrate other advanced clustering methods such as twin contrastive learning for online clustering, structured autoencoders for subspace clustering, and XAI beyond classification: interpretable neural clustering [30–33] for further models’ improvement and to analyze clustering techniques contribution.

**Author Contributions:** This manuscript was designed and written by A.K., T.K.W., R.O., S.M. and A.B.A. supervised the study and contributed to the analysis and discussion of the algorithm and experimental results. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Gachon University research fund of 2019(GCU-2019-0796).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to express their sincere gratitude and appreciation to the supervisor, Taeg Keun Whangbo (Gachon University), for his support, comments, remarks, and engagement over the period in which this manuscript was written. Moreover, the authors would like to thank the editor and anonymous referees for the constructive comments in improving the contents and presentation of this paper.

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

## References

1. Chen, Z.; Wang, Y.; Zhang, S.; Zhong, H.; Chen, L. Differentially private user-based collaborative filtering recommendation based on k-means clustering. *Expert Syst. Appl.* **2021**, *168*, 114366. [[CrossRef](#)]
2. Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. *Recommender Systems Handbook*; Springer: Berlin, Germany, 2011; ISBN 978-0-387-85819-7.
3. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutierrez, A. Recommender systems survey. *Knowl. Based Syst.* **2013**, *46*, 109–132. [[CrossRef](#)]
4. Goldberg, D.; Nichols, D.; Oki, B.M.; Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* **1992**, *35*, 61–70. [[CrossRef](#)]
5. Koren, Y.; Bell, R.; Volinskiy, C. Matrix factorization techniques for recommender systems. *IEEE Comput.* **2009**, *42*, 30–37. [[CrossRef](#)]
6. Kumar, A.; Sodera, N. Open problems in recommender systems diversity. In Proceedings of the International Conference on Computing, Communication and Automation (ICCCA2017), Greater Noida, India, 5–6 May 2017.
7. Guo, X.; Yin, S.-C.; Zhang, Y.-W.; Li, W.; He, Q. Cold start recommendation based on attribute-fused singular value decomposition. *IEEE Access* **2019**, *7*, 11349–11359. [[CrossRef](#)]

8. Ortega, F.; Hurtado, R.; Bobadilla, J.; Bojorque, R. Recommendation to groups of users the singularities concept. *IEEE Access* **2018**, *6*, 39745–39761. [[CrossRef](#)]
9. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep Learning based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* **2018**, *52*, 1–38. [[CrossRef](#)]
10. Darban, Z.Z.; Valipour, M.H. Graph-based Hybrid Recommendation System with Application to Movie Recommendation. *Expert Syst. Appl.* **2022**, *200*, 116850. [[CrossRef](#)]
11. Nouh, R.; Singh, M.; Singh, D. SafeDrive: Hybrid Recommendation System Architecture for Early Safety Predication Using Internet of Vehicles. *Sensors* **2021**, *21*, 3893. [[CrossRef](#)]
12. Alvarado-Uribe, J.; Gómez-Oliva, A.; Barrera-Animas, A.Y.; Molina, G.; Gonzalez-Mendoza, M.; Parra-Meroño, M.C.; Jara, A.J. HyRA: A Hybrid Recommendation Algorithm Focused on Smart POI. Ceutí as a Study Scenario. *Sensors* **2018**, *18*, 890. [[CrossRef](#)]
13. Xiaojun, L. An improved clustering-based collaborative filtering recommendation algorithm. *Cluster Comput* **2017**, *20*, 1281–1288. [[CrossRef](#)]
14. Bhaskaran, S.; Marappan, R.; Santhi, B. Design and Analysis of a Cluster-Based Intelligent Hybrid Recommendation System for E-Learning Applications. *Mathematics* **2021**, *9*, 197. [[CrossRef](#)]
15. Tran, C.; Kim, J.-Y.; Shin, W.-Y.; Kim, S.-W. Clustering-Based Collaborative Filtering Using an Incentivized/Penalized User Model. *IEEE Access*. **2019**, *7*, 62115–62125. [[CrossRef](#)]
16. Geng, X.; Zhang, H.; Bian, J.; Chua, T.-S. Learning Image and User Features for Recommendation in Social Networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4274–4282. [[CrossRef](#)]
17. Wang, J.-H.; Wu, Y.-T.; Wang, L. Predicting Implicit User Preferences with Multimodal Feature Fusion for Similar User Recommendation in Social Media. *Appl. Sci.* **2021**, *11*, 1064. [[CrossRef](#)]
18. D’Addio, R.M.; Domingues, M.A.; Manzato, M.G. Exploiting feature extraction techniques on users’ reviews for movies recommendation. *J. Braz. Comput. Soc.* **2017**, *23*, 1–16. [[CrossRef](#)]
19. Jayalakshmi, S.; Ganesh, N.; Čep, R.; Senthil Murugan, J. Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions. *Sensors* **2022**, *22*, 4904. [[CrossRef](#)]
20. Wang, T.; Li, J.; Zhou, J.; Li, M.; Guo, Y. Music Recommendation Based on “User-Points-Music” Cascade Model and Time Attenuation Analysis. *Electronics* **2022**, *11*, 3093. [[CrossRef](#)]
21. Sharma, B.; Hashmi, A.; Gupta, C.; Khalaf, O.I.; Abdulsahib, G.M.; Itani, M.M. Hybrid Sparrow Clustered (HSC) Algorithm for Top-N Recommendation System. *Symmetry* **2022**, *14*, 793. [[CrossRef](#)]
22. Widiyaningtyas, T.; Hidayah, I.; Adji, T.B. Recommendation Algorithm Using Clustering-Based UPCSim (CB-UPCSim). *Computers* **2021**, *10*, 123. [[CrossRef](#)]
23. Pérez-Núñez, P.; Díez, J.; Luaces, O.; Bahamonde, A. User encoding for clustering in very sparse recommender systems tasks. *Multimed. Tools Appl* **2022**, *81*, 2467–2488. [[CrossRef](#)]
24. Yang, L.; Liu, B.; Lin, L.; Xia, F.; Chen, K.; Yang, Q. Exploring Clustering of Bandits for Online Recommendation System. In Proceedings of the Fourteenth ACM Conference on Recommender Systems (RecSys ’20); Virtual Event, Brazil, 22–26 September 2020; ACM: New York, NY, USA, 2020; 10p. [[CrossRef](#)]
25. Li, J.; Li, C.; Liu, J.; Zhang, J.; Zhuo, L.; Wang, M. Personalized Mobile Video Recommendation Based on User Preference Modeling by Deep Features and Social Tags. *Appl. Sci.* **2019**, *9*, 3858. [[CrossRef](#)]
26. Zhu, R.; Yang, D.; Li, Y. Learning Improved Semantic Representations with Tree-Structured LSTM for Hashtag Recommendation: An Experimental Study. *Information* **2019**, *10*, 127. [[CrossRef](#)]
27. Tso-Sutter, K.H.L.; Marinho, L.B.; Schmidt-Thieme, L. Tag-aware recommender systems by fusion collaborative filtering algorithms. In Proceedings of the SAC ’08: 2008 ACM Symposium on Applied Computing, Fortaleza, Brazil, 16–20 March 2008.
28. Kutlimuratov, A.; Abdusalomov, A.; Whangbo, T.K. Evolving Hierarchical and Tag Information via the Deeply Enhanced Weighted Non-Negative Matrix Factorization of Rating Predictions. *Symmetry* **2020**, *12*, 1930. [[CrossRef](#)]
29. Maleszka, M.; Mianowska, B.; Nguyen, N.T. A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles. *Knowl. Based Syst.* **2013**, *47*, 2013. [[CrossRef](#)]
30. Li, Y.; Hu, P.; Liu, Z.; Peng, D.; Zhou, J.T.; Peng, X. Contrastive Clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020. [[CrossRef](#)]
31. Li, Y.; Yang, M.; Peng, D.; Li, T.; Huang, J.; Peng, X. Twin Contrastive Learning for Online Clustering. *Int. J. Comput. Vis.* **2022**, *130*, 2205–2221. [[CrossRef](#)]
32. Peng, X.; Feng, J.; Xiao, S.; Yau, W.-Y.; Zhou, J.T.; Yang, S. Structured AutoEncoders for Subspace Clustering. *IEEE Trans. Image Process.* **2018**, *27*, 5076–5086. [[CrossRef](#)]
33. Peng, X.; Li, Y.; Tsang, I.W.; Zhu, H.; Lv, J.; Zhou, J.T. XAI Beyond Classification: Interpretable Neural Clustering. *J. Mach. Learn. Res.* **2018**, *23*, 1–28. [[CrossRef](#)]
34. Vellaichamy, V.; Kalimuthu, V. Hybrid Collaborative Movie Recommender System Using Clustering and Bat Optimization. *Int. J. Intell. Eng. Syst.* **2017**, *10*, 38–47. [[CrossRef](#)]
35. Zhang, J.; Lin, Y.; Lin, M.; Liu, J. An effective collaborative filtering algorithm based on user preference clustering. *Appl. Intell.* **2016**, *45*, 230–240. [[CrossRef](#)]

36. Aytekin, T.; Karakaya, M.Ö. Clustering-based diversity improvement in top-N recommendation. *J. Intell. Inf. Syst.* **2014**, *42*, 1–18. [[CrossRef](#)]
37. Koosha, H.; Ghorbani, Z.; Nikfetrat, R. A Clustering-Classification Recommender System based on Firefly Algorithm. *J. AI Data Min.* **2022**, *10*, 103–116. [[CrossRef](#)]
38. Li, Z.; Xu, G.; Zha, J. A collaborative filtering recommendation algorithm based on user spectral clustering. *Comput. Technol.* **2014**, *24*, 59–67.
39. Jiang, M.; Zhang, Z.; Jiang, J.; Wang, Q.; Pei, Z. A collaborative filtering recommendation algorithm based on information theory and bi-clustering. *Neural Comput. Appl.* **2019**, *31*, 8279–8287. [[CrossRef](#)]
40. Yang, J.; Sun, Z.; Bozzon, A.; Zhang, J. Learning hierarchical feature influence for recommendation by recursive regularization. In Proceedings of the Recsys: 10th ACM Conference on Recommender System, Boston, MA, USA, 15–19 September 2016; pp. 51–58.
41. Kang, S.; Chung, K. Preference-Tree-Based Real-Time Recommendation System. *Entropy* **2022**, *24*, 503. [[CrossRef](#)]
42. Lu, K.; Zhang, G.; Li, R.; Zhang, S.; Wang, B. Exploiting and exploring hierarchical structure in music recommendation. In *AIRS 2012: Information Retrieval Technology*; Springer: Berlin, Germany, 2012; pp. 211–225.
43. Shi, W.; Wang, L.; Qin, J. Extracting user influence from ratings and trust for rating prediction in recommendations. *Sci. Rep.* **2020**, *10*, 13592. [[CrossRef](#)]
44. Nikolakopoulos, N.; Kouneli, M.A.; Garofalakis, J.D. Hierarchical Itemspace Rank: Exploiting hierarchy to alleviate sparsity in ranking-based recommendation. *J. Neurocomput.* **2015**, *163*, 126–136. [[CrossRef](#)]
45. Ilyosov, A.; Kutlimuratov, A.; Whangbo, T.-K. Deep-Sequence-Aware Candidate Generation for e-Learning System. *Processes* **2021**, *9*, 1454. [[CrossRef](#)]
46. Borrego, G.; González-López, S.; Palacio, R.R. Tags' Recommender to Classify Architectural Knowledge Applying Language Models. *Mathematics* **2022**, *10*, 446. [[CrossRef](#)]
47. Bagheri, E.; Ensan, F. Semantic tagging and linking of software engineering social content. *Autom. Softw. Eng.* **2016**, *23*, 147–190. [[CrossRef](#)]
48. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In Proceedings of the Fifth International Conference on Computer and Information Technology, Munich, Germany, 30–31 August 2002.
49. Shepitsen, A.; Gemmell, J.; Mobasher, M.; Burke, R. Personalized recommendation in social tagging systems using hierarchical clustering. In Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys, Lausanne, Switzerland, 23–25 October 2008.
50. Chung, F. *Spectral Graph Theory*; American Mathematical Society: Providence, RI, USA, 1997.
51. Abdusalomov, A.B.; Mukhiddinov, M.; Kutlimuratov, A.; Whangbo, T.K. Improved Real-Time Fire Warning System Based on Advanced Technologies for Visually Impaired People. *Sensors* **2022**, *22*, 7305. [[CrossRef](#)]
52. Trigeorgis, G.; Bousmalis, K.; Zaferiou, S.; Schuller, B. A deep semi-nmf model for learning hidden representations. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, 21–26 June 2014; pp. 1692–1700.
53. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.* **2001**, *13*, 556–562.
54. Ding, C.; Li, T.; Peng, W.; Park, H. Orthogonal nonnegative matrix t-factorizations for clustering. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; pp. 126–135.
55. Abdusalomov, A.; Baratov, N.; Kutlimuratov, A.; Whangbo, T.K. An Improvement of the Fire Detection and Classification Method Using YOLOv3 for Surveillance Systems. *Sensors* **2021**, *21*, 6519. [[CrossRef](#)] [[PubMed](#)]
56. Gu, Q.; Zhou, J.; Ding, C.H.Q. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In Proceedings of the 2010 SIAM International Conference on Data Mining, Columbus, OH, USA, 29 April–1 May 2010; pp. 199–210.
57. Bałchanowski, M.; Boryczka, U. Aggregation of Rankings Using Metaheuristics in Recommendation Systems. *Electronics* **2022**, *11*, 369. [[CrossRef](#)]
58. Alabdulrahman, R.; Viktor, H. Catering for unique tastes: Targeting grey-sheep users recommender systems through one-class machine learning. *Expert Syst. Appl.* **2021**, *166*, 114061. [[CrossRef](#)]
59. Bhaumik, M.; Attah, P.U.; Javed, F. Emotion Integrated Music Recommendation System Using Generative Adversarial Networks. *SMU Data Sci. Rev.* **2021**, *5*, 4.
60. Wakil, K.; Bakhtyar, R.; Ali, K.; Alaadin, K. Improving Web Movie Recommender System Based on Emotions. *Int. J. Adv. Comput. Sci. Appl.* **2015**, *6*, 218–226. [[CrossRef](#)]