Intelligent Systems with Applications 25 (2025) 200477

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

A hybrid recommendation system using association rule mining, i-ALS algorithm, and SVD++ approach: A case study of a B2B company

Thamer Saraei, Maha Benali *, Jean-Marc Frayret

Department of Mathematics and Industrial Engineering, Polytechnique Montréal, 2500 Chem. de Polytechnique, Montréal, QC H3T 1J4, Canada

ARTICLE INFO

Keywords:
Recommendation System
Collaborative Filtering
Association Rule Mining
Implicit Alternative Least Squares
Singular Value Decomposition
Business-To-Business

ABSTRACT

In the field of recommendation systems, collaborative filtering is a widely used technique. It provides recommendations to active users based on the ratings provided by similar users. However, this method may reduce the accuracy of user preference predictions and lead to lower-quality recommendations in cases of high data sparsity. This issue is often observed in the Business-to-Business (B2B) context, where user-generated reviews are often sparse. To overcome this challenge, we present a novel hybrid approach that explores product taxonomies and association rule mining combined with an advanced method for initialization. Our approach first involves generating a new explicit taxonomy based solely on textual product descriptions and extending the user–product matrix using association rule mining results. Second, complementary items are added to the user–item matrix based on users' purchasing behaviors, as emphasized by the extracted association rules. Finally, we use the implicit Alternating Least Squares (i-ALS) algorithm and initialize the latent factor matrices with values obtained through the singular value decomposition approach (BLS-SVD++). This hybrid approach is tested and compared with conventional approaches, considering a real-world case study of a distributor located in Quebec. The results obtained from feedback implicitly inferred from sales data demonstrated improved RS performance compared to conventional approaches.

1. Introduction

E-commerce sites offer customers a wide range of products, making it challenging for users to select the most suitable ones. Recommendation Systems (RSs) assist users by filtering items and displaying only those likely to be relevant to them (Najafabadi, Mahrin, Chuprat, & Sarkan, 2017). With a Top-N recommendation approach, for example, an RS provides a list of N items that are most likely to attract the user's attention (Nia, Lu, Zhang, & Ribeiro, 2019). RSs can employ Content-Based Filtering (CBF) or Collaborative Filtering (CF), depending on the type of data used to generate recommendations (Lü, Medo, Yeung, Zhang, Zhang, & Zhou, 2012).

CBF relies on features such as artist name, song genre, or release year (Balabanović & Shoham, 1997). CF, one of the more widely used approaches (Iijima & Ho, 2007), considers a user's purchase and review history. A user's history, which includes ratings of purchased items, video views, and other relevant interactions, plays a critical role. CF combines this information to generate recommendations tailored to the user's preferences and interests. Feedback in CF can be explicit, where users provide ratings (Kim & Mathes, 2001), or implicit, inferred from browsing or purchase history (Liu, Xiang, Zhao, & Yang, 2010). By incorporating both purchase and review history, CF enables a comprehensive understanding of user preferences, increasing recommendation effectiveness (Zuva, Ojo, Ngwira, Zuva, et al., 2012).

However, CF faces challenges such as the sparsity problem, which arises from the limited number of items that users interact with Liu et al. (2010), and the synonymy problem, which occurs when similar items have different names or descriptions (Liphoto, Du, & Ngwira, 2016). In the business-to-business (B2B) context, where transactions occur between two companies (e.g., a wholesaler and a retailer), an item can have multiple names and descriptions. Data sparsity is a critical challenge for collaborative filtering in the B2B context because of infrequent and specialized transactions. For instance, in our case study involving a B2B industrial procurement platform, the catalog consists of over 15,000 unique items. However, individual customers interact with fewer than 3% of these items on average.

To address these challenges, this paper proposes a novel approach to addressing the sparsity and synonymy issues encountered by Collaborative Filtering Recommendation Systems (CF-RSs) in B2B contexts. Our approach integrates techniques from the literature, combining product taxonomies, Association Rule Mining (ARM), and an advanced initialization method. Unlike traditional hybrid models, which typically focus

* Corresponding author.
E-mail addresses: thamer.saraei@polymtl.ca (T. Saraei), maha.benali@polymtl.ca (M. Benali), jean-marc.frayret@polymtl.ca (J.-M. Frayret).

https://doi.org/10.1016/j.iswa.2025.200477
Received 6 May 2024; Received in revised form 24 November 2024; Accepted 4 January 2025
Available online 9 January 2025

on combining CF with ARM, our method uniquely tailors integration to the B2B environment. By leveraging a sales database with over 15,000 items and diverse customer profiles, we develop a CF-based approach that enhances recommendation accuracy and better addresses the complexities of sparsity and synonymy in this context. The first step involves creating a denser user–item matrix by aggregating user preferences through an automated product taxonomy, further enhanced by ARM techniques. In the second step, complementary items are added to the user–item matrix based on ARM results. Finally, the Implicit Alternating Least Squares (i-ALS) method is employed and initialized with values obtained using the Singular Value Decomposition approach (BLS-SVD++). To evaluate the performance of the proposed approach, the precision of the top 3 recommended items is measured and compared to various baselines.

The rest of this paper is organized as follows. Related works are presented in Section 2. The proposed approach is described in detail in Section 3, while Section 4 presents the dataset and the conducted experiments. Results and comparisons with conventional approaches are discussed in Section 5. Finally, Section 6 concludes the article and suggests directions for future work.

2. Background and related works

2.1. Matrix factorization

Matrix factorization is a computational technique widely used in RSs (Wen, Ding, Liu, & Wang, 2014). This technique aims to uncover underlying structures in the data by decomposing the original user–item interaction matrix into latent factors. More specifically, it involves computing latent user and item factor matrices from the interaction matrix to uncover hidden patterns or features that are not explicitly observed. These latent factors represent abstract features or dimensions that contribute to relationships between users and items, enabling a more compact and meaningful representation of the original data (Chen & Peng, 2018).

The iterative Alternating Least Squares (i-ALS) algorithm is commonly used in matrix factorization due to its computational efficiency. Proposed by Hu, Koren, and Volinsky in 2008, i-ALS alternately updates the user and item matrices, keeping one fixed while updating the other, to converge on a solution that accurately captures the underlying patterns in the data (Koren, Bell, & Volinsky, 2009).

Latent factors derived through matrix factorization represent hidden characteristics in user–item interactions. These factors enable more accurate predictions of user preferences or item recommendations by capturing nuanced relationships underlying user–item interactions. In CF-RSs, this methodology has proven effective across various domains. Moreover, it addresses data sparsity and scalability challenges often encountered in recommendation tasks (Koren, 2008).

2.2. Product taxonomy

The use of taxonomies in RSs has been extensively studied in the literature. For instance, Maidel, Shoval, Shapira, and Taieb-Maimon (2008) proposed a taxonomy-based methodology for relevance classification in the e-paper domain. Ziegler, Lausen, and Schmidt-Thieme (2004) utilized a hierarchical taxonomy of items to enhance the accuracy of product recommendations. Similarly, Oppermann and Specht (1999) recommended museum visits by employing ICONCLASS, a comprehensive classification system for Western art themes. In the music domain, Mnih (2012) developed latent factor models that leverage a taxonomy to improve music recommendations.

Bayesian hierarchical models have also been employed to model user preferences for product attributes. For example, Ahmed et al. (2013) introduced a personalized Bayesian hierarchical model to capture user preferences for brand and price, while Lawrence, Almasi, Kotlyar, Viveros, and Duri (2001) employed a product taxonomy to model relationships between items in a supermarket RS. Furthermore, Kanagal et al. (2012) introduced latent factors for each internal node of the taxonomy and applied taxonomy-based priors to user and item latent factors.

The RSs proposed in these studies rely on predefined taxonomies, which limit their effectiveness. To address this limitation, Zhang, Ahmed, Josifovski, and Smola (2014) proposed a parametric model that automatically discovers item taxonomy structures and uses them to generate user priors. This model outperformed conventional approaches on two real-world datasets.

2.3. Association Rule Mining (ARM)

The ARM technique is widely used to enhance the performance of CF-RSs. By providing additional information about item associations, user preferences, and item co-occurrences, association rules effectively complement CF-based RSs. This technique employs association rules, which represent relationships between sets of items frequently occurring in users' transactions, to provide a descriptive rather than predictive understanding of the data. Several studies have explored the use of association rules in RSs (Leung, Chan, & Chung, 2008).

For instance, Leung, Chan, and Chung (2006) introduced a model based on fuzzy association rules, exploiting similarities between items within existing taxonomies to enhance recommendations. Kardan and Ebrahimi (2013) employed the ARM technique in a hybrid recommendation system to determine user similarity using implicit data collected from a newsgroup. Similarly, Liu et al. (2010) proposed extracting customer purchase behavior through sequential association rules, capturing the evolving nature of customer purchase patterns over time. Feng, Tian, Wang, and Li (2015) developed a novel recommendation technique by introducing a temporally overlapping community detection method based on time-weighted association rules.

None of these studies used the generated association rules to directly enhance the user–item matrix. For example, Schoinas and Tjortjis (2019) proposed a RS employing ARM, using a single confidence level as a threshold to select association rules for populating the user–item matrix. In contrast, this study proposes enriching the user–item matrix by comparing association rules derived from two different confidence levels.

2.4. Initialization of user–item factor matrices

Smilde, Geladi, and Bro (2005) showed that good initialization can improve both the speed and accuracy of matrix factorization algorithms. In the case of the i-ALS algorithm, as well as most matrix factorization approaches, the latent user and item factor matrices are randomly initialized. Then, an iterative optimization method is applied to update these matrices by minimizing the error of a cost function. Over several iterations, the matrix factorization methods converge to a local optimum influenced by the initial values.

An alternative approach to initializing the latent user and item factor matrices is presented by Donoho and Stodden (2003). The authors propose using the Singular Value Decomposition (SVD) approach to generate more accurate factor matrices, thereby increasing the overall accuracy of the matrix factorization algorithm. They also show that the choice of initial values affects not only the convergence time but also the quality of the resulting solution (Donoho & Stodden, 2003).

Recently, a new initialization technique based on a more advanced decomposition approach, known as BLS-SVD++, has been proposed by Wang, Sun, and Li (2020). Using the FilmTrust, MovieLens 1M, and 10M datasets, Wang et al. (2020) demonstrated that the BLS-SVD++ approach outperforms the SVD approach in terms of performance and convergence time, as it decomposes the user–item matrix into two matrices that individually capture the factors influencing users and items.

2.5. Paper contribution

Most of the existing studies focus on B2C applications with experimental datasets. This paper is among few studies considering a B2B real-world case study, which is an attempt to bridge the theory–practice gap for RSs. In addition, to the best of our knowledge, this is the first paper to propose a hybrid RS combining three techniques from the literature, which are the ARM, the i-ALS algorithm, and the BLS-SVD++ initialization approach. The proposed methodology can be generalizable to other real-world cases and is particularly relevant for overcoming the sparsity problem in B2B settings.

3. Methodology and implementation

In this section, we present the proposed methodology developed to address two main challenges commonly faced when dealing with transactional data in a B2B company: the synonymy problem and the sparsity problem. First, to solve the synonymy problem, we developed a tool that automatically generates a new item taxonomy to aggregate user preferences (step 1 in Fig. 1). Second, we propose enhancing the user–item matrix using the ARM technique to reduce sparsity in the matrix. Specifically, association rules (ARs) are extracted based on the Apriori algorithm (Agrawal et al., 1996), and missing values in the user–item matrix are filled by including complementary items for users (step 2 in Fig. 1). Finally, to improve accuracy, we propose using the initialization strategy based on the singular value decomposition approach BLS-SVD++ proposed by Wang et al. (2020) (step 3 in Fig. 1).

3.1. Creation of new item-taxonomy

A new item taxonomy is created by aggregating items with similar characteristics, which enables the collaborative filtering (CF) algorithm to identify synonyms. Specifically, the same ITEM FAMILY is assigned to two items with distinct Unique Identifiers (IDs) but with a high degree of similarity in their textual descriptions. For instance, consider two items, I1 and I2.

- I1: ID = 1001 and labeled as "steel bolt size 10 * 12 * 15"
- I2: ID = 1005 and labeled as "black steel bolt size 12 * 12 * 20"
- The ITEM_FAMILY of I1 and I2 is "Steel bolt"

Although the two items have different IDs, they share an identical description as steel bolts, indicating that they should not be treated as separate items. To address this, a tool was developed to facilitate the automatic extraction of the taxonomy. This tool uses the Natural Language Toolkit (NLTK), a leading platform for building Python programs, and consists mainly of two processing modules: "text splitting" and "sentence clustering" (see Fig. 2).

To generate an explicit product taxonomy, we preprocess the textual descriptions using several natural language processing (NLP) techniques. The preprocessing begins with cleaning the text, which includes converting all characters to lowercase, removing punctuation, and excluding non-Latin characters. The text is then tokenized into individual words using the Tokenizer package from the NLTK library,[1] a Python-based[2] toolkit for NLP tasks. After tokenization, each token is checked against NLTK's predefined list of stop words, which includes common words such as prepositions, pronouns, and conjunctions. These stop words are removed to eliminate non-informative content, leaving only meaningful terms.

Once the text has been cleaned and tokenized, we apply a clustering algorithm to group similar item descriptions into clusters, which form the basis of the taxonomy. The algorithm iteratively decides whether a new input sentence should be added to an existing cluster or form a new cluster. This decision is guided by a similarity function based on the Word Overlap (WO) metric defined in Eq. (1).

$$WO(D,C) = \frac{|W(D) \cap W(C)|}{|W(D) \cup W(C)|}, \tag{1}$$

where $W(D)$ and $W(C)$ are the sets of unique words in the description D and the cluster C, respectively. The metric quantifies the degree of overlap between the words in the input text and an existing cluster. If the WO score exceeds a predefined threshold, the description is assigned to the cluster; otherwise, a new cluster is initialized. The threshold is carefully tuned to achieve the desired level of granularity in clustering. By organizing descriptions with high semantic similarity, the clustering algorithm creates a structured taxonomy that reflects the relationships between products.

3.2. Enhancement of the user–item matrix

In order to improve the user–item matrix, we propose to populate the elements associated with a given user by considering both the products they have purchased and the complementary products they have not purchased. This process leverages ARM, which is particularly useful in this context because it identifies meaningful relationships between products based on transaction patterns, even in the absence of explicit user feedback. ARM is well-suited to B2B scenarios where purchase data often exhibit high sparsity, and users may rely on complementary or related items to fulfill their needs.

Association rules are extracted from historical transaction data using the Apriori algorithm (Agrawal et al., 1996), which identifies frequent itemsets that satisfy a minimum support threshold. For each frequent itemset, rules of the form $A \rightarrow B$ are generated, where A is the base product and B is the complementary product. Each rule is associated with a confidence level nc, which indicates the likelihood of product B appearing in transactions involving product A.

To apply these rules, we focus on a specific user, denoted as u, who has purchased product A n_{uiA} times but has not purchased product B. A score, n_{uiB}, is assigned to the complementary product B for user u, calculated as the product of the purchase frequency of A and the confidence level of the rule, as shown in Eq. (2).

$$n_{ui_B} = nc \times n_{ui_A} \tag{2}$$

This scoring approach integrates the relationships between base and complementary products derived from the rules and the user's individual purchase behavior. By filling in the missing elements of the user–item matrix with scores for complementary products, the method enhances recommendation relevance. ARM provides a structured way to infer meaningful associations between products, which is particularly advantageous in scenarios where explicit user feedback is unavailable or incomplete. This integration ensures that the recommendations align with both user preferences and the inherent relationships in the product catalog.

3.3. Initialization of the i-ALS algorithm

The i-ALS algorithm was selected for this study because of its effectiveness in handling scenarios with implicit feedback, where explicit user preferences are not provided. Unlike traditional ALS or other collaborative filtering techniques, i-ALS is tailored to optimize implicit feedback signals, such as purchase counts or click-through data. A key feature of i-ALS is its confidence weighting mechanism, which improves the model's ability to handle sparse datasets while reducing the influence of noise, making it particularly well-suited for complex real-world data. The initialization of the user factor matrix and the item factor matrix is crucial for the i-ALS algorithm, as it greatly affects its performance. While one possibility is to randomly initialize these matrices, it has been shown that choosing appropriate initializations

[1] https://www.nltk.org/.

[2] https://www.python.org/.

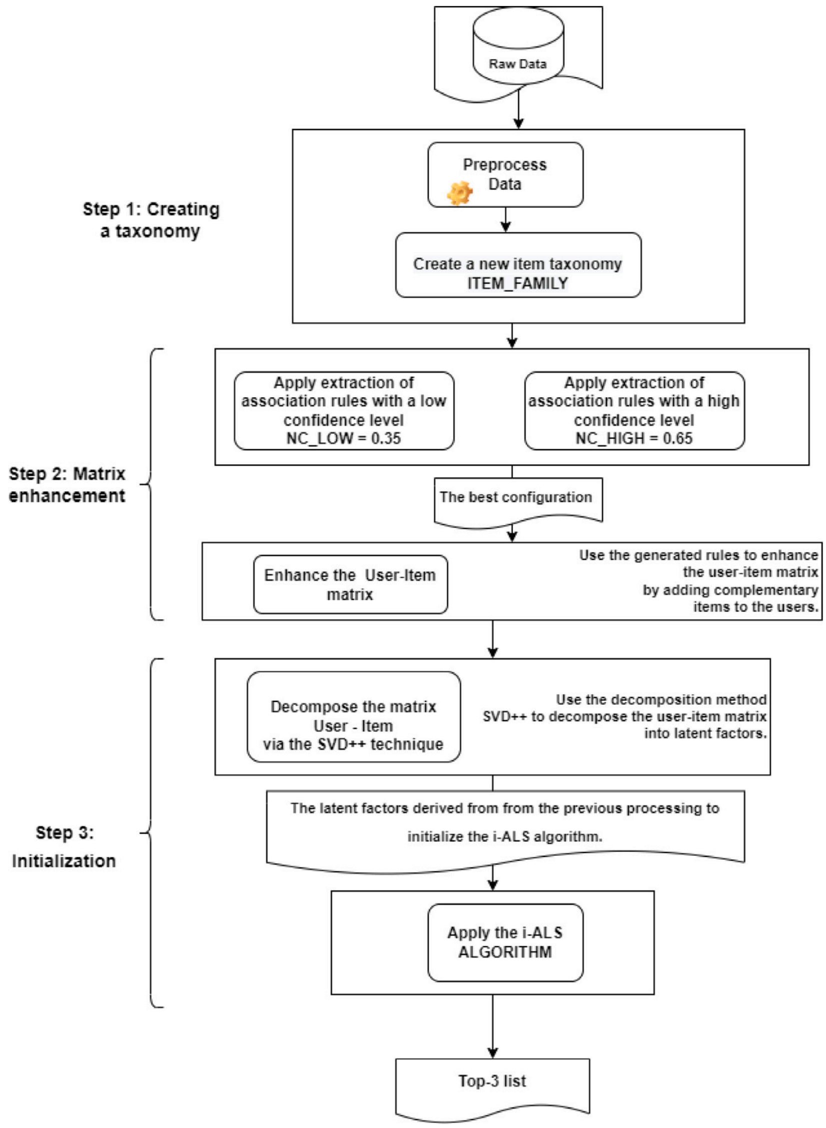

Fig. 1. Methodology steps.

can significantly speed up convergence and generally lead to superior solutions (Smilde et al., 2005).

In this study, we propose to initialize the user factor matrix and the item factor matrix using the results of the BLS-SVD++ approach. The BLS-SVD++ method builds upon traditional Singular Value Decomposition by incorporating both explicit feedback and implicit feedback into the decomposition process. This is achieved by augmenting the matrix factorization framework with additional terms that capture the implicit interactions between users and items, thereby improving the quality of the latent factor representations (Xian, Li, Li, & Li, 2017). This hybrid approach allows the initialization matrices to better reflect the underlying relationships in the data compared to standard SVD, which is typically used for explicit datasets.

The rationale behind using BLS-SVD++ initialization lies in its demonstrated ability to improve recommendation accuracy by leveraging implicit feedback information during the initialization phase. By starting with latent factors that already incorporate meaningful user–item interaction patterns, the i-ALS algorithm can achieve faster convergence and potentially better overall performance. For further details on the BLS-SVD++ method, the interested reader is referred to Xian et al., (2017).

4. Experiments

4.1. Dataset

In this paper, a real-world case study was conducted using a dataset obtained from a distributor offering fasteners and industrial hardware through their online store and six physical locations in Quebec province, Canada. A summary of the dataset's key characteristics is presented in Table 1. The dataset is a comprehensive collection of transactional data from 2007 to 2020. It includes records from more than 5000 customer companies and covers over 15,000 unique items available on the distributor's website. In total, the dataset comprises approximately 10 million transaction records, making it a rich source of information for analyzing customer purchasing behavior.

After analyzing the dataset, several statistical measures were derived to gain a deeper understanding of its characteristics. The dataset exhibits a sparsity level of 0.43%, indicating that the majority of entries in the item–customer matrix are zero, reflecting the large number of items that any single customer typically does not purchase.

In terms of customer behavior, the dataset provides valuable insights. On average, each customer has purchased approximately 50 items per month, resulting in a cumulative average of 1000 transactions

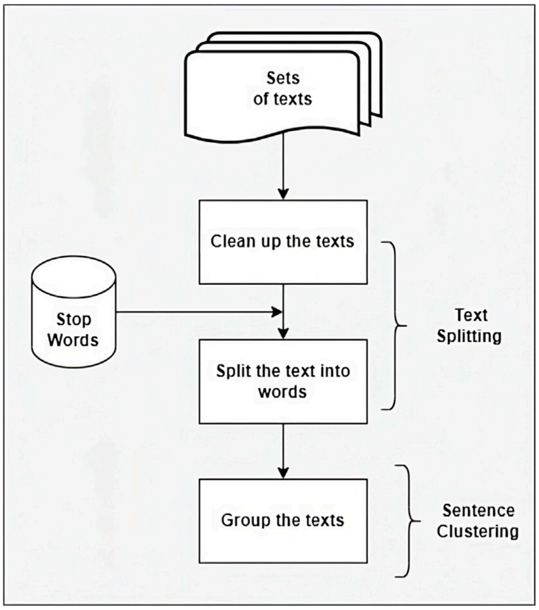

Fig. 2. Automatic extraction of taxonomy.

Table 1
Summary of dataset characteristics.

Characteristic	Value
Time period	2007–2020
Number of customer companies	Over 5000
Number of unique items	Over 15,000
Total number of transactions	Approximately 10 million
Average purchase frequency per user	50 items per month
Average transactions per user	1000
Matrix sparsity	0.43%

per user. Additionally, the dataset captures variations in purchasing patterns across different customer types, which were leveraged to refine the recommendation process.

This dataset serves as a strong validation tool, as its size, diversity, and sparsity reflect real-world challenges faced in deploying RSs in the B2B sector.

4.2. Performance measure

In this paper, we evaluate the performance of the ARM technique and the accuracy of the RS approaches tested. The goal of the ARM technique is to discover interesting relationships within a large dataset by identifying connections between users' historical navigation paths. In general, ARM can uncover user interests through rules of the form $A \rightarrow B$. An association rule ($A \rightarrow B$) means that users interested in A are likely to also be interested in B. In other words, the occurrence of item A (the antecedent of the rule) predicts the occurrence of item B (the consequent of the rule) based on user transactions in the dataset (Panda & Patra, 2009).

The metrics *support* and *confidence* are commonly used in ARM and are attributed to the paper by Agrawal, Imieliński, and Swami (1993), which introduced the Apriori algorithm for mining frequent item sets. Confidence is determined by computing the ratio of the support of the antecedent and consequent of an association rule to the support of the antecedent. The support of an item set is the proportion of transactions in the dataset that contain the item set. The support and confidence of an association rule $A \rightarrow B$ are defined by Eqs. (3) and (4), respectively.

$$\text{Support}(A \rightarrow B)(\%) = \frac{\text{Number of transactions containing both A and B}}{\text{Total number of transactions}} \tag{3}$$

$$\text{Confidence}(A \rightarrow B)(\%) = \frac{\text{Number of transactions containing both A and B}}{\text{Number of transactions containing A}} \tag{4}$$

For a Top-N recommendation approach, it is necessary to measure the Precision (P), which measures the proportion of recommended items that were selected by the user and is usually defined by Eq. (5).

$$P = \frac{|RL \cap SL|}{|RL|} \tag{5}$$

Here, RL represents the set of items recommended to a user, and SL represents the set of items selected by that user. Thus, P provides an overall assessment of the accuracy of all recommendations but does not take into account the order of the recommendations. To measure the Precision up to the kth recommended item, the metric Precision at cut-off k (P@k) can be used, as defined by Eq. (6).

$$P@k = \frac{1}{|U|} \times \sum_{u \in U} \frac{SL(u)@k}{k} \tag{6}$$

Here, $|U|$ represents the set of users, and SL(u)@k denotes the set of items selected by user u up to the kth recommended item. A higher value of P@k indicates a higher accuracy of the recommendation results up to the kth recommended item. In this study, considering the industrial partner's needs, we choose to measure the Precision (P@3). The training set consists of 70% of the user–item matrix, while the test set consists of the remaining 30%.

4.3. Baselines

To assess the performance of the approach proposed in Section 3, a comparison is made with four baselines that operate with implicit feedback. For the first baseline that we called Random, a random list of N items is passively recommended to the user. For the second baseline that we called Most Popular, the user is presented with the N most popular items. It is a widely used benchmark for implicit feedback-based product recommendation tasks. . In addition to Random and Most popular approaches, we considered the Bayesian Personalized Ranking (BPR) approach and an approach based on Neural Matrix Factorization (NeuMF).

The BPR approach is a collaborative filtering approach which focuses on personalized ranking, and optimizes the ranking of items for individual users based on their preferences (Rendle, Freudenthaler, Gantner, & Schmidt-Thieme, 2012). With this approach, the recommendation model encompasses the following components and details:

- **Parameters:** BPR utilizes crucial parameters, including the number of latent factors, learning rate, regularization strength, and the number of iterations. These parameters play a significant role in the training and optimization of the model. We performed a grid search over parameters such as the number of latent factors, learning rate, regularization strength, and iterations. Cross-validation was used to identify the best combination, balancing ranking quality and generalization.
- **Training Methodology:** We used a stochastic gradient descent (SGD) as the optimization algorithm. During the training process, user–item pairs are randomly selected, and the model parameters are optimized to maximize the ranking of positive items over negative items.
- **Implementation :** BPR relies on matrix factorization techniques to learn embeddings for users and items. The model leverages a pairwise ranking objective, which compares user preferences for positive and negative items to derive personalized rankings. It is important to provide specific details regarding the employed loss function and any additional enhancements or modifications introduced to the BPR algorithm.

Table 2
Design of experiments.

Expr.	Objective	Configurations tested	Results
1	Compare the performance of the i-ALS algorithm according to three levels of taxonomies	Three taxonomy levels: ITEM_ID, ITEM_SS_CATG, and ITEM_FAMILY	We calculate P@3 of the algorithm for each configuration. The best configuration will be used for the next experiments.
2	Evaluate the effect of introducing ARM on the performance of the i-ALS	ARM with a low confidence level (NC_LOW = 0.35) and ARM with a high confidence level (NC_HIGH = 0.65)	We calculate P@3 of the algorithm for each configuration. The best configuration will be used for the next experiments
3	Evaluate the effect of the initialization of the i-ALS algorithm	With Initialization and without Initialization (Random)	We calculate P@3 of the algorithm for each configuration. The best configuration will be used for the next experiments
4	Compare the proposed approach to four baselines	The proposed approach, Random, Most Popular, BPR, NeuMF	We calculate P@3 of the algorithm for each approach

NeuMF approach used a hybrid recommendation model that integrates matrix factorization with neural networks (He et al., 2017). Its goal is to capture both linear and nonlinear user–item interactions. The model includes the following components and details:

- **Parameters:** NeuMF encompasses parameters related to the architecture of the neural network, such as the number of hidden layers, hidden units per layer, activation functions, and dropout rates. Additionally, it may include parameters associated with matrix factorization, such as the number of latent factors. We similarly tuned the number of hidden layers, hidden units, activation functions, dropout rates, and latent factors. Dropout and L2 regularization were adjusted to prevent overfitting, and different learning rates were tested for stable convergence.
- **Training Methodology:** NeuMF frequently employs backpropagation with gradient descent as the optimization approach. The training process involves feeding user–item pairs into the model, calculating the loss based on predicted ratings, and iteratively updating the model parameters to minimize the loss. It is essential to provide specific details about the training algorithm, the schedule of the learning rate, and any adopted regularization techniques.
- **Implementation Details:** NeuMF combines matrix factorization and neural network components. Typically, it integrates the embeddings derived from the matrix factorization model with the hidden layers of the neural network to capture both collaborative and content-based information. Detailed architectural information, such as the structure of the neural network layers and the fusion mechanism, should be provided.

4.4. Design of experiments

Four experiments were conducted (see Table 2) . The first experiment aims to compare different configurations of the i-ALS algorithm from the approach proposed in Section 3 and to select the best configuration. The best configuration is then compared in Experiment 4 with the different baselines (Random, Most Popular, BPR, and MeuMF).

As presented in Table 2, the first experiments aim to compare the performance of the *i-ALS* algorithm across different levels of taxonomies (Level 1: Subcategory denoted by ITEM_SS_CATG, Level 2: Item family denoted by ITEM_FAMILY) while considering only the Unique Identifiers (IDs) of items (Level 3: ITEM_ID). To achieve this, the i-ALS algorithm is evaluated using each of these matrices:

- **The matrix ALS_ITEM_SS_CATG** is derived from the original database by calculating, for each user u, the frequency of purchases for items belonging to a specific subcategory ITEM_SS_CATG;
- **The matrix ALS_ITEM_FAMILY** is derived from the original database by calculating, for each user u, the frequency of purchases for items belonging to a specific type ITEM_FAMILY;
- **The matrix ALS_ITEM_ID** is derived from the original database by calculating, for each user u, the frequency of purchases for each ITEM_ID identifier.

Among the three matrices, the one that demonstrated the best algorithm performance is used for experiments 2,3 and 4.

Experiments 2 and 3 aim to investigate potential enhancements to the i-ALS algorithm using the selected matrix in experiment 1. In Experiment 2, we tested a range of confidence levels and selected thresholds of 0.35 and 0.65 to analyze the impact of rule density on the i-ALS algorithm's performance. The 0.35 threshold generated a substantial number of association rules, enriching the user–item matrix with complementary products, while the more stringent 0.65 threshold produced fewer, higher-confidence associations. These values allowed us to explore how varying rule densities affect recommendation accuracy. Higher confidence levels (above 0.65) produced negligible or no rules, making them unsuitable for our analysis.

In Experiment 3, we evaluate the impact of using the BLS-SVD++ method to initialize the i-ALS algorithm. BLS-SVD++ applies singular value decomposition to the historical user–item matrix, producing data-informed latent factors as initial values for i-ALS. This method reduces sparsity in the matrix and aligns latent factors with user interaction patterns, thereby improving convergence speed and accuracy compared to random initialization. Unlike standard SVD, which directly decomposes the matrix, BLS-SVD++ further optimizes the factors for compatibility with i-ALS, resulting in a more stable and efficient iterative process.

5. Results

The results of the different experiments are summarized in Fig. 3 and in the subsections below.

5.1. Experiment 1

The results obtained by applying the i-ALS algorithm to the three matrices (ALS_ITEM_ID, ALS_ITEM_FAMILY, and ALS_ITEM_SS_CATG) are compared in Fig. 4. It can be observed that the i-ALS algorithm shows superior performance when applied to the ALS_ITEM_FAMILY matrix. Specifically, the matrices based on taxonomy levels, namely the ALS_ITEM_FAMILY matrix and the ALS_ITEM_SS_CATG matrix, outperform the standard matrix (ALS_ITEM_ID) with improvements of **23.25** and **14.34%**, respectively. The difference in performance can be attributed to the different levels of data sparsity among the three matrices. The difference in performance can be attributed to the different levels of data sparsity among the three matrices. Table 3 highlights

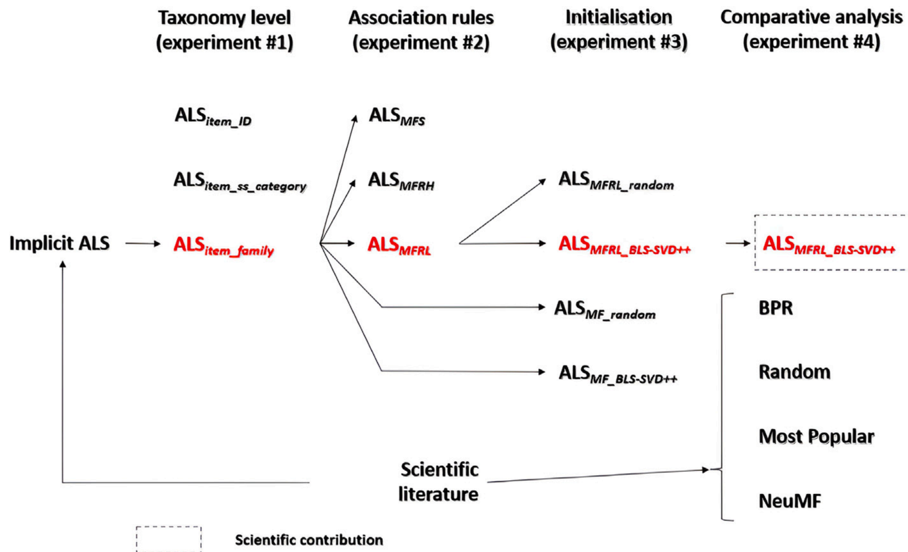

Fig. 3. Summary of experiments.

Table 3
Comparison of taxonomy levels in terms of sparsity, accuracy, and computational complexity.

Taxonomy level	Matrix name	Sparsity (%)	Accuracy improvement	Computational complexity
Item level	ALS_ITEM_ID	99.34	Baseline (0%)	Moderate
Subcategory level	ALS_ITEM_SS_CATG	98.70	+14.34%	Moderate
Family level	ALS_ITEM_FAMILY	**91.18**	+23.25%	Moderate

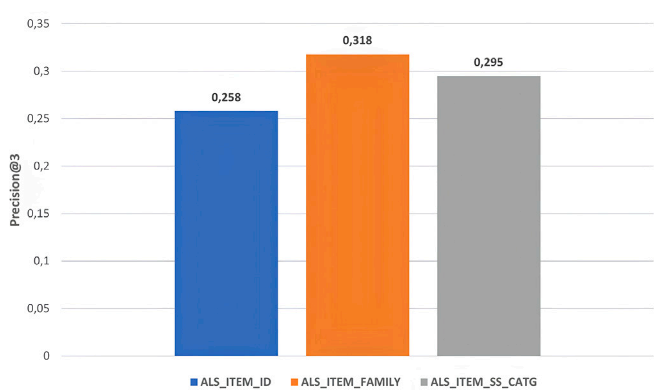

Fig. 4. Precision considering different taxonomy levels.

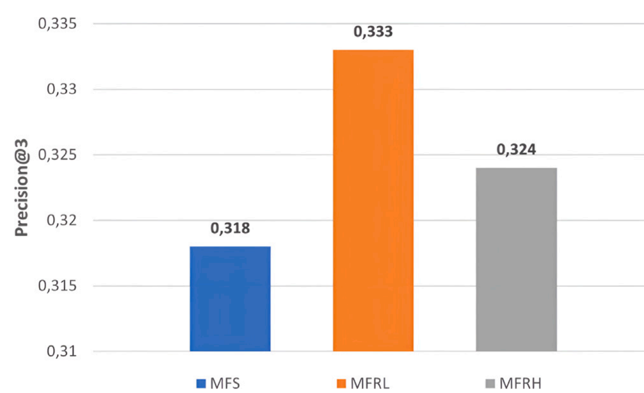

Fig. 5. Impacts of ARM on the i-ALS algorithm performance.

that the ALS_ITEM_FAMILY matrix provides the most balanced trade-off between sparsity, accuracy, and computational complexity. Thus, the ALS_ITEM_FAMILY matrix will be used for the next experiments.

5.2. Experiment 2

In order to evaluate the impact of introducing the ARM on the performance of i-ALS algorithm, three different configurations are tested (see Fig. 3):

- MFS : The algorithm i-ALS is applied to the **M**atrix ITEM_**F**AMILY **S**tandard (ALS_ITEM_FAMILY).
- MFRL : The algorithm i-ALS is applied to the **M**atrix ITEM_**F**AMILY **R**einforced by association rules, using a low confidence level (NC_**L**OW = 0.35).
- MFRH: The algorithm i-ALS is applied to the **M**atrix ITEM_**F**AMILY **R**einforced by association rules, using a high confidence level (NC_**H**IGH = 0.65).

Fig. 5 shows that the two configurations, MFRL and MFHL, which use ARM to reinforce the ITEM_FAMILY matrix, outperformed the standard configuration MFS. In particular, using a low confidence level resulted in the largest improvement of 4.7% compared to MFS. However, using a high confidence level did not lead to any further performance improvement. In fact, ARM with a high confidence level achieved only 1.88% improvement compared to MFS. This can be attributed to the fact that a high confidence level generates a smaller number of rules compared to a low confidence level. Therefore, by using a low confidence level, we can better enrich the user–item matrix and subsequently reduce the sparsity level of the matrix ALS_ITEM_FAMILY.

5.3. Experiment 3

In order to assess the impact of initializing the i-ALS algorithm with the outcome of the BLS-SVD++ decomposition technique, four methodologies were compared (see Fig. 3):

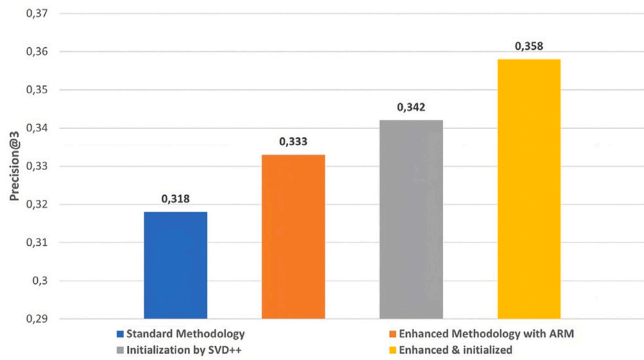

Fig. 6. Evaluation of the effect of initialization.

Table 4
Precision P@3 achieved by the different approaches.

	#Approaches	P@3
	Proposed approach	**0.358**
Baselines	BPR	0.316
	Random	0.09
	Most Popular	0.134
	NeuMF	0.314

Table 5
Runtime of the proposed method for datasets of varying sizes.

Dataset size	Taxonomy (s)	Rule mining (s)	I-ALS (s)	Total (s)
Small	2.5	1.3	5.0	8.8
Medium	18.4	12.1	26.7	57.2
Large	65.8	35.3	75.5	176.6

- Standard Methodology: This involves applying the i-ALS algorithm to the ALS_ITEM_FAMILY matrix with random initialization.
- Initialization by BLS-SVD++: This involves initializing the i-ALS algorithm with the results of the BLS-SVD++ algorithm and applying it to the ALS_ITEM_FAMILY matrix.
- Enhanced Methodology with ARM: This entails applying the i-ALS algorithm to the MFRL matrix, which is enhanced with the association rules with a low confidence level (motivated by results of experiment 2).
- Enhanced & Initialized: This involves initializing the i-ALS algorithm with the results of the BLS-SVD++ algorithm and applying it to the reinforced matrix MFRL.

Fig. 6 shows that all the applied methods effectively improved the accuracy of the algorithm. The proposed initialization method (Initialization with BLS-SVD++) improves the i-ALS algorithm by 7.54% compared to the random initialization (Standard Methodology). The (Enhanced & Initialized) approach achieves the highest improvement of 12.57%.

Thus far, we have demonstrated that better accuracy can be achieved by using: (i) the ALS_ITEM_FAMILY taxonomy, (ii) ARM rules with a low confidence level, and (iii) initialization with SVD++. Next, we compare this configuration with the four common baselines described in Section 4.3.

5.4. Final experiment

This experiment aims to compare the proposed approach with conventional approaches (see Fig. 3). Table 4 shows that the proposed approach exhibits superior performance compared to the four baseline models during the evaluation test period. This improvement indicates that the RS is capable of generating recommendations with greater accuracy. One of the principal factors contributing to this enhancement in performance is the automatic generation of a new taxonomy derived from the textual descriptions of items. The model's performance is notably enhanced by addressing two significant challenges in collaborative filtering: sparsity and synonymy.

The issue of sparsity is addressed by the ITEM FAMILY taxonomy, which facilitates a more comprehensive representation of user–item interactions. This results in a more robust model that is better able to capture user preferences, even in the absence of historical data. Conversely, the synonym problem is effectively addressed by the taxonomy's capacity to group items with similar semantics. This grouping ensures that the model is better able to recognize and recommend items that may not be explicitly linked but are related in a meaningful way based on their descriptions.

Furthermore, the hypothesis concerning non-random initialization based on Singular Value Decomposition (SVD) was tested, and the results indicate that such initialization significantly enhances the accuracy of the matrix factorization technique. The SVD-based initialization provides a superior starting point for the factorization process, resulting in more accurate latent factor estimations. This, in turn, enhances the overall recommendation quality.

The combination of the newly generated taxonomy and the optimized initialization technique contributes to the superior performance of the proposed model by addressing core limitations in collaborative filtering.

5.5. Computational complexity analysis

The computational complexity of the proposed method is detailed as follows:

- **Taxonomy Generation:** The taxonomy is constructed by aggregating items with similar characteristics based on their textual descriptions. This involves pairwise similarity computations for n items, which has a complexity of $O(n^2)$. To make this feasible for large datasets, efficient string similarity measures (cosine similarity) and clustering techniques are applied to group items into `ITEM_FAMILY` categories.
- **Association Rule Mining:** We employ the *Apriori algorithm* to mine frequent itemsets. Its complexity is $O(m \cdot 2^k)$, where m is the number of transactions and k is the average number of items per transaction.
- **Matrix Factorization:** For collaborative filtering, we utilize the *Implicit Alternating Least Squares (I-ALS)* algorithm. The complexity of I-ALS is $O(n \cdot d^2)$ per iteration, where n is the number of users or items, and d is the latent dimensionality. In our implementation, convergence is typically achieved within 500 iterations.

5.6. Runtime and scalability

We evaluated the runtime of the taxonomy generation, association rule mining, and matrix factorization components using datasets of varying sizes. Results are summarized in Table 5.

- **Taxonomy Generation:** Runtime scales quadratically with the number of items due to pairwise similarity calculations. This step is efficient for datasets with up to 15 000 items but may require optimization (e.g., parallel processing) for larger datasets.
- **Association Rule Mining:** The Apriori algorithm's runtime increases exponentially with the average itemset size k. Setting a minimum support threshold effectively limits this growth.
- **Matrix Factorization:** I-ALS scales linearly with the number of users/items and quadratically with the latent dimension d. It demonstrates practical runtime for typical latent dimensions ($d \leq 100$).

5.7. Practical implications

The analysis shows that the proposed method is computationally feasible for datasets of moderate size. Scalability challenges may arise for very large item catalogs or transaction datasets, particularly in the taxonomy generation and rule mining steps. Possible improvements include:

- Optimizing taxonomy generation using approximate string matching or distributed computation.
- Employing sampling or dimensionality reduction techniques to accelerate Apriori.
- Leveraging parallel or distributed I-ALS implementations for large-scale datasets.

6. Conclusion and perspectives

The focus of this work has been the development of a collaborative filtering-based implicit feedback recommendation system, with particular emphasis on its application in a real-world case study in the B2B context. The challenges associated with data sparsity and synonym ambiguity are prevalent in collaborative filtering, affecting the quality and accuracy of recommendations. The user–item matrix often contains sparse entries and missing data, which require effective solutions.

To address these challenges, a hybrid approach combining three techniques from the literature — ARM, the i-ALS algorithm, and the BLS-SVD++ initialization approach — was proposed. An explicit taxonomy was automatically generated based solely on the textual descriptions of the items, contributing to the improvement of the user–item matrix. In addition, association rule extraction techniques were used to identify complementary items for users, based on their purchases of base items as indicated by the extracted rules. Furthermore, our approach included a matrix factorization algorithm initialized with computed values obtained from the BLS-SVD++ approach.

Through experiments conducted on real-world data, our proposed approach showed promising results, confirming the effectiveness of using taxonomy and association rule extraction techniques to overcome the challenges of sparsity and synonym ambiguity. In particular, the initialization of the primary algorithm, i-ALS, showed superior accuracy compared to random initialization methods.

The experimental results also showed that our approach outperformed existing approaches in the literature in terms of accuracy, highlighting its potential for improving the performance of recommendation systems in a B2B context. The practical implications of this study are particularly relevant for recommendation systems used in B2B settings and can be generalized to other real-world cases.

In this study, only an offline evaluation could be carried out on the partner's data. Once the proposed recommendation system is implemented in the industrial partner's online store, online experiments can be conducted to measure not only the predictive power of the proposed recommendation system but also its influence on user behavior.

Furthermore, the offline evaluation has already provided preliminary insights into potential areas for refinement, such as tailoring recommendations to seasonal trends, which will be further validated through online implementation.

While this study focuses on a specific B2B distributor case, we recognize that scaling the approach to larger datasets or different industries presents challenges. Future work will explore the scalability of the method and its adaptability to various industries, ensuring its robustness across diverse real-world contexts.

Emerging metaheuristic optimization algorithms, such as Particle Swarm Optimization (Gad, 2022), Teaching-Learning-Based Optimization (Xue & Wu, 2019), and Manta Ray Foraging Optimization (Soleimanian Gharehchopogh, Ghafouri, Namazi, & Arasteh, 2024), present promising opportunities to enhance matrix factorization techniques like i-ALS. These methods can be utilized for initializing factorized matrices, optimizing hyperparameters, or addressing non-convex loss functions, thereby improving model robustness and convergence. While challenges such as computational overhead and scalability remain, integrating these algorithms into i-ALS frameworks could open new avenues for handling sparse implicit feedback data effectively, offering a potential direction for future research.

CRediT authorship contribution statement

Thamer Saraei: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Maha Benali:** Validation, Supervision, Writing – review & editing. **Jean-Marc Frayret:** Validation, Supervision, Writing – review & editing.

Acknowledgments

The authors thank the industrial partner for providing the real-world data used in the case study. We also thank MITACS Canada and the industrial partner for financial support.

Data availability

The data that has been used is confidential.

References

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on management of data* (pp. 207–216).

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I., et al. (1996). Fast discovery of association rules.. *Advances in Knowledge Discovery and Data Mining*, *12*(1), 307–328.

Ahmed, A., Kanagal, B., Pandey, S., Josifovski, V., Pueyo, L. G., & Yuan, J. (2013). Latent factor models with additive and hierarchically-smoothed user preferences. In *Proceedings of the sixth ACM international conference on web search and data mining* (pp. 385–394).

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, *40*(3), 66–72.

Chen, S., & Peng, Y. (2018). Matrix factorization for recommendation with explicit and implicit feedback. *Knowledge-Based Systems*, *158*, 109–117.

Donoho, D., & Stodden, V. (2003). When does non-negative matrix factorization give a correct decomposition into parts? *Advances in Neural Information Processing Systems*, *16*.

Feng, H., Tian, J., Wang, H. J., & Li, M. (2015). Personalized recommendations based on time-weighted overlapping community detection. *Information & Management*, *52*(7), 789–800.

Gad, A. (2022). Particle swarm optimization algorithm and its applications: A systematic review. *Archives of Computational Methods in Engineering*, *29*, 2531–2561. http://dx.doi.org/10.1007/s11831-021-09694-4.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182).

Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining* (pp. 263–272). IEEE.

Iijima, J., & Ho, S. (2007). Common structure and properties of filtering systems. *Electronic Commerce Research and Applications*, *6*(2), 139–145.

Kanagal, B., Ahmed, A., Pandey, S., Josifovski, V., Yuan, J., & Pueyo, L. G. (2012). Supercharging recommender systems using taxonomies for learning user purchase behavior. CoRR, abs/1207.0136. arXiv:1207.0136. URL: http://arxiv.org/abs/1207.0136.

Kardan, A. A., & Ebrahimi, M. (2013). A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. *Information Sciences*, *219*, 93–110.

Kim, H.-R., & Mathes, G. (2001). Explicit vs. implicit corrective feedback. *The Korea TESOL Journal*, *4*(1), 57–72.

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 426–434).

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M., & Duri, S. S. (2001). Personalization of supermarket product recommendations. In *Applications of data mining to electronic commerce* (pp. 11–32). Springer.

Leung, C. W.-k., Chan, S. C.-f., & Chung, F.-l. (2006). A collaborative filtering framework based on fuzzy association rules and multiple-level similarity. *Knowledge and Information Systems*, *10*(3), 357–381.

Leung, C. W.-k., Chan, S. C.-f., & Chung, F.-l. (2008). An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, *21*(7), 515–529.

Liphoto, M., Du, C., & Ngwira, S. (2016). A survey on recommender systems. In *2016 international conference on advances in computing and communication engineering* (pp. 276–280). IEEE.

Liu, N. N., Xiang, E. W., Zhao, M., & Yang, Q. (2010). Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 1445–1448).

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K., & Zhou, T. (2012). Recommender systems. *Physics Reports*, *519*(1), 1–49.

Maidel, V., Shoval, P., Shapira, B., & Taieb-Maimon, M. (2008). Evaluation of an ontology-content based filtering method for a personalized newspaper. In *Proceedings of the 2008 ACM conference on recommender systems* (pp. 91–98).

Mnih, A. (2012). Taxonomy-informed latent factor models for implicit feedback. In *Proceedings of KDD cup 2011* (pp. 169–181). PMLR.

Najafabadi, M. K., Mahrin, M. N., Chuprat, S., & Sarkan, H. M. (2017). Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Computers in Human Behavior*, *67*, 113–128.

Nia, A. G., Lu, J., Zhang, Q., & Ribeiro, M. (2019). A framework for a large-scale B2B recommender system. In *2019 IEEE 14th international conference on intelligent systems and knowledge engineering* (pp. 337–343). http://dx.doi.org/10.1109/ISKE47853.2019.9170298.

Oppermann, R., & Specht, M. (1999). A nomadic information system for adaptive exhibition guidance. *Archives and Museum Informatics*, *13*(2), 127–138.

Panda, M., & Patra, M. R. (2009). Mining association rules for constructing network intrusion detection model. *International Journal of Applied Engineering Research*, *4*(3), 381–399.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.

Schoinas, I., & Tjortjis, C. (2019). MuSIF: a product recommendation system based on multi-source implicit feedback. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 660–672). Springer.

Smilde, A. K., Geladi, P., & Bro, R. (2005). *Multi-way analysis: applications in the chemical sciences*. John Wiley & Sons.

Soleimanian Gharehchopogh, F., Ghafouri, S., Namazi, M., & Arasteh, B. (2024). Advances in manta ray foraging optimization: A comprehensive survey. *Journal of Bionic Engineering*, *21*, http://dx.doi.org/10.1007/s42235-024-00481-y.

Wang, S., Sun, G., & Li, Y. (2020). Svd++ recommendation algorithm based on backtracking. *Information*, *11*, 369. http://dx.doi.org/10.3390/info11070369.

Wen, H., Ding, G., Liu, C., & Wang, J. (2014). Matrix factorization meets cosine similarity: addressing sparsity problem in collaborative filtering recommender system. In *Web technologies and applications: 16th Asia-Pacific web conference, APWeb 2014, Changsha, China, September 5-7, 2014. proceedings 16* (pp. 306–317). Springer.

Xian, Z., Li, Q., Li, G., & Li, L. (2017). New collaborative filtering algorithms based on svd++ and differential privacy. *Mathematical Problems in Engineering*, *2017*.

Xue, R., & Wu, Z. (2019). A survey of application and classification on teaching-learning-based optimization algorithm. *IEEE Access*, *PP*, http://dx.doi.org/10.1109/ACCESS.2019.2960388, 1–1.

Zhang, Y., Ahmed, A., Josifovski, V., & Smola, A. (2014). Taxonomy discovery for personalized recommendation. In *Proceedings of the 7th ACM international conference on web search and data mining* (pp. 243–252). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2556195.2556236.

Ziegler, C.-N., Lausen, G., & Schmidt-Thieme, L. (2004). Taxonomy-driven computation of product recommendations. In *Proceedings of the thirteenth ACM international conference on information and knowledge management* (pp. 406–415).

Zuva, T., Ojo, S. O., Ngwira, S., Zuva, K., et al. (2012). A survey of recommender systems techniques, challenges and evaluation metrics. *International Journal of Emerging Technology and Advanced Engineering*, *2*(11), 382–386.