

A Service of



Leibniz-Informationszentrum Wirtschaft Leibniz Information Centre for Economics

Schmidtke, Marcel; Neeb, Jannik; Wöhner, Thomas

#### Research Report Collaborative filtering recommender systems: Methods, strengths and weaknesses

Wirtschaftswissenschaftliche Schriften, No. 01/2025

#### **Provided in Cooperation with:**

Ernst-Abbe-Hochschule Jena, University of Applied Sciences, Department for Business Administration

*Suggested Citation:* Schmidtke, Marcel; Neeb, Jannik; Wöhner, Thomas (2025) : Collaborative filtering recommender systems: Methods, strengths and weaknesses, Wirtschaftswissenschaftliche Schriften, No. 01/2025, ISBN 978-3-932886-48-5, Verlag Ernst-Abbe-Hochschule Jena, Jena

This Version is available at: https://hdl.handle.net/10419/315753

#### Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

#### Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



## WWW.ECONSTOR.EU



## Wirtschaftswissenschaftliche Schriften

## Collaborative Filtering Recommender Systems:

Methods, Strengths and Weaknesses

Marcel Schmidtke Jannik Neeb Thomas Wöhner

Heft 01 / 2025

Department for Business Administration Fachbereich Betriebswirtschaft

# Collaborative Filtering Recommender Systems:

### Methods, Strengths and Weaknesses

Marcel Schmidtke Jannik Neeb Thomas Wöhner



## Reihe:Wirtschaftswissenschaftliche Schriften<br/>Jahrgang 2025, Heft 01Herausgeber:Thomas Wöhner

Redaktion: Prof. Dr. Frank Pothen

ISSN 1868-1689 eISSN (PDF) 1868-1697 ISBN 978-3-932886-47-8 eISBN (PDF) 978-3-932886-48-5

© 2025 Verlag Ernst-Abbe-Hochschule Jena Carl-Zeiss-Promenade 2, 07745 Jena

Ernst-Abbe-Hochschule Jena, University of Applied Sciences Fachbereich Betriebswirtschaft Carl-Zeiss-Promenade 2, 07745 Jena Tel.: 03641-205-550

Die vorliegende Publikation wurde mit größter Sorgfalt erstellt, Verfasser und Herausgeber können für den Inhalt jedoch keine Gewähr übernehmen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Dieses Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Vervielfältigung, Übersetzung, Mikroverfilmung und Einspeicherung in elektronische Systeme des gesamten Werkes oder Teilen daraus bedarf – auch für Unterrichtszwecke – der vorherigen Zustimmung der Ernst-Abbe-Hochschule Jena, Fachbereich Betriebswirtschaft und der Autor:innen.

Printed in Germany

Wirtschaftswissenschaftliche Schriften Heft 01/2025

#### **Collaborative Filtering Recommender Systems:**

#### Methods, Strengths and Weaknesses

Marcel Schmidtke, Jannik Neeb, Thomas Wöhner

#### Abstract

Recommender systems are indispensable in e-business due to the extensive product range and the large number of niche articles. Collaborative filtering (CF) algorithms play a central role in generating personalised recommendations. There are numerous CF approaches in the literature that have specific advantages and limitations. The aim of this article is to provide a comprehensive overview of the characteristics of model- and memory-based CF methods through a systematic literature analysis of 62 scientific papers.

The analysis shows that memory-based approaches are convincing due to their simple implementation and interpretability but exhibit scaling problems as well as susceptibility to data sparsity and the cold start problem. Model-based methods, on the other hand, offer greater scalability and robustness against data sparsity, but require a more complex implementation, higher computing power and complicate the interpretation of results.

**Keywords**: Recommender systems, collaborative filtering, literature review, memory-based, model-based

#### Zusammenfassung

Recommender-Systeme sind im E-Business aufgrund des umfangreichen Produktsortiments und der Vielzahl an Nischenartikeln unverzichtbar. Dabei nehmen Collaborative-Filtering-Algorithmen (CF) eine zentrale Rolle bei der Generierung personalisierten Empfehlungen ein. In der Literatur existieren zahlreiche CF-Ansätze, die spezifische Vorteile und Limitationen aufweisen. Ziel dieses Artikels ist es, durch eine systematische Literaturanalyse von 62 wissenschaftlichen Arbeiten, einen umfassenden Überblick über die Merkmale von model- und speicherbasierten CF-Methoden zu geben.

Die Analyse zeigt, dass speicherbasierte Ansätze durch einfache Implementierung und Interpretierbarkeit überzeugen, jedoch Skalierungsprobleme sowie eine Anfälligkeit gegenüber Datenknappheit und Kaltstartproblemen aufweisen. Modellbasierte Methoden bieten dagegen eine höhere Skalierbarkeit und Robustheit gegenüber Datenknappheit, erfordern jedoch eine komplexere Implementierung, mehr Rechenleistung und erschweren die Ergebnis-interpretation.

**Schlagwörter**: Empfehlungssysteme, kollaboratives Filtern, systematische Literaturanalyse, speicherbasiert, modellbasiert

E-Mail: <u>marcel.schmidtke@stud.eah-jena.de</u>, <u>jannik.neeb@eah-jena.de</u>, <u>thomas.woehner@eah-jena.de</u>

#### Contents

List of AbbreviationsII			
1	Intro	duction1	
	1.1	Problem Definition1	
	1.2	Objective1	
	1.3	Methodology and Approach1	
2	First Principles of Collaborative Filtering2		
3 Memory-Based Collaborative Filtering		nory-Based Collaborative Filtering	
	3.1	User-Based Collaborative Filtering	
	3.1.1	Concept of User-Based Collaborative Filtering	
	3.1.2	Strengths and Weaknesses of User-Based Collaborative Filtering	
	3.2	Item-Based Collaborative Filtering	
	3.2.1	Concept of Item-Based Collaborative Filtering	
	3.2.2	2. Strengths and Weaknesses of Item-Based Collaborative Filtering	
4	Mod	el-based Collaborative Filtering9	
	4.1	Matrix Factorization in Collaborative Filtering9	
	4.1.1	Latent Factor Models9	
	4.1.2	Matrix Factorization Techniques10	
	4.1.3	Strengths and Weaknesses of Matrix Factorization11	
	4.2	Probabilistic Graphical Models	
	4.2.1	Concept of Probabilistic Graphical Models13	
	4.2.2	Strengths and Weaknesses of Probabilistic Graphical Models14	
5	Cone	Concluding Remarks1	
	5.1	Summary15	
	5.2	Critical Appraisal and Outlook16	
	Refe	rencesV	

#### List of Abbreviations

Abbreviation	Definition
ACOS	Adjusted cosine similarity
ALS	Alternating least squares
BPR	Bayesian personalized ranking
CF	Collaborative filtering
CS	Cosine similarity
IBCF	Item-based collaborative filtering
LF	Latent factor
LFM	Latent factor model
MCA	Mean-centered approach
MF	Matrix factorization
NMF	Non-negative matrix factorization
PCC	Pearson correlation coefficient
PGM	Probabilistic graphical model
RS	Recommender system
RBM	Restricted Boltzmann machine
SGD	Stochastic gradient descent
SVD	Singular value decomposition
UBCF	User-based collaborative filtering
UII	User-item interaction
UIM	User-item interaction matrix

#### **1** Introduction

#### **1.1 Problem Definition**

Recommender systems (RS) have become indispensable in various industries, particularly in e-commerce and entertainment, where they address information overload and significantly enhance user experiences.<sup>1</sup> By suggesting relevant products, services, or content, these systems help users navigate vast amounts of information efficiently, improving engagement and satisfaction.<sup>2</sup> However, the effectiveness of collaborative filtering (CF) hinges on the chosen method and its inherent strengths and weaknesses, which vary across different algorithms.<sup>3</sup> The literature on CF is extensive, yet a comprehensive understanding of the diverse methods, their advantages and limitations, and their applicability to real-world constraints remains elusive.<sup>4</sup> This paper addresses this gap by providing a systematic overview of CF techniques, analyzing their theoretical foundations, and identifying their inherent characteristics. The central research question guiding this investigation is: "How do the strengths and weaknesses of memory- and model-based CF methods affect their ability to address real-world constraints?"

#### 1.2 Objective

The objective of this paper is to conduct a thorough analysis of various CF approaches, comparing their strengths and weaknesses in relation to real-world constraints.

#### 1.3 Methodology and Approach

This research is grounded in a thorough review of relevant literature, encompassing peerreviewed journal articles, books, and conference proceedings sourced from reputable academic databases like Google Scholar, IEEE Xplore, and ACM Digital Library. A systematic search strategy was employed, utilizing different keyword combinations and their variations.<sup>5</sup> Initial searches yielded a vast number of results, which were systematically refined through multiple stages of screening, including title/abstract review, full-text assessment, and relevance filtering. Ultimately, 62 high-quality sources were selected based on their relevance, recency, and quality metrics like VHB-JOURQUAL3, Scimago Journal Rank, and

<sup>&</sup>lt;sup>1</sup> Cf. Jannach, Zanker and Felderer (2010), pp. 13 ff; cf. Resnick/ Varian (1997), pp. 56 ff.

<sup>&</sup>lt;sup>2</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff; cf. Resnick/ Varian (1997), pp. 56 ff.

<sup>&</sup>lt;sup>3</sup> Cf. Goldberg, Nichols, Oki and Terry (1992), pp. 61 ff; cf. Schafer, Frankowski, Herlocker and Sen (2007), pp. 291 ff.

<sup>&</sup>lt;sup>4</sup> Cf. Koren/ Bell (2015), pp. 77 ff; cf. Schein, Popescul, Ungar and Pennock (2002), pp. 253 ff.

<sup>&</sup>lt;sup>5</sup> Cf. Webster/ Watson (2002), pp. 13 ff.

CORE ranking. To ensure comprehensibility and conciseness, the paper focuses on analyzing a selection of common memory- and model-based CF methods. Therefore, techniques like clustering, association rule mining, regression, ensemble methods, and hybrid approaches are excluded from this study to limit the scope.

This paper presents a structured exploration of CF to achieve its objectives, starting with the first principles in Chapter 2. Chapter 3 delves into memory-based CF techniques, specifically user-based and item-based approaches, analyzing their concepts, strengths, and weaknesses. Chapter 4 explores model-based CF, focusing on matrix factorization (MF) and probabilistic graphical models (PGMs), elucidating their principles, advantages, and limitations. Finally, Chapter 5 concludes the paper with a summary of key findings, a critical appraisal, and an outlook on future research directions.

#### 2 First Principles of Collaborative Filtering

CF is a popular and effective technique used in modern RS, enabling personalized content delivery based on user interactions.<sup>6</sup> RS are advanced algorithms designed to help users discover items or content matching their individual preferences and needs.<sup>7</sup> These systems analyze user data, including past interactions and preferences, to provide personalized recommendations, enhancing user experience and satisfaction.<sup>8</sup> The fundamental idea behind CF is to leverage user-item interactions (UII) to generate recommendations.<sup>9</sup> This approach assumes that users who have agreed on items in the past will likely agree on similar items in the future.<sup>10</sup> The basis for personalized recommendations in CF is the UII matrix (UIM).<sup>11</sup> In this matrix, rows represent users, columns represent items, and entries capture various types of UIIs.<sup>12</sup> These interactions can be expressed explicitly through ratings or implicitly through behaviors like clicks or purchases.<sup>13</sup> Ratings can take various forms: continuous ratings, where values fall within a range; interval-based ratings, using a discrete set of or-

<sup>&</sup>lt;sup>6</sup> Cf. Aggarwal (2016), p. 8; cf. Schafer, Frankowski, Herlocker and Sen (2007), pp. 291 ff.

<sup>&</sup>lt;sup>7</sup> Cf. Aggarwal (2016), pp. 1 f; cf. Goldberg, Nichols, Oki and Terry (1992), pp. 61 ff; cf. Resnick/ Varian (1997), pp. 56 ff.

<sup>&</sup>lt;sup>8</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff; cf. Konstan/ Riedl (2012), pp. 106 ff; cf. Lu, Wu, Mao, Wang and Zhang (2015), pp. 12 ff.

<sup>&</sup>lt;sup>9</sup> Cf. Aggarwal (2016), p. 8; cf. Schafer, Frankowski, Herlocker and Sen (2007), pp. 291 ff.

<sup>&</sup>lt;sup>10</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff; cf. Herlocker, Konstan, Terveen and Riedl (2004), pp. 5 ff.

<sup>&</sup>lt;sup>11</sup> Cf. Aggarwal (2016), pp. 2 f; cf. Bobadilla, Ortega, Hernando and Gutiérrez (2013), pp. 109 ff.

<sup>&</sup>lt;sup>12</sup> Cf. Aggarwal (2016), pp. 2 f; cf. Bobadilla, Ortega, Hernando and Gutiérrez (2013), pp. 109 ff.

<sup>&</sup>lt;sup>13</sup> Cf. Aggarwal (2016), pp. 10 f; cf. Ekstrand, Riedl and Konstan (2011), pp. 81 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 ff.

dered numbers; ordinal ratings, reflecting ranked categorical responses; binary ratings, indicating like or dislike; and unary ratings, which only indicate a positive preference.<sup>14</sup> CF algorithms analyze the UIM to identify patterns and similarities in user preferences and item characteristics, enabling personalized recommendations for items users have not vet interacted with.<sup>15</sup> CF methods can be broadly classified into two main types: memory-based and model-based.<sup>16</sup> Memory-based CF, also known as neighborhood-based CF, is a technique used in recommender systems that relies on the entire user-item interaction matrix (UIM) to make recommendations based on similarities between users or items.<sup>17</sup> This approach can be further classified into user-based CF (UBCF) and item-based CF (IBCF).<sup>18</sup> In contrast, model-based CF involves creating a predictive model from the UIM.<sup>19</sup> These models are trained on the UII data and can generalize from the patterns they find to give recommendations.<sup>20</sup> CF differs from other recommendation techniques, such as content-based filtering. Content-based filtering recommends items similar to those a user has liked in the past, focusing on item attributes such as genre, type, or description without considering other users' preferences.<sup>21</sup> CF can overcome certain constraints of content-based filtering, such as limited content analysis, where items may not be sufficiently described by available data.<sup>22</sup>

#### 3 Memory-Based Collaborative Filtering

#### 3.1 User-Based Collaborative Filtering

#### 3.1.1 Concept of User-Based Collaborative Filtering

UBCF is a popular recommendation technique that predicts a user's rating for an item based on the ratings given by similar users.<sup>23</sup> For example, if User A and User B have rated several

<sup>&</sup>lt;sup>14</sup> Cf. Aggarwal (2016), pp. 10 f; cf. Koren/ Bell (2015), pp. 77 ff.

<sup>&</sup>lt;sup>15</sup> Cf. Aggarwal (2016), p. 8; cf. Ricci, Rokach and Shapira (2011), pp. 1 ff; cf. Ekstrand, Riedl and Konstan (2011), pp. 81 ff.

<sup>&</sup>lt;sup>16</sup> Cf. Papadakis, Ioannou, Palpanas, Niederee and Koutrika (2022), pp. 40 ff.

<sup>&</sup>lt;sup>17</sup> Cf. Aggarwal (2016), p. 9; cf. Bobadilla, Ortega, Hernando and Gutiérrez (2013), pp. 109 ff.

<sup>&</sup>lt;sup>18</sup> Cf. Aggarwal (2016), pp. 10 f; cf. Resnick, Iacovou, Suchak, Bergstrom and Riedl (1994), pp. 175 ff; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

<sup>&</sup>lt;sup>19</sup> Cf. Aggarwal (2016), p. 9; cf. Koren/ Bell (2015), pp. 77 ff; cf. Lops, Gemmis and Semeraro (2011), pp. 73 ff.

<sup>&</sup>lt;sup>20</sup> Cf. Aggarwal (2016), p. 9; cf. Koren/ Bell (2015), pp. 77 ff; cf. Lops, Gemmis and Semeraro (2011), pp. 73 ff.

<sup>&</sup>lt;sup>21</sup> Cf. Lops, Gemmis and Semeraro (2011), pp. 73 ff; cf. Resnick/ Varian (1997), pp. 56 ff; cf. Zhu, Yu, Chi and Gong (2014), pp. 387 ff.

<sup>&</sup>lt;sup>22</sup> Cf. Koren/ Bell (2015), pp. 77 ff; cf. Lops, Gemmis and Semeraro (2011), pp. 73 ff; cf. Zhu, Yu, Chi and Gong (2014), pp. 387 ff.

<sup>&</sup>lt;sup>23</sup> Cf. Aggarwal (2016), p. 9; cf. Resnick, Iacovou, Suchak, Bergstrom and Riedl (1994), pp. 175 ff.

movies similarly, User A would likely enjoy a new movie that User B has rated highly.<sup>24</sup> UBCF's recommendation methodology begins by collecting historical UII data, typically represented in a UIM.<sup>25</sup> Subsequently, the similarity between users is calculated using metrics like cosine similarity (CS) or Pearson correlation coefficient (PCC).<sup>26</sup> CS measures the cosine of the angle between two vectors, which represent user preferences, in a multi-dimensional space.<sup>27</sup> CS is a widely used metric due to its simplicity and is calculated as follows:

Cosine Similarity 
$$(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{v,i}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_v} r_{vi}^2}}$$

where  $r_{ui}$  and  $r_{vi}$  are the elements of vectors  $I_u$  and  $I_v$ , respectively.<sup>28</sup> A higher CS score indicates more similar preferences between users, which is useful for making recommendations.<sup>29</sup> PCC calculates user similarity based on the linear dependence of their ratings and is particularly useful in handling differences in user rating behaviors.<sup>30</sup> The PCC between two users *u* and *v* is calculated as follows:

$$Pearson\ Correlation\ (u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u) \cdot (r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

where  $r_{ui}$  and  $r_{vi}$  are the ratings of users *u* and v for item *i*, and  $\bar{r}_u$  and  $\bar{r}_v$  are the mean ratings of users *u* and *v*, respectively.<sup>31</sup> Once similarities are calculated, the next step involves selecting the top-N most similar users (nearest neighbors) to the target user.<sup>32</sup> The choice of *N* impacts the recommendation quality, as a larger *N* may include less relevant users, while a smaller *N* may omit useful information.<sup>33</sup> The target user's rating for an unseen item is then predicted by aggregating the neighbors' ratings, often using a weighted average where weights are the similarity scores, or the mean-centered approach (MCA).<sup>34</sup> The MCA adjusts

<sup>&</sup>lt;sup>24</sup> Cf. Aggarwal (2016), p. 9.

<sup>&</sup>lt;sup>25</sup> Cf. Ekstrand, Riedl and Konstan (2011), pp. 81 ff; cf. Resnick, Iacovou, Suchak, Bergstrom and Riedl (1994), pp. 175 ff.

<sup>&</sup>lt;sup>26</sup> Cf. Aggarwal (2016), pp. 34 ff; cf. Ahn (2008), pp. 38 f.

<sup>&</sup>lt;sup>27</sup> Cf. Ahn (2008), p. 39.

<sup>&</sup>lt;sup>28</sup> Cf. Ekstrand, Riedl and Konstan (2011), pp. 81 ff; cf. Jannach, Zanker and Felderer (2010), pp. 13 ff.

<sup>&</sup>lt;sup>29</sup> Cf. Ahn (2008), p. 39; cf. Jannach, Zanker and Felderer (2010), pp. 13 ff.

<sup>&</sup>lt;sup>30</sup> Cf. Ahn (2008), p. 39; cf. Jannach, Zanker and Felderer (2010), pp. 13 ff; cf. Khojamli/ Razmara (2021), pp. 2 f.

<sup>&</sup>lt;sup>31</sup> Cf. Aggarwal (2016), pp. 34 ff; cf. Khojamli/ Razmara (2021), pp. 2 f; cf. Resnick, Iacovou, Suchak, Bergstrom and Riedl (1994), pp. 175 ff.

<sup>&</sup>lt;sup>32</sup> Cf. Desrosiers/ Karypis (2011), pp. 107 ff.

<sup>&</sup>lt;sup>33</sup> Cf. Herlocker, Konstan, Borchers and Riedl (1999), pp. 230 ff.

<sup>&</sup>lt;sup>34</sup> Cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

ratings by subtracting the user's average rating before aggregation, mitigating individual biases and enhancing prediction accuracy in diverse datasets.<sup>35</sup> The formula is as follows:

$$\hat{r}_{ui} = \bar{r}_{v} + \frac{\sum_{v \in N(u)} sim(u, v) \cdot (r_{vi} - \bar{r}_{v})}{\sum_{v \in N(u)} sim(u, v)}$$

where  $\hat{r}_{ui}$  is the predicted rating,  $\bar{r}_u$  is the average rating of user u, N(u) is the set of neighbors for user u, sim(u, v) is the similarity score between users u and v,  $r_{vi}$  is the rating of user v for item i, and  $\bar{r}_v$  average rating of user v.<sup>36</sup> Lastly, a list of recommended items is generated for the active user based on the highest predicted ratings.<sup>37</sup>

#### 3.1.2 Strengths and Weaknesses of User-Based Collaborative Filtering

UBCF has strengths that contribute to its widespread use and effectiveness in RS: The core idea of finding similar users and making predictions based on their preferences is intuitive and straightforward to implement, as it does not require complex algorithms.<sup>38</sup> This simplicity makes UBCF a popular choice for RS, particularly in early-stage deployments or in environments with limited technical resources.<sup>39</sup> Furthermore, UBCF demonstrates superior performance in datasets with dense UIIs, where the abundance of user ratings facilitates accurate similarity calculations and enhances recommendation quality.<sup>40</sup> Additionally, UBCF's flexibility to adapt to various domains and interaction data types, including movies, books, and e-commerce products, makes it a versatile tool for personalized recommendations across different industries.<sup>41</sup> Finally, UBCF directly utilizes user interactions, which often reflect genuine user preferences.<sup>42</sup> This direct interaction ensures that the recommendations are grounded in actual user behavior rather than inferred attributes, leading to more accurate and relevant suggestions that are interpretable.<sup>43</sup>

One of the significant challenges with UBCF is its scalability, as its computational complexity grows exponentially with an increasing number of users and items, potentially causing

<sup>&</sup>lt;sup>35</sup> Cf. Aggarwal (2016), pp. 33 ff; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

<sup>&</sup>lt;sup>36</sup> Cf. Aggarwal (2016), p. 36; cf. Ahn (2008), p. 39; cf. Herlocker, Konstan, Borchers and Riedl (1999), pp. 230 ff.

<sup>&</sup>lt;sup>37</sup> Cf. Aggarwal (2016), pp. 33 ff; cf. Resnick, Iacovou, Suchak, Bergstrom and Riedl (1994), pp. 175 ff.

 <sup>&</sup>lt;sup>38</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff; cf. Herlocker, Konstan, Borchers and Riedl (1999), pp. 230 ff.
<sup>39</sup> Cf. Aggarwal (2016), pp. 44; cf. Walker, Glance and Rodriguez (2004), pp. 365 ff.

<sup>&</sup>lt;sup>40</sup> Cf. Resnick, Iacovou, Suchak, Bergstrom and Riedl (1994), pp. 175 ff; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

<sup>&</sup>lt;sup>41</sup> Cf. Herlocker, Konstan, Terveen and Riedl (2004), pp. 5 ff; cf. Su/ Khoshgoftaar (2009), pp. 1 ff.

<sup>&</sup>lt;sup>42</sup> Cf. Linden, Smith and York (2003), pp. 76 ff.

<sup>&</sup>lt;sup>43</sup> Cf. Linden, Smith and York (2003), pp. 76 ff.

performance bottlenecks in large-scale systems.<sup>44</sup> UBCF is sensitive to data sparsity, a common issue in new or niche domains, where users have rated few items, making it difficult to find similar users and leading to poor recommendations.<sup>45</sup> This sensitivity also exacerbates the cold-start problem for new users and items, as their lack of interaction history hinders the system's ability to provide accurate recommendations.<sup>46</sup> Moreover, UBCF can exhibit a bias towards popular items, as they are more likely to be rated by similar users, potentially exacerbating the long tail problem and limiting user exposure to a wider range of potentially relevant items.<sup>47</sup> UBCF's reliance on historical data can hinder its ability to adapt to changing user preferences and novel items, requiring frequent updates to maintain recommendation relevance and accuracy, which can be computationally intensive.<sup>48</sup>

#### 3.2 Item-Based Collaborative Filtering

#### 3.2.1 Concept of Item-Based Collaborative Filtering

IBCF generates recommendations by predicting a user's rating for an unseen item based on item-to-item similarities.<sup>49</sup> For instance, if a user liked a specific book, IBCF will recommend other books liked by users who also liked the initial book.<sup>50</sup> The methodology of IBCF is as follows: The initial steps of data collection and matrix representation in IBCF are similar to those in UBCF.<sup>51</sup> The similarity computation is analogous to those used in UBCF, with the distinction that similarities are calculated between items *i* and *j* instead of users.<sup>52</sup> Common similarity metrics include PCC, CS, and adjusted Cosine Similarity (ACOS).<sup>53</sup> ACOS is often preferred due to its superior performance in accounting for individual user rating tendencies.<sup>54</sup> By subtracting the user's average rating before calculating similarity,

<sup>&</sup>lt;sup>44</sup> Cf. Linden, Smith and York (2003), pp. 76 ff.

<sup>&</sup>lt;sup>45</sup> Cf. Breese, Heckerman and Kadie (1998), pp. 43 ff; cf. Singh (2020), pp. 15 ff.

<sup>&</sup>lt;sup>46</sup> Cf. Ahn (2008), pp. 37 ff; cf. Schein, Popescul, Ungar and Pennock (2002), pp. 253 ff.

<sup>&</sup>lt;sup>47</sup> Cf. Adomavicius/ Kwon (2007), pp. 48 ff; cf. Aggarwal (2016), p. 49; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

<sup>&</sup>lt;sup>48</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 f.

<sup>&</sup>lt;sup>49</sup> Cf. Linden, Smith and York (2003), pp. 76 ff; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

<sup>&</sup>lt;sup>50</sup> Cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff; cf. Zhu, Yu, Chi and Gong (2014), pp. 387 ff.

<sup>&</sup>lt;sup>51</sup> Cf. Aggarwal (2016), p. 40; cf. Deshpande/ Karypis (2004), pp. 143 ff; cf. Resnick, Iacovou, Suchak, Bergstrom and Riedl (1994), pp. 175 ff.

<sup>&</sup>lt;sup>52</sup> Cf. Aggarwal (2016), p. 40; cf. Ekstrand, Riedl and Konstan (2011), pp. 81 ff; cf. Linden, Smith and York (2003), pp. 76 ff.

<sup>&</sup>lt;sup>53</sup> Cf. Ahn (2008), pp. 38 f; cf. Khojamli/ Razmara (2021), pp. 2 f; cf. Sarwar, Karypis, Konstan and Riedl (2001), p. 288.

<sup>&</sup>lt;sup>54</sup> Cf. Aggarwal (2016), p. 40; cf. Khojamli/ Razmara (2021), pp. 2 f; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

ACOS neutralizes biases from users who consistently rate items higher or lower, providing more accurate measures.<sup>55</sup> The ACOS between items *i* and *j* is given by:

Adjusted Cosine Similarity (i, j) = 
$$\frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_u) \cdot (r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_u)^2} \cdot \sqrt{\sum_{u \in U_{ij}} (r_{uj} - \bar{r}_u)^2}}$$

where  $r_{ui}$  is the rating of user u for item i,  $\bar{r}_u$  is the average rating of user u, and  $U_{ij}$  is the set of users who have rated both items i and j.<sup>56</sup> Subsequently, the top-N most similar items are selected for each target item, forming its neighborhood.<sup>57</sup> The target item's rating is then predicted by aggregating the weighted ratings of its neighbors.<sup>58</sup> The predicted rating  $\hat{r}_{ui}$  for user u on item i can be calculated as:

$$\hat{r}_{ui} = \frac{\sum_{v \in N(u)} sim(u, v) \cdot (r_{vi} - \bar{r}_v)}{\sum_{v \in N(u)} sim(u, v)}$$

where N(i) is the set of items similar to item *i*.<sup>59</sup> Finally, a list of recommended items with the highest predicted ratings that the user has not interacted with is generated.<sup>60</sup>

#### 3.2.2 Strengths and Weaknesses of Item-Based Collaborative Filtering

One key strength of IBCF is its ability to produce accurate and relevant recommendations because items generally exhibit more stable relationships over time compared to the transient preferences of individual users.<sup>61</sup> Furthermore, IBCF reduces computational complexity, by focusing on a smaller set of similar items rather than a vast user pool, making it advantageous for large-scale applications like e-commerce and streaming services.<sup>62</sup> Precomputing and storing item similarities also enables efficient real-time recommendations.<sup>63</sup> IBCF is also robust to the sparsity problem common in recommendation scenarios, where users typically interact with only a few items.<sup>64</sup> By utilizing the dense item-item similarity matrix, IBCF can generate recommendations even with limited user data, enhancing its ability to serve new or infrequent users.<sup>65</sup> Lastly, IBCF offers enhanced interpretability and explainability

<sup>55</sup> Cf. Aggarwal (2016), pp. 29 ff.

<sup>&</sup>lt;sup>56</sup> Cf. Aggarwal (2016), p. 40.

<sup>&</sup>lt;sup>57</sup> Cf. Herlocker, Konstan, Borchers and Riedl (1999), pp. 230 ff; cf. Ning/ Karypis (2011), pp. 497 ff.

<sup>&</sup>lt;sup>58</sup> Cf. Aggarwal (2016), p. 40; cf. Sarwar, Karypis, Konstan and Riedl (2001), p. 289.

<sup>&</sup>lt;sup>59</sup> Cf. Aggarwal (2016), p. 40; cf. Sarwar, Karypis, Konstan and Riedl (2001), p. 289.

<sup>&</sup>lt;sup>60</sup> Cf. Linden, Smith and York (2003), pp. 76 ff.

<sup>&</sup>lt;sup>61</sup> Cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff; cf. Deshpande/ Karypis (2004), pp. 143 ff.

<sup>&</sup>lt;sup>62</sup> Cf. Deshpande/ Karypis (2004), pp. 143 ff.

<sup>63</sup> Cf. Karypis (2001), pp. 247 ff.

<sup>&</sup>lt;sup>64</sup> Cf. Desrosiers/ Karypis (2011), pp. 107 ff.

<sup>65</sup> Cf. Aggarwal (2016), pp. 42 ff; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

in recommendations.<sup>66</sup> By focusing on item similarities, it is easier to justify recommendations based on past user interactions, enhancing user trust and satisfaction.<sup>67</sup>

Despite its strengths, IBCF has inherent weaknesses that can impact its effectiveness in certain contexts. IBCF relies on the assumption that past item interaction can accurately predict future user preferences, which may not hold in dynamic environments where tastes evolve or novel items are introduced frequently, potentially leading to outdated recommendations.<sup>68</sup> Moreover, IBCF faces challenges with the cold start problem for novel items lacking sufficient ratings, as it relies on historical interaction data to compute item similarities.<sup>69</sup> This limitation hinders the discovery of novel items and leads to recommendations that are too similar to past user interactions, potentially reducing serendipity.<sup>70</sup> Furthermore, IBCF's focus on item-item similarities without directly accounting for individual user preferences may lead to less personalized recommendations for users with diverse and dynamic tastes compared to hybrid or user-based approaches.<sup>71</sup> Additionally, IBCF may lead to biased recommendations if certain items have disproportionately high ratings, potentially overshadowing more relevant options for users.<sup>72</sup>

In summary, memory-based CF methods offer powerful and intuitive approaches for generating personalized recommendations.<sup>73</sup> However, both methods face significant challenges, such as the popularity bias and the cold start problem, which need to be addressed for optimal performance.<sup>74</sup>

<sup>&</sup>lt;sup>66</sup> Cf. Linden, Smith and York (2003), pp. 76 ff.

cf. Herlocker, Konstan, Terveen and Riedl (2000), pp. 241 ff.

<sup>&</sup>lt;sup>68</sup> Cf. Aggarwal (2016), pp. 29 ff.

<sup>&</sup>lt;sup>69</sup> Cf. Ahn (2008), pp. 37 ff; cf. Lika, Kolomvatsos and Hadjiefthymiades (2014), pp. 2065 ff; cf. Schein, Popescul, Ungar and Pennock (2002), pp. 253 ff.

<sup>&</sup>lt;sup>70</sup> Cf. Celma (2010), pp. 3 ff.

<sup>&</sup>lt;sup>71</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff.

<sup>&</sup>lt;sup>72</sup> Cf. Abdollahpouri, Burke and Mobasher (2017), pp. 42 ff; cf. Linden, Smith and York (2003), pp. 76 ff.

<sup>&</sup>lt;sup>73</sup> Cf. Aggarwal (2016), pp. 29 ff; cf. Deshpande/ Karypis (2004), pp. 143 ff; cf. Linden, Smith and York (2003), pp. 76 ff.

<sup>&</sup>lt;sup>74</sup> Cf. Abdollahpouri, Burke and Mobasher (2017), pp. 42 ff; cf. Aggarwal (2016), pp. 44 ff; cf. Lika, Ko-lomvatsos and Hadjiefthymiades (2014), pp. 2065 ff.

#### 4 Model-based Collaborative Filtering

#### 4.1 Matrix Factorization in Collaborative Filtering

#### 4.1.1 Latent Factor Models

Model-based collaborative filtering (CF) is a sophisticated approach within recommender systems that leverages predictive models trained on historical UII data to forecast user preferences and generate recommendations.<sup>75</sup> Latent factor models (LFM) are a core component of model-based CF.<sup>76</sup> These models aim to discover the underlying factors that influence user preferences and item characteristics by decomposing the UIM into lower-dimensional representations.<sup>77</sup> These latent factors (LFs) capture hidden patterns and relationships between users and items, enabling the model to predict missing ratings and generate recommendations.<sup>78</sup> For example, in a movie RS, LFs might capture genres, directors, actors, or other abstract qualities that influence user preferences.<sup>79</sup> By learning these factors from the data, the model can make more accurate and personalized recommendations.<sup>80</sup> In LFM, each user and item is represented as a vector in a shared latent factor space.<sup>81</sup> The interaction between a user and an item is modeled as the dot product of their respective vectors.<sup>82</sup> This approach helps in uncovering the latent features that drive UIIs, such as a user's affinity for certain genres or an item's popularity among certain demographics.<sup>83</sup> The mathematical formulation of LFM typically involves MF techniques.84 Therefore, the original interaction matrix R (with dimensions  $n \times m$ , where m is the number of users and n is the number of items) is approximated by the product of two lower-dimensional matrices *P* and *Q*:

 $R \approx P \times Q^T$ 

<sup>&</sup>lt;sup>75</sup> Cf. Hofmann (2004), pp. 89 ff; cf. Shi, Larson and Hanjalic (2014), pp. 4 f.

<sup>&</sup>lt;sup>76</sup> Cf. Hofmann (2004), pp. 89 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 f.

<sup>&</sup>lt;sup>77</sup> Cf. Abdi, Okeyo and Mwangi (2018), pp. 1 ff; cf. Chen, Hua, Chang, Wang, Zhang and Kong (2018), pp. 64302 ff; cf. Shi, Larson and Hanjalic (2014), pp. 4 f.

<sup>&</sup>lt;sup>78</sup> Cf. Chen, Hua, Chang, Wang, Zhang and Kong (2018), pp. 64302 ff; cf. Hofmann (2004), pp. 89 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 ff; cf. Shi, Larson and Hanjalic (2014), pp. 1 ff.

<sup>&</sup>lt;sup>79</sup> Cf. Aggarwal/ Parthasarathy (2001), pp. 227 ff; cf. Mongia, Jhamb, Chouzenoux and Majumdar (2020), pp. 2 ff; cf. Shi, Larson and Hanjalic (2014), pp. 1 ff.

<sup>&</sup>lt;sup>80</sup> Cf. Mongia, Jhamb, Chouzenoux and Majumdar (2020), pp. 2 ff.

<sup>&</sup>lt;sup>81</sup> Cf. Hofmann (2004), pp. 89 ff.

<sup>&</sup>lt;sup>82</sup> Cf. Abdi, Okeyo and Mwangi (2018), pp. 1 ff; cf. Fang, Zhang, Hu, Xu, Yang and Liu (2020), pp. 73 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 ff.

<sup>&</sup>lt;sup>83</sup> Cf. Hofmann (2004), pp. 89 ff.

<sup>&</sup>lt;sup>84</sup> Cf. Abdi, Okeyo and Mwangi (2018), pp. 1 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 ff.

Here, *P* is an  $m \times k$  matrix where each row represents a user in the *k*-dimensional LF space, and *Q* is an  $n \times k$  matrix where each row represents an item in the same *k*-dimensional space.<sup>85</sup> The predicted interaction  $\hat{r}_{ui}$  between user *u* and item *i* is given by:

$$\hat{r}_{u,i} = P_u \cdot Q_i^T$$

where  $P_u$  is the vector of LFs for user u, and  $Q_i$  is the vector of LFs for item i.<sup>86</sup> LFM aim to minimize prediction error between actual and estimated interactions using loss functions like root mean squared error or mean absolute error.<sup>87</sup> To enhance prediction accuracy and model performance, optimization techniques like stochastic gradient descent (SGD) and alternating least squares (ALS) are used, as they iteratively update to find the optimal values of the feature vectors.<sup>88</sup>

#### 4.1.2 Matrix Factorization Techniques

Singular value decomposition (SVD) is a fundamental linear algebra technique that has been widely applied to MF for CF.<sup>89</sup> In the context of CF RS, SVD decomposes the user-item rating matrix (R) into three matrices:

$$R = U\Sigma V^T$$

where U represents user LFs,  $\Sigma$  is a diagonal matrix containing singular values, and V<sup>T</sup> represents item LFs.<sup>90</sup> This decomposition helps in capturing the most critical latent factors that explain the observed interactions.<sup>91</sup> The diagonal entries in  $\Sigma$  represent the singular values that denote the strength of each latent factor.<sup>92</sup> By truncating  $\Sigma$  to keep only the top *k* singular values, SVD effectively reduces the dimensionality of the problem, while preserving essential information about user preferences and item characteristics.<sup>93</sup> The predicted rating for a user-item pair can be computed as the dot product of the corresponding row in U and column in V<sup>T</sup>, weighted by the corresponding singular value in  $\Sigma$ .<sup>94</sup> This approach not only enhances

<sup>&</sup>lt;sup>85</sup> Cf. Abdi, Okeyo and Mwangi (2018), pp. 1 ff; cf. Fang, Zhang, Hu, Xu, Yang and Liu (2020), pp. 73 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 ff.

<sup>86</sup> Cf. Hofmann (2004), pp. 89 ff.

<sup>&</sup>lt;sup>87</sup> Cf. Chen, Hua, Chang, Wang, Zhang and Kong (2018), pp. 64306 f; cf. Fang, Zhang, Hu, Xu, Yang and Liu (2020), pp. 73 ff.

<sup>&</sup>lt;sup>88</sup> Cf. Koren, Bell and Volinsky (2009), pp. 32 f.

<sup>&</sup>lt;sup>89</sup> Cf. Koren (2008), p. 427; cf. Sarwar, Karypis, Konstan and Riedl (2000), pp. 82 ff.

<sup>&</sup>lt;sup>90</sup> Cf. Aggarwal (2016), pp. 29 ff.

<sup>&</sup>lt;sup>91</sup> Cf. Koren (2008), p. 426; cf. Sarwar, Karypis, Konstan and Riedl (2000), pp. 82 ff.

<sup>&</sup>lt;sup>92</sup> Cf. Koren (2008), p. 426; cf. Sarwar, Karypis, Konstan and Riedl (2000), pp. 82 ff.

<sup>&</sup>lt;sup>93</sup> Cf. Koren (2008), p. 426; cf. Sarwar, Karypis, Konstan and Riedl (2000), pp. 82 ff.

<sup>&</sup>lt;sup>94</sup> Cf. Koren, Bell and Volinsky (2009), pp. 32 f.

computational efficiency but also improves the robustness of the recommendations by mitigating overfitting.<sup>95</sup> While SVD offers an elegant solution for MF, its computational complexity can be prohibitive for large datasets.<sup>96</sup>

Non-negative MF (NMF) is another popular technique that imposes a non-negativity constraint on the factorized matrices.<sup>97</sup> This constraint aligns with the intuitive notion that user preferences and item attributes are inherently positive.<sup>98</sup> NMF often yields more interpretable LFs, as they represent additive combinations of user or item attributes, which can be valuable in understanding the underlying reasons behind recommendations.<sup>99</sup>

#### 4.1.3 Strengths and Weaknesses of Matrix Factorization

MF boasts key strengths that have solidified its position as a dominant technique in CF: Due to dimensionality reduction, MF techniques elegantly tackle the issue of sparse interaction data.<sup>100</sup> By uncovering LFs that generalize across users and items, MF enables the model to make informed predictions even for interactions that have not been explicitly observed, leading to more comprehensive and personalized recommendations.<sup>101</sup> Additionally, MF algorithms exhibit impressive scalability to massive datasets.<sup>102</sup> This scalability is essential in real-world RS that often deal with millions of users and items.<sup>103</sup> The ability to handle such scale allows MF-based systems to provide timely and relevant recommendations without compromising on computational efficiency.<sup>104</sup> Moreover, the LFs extracted through MF often carry meaningful interpretations.<sup>105</sup> For instance, in a movie RS, LFs might correspond to genres, actors, or directors.<sup>106</sup>. This interpretability not only enhances the transparency of recommendations but also enables the incorporation of domain knowledge to further refine

<sup>95</sup> Cf. Koren (2008), p. 426; cf. Sarwar, Karypis, Konstan and Riedl (2000), pp. 82 ff.

<sup>&</sup>lt;sup>96</sup> Cf. Koren (2008), p. 426.

<sup>97</sup> Cf. Lee/ Seung (1999), pp. 788 ff.

<sup>98</sup> Cf. Lee/ Seung (1999), pp. 788 ff. Cf. Hofmann (2004), pp. 89 ff

<sup>&</sup>lt;sup>99</sup> Cf. Hofmann (2004), pp. 89 ff.

<sup>&</sup>lt;sup>100</sup> Cf. Koren (2008), p. 427; cf. Sarwar, Karypis, Konstan and Riedl (2000), pp. 82 ff.

<sup>&</sup>lt;sup>101</sup> Cf. Abdi, Okeyo and Mwangi (2018), pp. 1 ff; cf. Koren (2008), p. 427; cf. Koren, Bell and Volinsky (2009), pp. 30 ff.

<sup>&</sup>lt;sup>102</sup> Cf. Koren (2008), p. 427; cf. Zhou, Wilkinson, Schreiber and Pan (2008), pp. 337 ff.

<sup>&</sup>lt;sup>103</sup> Cf. Bell/ Koren (2007), pp. 75 ff; cf. Koren (2008), p. 427; cf. Zhou, Wilkinson, Schreiber and Pan (2008), pp. 337 ff.

<sup>&</sup>lt;sup>104</sup> Cf. Koren, Bell and Volinsky (2009), pp. 30 ff; cf. Yu, Hsieh, Si and Dhillon (2014), pp. 793 ff.

<sup>&</sup>lt;sup>105</sup> Cf. Koren, Bell and Volinsky (2009), pp. 30 ff.

<sup>&</sup>lt;sup>106</sup> Cf. Mongia, Jhamb, Chouzenoux and Majumdar (2020), pp. 2 ff.

the model.<sup>107</sup> MF frameworks also offer a high degree of flexibility, allowing for the incorporation of various regularization techniques, bias terms, and even temporal dynamics.<sup>108</sup> This adaptability enables researchers and practitioners to tailor MF models to specific domains and applications, ensuring optimal performance and relevance.<sup>109</sup>

A primary challenge for MF-based RS is the cold-start problem.<sup>110</sup> This situation arises when new users or items lack interaction history, making it difficult for the model to generate accurate recommendations, which can lead to suboptimal user experiences and hinder the adoption of new items.<sup>111</sup> Additionally, MF models heavily depend on the choice of hyperparameters, such as the number of LFs, regularization parameters, and learning rate. Improper tuning can result in overfitting or underfitting, affecting the model's accuracy and generalization capabilities.<sup>112</sup> Although scalable, MF techniques can be computationally intensive, especially with very large datasets or when frequent model retraining is required.<sup>113</sup> This complexity can pose challenges in real-time recommendation scenarios.<sup>114</sup> Unlike simpler models with transparent rationales, MF's LFs do not have clear, intuitive meanings, hindering the ability to explain recommendations to users, which is crucial for user trust and satisfaction.<sup>115</sup> MF-based systems may also amplify popularity bias, where popular items receive disproportionately more recommendations than less popular ones.<sup>116</sup> This can reduce diversity in recommendations and limit users' exposure to potentially relevant but lessknown items.<sup>117</sup>

<sup>&</sup>lt;sup>107</sup> Cf. Koren (2009), p. 451, pp. 447 ff; cf. Lin Qiu, Sheng Gao, Qinjie Lyu, Jun Guo and Patrick Gallinari (2018), pp. 144 ff; cf. Mongia, Jhamb, Chouzenoux and Majumdar (2020), p. 3.

<sup>&</sup>lt;sup>108</sup> Cf. Abdi, Okeyo and Mwangi (2018), pp. 1 ff; cf. Koren (2009), p. 451; cf. Mongia, Jhamb, Chouzenoux and Majumdar (2020), p. 3.

<sup>&</sup>lt;sup>109</sup> Cf. Koren (2009), p. 451; cf. Rendle (2012), p. 3.

<sup>&</sup>lt;sup>110</sup> Cf. Ahn (2008), pp. 37 ff; cf. Lika, Kolomvatsos and Hadjiefthymiades (2014), pp. 2065 ff.

<sup>&</sup>lt;sup>111</sup> Cf. Abdi, Okeyo and Mwangi (2018), pp. 1 ff; cf. Schein, Popescul, Ungar and Pennock (2002), pp. 253 ff. <sup>112</sup> Cf. Konstan/ Riedl (2012), pp. 104 f; cf. Koren, Bell and Volinsky (2009), pp. 30 ff; cf. Rendle (2012), p.

<sup>&</sup>lt;sup>113</sup> Cf. Chen, Hua, Chang, Wang, Zhang and Kong (2018), pp. 64305 f; cf. Konstan/ Riedl (2012), pp. 104 f; cf. Rendle (2012), p. 7.

<sup>&</sup>lt;sup>114</sup> Cf. Chen, Hua, Chang, Wang, Zhang and Kong (2018), pp. 64305 f; cf. Konstan/ Riedl (2012), pp. 104 f; cf. Zhou, Wilkinson, Schreiber and Pan (2008), p. 340.

<sup>&</sup>lt;sup>115</sup> Cf. Chen, Hua, Chang, Wang, Zhang and Kong (2018), p. 64307; cf. Konstan/ Riedl (2012), p. 102; cf. Koren (2009), p. 453.

<sup>&</sup>lt;sup>116</sup> Cf. Abdollahpouri, Burke and Mobasher (2017), pp. 42 ff; cf. Chen, Hua, Chang, Wang, Zhang and Kong (2018), p. 64307.

<sup>&</sup>lt;sup>117</sup> Cf. Abdollahpouri, Burke and Mobasher (2017), pp. 42 ff; cf. Chen, Hua, Chang, Wang, Zhang and Kong (2018), p. 64307.

#### 4.2 Probabilistic Graphical Models

#### 4.2.1 Concept of Probabilistic Graphical Models

PGMs provide a robust framework for modeling complex relationships between users, items, and their interactions.<sup>118</sup> By leveraging probabilistic reasoning and graph theory, PGMs offer a compact representation of dependencies among random variables.<sup>119</sup> A notable application of PGMs in CF is the use of Bayesian networks and Markov networks to model user preferences and item relationships.<sup>120</sup> These models are particularly powerful at handling sparse data and incorporating various sources of uncertainty, making them suitable for complex recommendation tasks.<sup>121</sup> Two notable PGMs used in CF are Bayesian personalized ranking (BPR) and restricted Boltzmann machines (RBM).<sup>122</sup>

BPR is an optimization criterion designed specifically to address the ranking problem in CF.<sup>123</sup> Unlike traditional MF techniques that predict explicit ratings, BPR focuses on optimizing the ranking of items, which is often more indicative of user preferences.<sup>124</sup> BPR operates on the principle of pairwise ranking, optimizing the recommendation model to rank a user's preferred items higher than non-preferred items.<sup>125</sup> It uses a Bayesian framework to incorporate prior knowledge and generate personalized rankings by maximizing the posterior probability.<sup>126</sup> BPR utilizes SGD for optimization, making it scalable to large datasets, therefore improving recommendation accuracy, especially in scenarios involving implicit feedback.<sup>127</sup>

RBMs are another class of probabilistic graphical models effectively applied to CF.<sup>128</sup> RBMs are generative stochastic neural networks that can learn a probability distribution over a set of inputs.<sup>129</sup> In CF, RBMs model the distribution of user ratings, capturing complex patterns and dependencies in the data.<sup>130</sup> An RBM consists of visible and hidden layers, where visible

<sup>&</sup>lt;sup>118</sup> Cf. Jordan (2003), pp. 3 f; cf. Koller/ Friedman (2009), pp. 3 f.

<sup>&</sup>lt;sup>119</sup> Cf. Koller/ Friedman (2009), pp. 3 f.

<sup>&</sup>lt;sup>120</sup> Cf. Friedman, Geiger and Goldszmidt (1997), pp. 131 ff; cf. Koller/ Friedman (2009), pp. 134 ff.

<sup>&</sup>lt;sup>121</sup> Cf. Jordan (2003), pp. 8 ff; cf. Zhou, Wilkinson, Schreiber and Pan (2008), pp. 337 ff.

<sup>&</sup>lt;sup>122</sup> Cf. Hinton (2012), pp. 599 ff; cf. Rendle, Freudenthaler, Gantner and Schmidt-Thieme (2009), pp. 452 ff.

<sup>&</sup>lt;sup>123</sup> Cf. Lee, Kim, Choi, He and Kim (2021), pp. 255 ff.

<sup>&</sup>lt;sup>124</sup> Cf. Rendle, Freudenthaler, Gantner and Schmidt-Thieme (2009), pp. 452 ff.

<sup>&</sup>lt;sup>125</sup> Cf. Rendle, Freudenthaler, Gantner and Schmidt-Thieme (2009), pp. 452 ff.

<sup>&</sup>lt;sup>126</sup> Cf. Lee, Kim, Choi, He and Kim (2021), pp. 255 ff; cf. Papadakis, Ioannou, Palpanas, Niederee and Koutrika (2022), p. 54.

<sup>&</sup>lt;sup>127</sup> Cf. Hu, Koren and Volinsky (2008), pp. 263 ff; cf. Rendle, Freudenthaler, Gantner and Schmidt-Thieme (2009), pp. 452 ff.

<sup>&</sup>lt;sup>128</sup> Cf. Papadakis, Ioannou, Palpanas, Niederee and Koutrika (2022), pp. 50 ff.

<sup>&</sup>lt;sup>129</sup> Cf. Salakhutdinov, Mnih and Hinton (2007), pp. 791 ff.

<sup>&</sup>lt;sup>130</sup> Cf. Salakhutdinov, Mnih and Hinton (2007), pp. 791 ff.

units represent UII data and hidden units capture LF.<sup>131</sup> The interaction between these layers is governed by a set of weights, learned during training.<sup>132</sup> Once trained, the RBM can infer the hidden representation of users and items, which can then be used to predict missing ratings and generate recommendations.<sup>133</sup> RBMs have demonstrated their efficacy in various recommendation tasks due to their ability to model non-linear relationships and handle high-dimensional data.<sup>134</sup>

#### 4.2.2 Strengths and Weaknesses of Probabilistic Graphical Models

PGMs excel in representing a wide range of dependencies among variables, making them suitable for capturing intricate relationships between users and items.<sup>135</sup> Additionally, PGMs inherently incorporate probabilistic reasoning, allowing them to manage uncertainty effectively.<sup>136</sup> This feature is particularly useful in CF, where user preferences and item attributes are often uncertain or partially observed.<sup>137</sup> Techniques such as variational inference and Markov chain Monte Carlo methods enable PGMs to process large-scale CF tasks without significant computational overhead.<sup>138</sup> Furthermore, PGMs allow for the integration of domain knowledge into the model through the structure of the graph and the choice of conditional dependencies.<sup>139</sup> This flexibility can enhance the accuracy and interpretability of the recommendations.<sup>140</sup> Finally, PGMs can make robust inferences, even in sparse datasets, to understand user preferences and item characteristics by modeling the dependencies between observed and unobserved variables probabilistically.<sup>141</sup>

<sup>&</sup>lt;sup>131</sup> Cf. Salakhutdinov, Mnih and Hinton (2007), pp. 791 ff. Cf. Murphy (2012), pp. 987 ff

<sup>&</sup>lt;sup>132</sup> Cf. Hinton (2012), pp. 599 ff; cf. Murphy (2012), pp. 987 ff.

<sup>&</sup>lt;sup>133</sup> Cf. Hinton (2012), pp. 599 ff; cf. Murphy (2012), pp. 987 ff.

<sup>&</sup>lt;sup>134</sup> Cf. Hinton (2012), pp. 599 ff; cf. Salakhutdinov, Mnih and Hinton (2007), pp. 791 ff.

<sup>&</sup>lt;sup>135</sup> Cf. Jordan (2003), pp. 12 ff.

<sup>&</sup>lt;sup>136</sup> Cf. Das, Datar, Garg and Rajaram (2007), pp. 271 ff.

<sup>&</sup>lt;sup>137</sup> Cf. Zhou, Wilkinson, Schreiber and Pan (2008), pp. 337 ff.

<sup>&</sup>lt;sup>138</sup> Cf. Blei, Kucukelbir and McAuliffe (2017), pp. 859 ff; cf. Koller/ Friedman (2009), pp. 134 ff; cf. Zhou, Wilkinson, Schreiber and Pan (2008), pp. 337 ff.

<sup>&</sup>lt;sup>139</sup> Cf. Koller/ Friedman (2009), pp. 3 f.

<sup>&</sup>lt;sup>140</sup> Cf. Friedman, Geiger and Goldszmidt (1997), pp. 131 ff; cf. Rendle, Freudenthaler, Gantner and Schmidt-Thieme (2009), pp. 452 ff.

<sup>&</sup>lt;sup>141</sup> Cf. Friedman, Geiger and Goldszmidt (1997), pp. 131 ff; cf. Murphy (2012), pp. 987 ff; cf. Jing, Liang, Lin and Tsao (2014), pp. 250 ff.

One of the main drawbacks of PGMs is their computational complexity.<sup>142</sup> The inference and learning processes, especially in large-scale applications, can be intensive and time-consuming, posing challenges in environments requiring real-time recommendations.<sup>143</sup> Additionally, designing an appropriate PGM involves selecting the right model structure and estimating numerous parameters, demanding substantial expertise and time.<sup>144</sup> Furthermore, PGMs require a large amount of training data to accurately estimate conditional probability distributions and learn the network structure.<sup>145</sup> With limited data, models may overfit or fail to capture underlying dependencies accurately.<sup>146</sup> While PGMs can incorporate domain knowledge, the resulting models may be difficult to interpret, especially when involving many variables and complex dependencies, hindering the ability to explain recommendations to users.<sup>147</sup>

#### 5 Concluding Remarks

#### 5.1 Summary

This literacy paper has conducted a comprehensive analysis of CF RS, focusing on both memory-based and model-based techniques. The main findings highlight the distinct advantages and limitations of each approach in addressing real-world constraints. Memory-based CF methods, such as UBCF and IBCF, are straightforward to implement and interpret.<sup>148</sup> They excel in environments with dense UII data, providing high-quality recommendations by leveraging direct user- or item similarities.<sup>149</sup> However, these methods face significant challenges related to popularity bias, scalability, data sparsity, and the cold-start problem, which can limit their effectiveness in larger or more dynamic systems.<sup>150</sup> Model-based CF techniques, including MF and PGMs, offer robust solutions to certain limitations

<sup>&</sup>lt;sup>142</sup> Cf. Das, Datar, Garg and Rajaram (2007), p. 277; cf. Zhou, Wilkinson, Schreiber and Pan (2008), pp. 337 ff.

<sup>&</sup>lt;sup>143</sup> Cf. Das, Datar, Garg and Rajaram (2007), p. 277; cf. Zhou, Wilkinson, Schreiber and Pan (2008), pp. 337 ff. <sup>144</sup> Cf. Helge Langseth/ Thomas D. Nielsen (2015), pp. 1 ff; cf. Murphy (2012), pp. 987 ff.

<sup>&</sup>lt;sup>145</sup> Cf. Rendle, Freudenthaler, Gantner and Schmidt-Thieme (2009), pp. 452 ff; cf. Salakhutdinov, Mnih and Hinton (2007), pp. 791 ff.

<sup>&</sup>lt;sup>146</sup> Cf. Helge Langseth/ Thomas D. Nielsen (2015), pp. 1 ff; cf. Li/ Yeung (2011), pp. 803 ff.

<sup>&</sup>lt;sup>147</sup> Cf. Heckerman, Geiger and Chickering (1995), pp. 197 ff; cf. Hinton (2012), pp. 599 ff.

<sup>&</sup>lt;sup>148</sup> Cf. Aggarwal (2016), pp. 29 ff; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

<sup>&</sup>lt;sup>149</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff; cf. Aggarwal (2016), pp. 29 ff; cf. Sarwar, Karypis, Konstan and Riedl (2001), pp. 285 ff.

<sup>&</sup>lt;sup>150</sup> Cf. Abdollahpouri, Burke and Mobasher (2017), pp. 42 ff; cf. Aggarwal (2016), pp. 44 ff; cf. Schein, Popescul, Ungar and Pennock (2002), pp. 253 ff; cf. Singh (2020), pp. 15 ff.

of memory-based methods.<sup>151</sup> These techniques are highly scalable and capable of uncovering LFs that drive user preferences, thus providing more personalized recommendations.<sup>152</sup> Despite these strengths, model-based methods can be computationally intensive and require careful tuning of hyperparameters.<sup>153</sup> Additionally, their complexity can hinder interpretability, making it difficult to explain recommendations to end-users.<sup>154</sup> In answering the research question, it is evident that while memory-based CF methods provide simplicity<sup>155</sup> and direct interpretability<sup>156</sup>, their scalability and performance are often compromised in sparse and dynamic environments.<sup>157</sup> Conversely, model-based methods, although more complex and computationally demanding, offer greater scalability and robustness to sparsity, making them more suitable for large-scale applications.<sup>158</sup> The practical implications of these findings suggest that the choice of the CF method should be guided by the specific constraints and requirements of the application domain. For instance, memory-based methods might be preferable for small- to medium-sized systems with rich interaction data, while model-based methods are better suited for large-scale platforms requiring high levels of personalization and adaptability.

#### 5.2 Critical Appraisal and Outlook

While this paper provides valuable insights, it is important to acknowledge its limitations. The reliance on secondary sources and the lack of empirical validation may affect the generalizability of the findings. Future research should aim to address these limitations through empirical studies that evaluate the conclusions in real-world settings. Specific directions for future research include conducting empirical studies to validate the theoretical insights presented in this paper, exploring new or hybrid CF methods that combine the strengths of both

<sup>&</sup>lt;sup>151</sup> Cf. Ahn (2008), pp. 37 ff; cf. Herlocker, Konstan, Terveen and Riedl (2004), pp. 5 ff; cf. Schein, Popescul, Ungar and Pennock (2002), pp. 253 ff; cf. Singh (2020), pp. 15 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 ff.

<sup>&</sup>lt;sup>152</sup> Cf. Helge Langseth/ Thomas D. Nielsen (2015), pp. 1 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 ff; cf. Shi, Larson and Hanjalic (2014), pp. 1 ff.

<sup>&</sup>lt;sup>153</sup> Cf. Konstan/ Riedl (2012), pp. 104 f; cf. Koren, Bell and Volinsky (2009), pp. 30 ff; cf. Rendle (2012), p. 5.

<sup>&</sup>lt;sup>154</sup> Cf. Konstan/ Riedl (2012), pp. 101 ff; cf. Koren, Bell and Volinsky (2009), pp. 30 ff; cf. Salakhutdinov, Mnih and Hinton (2007), pp. 791 ff.

<sup>&</sup>lt;sup>155</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff; cf. Aggarwal (2016), p. 44; cf. Herlocker, Konstan, Borchers and Riedl (1999), pp. 230 ff; cf. Walker, Glance and Rodriguez (2004), pp. 365 ff.

<sup>&</sup>lt;sup>156</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff; cf. Herlocker, Konstan, Borchers and Riedl (1999), pp. 230 ff; cf. Linden, Smith and York (2003), pp. 76 ff.

<sup>&</sup>lt;sup>157</sup> Cf. Aggarwal (2016), p. 44; cf. Ahn (2008), pp. 37 ff; cf. Breese, Heckerman and Kadie (1998), pp. 43 ff; cf. Schein, Popescul, Ungar and Pennock (2002), pp. 253 ff; cf. Singh (2020), pp. 15 ff.

<sup>&</sup>lt;sup>158</sup> Cf. Das, Datar, Garg and Rajaram (2007), pp. 271 ff; cf. Jing, Liang, Lin and Tsao (2014), pp. 250 ff; cf. Koren (2008), p. 427; cf. Sarwar, Karypis, Konstan and Riedl (2000), pp. 82 ff.

memory-based and model-based approaches, and investigating the application of CF techniques in specific domains such as healthcare, education, and social media.<sup>159</sup> The potential impact of these proposed research directions on the field of RS is significant. Empirical validation can strengthen the theoretical foundations and provide actionable insights for practitioners. Exploring specific application domains can lead to tailored CF solutions that address unique challenges and opportunities within those fields.

<sup>&</sup>lt;sup>159</sup> Cf. Adomavicius/ Tuzhilin (2005), pp. 734 ff; cf. Bell/ Koren (2007), pp. 75 ff.

#### References

Abdi, M. H., Okeyo, G. and Mwangi, R. (2018): Matrix Factorization Techniques for Context-Aware Collaborative Filtering Recommender Systems – A Survey. In: *Computer and Information Science*, Vol. 11 No. 2, 2018, pp. 1–10.

Abdollahpouri, H., Burke, R. and Mobasher, B. (2017): Controlling popularity bias in learning-to-rank recommendation. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 42–46.

Adomavicius, G. and Tuzhilin, A. (2005): Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. In: *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17 No. 6, 2005, pp. 734–749.

Adomavicius, G. and Kwon, Y. (2007): New recommendation techniques for multicriteria rating systems. In: *IEEE Intelligent Systems*, Vol. 22 No. 3, 2007, pp. 48–55.

Aggarwal, C. and Parthasarathy, S. (2001): Mining massively incomplete data sets by conceptual reconstruction. In *ACM KDD Conference*, pp. 227–232.

Aggarwal, C. C. (2016): Recommender Systems – The Textbook. Springer 2016.

Ahn, H. J. (2008): A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. In: *Information Sciences*, Vol. 178 No. 1, 2008, pp. 37–51.

Bell, R. and Koren, Y. (2007): Lessons from the Netflix prize challenge. In: *ACM SIGKDD Explorations Newsletter*, Vol. 9 No. 2, 2007, pp. 75–79.

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017): Variational inference – A review for statisticians. In: *Journal of the American Statistical Association*, Vol. 112 No. 518, 2017, pp. 859–877.

Bobadilla, J., Ortega, F., Hernando, A. and Gutiérrez, A. (2013): Recommender systems survey. In: *Knowledge-Based Systems*, Vol. 46, 2013, pp. 109–132.

Breese, J. S., Heckerman, D. and Kadie, C. (1998): Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43–52.

Celma, Ò. (2010): Music recommendation and discovery. Springer 2010.

Chen, R., Hua, Q., Chang, Y.-S., Wang, B., Zhang, L. and Kong, X. (2018): A Survey of Collaborative Filtering-Based Recommender Systems – From Traditional Methods to Hybrid Methods Based on Social Networks. In: *IEEE Access*, Vol. 6, 2018, pp. 64301–64320.

Das, A. S., Datar, M., Garg, A. and Rajaram, S. (2007): Google news personalization – Scalable online collaborative filtering. In: *Proceedings of the 16th International Conference on World Wide Web*, pp. 271–280.

Deshpande, M. and Karypis, G. (2004): Item-based top-N recommendation algorithms. In: *ACM Transactions on Information Systems*, Vol. 22 No. 1, 2004, pp. 143–177.

Desrosiers, C. and Karypis, G. (2011): A Comprehensive Survey of Neighborhoodbased Recommendation Methods. In: Ricci F., Rokach L., Shapira B., Kantor P. B. (Eds.): Recommender Systems Handbook, Boston Springer 2011, pp. 107–144.

Ekstrand, M. D., Riedl, J. T. and Konstan, J. A. (2011): Collaborative filtering recommender systems. In: *Foundations and Trends in Human-Computer Interaction*, Vol. 4 No. 2, 2011, pp. 81–173.

Fang, J., Zhang, X., Hu, Y., Xu, Y., Yang, M.-H. and Liu, J. (2020): Probabilistic Latent Factor Model for Collaborative Filtering with Bayesian Inference. In 2020 25th International Conference on Pattern Recognition, pp. 73–80.

Friedman, N., Geiger, D. and Goldszmidt, M. (1997): Bayesian network classifiers. In: *Machine Learning*, Vol. 29 No. 2-3, 1997, pp. 131–163.

Goldberg, D., Nichols, D., Oki, B. M. and Terry, D. (1992): Using collaborative filtering to weave an information tapestry. In: *Communications of the ACM*, Vol. 35 No. 12, 1992, pp. 61–70.

Heckerman, D., Geiger, D. and Chickering, D. M. (1995): Learning Bayesian networks: The combination of knowledge and statistical data. In: *Machine Learning*, Vol. 20 No. 3, 1995, pp. 197–243.

Helge Langseth and Thomas D. Nielsen (2015): Scalable learning of probabilistic latent models for collaborative filtering. In: *Decision Support Systems*, Vol. 74 2015, pp. 1–11.

Herlocker, J. L., Konstan, J. A., Borchers, A. and Riedl, J. (1999): An algorithmic framework for performing collaborative filtering. In: *Proceedings of the 22nd Annual* 

International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 230–237.

Herlocker, J. L., Konstan, J. A., Terveen, L. G. and Riedl, J. T. (2000): Explaining collaborative filtering recommendations. In: *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, pp. 241–250.

Herlocker, J. L., Konstan, J. A., Terveen, L. G. and Riedl, J. T. (2004): Evaluating collaborative filtering recommender systems. In: *ACM Transactions on Information Systems*, Vol. 22 No. 1, 2004, pp. 5–53.

Hinton, G. E. (2012): A practical guide to training restricted Boltzmann machines. In: Montavon, G., Orr, G.B., Müller, K. R. (Eds.): Neural Networks – Tricks of the Trade, Berlin Heidelberg Springer 2012, pp. 599–619.

Hofmann, T. (2004): Latent semantic models for collaborative filtering. In: *ACM Transactions on Information Systems*, Vol. 22 No. 1, 2004, pp. 89–115.

Hu, Y., Koren, Y. and Volinsky, C. (2008): Collaborative filtering for implicit feedback datasets. In: 2008 8th IEEE international conference on data mining, pp. 263–272.

Jannach, D., Zanker, M. and Felderer, M. (2010): Recommender systems – An introduction. Cambridge University Press 2010.

Jing, H., Liang, A.-C., Lin, S.-D. and Tsao, Y. (2014): A Transfer Probabilistic Collective Factorization Model to Handle Sparse Data in Collaborative Filtering. In *2014 IEEE International Conference on Data Mining*, pp. 250–259.

Jordan, M. I. (2003): An Introduction to Probabilistic Graphical Models. Berkeley University of California 2003.

Karypis, G. (2001): Evaluation of item-based top-N recommendation algorithms. In: *Proceedings of the 10th International Conference on Information and Knowledge Management*, pp. 247–254.

Khojamli, H. and Razmara, J. (2021): Survey of similarity functions on neighborhoodbased collaborative filtering. In: *Expert Systems with Applications*, Vol. 185, 2021, pp. 1–28.

Koller, D. and Friedman, N. (2009): Probabilistic Graphical Models – Principles and Techniques. MIT press 2009.

Konstan, J. A. and Riedl, J. (2012): Recommender systems: From algorithms to user experience. In: *User Modeling and User-Adapted Interaction*, Vol. 22 No. 1-2 2012, pp. 101–123.

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. (2009): Collaborative filtering with temporal dynamics. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 447–456.

Koren, Y., Bell, R. and Volinsky, C. (2009): Matrix Factorization Techniques for Recommender Systems. In: *IEEE Computer*, Vol. 42 No. 8, 2009, pp. 30–37.

Koren, Y. and Bell, R. (2015): Advances in collaborative filtering. In: Ricci F., Rokach L., Shapira B., Kantor P. B. (Eds.): Recommender Systems Handbook, Boston Springer 2015, pp. 77–118.

Lee, D. D. and Seung, H. S. (1999): Learning the parts of objects by non-negative matrix factorization. In: *Nature*, Vol. 401 No. 6755, 1999, pp. 788–791.

Lee, Y.-C., Kim, T., Choi, J., He, X. and Kim, S.-W. (2021): M-BPR: A novel approach to improving BPR for recommendation with multi-type pair-wise preferences. In: *Information Sciences*, Vol. 547, 2021, pp. 255–270.

Lika, B., Kolomvatsos, K. and Hadjiefthymiades, S. (2014): Facing the cold start problem in recommender systems. In: *Expert Systems with Applications*, Vol. 41 No. 4, 2014, pp. 2065–2073.

Qiu, L., Gao, S., Lyu, Q., Guo, J., and Gallinari, P. (2018): A novel non-Gaussian embedding based model for recommender systems. In: *Neurocomputing*, Vol. 278, 2018, pp. 144–152.

Linden, G., Smith, B. and York, J. (2003): Amazon.com recommendations: Item-toitem collaborative filtering. In: *IEEE Internet Computing*, Vol. 7 No. 1, 2003, pp. 76– 80.

Li, W.-J. and Yeung, D.-Y. (2011): Social Relations Model for Collaborative Filtering. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 25 No. 1, 2011, pp. 803–808. Lops, P., Gemmis, M. and Semeraro, G. (2011): Content-based recommender systems – State of the art and trends. In: Ricci F., Rokach L., Shapira B., Kantor P. B. (Eds.): Recommender Systems Handbook, Boston Springer 2011, pp. 73–105.

Lu, J., Wu, D., Mao, M., Wang, W. and Zhang, G. (2015): Recommender system application developments: A survey. In: *Decision Support Systems*, Vol. 74, 2015, pp. 12–32.

Mongia, A., Jhamb, N., Chouzenoux, E. and Majumdar, A. (2020): Deep latent factor model for collaborative filtering. In: *Signal Processing*, Vol. 169, 2020, pp. 1–8.

Murphy, K. P. (2012): Machine Learning – A Probabilistic Perspective, Boston MIT Press 2012.

Ning, X. and Karypis, G. (2011): SLIM – Sparse Linear Methods for Top-N Recommender Systems. In: 2011 IEEE 11th International Conference on Data Mining, pp. 497–506.

Papadakis, G., Ioannou, E., Palpanas, T., Niederee, C. and Koutrika, G. (2022): The four generations of entity resolution. In: *Data Engineering Bulletin*, Vol. 45 No. 2, 2022, pp. 40–61.

Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L. (2009): BPR – Bayesian personalized ranking from implicit feedback. In: *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 452–461.

Rendle, S. (2012): Factorization Machines with libFM. In: *ACM Transactions on Intelligent Systems and Technology*, Vol. 3 No. 3, 2012, pp. 1–22.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994): GroupLens – An open architecture for collaborative filtering of netnews. In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pp. 175–186.

Resnick, P. and Varian, H. R. (1997): Recommender systems. In: *Communications of the ACM*, Vol. 40 No. 3, 1997, pp. 56–58.

Ricci, F., Rokach, L. and Shapira, B. (2011): Introduction to recommender systems handbook. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. B. (Eds.): Recommender Systems Handbook, Boston Springer 2011, pp. 1–35.

Salakhutdinov, R., Mnih, A. and Hinton, G. (2007): Restricted Boltzmann machines for collaborative filtering. In: *Proceedings of the 24th International Conference on Machine Learning*, pp. 791–798.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2000): Application of dimensionality reduction in recommender system—a case study. In: *Proceedings of the ACM WebKDD Workshop*, pp. 82–90.

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001): Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295.

Schafer, J. B., Frankowski, D., Herlocker, J. and Sen, S. (2007): Collaborative Filtering Recommender Systems. In: Brusilovsky P., Kobsa A., Nejdl W. (Eds.): The Adaptive Web, Berlin Heidelberg Springer 2007, pp. 291–324.

Schein, A. I., Popescul, A., Ungar, L. H. and Pennock, D. M. (2002): Methods and metrics for cold-start recommendations. In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 253–260.

Shi, Y., Larson, M. and Hanjalic, A. (2014): Collaborative filtering beyond the useritem matrix – A survey of the state of the art and future challenges. In: *ACM Computing Surveys*, Vol. 47 No. 1, 2014, pp. 1–45.

Singh, A. P. (2020): Handling sparsity problem in recommender systems using association retrieval approach. In: *International Journal of Computer Sciences and Engineering*, Vol. 8 No. 5, 2020, pp. 15–20.

Su, X. and Khoshgoftaar, T. M. (2009): A survey of collaborative filtering techniques. In: *Advances in Artificial Intelligence* 2009, pp. 1–19.

Walker, K., Glance, N. S. and Rodriguez, M. A. (2004): Scalable collaborative filtering algorithms for mining large data sets. In: *Journal of Computer Science and Technology*, Vol. 19 No. 4, 2004, pp. 365–375.

Webster, J. and Watson, R. T. (2002): Analyzing the past to prepare for the future – Writing a literature review. In: *MIS Quarterly*, Vol. 26 No. 2, 2002, pp. 13–23.

Yu, H. F., Hsieh, C. J., Si, S. and Dhillon, I. S. (2014): Parallel matrix factorization for recommender systems. In: *Knowledge and Information Systems*, Vol. 41, 2014, pp. 793–819.

Zhou, Y., Wilkinson, D., Schreiber, R. and Pan, R. (2008): Large-Scale Parallel Collaborative Filtering for the Netflix Prize, Springer 2008, pp. 337–348.

Zhu, S., Yu, Y., Chi, Y. and Gong, Y. (2014): Combining content and collaborative filters for improved recommendation. In: *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 387–390.