RecSys Pertaining to Research Information with Collaborative Filtering Methods: Characteristics and Challenges

Otmane Azeroual 1,*, and Tibor Koltay 2

1 German Centre for Higher Education Research and Science Studies (DZHW), Schützenstraße 6a, 10117 Berlin, Germany
2 Institute of Learning Technologies, Eszterházy Károly Catholic University, HU-3300 Eger, Hungary; koltay.tibor@uni-eszterhazy.hu
* Correspondence: azeroual@dzhw.eu; Tel.: +49-30-2064177-38

Abstract: Recommendation (recommender) systems have played an increasingly important role in both research and industry in recent years. In the area of publication data, for example, there is a strong need to help people find the right research information through recommendations and scientific reports. The difference between search engines and recommendation systems is that search engines help us find something we already know, while recommendation systems are more likely to help us find new items. An essential function of recommendation systems is to support users in their decision making. Recommendation systems are information systems that can be categorized into decision support systems, as long as they are used for decision making and are intended to support people instead of replacing them. This paper deals with recommendation systems for research information, especially publication data. We discuss and analyze the challenges and peculiarities of implementing recommender systems for the scientific exchange of research information. For this purpose, data mining techniques are examined and a concept for a recommendation system for research information is developed. Our aim is to investigate to what extent a recommendation system based on a collaborative filtering approach with cookies is possible. The data source is publication data extracted from cookies in the Web of Science database. The results of our investigation show that a collaborative filtering process is suitable for publication data and that recommendations can be generated with user information. In addition, we have seen that collaborative filtering is an important element that can solve a practical problem by sifting through large amounts of dynamically generated information to provide users with personalized content and services.

Keywords: research information; recommender systems; decision support systems; publication data; knowledge-based recommendation; filter process; data mining; collaborative recommendation algorithms

1. Introduction

In the course of digitization, humanity is currently in an information age. The Internet has become an important part of society and offers the possibility of accessing information at almost any time. Search engines also help users to filter or narrow down information from the Internet that actually or roughly corresponds to their search. Institutions and libraries are faced with the challenge of controlling and delivering their large range of publications as quickly as possible, which satisfies the needs of the user or arouses their interest in or need for it. Recommender systems (RecSys) help institutions to provide personalized and individually adapted publications on the basis of the analysis of large amounts of data, which, for example, a researcher may leave on a website. They are thus becoming a more and more important part of marketing strategies. Since the mid-1990s, recommendation systems have developed into an independent research area. In recent years, the interest in recommendation systems has increased further and research has been carried out in the area of publication data in order to recommend relevant articles for the needs of the users as references [1,2].
Although some academic search engines such as Google Scholar or Semantic Scholar can effectively help users find papers based on the keywords and constraints entered, the results returned may not always meet requirements because it is difficult to understand users’ needs [3,4]. Recommendation systems were an essential part of the first big e-commerce companies such as Amazon or Ebay and can also be used in other areas [5]. In the area of publication data, for example, there is a strong need to help people find the right research information through recommendations and scientific reports.

In a broader context, however, recommendation systems are being blamed for the destabilization of users’ informational ecosystems because of their role in the formation of filter bubbles [6] and echo chambers [7]. In essence, both aim to prevent users from consuming information coming from outside their sphere. Nonetheless, their appearance is not primarily caused by technology, but by deeper social and political issues [8]. Moreover, recommendation systems may prove useful for treating the symptoms of information overload, which is a set of objective and subjective difficulties caused by the amount and complexity of information available, as well as its diversity, complexity, and novelty. With the growing amount of various types of data available and the recognition of their importance, there is the problem of data overload as well [9].

In this paper, a prototypical concept for the implementation of a recommendation system for publication data is described. In addition, this work mainly deals with data mining methods (such as clustering with k-means procedures) in relation to recommendation systems and attempts to answer the following research question:

• To what extent is making a recommendation possible if it is based on a collaborative filtering approach directed to research information—in particular, publication data?

The feature space for the k-means algorithm is based on the fact that the implemented method of collaborative filtering is carried out as clustering on N-dimensional vectors (where N is the number of publications).

Research information is metadata that describes in a structured way which research activities take place at a research institution. This can be information about ongoing and completed research projects, publications, or other research activities. The data source used is data extracted from cookies from the publication data of the Web of Science database, which we have access to at the German Center for Higher Education Research and Science Studies (DZHW). The publication data consist of 60 columns and 1 million lines.

This paper is divided into seven sections. Section 1 is the introduction. Section 2 provides an overview of the recommendation systems and the state of the art in this area. Then, the types of recommendation systems and their advantages and disadvantages are presented. Section 3 deals with the challenges and special features that result from the application of the collaborative filtering approach. Section 4 describes the materials and methods used in this paper. In Section 5, the procedure, data processing techniques, and technical environment are detailed and a model is created to serve as an aid to research institutions. Finally, the implementation of this model using the KNIME software (https://www.knime.com/ accessed on 12 December 2021) is described and the results are presented. In Section 6, the important results are summarized and an outlook on future work is given.

2. State of the Art: RecSys and Collaborative Filtering Processes

Our aim is to analyze which recommendation systems exist in the literature and how this potential has been utilized by various authors working on this topic. Through our literature review, we found that many recommendation algorithms have been proposed and tested by authors in different contexts, but not in the context of research information. The use of recommendation systems in the field of research information processing is of great interest to the research information community as well as to research institutions and researchers. There is no uniform definition of recommendation systems used for research purposes. They take on different meaning depending on the requirements and applications involved. The origins of recommendation systems lie in the research areas of
information retrieval, cognitive science, and machine learning [10]. Most of the studies on recommendation systems, such as [11–13], address the underlying procedures used. Since research in this area began, various recommendation algorithms have been proposed and tested. Based on [14,15], the following six categories of recommendation systems can be distinguished. Their functionality is shown in Figure 1.

![Figure 1. How a RecSys works (based on [16]).](image-url)

1. **Non-personalized filtering**: The recommendations are the same for all customers. This type of filtering materializes, for example, in systems, where the most popular products are recommended.
   - Advantages: Non-personalized filtering is easy to implement, since the recommendations consist of popular or highly rated articles and the data required for these recommendations can therefore be easily captured [17].
   - Disadvantages: As filtering is not personalized, every user receives the same recommendation; therefore, such recommendations may not apply to all users [17].

2. **Collaborative filtering**: Recommendations are based on historical reviews of users or data. Traditionally, either items rated highly by similar users or articles that are similar to already highly rated items are recommended.
   - Advantages: The greatest advantage of collaborative filtering systems is that they do not need any reference to the content of the article. This means that they are completely independent of any kind of information involved and the item descriptions. All that is required to recommend the item is a name and the associated rating.
   - Disadvantages: Collaborative filtering is basically based on statistical methods and requires a certain degree of consistency in user ratings. As a result, a very high density of ratings is required in order for this to function properly.

3. **Content-based filtering**: Recommendations are based on the content or properties of articles. Articles that are similar in content to preferred articles are recommended.
   - Advantages: Content-based filtering uses article-to-article correlation [18]. The user is offered articles that would be suitable according to their user profile. This knowledge is derived from the profiles of the individual user. With this method, the cold start problem is minimized, and since there is no new item problem only the new user problem remains. There are no privacy problems because
one’s own ratings and preferences are not visible to other users. Additionally, the detailed preferences of users can be taken into account.

- **Disadvantages:** Content-based filtering and its systems are limited by the content-descriptive characteristics of the items needing to be evaluated. For example, a content-based recommendation system for publications can only be based on article descriptions, such as the author’s name, publication title, journal, volume, issue, year, etc., because the publication itself cannot be interpreted by the system. This means that only fully described publications (namely, all publications that are included in the calculation) can lead to a successful recommendation. One also needs to have enough descriptors to avoid meta-evaluations such as those seen in collaborative filtering. There is also the problem that not all aspects of an article can be formally described.

4. **Knowledge-based filtering:** Recommendations are made based on specialist knowledge of how certain properties of an item influence the satisfaction of user needs. For this purpose, similarity functions or knowledge databases with explicit rules are used.

- **Advantages:** With this method, users have the advantage of finding articles based on the restrictions with which they are familiar or articles that are similar and those that meet the criteria set [18].
- **Disadvantages:** The system must have access to a knowledge database that the information can be easily identified or derived from. In addition, there is the challenge of making a good recommendation to investigate which article properties are decisive for the user [18].

5. **Demographic filtering:** Recommendations are based on sociodemographic data. This method only uses information about the user, such as their gender, age, and employment status. The system thus generates recommendations based on demographic similarities.

- **Advantages:** In contrast to other filtering methods, this method does not require historical data [19].
- **Disadvantages:** Collecting all of the demographic information can conflict with the requirements of data protection, as it is sensitive and personal data. In addition, this is one of the traditional techniques used to create the profiles of users and no advanced data mining techniques are used [10].

6. **Hybrid filtering:** A combination of the filters already presented are used to generate recommendations. Recommendations consist of combinations of several filter methods, such as collaborative filtering and content-based filtering.

- **Advantages:** The aim of hybrid filtering is to bring different models together in order to increase performance and minimize the system-specific and inherent problems of recommendation systems.
- **Disadvantages:** With hybrid filtering, the quality of the description of the items and their quality depends on the number of ratings.

Collaborative recommendation algorithms based on historical evaluation data are the most widely used techniques [20, 21]. The literature research in the Web of Science showed that the topic of recommendation systems and the methodology of collaborative filtering in the field of research information processing receives too little attention nowadays and is therefore the focus of the research in this work. The high-quality theoretical elaboration of the related issues opens up a wide range of possibilities for applied research and experiments with research information collections. Collaborative filters can make use of the collective strength of publication data to generate recommendations. The idea behind such procedures is that unknown evaluations can be predicted because known evaluations from different users and/or publications often correlate strongly with one another. Let there be two scientific users A and B, who have very similar research areas. If the submitted ratings are similar, this will be recognized by the recommendation algorithm. In such cases, it is very likely that the ratings given by only one of the two users are also similar. This presumption of similarity is used to draw conclusions about unknown evaluations.
Figure 2 shows the schematic flow of a collaborative recommendation process. On the basis of an m \times n user-publication matrix, a collaborative filter algorithm generates either recommendations or forecasts, which are presented to the users in the form of a top N recommendation list.

Two types of procedures are used in collaborative recommendation systems [22,23]:

- Memory-based, where a user rating of an item is predicted on the basis of all previous ratings of the other users.
- Model-based with the help of stochastic procedures and a training data set, attempting to find patterns in the evaluations of the various users and to create a model based on them. The learned model is then used to propose new articles. In the case of model-based methods, methods from the field of machine learning and data mining are used as a forecasting model. Examples of such methods are decision trees, rule-based models, Bayesian methods, and latent factor models.

The collaborative filtering concept can be implemented with two different methods. In the case of the item-based neighborhood method, the similarity of the articles and evaluation patterns of the user can be used to create recommendations. If two different items are rated good or bad by the same users, the items are considered the same, assuming that the users have the identical preferences for the same items. This method considers the set of items evaluated by a user, calculates from this the usefulness of a target item, and decides whether it should be suggested to the user or not. Figure 3 shows a user × item matrix in which the user’s u_1 and u_2 rated two items i_j and i_k.

In the user-based neighborhood method, similarities are defined on the basis of similar users. The similarities or neighborhoods between the users are determined by the evaluation behavior. Based on n users and m recommendation items, the matrix R = (r_{ij}) with i = 1 \ldots n and j = 1 \ldots m is generated. Here, the value r_{ij} represents the evaluation of item j by user i. The rating can be explicitly good/bad or implicitly read/not read. The aim is thus to recommend a recommendation item i_y for a user U_x by the users U = \{U_1, \ldots, U_n\}, using the U_y who are most similar and have given good ratings to items that the user U_x has not yet rated. The Pearson correlation is often used for this method in order to calculate the similarities between two users U_1 and U_x. If the k nearest neighbors have been found, the rating is calculated from the sum of the weighted average of the rating across the k nearest neighbors.

Figure 4 shows the (user × item) matrix that is used to calculate the similarity between users based on jointly rated items.
Recommendation systems often have to operate in environments that present a number of challenges. Nevertheless, in the interests of the scientific institutions, they should quickly generate precise recommendations. The extent to which collaborative recommendation processes are able to do this depends on how they deal with the following challenges, which are also part of their characteristics.

Insufficient data: When using the recommendation system, it is imperative that sufficient data are available. However, this does not depend on the pure amount of data. It is much more important that these meet a certain quality standard, because recommendations can only be made with sufficiently high-quality information. Therefore, the institutions should attach great importance to the right choice and the collection of suitable data from the start.

This gives rise to a problem that is well known in the application area of recommendation systems: the cold start problem [24]. The cold start problem describes the problem that occurs when a new user or item is added to the system [24]. In such situations, there are no known likes or dislikes for any user, nor has any user rated a new item. There is therefore no data basis on which recommendations for a user or an article can be calculated.
Ambiguity in the database: If the labeling is inaccurate, it can lead to the same items with different labels being incorrectly rated. As a result, publications may be suggested to a user that do not meet their needs.

Scalability: The huge volume of publications overwhelms many research institutions these days. In this respect, it is becoming necessary to use increasingly large data sets for recommendation systems. Collaborative filtering methods have scaling problems due to the increasing number of ways in which to solve this challenge. On the one hand, there is an incremental singular value of decomposition [25], which calculates new forecasts after adding additional evaluations without recalculating the dimension-reduced model [26]. This means that the system is not forced to recalculate the entire algorithm after every update of the database. However, it should be noted that the expensive matrix factorization has to be carried out.

Storage-based approaches, such as the article-based collaborative filtering method, can also achieve a good scalability. Bayesian collaborative filtering methods address the scaling problem by calculating forecasts based on known evaluations [27,28]. Cluster methods, on the other hand, do not search for similar users within the entire database, but rather generate recommendations by searching within small clusters [29]. However, the increased scalability is at the expense of the forecast accuracy.

Gray Sheep: Gray sheep are users whose preferences are neither consistent nor disagreeable with the ratings of other user groups and therefore do not benefit from the collaborative recommendation process. In contrast, the group of black sheep includes users whose unique research areas make recommendations almost impossible. To solve the gray sheep problem, Claypool et al. [30] used a hybrid approach of content-based and collaborative methods that calculated a weighted average of both forecasts. The weights were set in order to generate an optimal mix for each user.

Shilling Attacks: In some cases, all visitors can leave reviews regardless of whether the publication is selected or not. This is a way of ensuring that your own articles are recommended despite receiving bad reviews. This phenomenon is referred to in the literature as a Schilling attack [31]. Bell and Koren [32] proposed a solution to the problem of memory-based methods. To do this, they removed global effects during data normalization and used their residuals to look for neighborhoods.

There are different recommendation systems that should have different properties; these are discussed below:

- Relevance: The filtered set of all articles should be relevant to the user.
- Novelty: The recommendations determined should be novel to the user—i.e., after a certain contact frequency without user interaction, the recommendation should be replaced by a new one.
- Discovery: Recommendation systems should deliver results that are unknown and surprising to the user, but at the same time interesting.
- Diversity: The recommendations from the recommendation system should have a certain diversity, i.e., even if the user has only looked at publications from the IT area so far, they should not only see publications from the IT area as recommendations.

4. Materials and Methods

As mentioned in the literature, recommendation systems are essential for different platforms and databases [21,33]. Publication databases such as Web of Science, from which cookies are taken, currently offers a filter function in which the user can filter articles by category. In the context of this database, it should be examined whether recommendations can be generated from the cookies collected. It should also be checked whether it is possible to make a recommendation using the collaborative filtering approach. In order to generate a recommendation, knowledge must be generated from the publication data. For the collaborative filtering approach, it is important to use the data to create profiles of users who are similar in terms of their interests and behavior. To achieve this, the publication data from the cookies must be processed and transformed. Distance functions
are used to calculate the similarities of profiles. Profiles should then be clustered. For this purpose, it should be checked which algorithms from the data mining environment are suitable. Articles that have not yet been read should then be recommended to a user. The recommendation is thus based on the similar behaviors and interests of users. The procedure is based on this process, which is shown in Figure 5. Recommendation systems use data mining techniques with the aim of generating patterns from the data sources. The characteristics of these patterns are that they are valid, unknown, potentially useful, and easy to understand.

Figure 5. Knowledge discovery in databases (KDD).

Data mining means digging for data with the aim of generating knowledge. Data and knowledge are regarded as gold by companies today, since sales and profits can be generated from them. The information overload and storage of large amounts of data can obscure interesting relationships between data. The entire process from data selection to knowledge generation is referred to as knowledge discovery in databases. Data mining is therefore a phase in the KDD process. The methods of data mining are divided into different categories in the literature. Some methods, techniques, and algorithms considered in our work (e.g., through clustering methods such as k-means) are discussed in [34]. Therefore, we do not go into the details of how they work here. The publication data, which are used for the KDD process, are shown in Figure 6. This includes publication data from a Web of Science publication database, which is based on cookies. Cookies can be understood as a supplement to HTTP [35]. They contain text information that is generated when a client accesses a website. Cookies can be saved on the client’s browser. In order to use a website again, the cookies can be read out directly by the web server or transmitted to the server via a script [35]. The tasks include, for example, identifying the client and storing shopping carts and registrations [35].

Figure 6. Publication data from Web of Science.
5. Results

The data were made available by the Web of Science database in the file type table. The raw file was 100 MB in size and contained 60 columns and 1 million lines. All publication data were in the data type string and integer. A higher number of cookie records leads to a more relevant analysis because each line represents a session cookie in which a user has accessed an article in the publication database. Each article call is represented in one line. The cookies used in this work have been made available in a hard drive. The complete raw data record and all subsequent processed data records are shown in Figure 6.

This relevant information was extracted from cookies and reduced to these dimensions.

- First name;
- Last name;
- Gender;
- Date of birth;
- ORCID;
- DOI;
- Title.

This is the information that could be interpreted and was taken into account in the further course of this work. For the KDD process, we used KNIME, which is an open-source data mining software developed by the University of Konstanz. KNIME is compatible with all operating systems. More information about the KNIME tool can be found in [36].

The publication data were examined for missing, noisy, incorrect, and inconsistent data. A total of 153,065 null values were found; they were replaced by the string unknown in the device column and remained in the data record. There were a lot of noisy, inconsistent, and incorrect data and outliers that were corrected. Sufficient data are required to make a meaningful recommendation.

As can be seen in Figure 7, 16,287 users only read the publication entitled “METHODS, HYDRAULICS AND MASS EXCHANGE IN INDUSTRIAL PLATE COLUMNS”. Additionally, 15,639 users only read the publication entitled “Expectations and realities of GPS animal location collars: results of three years in the field”.

Figure 7. How often the articles are read.
Each line of the current data record represents an article view by a user. The user can be identified by the AuthorID. In order to recognize which articles a user has read, the previous table was transformed, as can be seen in Figure 7. The new columns of the new table also contain all categories that are intended to indicate the content of the articles (e.g., in the field of medicine, IT, energy, etc.) and have a connection with the title of the article. The values represent the number of articles read in the respective category. This means that there are a total of 10,467 users.

The box plot analysis (see Figure 8) shows that the number of articles read varies greatly. New features were added to the table shown here. All users who had read less than 24 articles were assigned a reading record of 1, regardless of the number of documents read. Average readers were assigned reading record numbers between 24 and 75. In addition, prolific readers were given reading record numbers of over 75. A new column reading activity was added to the profile of the readers. The entire corpus was analyzed for new feature extractions from the data.

Figure 8. Box plot number of articles read per user.

The values of the available data are numerical, so the k-means algorithm as described above is suitable for the collaborative filter approach. In addition, the k-means algorithm for partitioned clustering methods is widely used. The clusters found represent the basis for collaborative filtering. A well-known and established distance function in preparation for the k-means algorithm is the Euclidean distance function. Therefore, the Euclidean distance is calculated and then the k-means algorithm is applied with the choice for \( k = 9 \) based on the 9 first-level categories. The iterations were varied so that after 30 iterations the results no longer changed. Each cluster represents a specific user behavior pattern. As a result, a new cluster column is added to each profile, indicating which cluster this profile belongs to (see Figure 9).

The results of the k-means algorithm show that a profile belongs to the respective cluster. Many different articles can be found in the corpus of the clusters, some of which overlap with other clusters. The question arises here as to which articles are to be recommended exactly. For this purpose, the respective clusters are being reduced. In this cluster, the Top 10 most frequently read articles are selected. It then checks which articles users have read and which they have not. The articles that are not read are recommended to the user. The recommendation is thus based on the respective profiles and their affiliation to a specific cluster profile, which is generated from their reading behavior.
The data used in this paper were mainly based on cookies. Some information could be taken from the cookies, but the assumption made here is that an AuthorID is also assigned to a profile. Since the cookies are usually automatically deleted when the browser is closed, a new AuthorID is assigned to the same user when the page is called up again. This assumption was sufficient for the prototypical approach, but for a productive system it would be better to differentiate between users who have created an account and those without an account. The existing cookies show a total of 153,065 (15.30%) zero values. This is due to the fact that the categories could not be determined from the respective cookie. Data processing can be considerably simplified if there is access to the publication database, and this provides the opportunity to work with structured data.

The evaluation of cluster processes can be described as difficult in general. It is easier if examples have already been given and the investigation is based on how similar the clusters are to the already given clusters. With a cluster procedure, however, the target clusters are usually not known. The model can be rated when the expected result is available for comparison. In the context of this paper, recommendations were generated through the KDD process and the use of the respective methods. The model can, e.g., be evaluated by the accuracy, such as \[ \text{accuracy} = \frac{\text{number of successful recommendations}}{\text{number of recommendations}} \]. This calculation is used in many different KDD processes, including Burke [12]. The relative proportion of the incorrectly classified data for the entire assignment is then calculated here. However, the prerequisite for this method is that the expected result is available for comparison. In the case of a recommendation whether a user buys a publication from a scientific journal can be observed. In this case, whether a user reads or clicks on the recommended article on the publication database website should be observed.

For the evaluation of accuracy, the F1 score can also be determined as follows:

\[ F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]

Precision returns the proportion of relevant documents relative to all documents in the result set. Recall shows the ability of the search method to discover all relevant documents in the corpus. The calculation of precision and recall was as follows:

\[ \text{precision} = \frac{\text{(Number of relevant documents found)}}{\text{(Total number of documents found)}} \]

\[ \text{recall} = \frac{\text{(Number of relevant documents found)}}{\text{(Total number of relevant documents)}} \]

A comparison can then be made here, as can be seen in Figure 10.
In the context of this work, the timestamps were not taken into account. The topicality of an article can be recognized from the timestamps. After all, current publications are mostly relevant. In addition, the recommendation structure of the top 10 articles can also be improved. There could be a ranking among the Top 10 that shows which article within the Top 10 was read most often.

6. Conclusions and Future Work

The paper gave an insight into the best-known concepts and procedures from the subject areas of recommendation systems and data mining in the field of research information and indicated which institutions should consider their use. In addition, a prototypical approach for a recommendation system with the collaborative filtering process based on cookies, developed based on the publication data from Web of Science, was presented. With the dataset of cookies used for this paper, recommendations could be generated through a collaborative filtering approach. Similar users were identified by means of a clustering process and their profile created based on information about the articles they read. Recommendations from their cluster were then made for users. There is no evaluation here because the model developed must be observed in the production environment. In addition, it can be said that the data source provides little information due to the zero values that are relevant for profiling users. This study shows that a collaborative filtering process is suitable for publication data and that recommendations can be generated with little user information. Web mining and data mining in relation to research information remain exciting topics that have not yet had their potential fully exploited. From the selection of approaches to the analysis of user behavior and algorithms, new approaches are constantly being published. In addition, research institutions have to invest in these subject areas because the developments in the publication databases and repositories show that flexible structures are becoming more and more important and that users expect corresponding publications and recommendations.

In future work, we will consider the use of artificial intelligence (AI) and machine learning to generate recommendations because they offer a comprehensive portfolio of algorithms that can be used in recommendation systems to predict ratings or interactions based on item and user attributes [37] in the context of publication data. Based on this premise, we will propose a concrete example of a recommendation system that is able to find and recommend relevant but not frequently accessed publications.

Figure 10. F1 Score matrix recommendation.
Author Contributions: O.A. and T.K. contributed to the design and implementation of the research to the analysis of the results and to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data available on request from the authors.

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

References

1. Beel, J.; Langer, S.; Genzmehr, M.; Gipp, B.; Breitinger, C.; Nürnberger, A. Research paper recommender system evaluation: A quantitative literature survey. In Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation, Hong Kong, China, 12 October 2013; pp. 15–22. [CrossRef]
2. Salton, G. Associative document retrieval techniques using bibliographic information. J. ACM 1963, 10, 440–457. [CrossRef]
3. Yang, Q.; Li, Z.; Liu, A.; Liu, G.; Zhao, L.; Zhang, X.; Zhang, M.; Zhou, X. A novel hybrid publication recommendation system using compound information. World Wide Web 2019, 22, 2499–2517. [CrossRef]
4. Ricci, F.; Rokach, L.; Shapira, B. Introduction to Recommender Systems Handbook. In Recommender Systems Handbook; Ricci, F., Rokach, L., Shapira, B., Kantor, P., Eds.; Springer: Boston, MA, USA, 2011. [CrossRef]
5. Wang, D.; Liang, Y.; Xu, D.; Feng, X.; Guan, R. A content-based recommender system for computer science publications. Knowl.-Based Syst. 2018, 157, 1–9. [CrossRef]
6. Pariser, E. The Filter Bubble: What the Internet Is Hiding from You; Penguin UK Books: London, UK, 2012.
7. Sunstein, C.R. Republic.com 2.0; Princeton University Press: Princeton, NJ, USA, 2009.
8. Claes, A.; Philippette, T. Defining critical data literacy for recommender systems: A media-grounded approach. J. Media Lit. Educ. 2020, 12, 17–29. [CrossRef]
9. Bawden, D.; Robinson, L. Information Overload: An introduction. In Oxford Encyclopedia of Political Decision Making; Oxford University Press: Oxford, UK, 2020. [CrossRef]
10. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. 2005, 17, 734–749. [CrossRef]
11. Bogers, T.; van den Bosch, A. Recommending scientific articles using citeulike. In Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys ’08), Lausanne, Switzerland, 23–25 October 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 287–290. [CrossRef]
12. Pera, M.S.; Ng, Y.K. A personalized recommendation system on scholarly publications. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM ’11), Glasgow, UK, 24–28 October 2011; Association for Computing Machinery: New York, NY, USA, 2011; pp. 2133–2136. [CrossRef]
13. Raamkumar, A.S.; Foo, S. Multi-method Evaluation in Scientific Paper Recommender Systems. In Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP ’18); Association for Computing Machinery: New York, NY, USA, 2018; pp. 179–182. [CrossRef]
14. Francesco Ricci, F.; Rokach, L.; Shapira, B. Recommender Systems Handbook; Springer: Boston, MA, USA, 2015. [CrossRef]
15. Aggarwal, C.C. Recommender Systems: The Textbook; Springer International Publishing AG: Cham, Switzerland, 2016. [CrossRef]
16. Hohfeld, S.; Kwiatkowski, M. Empfehlungssysteme aus informationswissenschaftlicher Sicht-State of the Art. Inf. Wiss. Und Prax. 2007, 58, 265–276.
17. Fan, Y.; Shen, Y.; Mai, J. Study of the Model of E-commerce Personalized Recommendation System Based on Data Mining. In Proceedings of the International Symposium on Electronic Commerce and Security, Guangzhou, China, 3–5 August 2008; pp. 647–651. [CrossRef]
18. Burke, R. Integrating knowledge-based and collaborative-filtering recommender systems. In Proceedings of the Workshop on AI and Electronic Commerce; 1999; pp. 69–72. Available online: https://aaaipress.org/Papers/Workshops/1999/WS-99-01/WS99-01-01.pdf (accessed on 25 December 2021).
19. Burke, R. Hybrid recommender systems: Survey and experiments. User Modeling User-Adapt. Interact. J. Pers. Res. 2002, 12, 331–370. [CrossRef]
20. Javed, U.; Shaukat, K.; Hameed, I.A.; Iqbal, F.; Alam, T.M.; Luo, S. A Review of Content-Based and Context-Based Recommendation Systems. Int. J. Emerg. Technol. Learn. 2021, 16, 274–306. Available online: https://online-journals.org/index.php/i-jet/article/view/18851 (accessed on 25 December 2021).
21. Isinkaye, F.O.; Folajimi, Y.O.; Ojokoh, B.A. Recommendation systems: Principles, methods and evaluation. Egypt. Inform. J. 2015, 16, 261–273. [CrossRef]
22. Sarwar, B.; Karypis, G.; Konstan, J.; Reidl, J. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International World Wide Web Conference; 2001; pp. 285–295. Available online: https://dl.acm.org/doi/10.1145/371920.372071 (accessed on 25 December 2021).
23. Breese, J.S.; Heckerman, D.; Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI’98, Madison, WI, USA, 24–26 July 1998; Morgan Kaufmann.
24. Lika, B.; Kolomvatsos, K.; Hadjieithymiades, S. Facing the cold start problem in recommender systems. Expert Syst. Appl. 2014, 41, 2065–2073. [CrossRef]

25. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Incremental SVD-Based Algorithms for Highly Scalable Recommender Systems. In Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT ’02); 2002. Available online: http://glaros.dtc.umn.edu/gkhome/node/129 (accessed on 9 January 2022).

26. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 1990, 41, 391–407. [CrossRef]

27. Miyahara, K.; Pazzani, M.J. Collaborative Filtering with the Simple Bayesian Classifier. In PRICAI 2000 Topics in Artificial Intelligence. PRICAI 2000; Lecture Notes in Computer Science; Mizoguchi, R., Slaney, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1886. [CrossRef]

28. Miyahara, K.; Pazzani, M.J. Improvement of collaborative filtering with the simple Bayesian classifier. Inf. Process. Soc. Jpn. 2002, 43, 3429–3437.

29. Chee, S.H.S.; Han, J.; Wang, K. RecTree: An Efficient Collaborative Filtering Method. In Data Warehousing and Knowledge Discovery. DaWaK 2001; Kambayashi, Y., Winiwarter, W., Arikawa, M., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2114, pp. 141–151. [CrossRef]

30. Claypool, M.; Gokhale, A.; Miranda, T.; Murnikov, P.; Netes, D.; Sartin, M. Combining Content-Based and Collaborative Filters in an Online Newspaper. In Proceedings of the ACM SIGIR ’99 Workshop on Recommender Systems: Algorithms and Evaluation, ACM, Berkeley, CA, USA, 19 August 1999.

31. Resnick, P.; Varian, H.R. Recommender systems. Commun. ACM 1997, 40, 56–58. [CrossRef]

32. Bell, R.; Koren, Y. Improved neighborhood-based collaborative filtering. In Proceedings of the KDD Cup and Workshop, San Jose, CA, USA, 12 August 2007, pp. 7–14.

33. Kanungo, A.; Kamath, S.; Gosai, S.; Mishra, R. Cross Platform Recommendation System. In ICDSMLA 2019. Lecture Notes in Electrical Engineering; Kumar, A., Paprzycki, M., Gunjan, V., Eds.; Springer: Singapore, 2020; Volume 601. [CrossRef]

34. Azeroual, O. Text and Data Quality Mining in CRIS. Information 2019, 10, 374. [CrossRef]

35. Kristol, D.M. HTTP Cookies: Standards, privacy, and politics. ACM Trans. Internet Technol. 2001, 1, 151–198. [CrossRef]

36. Feltrin, L. KNIME an Open Source Solution for Predictive Analytics in the Geosciences [Software and Data Sets]. IEEE Geosci. Remote Sens. Mag. 2015, 3, 28–38. [CrossRef]

37. Nawrocka, A.; Kot, A.; Nawrocki, M. Application of machine learning in recommendation systems. In Proceedings of the 19th International Carpathian Control Conference (ICCC), Szilvasvarad, Hungary, 28–31 May 2018; pp. 328–331. [CrossRef]