

Recommender system

A **recommender system** or a **recommendation system** (sometimes replacing 'system' with a synonym such as platform or engine) is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.^{[1][2]} They are primarily used in commercial applications.

Recommender systems are utilized in a variety of areas, and are most commonly recognized as playlist generators for video and music services like Netflix, YouTube and Spotify, product recommenders for services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter.^[3] These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries. There are also popular recommender systems for specific topics like restaurants and online dating. Recommender systems have been developed to explore research articles and experts,^[4] collaborators,^[5] financial services,^[6] and life insurance.

Contents

Overview

Approaches

- Collaborative filtering
- Content-based filtering
- Multi-criteria recommender systems
- Risk-aware recommender systems
- Mobile recommender systems
- Hybrid recommender systems

The Netflix Prize

Performance measures

- Beyond accuracy
- Reproducibility in recommender system research

See also

References

Further reading

External links

Overview

Recommender systems usually make use of either or both collaborative filtering and content-based filtering (also known as the personality-based approach),^[7] as well as other systems such as knowledge-based systems. Collaborative filtering approaches build a model from a user's past behavior (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in.^[8] Content-based filtering approaches utilize a series of discrete, pre-tagged characteristics of an item in order to recommend additional items with similar properties.^[9] Current recommender systems typically combine one or more approaches into a hybrid system.

The differences between collaborative and content-based filtering can be demonstrated by comparing two early music recommender systems – [Last.fm](#) and [Pandora Radio](#).

- Last.fm creates a "station" of recommended songs by observing what bands and individual tracks the user has listened to on a regular basis and comparing those against the listening behavior of other users. Last.fm will play tracks that do not appear in the user's library, but are often played by other users with similar interests. As this approach leverages the behavior of users, it is an example of a collaborative filtering technique.
- Pandora uses the properties of a song or artist (a subset of the 400 attributes provided by the [Music Genome Project](#)) to seed a "station" that plays music with similar properties. User feedback is used to refine the station's results, deemphasizing certain attributes when a user "dislikes" a particular song and emphasizing other attributes when a user "likes" a song. This is an example of a content-based approach.

Each type of system has its strengths and weaknesses. In the above example, Last.fm requires a large amount of information about a user to make accurate recommendations. This is an example of the cold start problem, and is common in collaborative filtering systems.^{[10][11][12]} Whereas Pandora needs very little information to start, it is far more limited in scope (for example, it can only make recommendations that are similar to the original seed).

Recommender systems are a useful alternative to search algorithms since they help users discover items they might not have found otherwise. Of note, recommender systems are often implemented using search engines indexing non-traditional data.

Recommender systems were first mentioned in a technical report as a "digital bookshelf" in 1990 by [Jussi Karlgren](#) at Columbia University,^[13] and implemented at scale and worked through in technical reports and publications from 1994 onwards by Jussi Karlgren, then at SICS,^{[14] [15]} and research groups led by [Pattie Maes](#) at MIT,^[16] Will Hill at Bellcore,^[17] and [Paul Resnick](#), also at MIT^{[18] [19]} whose work with GroupLens was awarded the 2010 [ACM Software Systems Award](#).

Montaner provided the first overview of recommender systems from an intelligent agent perspective.^[20] Adomavicius provided a new, alternate overview of recommender systems.^[21] Herlocker provides an additional overview of evaluation techniques for recommender systems,^[22] and Beel et al. discussed the problems of offline evaluations.^[23] Beel et al. have also provided literature surveys on available research paper recommender systems and existing challenges.^{[24][25][26]}

Recommender systems have been the focus of several granted patents. ^{[27][28][29][30][31]}

Approaches

Collaborative filtering

One approach to the design of recommender systems that has wide use is collaborative filtering.^[32] Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. The system generates recommendations using only information about rating profiles for different users or items. By locating peer users/items with a rating history similar to the current user or item, they generate recommendations using this neighborhood. The user- and item-based nearest neighbor algorithms can be combined to deal with the cold start problem and improve recommendation results using this data.^[33] Collaborative filtering methods are classified as memory-based and model-based. A well-known example of memory-based approaches is the user-based algorithm,^[34] while that of model-based approaches is the Kernel-Mapping Recommender.^[35]



An example of collaborative filtering based on a ratings system

A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. Many algorithms have been used in measuring user similarity or item similarity in recommender systems. For example, the k-nearest neighbor (k-NN) approach^[36] and the Pearson Correlation as first implemented by Allen.^[37]

When building a model from a user's behavior, a distinction is often made between explicit and implicit forms of data collection.

Examples of explicit data collection include the following:

- Asking a user to rate an item on a sliding scale.
- Asking a user to search.
- Asking a user to rank a collection of items from favorite to least favorite.
- Presenting two items to a user and asking him/her to choose the better one of them.
- Asking a user to create a list of items that he/she likes (see Rocchio classification or other similar techniques).

Examples of implicit data collection include the following:

- Observing the items that a user views in an online store.
- Analyzing item/user viewing times.^[38]
- Keeping a record of the items that a user purchases online.
- Obtaining a list of items that a user has listened to or watched on his/her computer.
- Analyzing the user's social network and discovering similar likes and dislikes.

Collaborative filtering approaches often suffer from three problems: cold start, scalability, and sparsity.^[39]

- **Cold start:** For a new user or item, there isn't enough data to make accurate recommendations.^{[10][11]}
- **Scalability:** In many of the environments in which these systems make recommendations, there are millions of users and products. Thus, a large amount of computation power is often necessary to calculate recommendations.
- **Sparsity:** The number of items sold on major e-commerce sites is extremely large. The most active users will only have rated a small subset of the overall database. Thus, even the most popular items have very few ratings.

One of the most famous examples of collaborative filtering is item-to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by Amazon.com's recommender system.^[40]

Many social networks originally used collaborative filtering to recommend new friends, groups, and other social connections by examining the network of connections between a user and their friends.^[1] Collaborative filtering is still used as part of hybrid systems.

Content-based filtering

Another common approach when designing recommender systems is **content-based filtering**. Content-based filtering methods are based on a description of the item and a profile of the user's preferences.^{[41][42]} These methods are best suited to situations where there is known data on an item (name, location, description, etc.), but not on the user. Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on product features.

In this system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past, or is examining in the present. It does not rely on a user sign-in mechanism to generate this often temporary profile. In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. This approach has its roots in information retrieval and information filtering research.

To create a user profile, the system mostly focuses on two types of information:

1. A model of the user's preference.

2. A history of the user's interaction with the recommender system.

Basically, these methods use an item profile (i.e., a set of discrete attributes and features) characterizing the item within the system. To abstract the features of the items in the system, an item presentation algorithm is applied. A widely used algorithm is the tf-idf representation (also called vector space representation)^[43]. The system creates a content-based profile of users based on a weighted vector of item features. The weights denote the importance of each feature to the user and can be computed from individually rated content vectors using a variety of techniques. Simple approaches use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian Classifiers, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability that the user is going to like the item.^[44]

A key issue with content-based filtering is whether the system is able to learn user preferences from users' actions regarding one content source and use them across other content types. When the system is limited to recommending content of the same type as the user is already using, the value from the recommendation system is significantly less than when other content types from other services can be recommended. For example, recommending news articles based on browsing of news is useful, but would be much more useful when music, videos, products, discussions etc. from different services can be recommended based on news browsing. To overcome this, most content-based recommender systems now use some form of hybrid system.

Content based recommender systems can also include opinion-based recommender systems. In some cases, users are allowed to leave text review or feedback on the items. These user-generated texts are implicit data for the recommender system because they are potentially rich resource of both feature/aspects of the item, and users' evaluation/sentiment to the item. Features extracted from the user-generated reviews are improved meta-data of items, because as they also reflect aspects of the item like meta-data, extracted features are widely concerned by the users. Sentiments extracted from the reviews can be seen as users' rating scores on the corresponding features. Popular approaches of opinion-based recommender system utilize various techniques including text mining, information retrieval, sentiment analysis (see also Multimodal sentiment analysis) and deep learning^[45].

Multi-criteria recommender systems

Multi-criteria recommender systems (MCRS) can be defined as recommender systems that incorporate preference information upon multiple criteria. Instead of developing recommendation techniques based on a single criterion values, the overall preference of user u for the item i , these systems try to predict a rating for unexplored items of u by exploiting preference information on multiple criteria that affect this overall preference value. Several researchers approach MCRS as a multi-criteria decision making (MCDM) problem, and apply MCDM methods and techniques to implement MCRS systems.^[46] See this chapter^[47] for an extended introduction.

Risk-aware recommender systems

The majority of existing approaches to recommender systems focus on recommending the most relevant content to users using contextual information, yet do not take into account the risk of disturbing the user with unwanted notifications. It is important to consider the risk of upsetting the user by pushing recommendations in certain circumstances, for instance, during a professional meeting, early morning, or late at night. Therefore, the performance of the recommender system depends in part on the degree to which it has incorporated the risk into the recommendation process. One option to manage this issue is *DRARS*, a system which models the context-aware recommendation as a bandit problem. This system combines a content-based technique and a contextual bandit algorithm.^[48]

Mobile recommender systems

Mobile recommender systems make use of internet-accessing smart phones to offer personalized, context-sensitive recommendations. This is a particularly difficult area of research as mobile data is more complex than data that recommender systems often have to deal with. It is heterogeneous, noisy, requires spatial and temporal auto-correlation, and has validation and generality problems.^[49]

There are three factors that could affect the mobile recommender systems and the accuracy of prediction results: the context, the recommendation method and privacy.^[50] Additionally, mobile recommender systems suffer from a transplantation problem – recommendations may not apply in all regions (for instance, it would be unwise to recommend a recipe in an area where all of the ingredients may not be available).

One example of a mobile recommender system are the approaches taken by companies such as Uber and Lyft to generate driving routes for taxi drivers in a city.^[49] This system uses GPS data of the routes that taxi drivers take while working, which includes location (latitude and longitude), time stamps, and operational status (with or without passengers). It uses this data to recommend a list of pickup points along a route, with the goal of optimizing occupancy times and profits.

Mobile recommendation systems have also been successfully built using the "Web of Data" as a source for structured information. A good example of such system is SMARTMUSEUM^[51] The system uses semantic modelling, information retrieval, and machine learning techniques in order to recommend content matching user interests, even when presented with sparse or minimal user data.

Hybrid recommender systems

Most recommender systems now use a hybrid approach, combining collaborative filtering, content-based filtering, and other approaches . There is no reason why several different techniques of the same type could not be hybridized. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model (see^[21] for a complete review of recommender systems). Several studies that empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrated that the hybrid methods can provide more accurate recommendations than pure approaches. These methods can also be used to overcome some of the common problems in recommender systems such as cold start and the sparsity problem, as well as the knowledge engineering bottleneck in knowledge-based approaches.^[52]

Netflix is a good example of the use of hybrid recommender systems.^[53] The website makes recommendations by comparing the watching and searching habits of similar users (i.e., collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering).

Some hybridization techniques include:

- **Weighted:** Combining the score of different recommendation components numerically.
- **Switching:** Choosing among recommendation components and applying the selected one.
- **Mixed:** Recommendations from different recommenders are presented together to give the recommendation.
- **Feature Combination:** Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.
- **Feature Augmentation:** Computing a feature or set of features, which is then part of the input to the next technique.
- **Cascade:** Recommenders are given strict priority, with the lower priority ones breaking ties in the scoring of the higher ones.
- **Meta-level:** One recommendation technique is applied and produces some sort of model, which is then the input used by the next technique.^[54]

The Netflix Prize

One of the events that energized research in recommender systems was the Netflix Prize. From 2006 to 2009, Netflix sponsored a competition, offering a grand prize of \$1,000,000 to the team that could take an offered dataset of over 100 million movie ratings and return recommendations that were 10% more accurate than those offered by the company's existing recommender system. This competition energized the search for new and more accurate algorithms. On 21 September 2009, the grand prize of US\$1,000,000 was given to the BellKor's Pragmatic Chaos team using tiebreaking rules.^[55]

The most accurate algorithm in 2007 used an ensemble method of 107 different algorithmic approaches, blended into a single prediction. As stated by the winners, Bell et al.:^[56]

Predictive accuracy is substantially improved when blending multiple predictors. *Our experience is that most efforts should be concentrated in deriving substantially different approaches, rather than refining a single technique.* Consequently, our solution is an ensemble of many methods.

Many benefits accrued to the web due to the Netflix project. Some teams have taken their technology and applied it to other markets. Some members from the team that finished second place founded Gravity R&D, a recommendation engine that's active in the RecSys community.^{[55][57]} 4-Tell, Inc. created a Netflix project-derived solution for ecommerce websites.

A number of privacy issues arose around the dataset offered by Netflix for the Netflix Prize competition. Although the data sets were anonymized in order to preserve customer privacy, in 2007 two researchers from the University of Texas were able to identify individual users by matching the data sets with film ratings on the Internet Movie Database.^[58] As a result, in December 2009, an anonymous Netflix user sued Netflix in *Doe v. Netflix*, alleging that Netflix had violated United States fair trade laws and the Video Privacy Protection Act by releasing the datasets.^[59] This, as well as concerns from the Federal Trade Commission, led to the cancellation of a second Netflix Prize competition in 2010.^[60]

Performance measures

Evaluation is important in assessing the effectiveness of recommendation algorithms. To measure the effectiveness of recommender systems, and compare different approaches, three types of evaluations are available: user studies, online evaluations (A/B tests), and offline evaluations.^[23]

The commonly used metrics are the mean squared error and root mean squared error, the latter having been used in the Netflix Prize. The information retrieval metrics such as precision and recall or DCG are useful to assess the quality of a recommendation method. Diversity, novelty, and coverage are also considered as important aspects in evaluation.^[61] However, many of the classic evaluation measures are highly criticized.^[62]

User studies are rather small scale. A few dozens or hundreds of users are presented recommendations created by different recommendation approaches, and then the users judge which recommendations are best. In A/B tests, recommendations are shown to typically thousands of users of a real product, and the recommender system randomly picks at least two different recommendation approaches to generate recommendations. The effectiveness is measured with implicit measures of effectiveness such as conversion rate or click-through rate. Offline evaluations are based on historic data, e.g. a dataset that contains information about how users previously rated movies.^[63]

The effectiveness of recommendation approaches is then measured based on how well a recommendation approach can predict the users' ratings in the dataset. While a rating is an explicit expression of whether a user liked a movie, such information is not available in all domains. For instance, in the domain of citation recommender systems, users typically do not rate a citation or recommended article. In such cases, offline evaluations may use implicit measures of effectiveness. For instance, it may be assumed that a recommender system is effective that is able to recommend as many articles as possible that are contained in a research article's reference list. However, this kind of offline evaluations is seen critical by many researchers.^{[64][65][23]} For instance, it has been shown that results of offline evaluations have low correlation with results from user studies or A/B tests.^{[65][66]} A dataset popular for offline evaluation has been shown to contain duplicate data and thus to lead to wrong conclusions in the evaluation of algorithms.^[67] Often, results of so-called offline evaluations do not correlate with actually assessed user-satisfaction.^[68] Researchers have concluded that the results of offline evaluations should be viewed critically.

Beyond accuracy

Typically, research on recommender systems is concerned about finding the most accurate recommendation algorithms. However, there are a number of factors that are also important.

- **Diversity** – Users tend to be more satisfied with recommendations when there is a higher intra-list diversity, e.g. items from different artists.^[69]
- **Recommender persistence** – In some situations, it is more effective to re-show recommendations,^[70] or let users re-rate items,^[71] than showing new items. There are several reasons for this. Users may ignore items when they are shown for the first time, for instance, because they had no time to inspect the recommendations carefully.
- **Privacy** – Recommender systems usually have to deal with privacy concerns^[72] because users have to reveal sensitive information. Building user profiles using collaborative filtering can be problematic from a privacy point of view. Many European countries have a strong culture of data privacy, and every attempt to introduce any level of user profiling can result in a negative customer response. Much research has been conducted on ongoing privacy issues in this space. The Netflix Prize is particularly notable for the detailed personal information released in its dataset. Ramakrishnan et al. have conducted an extensive overview of the trade-offs between personalization and privacy and found that the combination of weak ties (an unexpected connection that provides serendipitous recommendations) and other data sources can be used to uncover identities of users in an anonymized dataset.^[73]
- **User demographics** – Beel et al. found that user demographics may influence how satisfied users are with recommendations.^[74] In their paper they show that elderly users tend to be more interested in recommendations than younger users.
- **Robustness** – When users can participate in the recommender system, the issue of fraud must be addressed.^[75]
- **Serendipity** – Serendipity is a measure of "how surprising the recommendations are".^[76] For instance, a recommender system that recommends milk to a customer in a grocery store might be perfectly accurate, but it is not a good recommendation because it is an obvious item for the customer to buy.
- **Trust** – A recommender system is of little value for a user if the user does not trust the system.^[77] Trust can be built by a recommender system by explaining how it generates recommendations, and why it recommends an item.
- **Labelling** – User satisfaction with recommendations may be influenced by the labeling of the recommendations.^[78] For instance, in the cited study click-through rate (CTR) for recommendations labeled as "Sponsored" were lower (CTR=5.93%) than CTR for identical recommendations labeled as "Organic" (CTR=8.86%). Recommendations with no label performed best (CTR=9.87%) in that study.

Reproducibility in recommender system research

Previous research has had little impact on the practical application of recommender systems. By 2011, Ekstrand, Konstan, et al. criticized that "it is currently difficult to reproduce and extend recommender systems research results," and that evaluations are "not handled consistently".^[79] Konstan and Adomavicius conclude that "the Recommender Systems research community is facing a crisis where a significant number of papers present results that contribute little to collective knowledge [...] often because the research lacks the [...] evaluation to be properly judged and, hence, to provide meaningful contributions."^[80] As a consequence, much research about recommender systems can be considered as not reproducible.^[81] Hence, operators of recommender systems find little guidance in the current research for answering the question, which recommendation approaches to use in a recommender systems. Said & Bellogín conducted a study of papers published in the field, as well as benchmarked some of the most popular frameworks for recommendation and found large inconsistencies in results, even when the same algorithms and data sets were used.^[82] Some researchers demonstrated that minor variations in the recommendation algorithms or scenarios led to strong changes in the effectiveness of a recommender system. They conclude that seven actions are necessary to improve the current situation:^[81] "(1) survey other research fields and learn from them, (2) find a common understanding of reproducibility, (3) identify and understand the determinants that affect reproducibility, (4) conduct more comprehensive experiments (5) modernize publication practices, (6) foster the development and use of recommendation frameworks, and (7) establish best-practice guidelines for recommender-systems research."

See also

-
- [Rating site](#)
 - [Cold start](#)

- [Collaborative filtering](#)
- [Collective intelligence](#)
- [Content discovery platform](#)
- [Enterprise bookmarking](#)
- [Filter bubble](#)
- [Personalized marketing](#)
- [Preference elicitation](#)
- [Product finder](#)
- [Configurator](#)
- [Pattern recognition](#)

References

1. Francesco Ricci and Lior Rokach and Bracha Shapira, [Introduction to Recommender Systems Handbook](http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf) (<http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf>), *Recommender Systems Handbook*, Springer, 2011, pp. 1-35
2. "Facebook, Pandora Lead Rise of Recommendation Engines - TIME" (<http://www.time.com/time/magazine/article/0,9171,1992403,00.html>). *TIME.com*. 27 May 2010. Retrieved 1 June 2015.
3. Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Bosagh Zadeh [WTF:The who-to-follow system at Twitter](http://dl.acm.org/citation.cfm?id=2488433) (<http://dl.acm.org/citation.cfm?id=2488433>), *Proceedings of the 22nd international conference on World Wide Web*
4. H. Chen, A. G. Ororbia II, C. L. Giles [ExpertSeer: a Keyphrase Based Expert Recommender for Digital Libraries](http://arxiv.org/pdf/1511.02058.pdf) (<http://arxiv.org/pdf/1511.02058.pdf>), in *arXiv preprint 2015*
5. H. Chen, L. Gou, X. Zhang, C. Giles [Collabseer: a search engine for collaboration discovery](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.302.8659&rep=rep1&type=pdf) (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.302.8659&rep=rep1&type=pdf>), in *ACM/IEEE Joint Conference on Digital Libraries (JCDL) 2011*
6. Alexander Felfernig, Klaus Isak, Kalman Szabo, Peter Zachar, [The VITA Financial Services Sales Support Environment](http://www.aaai.org/Papers/AAAI/2007/AAAI07-274.pdf) (<http://www.aaai.org/Papers/AAAI/2007/AAAI07-274.pdf>), in *AAAI/IAAI 2007*, pp. 1692-1699, Vancouver, Canada, 2007.
7. Hosein Jafarkarimi; A.T.H. Sim and R. Saadatdoost [A Naïve Recommendation Model for Large Databases](http://www.ijiet.org/show-31-222-1.html) (<http://www.ijiet.org/show-31-222-1.html>), *International Journal of Information and Education Technology*, June 2012
8. Prem Melville and Vikas Sindhwani, [Recommender Systems](http://www.prem-melville.com/publications/recommender-systems-eml2010.pdf) (<http://www.prem-melville.com/publications/recommender-systems-eml2010.pdf>), *Encyclopedia of Machine Learning*, 2010.
9. R. J. Mooney & L. Roy (1999). *Content-based book recommendation using learning for text categorization*. In *Workshop Recom. Sys.: Algo. and Evaluation*.
10. Rubens, Neil; Elahi (<https://www.linkedin.com/in/mehdielahi>), Mehdi; Sugiyama, Masashi; Kaplan, Dain (2016). "Active Learning in Recommender Systems" (<https://rd.springer.com/book/10.1007/978-1-4899-7637-6>). In Ricci, Francesco; Rokach, Lior; Shapira, Bracha (eds.). *Recommender Systems Handbook* (2 ed.). Springer US. doi:10.1007/978-1-4899-7637-6_24 (https://doi.org/10.1007%2F978-1-4899-7637-6_24). ISBN 978-1-4899-7637-6.
11. Elahi (<https://www.linkedin.com/in/mehdielahi>), Mehdi; Ricci (<http://www.inf.unibz.it/~ricci/>), Francesco; Rubens, Neil (2016). "A survey of active learning in collaborative filtering recommender systems" (<https://www.researchgate.net/publication/303781992>). *Computer Science Review*. **20**: 29–50. doi:10.1016/j.cosrev.2016.05.002 (<https://doi.org/10.1016%2Fj.cosrev.2016.05.002>).
12. Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, David M. Pennock (2002). *Methods and Metrics for Cold-Start Recommendations* (<https://archive.org/details/sigir2002proceed0000inte>). *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*. New York City, New York: ACM. pp. 253–260. ISBN 1-58113-561-0. Retrieved 2008-02-02.
13. Karlgren, Jussi. 1990. "An Algebra for Recommendations." *Syslab Working Paper 179* (1990).
14. Karlgren, Jussi. "Newsgroup Clustering Based On User Behavior-A Recommendation Algebra (http://soda.swedish-ict.se/2225/2/T94_04.pdf)." *SICS Research Report* (1994).
15. Karlgren, Jussi (October 2017). "A digital bookshelf: original work on recommender systems" (<https://jussikarlgrén.wordpress.com/2017/10/01/a-digital-bookshelf-original-work-on-recommender-systems/>). Retrieved 27 October 2017.
16. Shardanand, Upendra, and Pattie Maes. "Social information filtering: algorithms for automating "word of mouth" (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.6583&rep=rep1&type=pdf>)." In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 210-217. ACM Press/Addison-Wesley Publishing Co., 1995.

17. Hill, Will, Larry Stead, Mark Rosenstein, and George Furnas. "Recommending and evaluating choices in a virtual community of use (<http://zhang.ist.psu.edu/teaching/501/readings/Hill.pdf>)." In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 194-201. ACM Press/Addison-Wesley Publishing Co., 1995.
18. Resnick, Paul, Neophytos Iacovou, Mitesh Suchak, Peter Bergström, and John Riedl. "GroupLens: an open architecture for collaborative filtering of netnews (<https://sites.ualberta.ca/~golmoham/SW/web%20mining%2023Jan2008/GroupLens%20An%20Open%20Architecture%20for%20Collaborating%20Filtering%20%20of%20Netnews.pdf>)." In Proceedings of the 1994 ACM conference on Computer supported cooperative work, pp. 175-186. ACM, 1994.
19. Resnick, Paul, and Hal R. Varian. "Recommender systems." *Communications of the ACM* 40, no. 3 (1997): 56-58.
20. Montaner, M.; Lopez, B.; de la Rosa, J. L. (June 2003). "A Taxonomy of Recommender Agents on the Internet". *Artificial Intelligence Review*. **19** (4): 285–330. doi:10.1023/A:1022850703159 (<https://doi.org/10.1023%2FA%3A1022850703159>)..
21. Adomavicius, G.; Tuzhilin, A. (June 2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions" (<http://portal.acm.org/citation.cfm?id=1070611.1070751>). *IEEE Transactions on Knowledge and Data Engineering*. **17** (6): 734–749. CiteSeerX 10.1.1.107.2790 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.107.2790>). doi:10.1109/TKDE.2005.99 (<https://doi.org/10.1109%2FTKDE.2005.99>)..
22. Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; Riedl, J. T. (January 2004). "Evaluating collaborative filtering recommender systems". *ACM Trans. Inf. Syst.* **22** (1): 5–53. CiteSeerX 10.1.1.78.8384 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.78.8384>). doi:10.1145/963770.963772 (<https://doi.org/10.1145%2F963770.963772>)..
23. Beel, J.; Genzmehr, M.; Gipp, B. (October 2013). "A Comparative Analysis of Offline and Online Evaluations and Discussion of Research Paper Recommender System Evaluation" (http://docear.org/papers/a_comparative_analysis_of_offline_and_online_evaluations_and_discussion_of_research_paper_recommender_system_evaluation.pdf) (PDF). *Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM Recommender System Conference (RecSys)*.
24. Beel, J.; Langer, S.; Genzmehr, M.; Gipp, B.; Breiting, C. (October 2013). "Research Paper Recommender System Evaluation: A Quantitative Literature Survey" (http://docear.org/papers/research_paper_recommender_system_evaluation--a_quantitative_literature_survey.pdf) (PDF). *Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM Recommender System Conference (RecSys)*.
25. Beel, J.; Gipp, B.; Langer, S.; Breiting, C. (26 July 2015). "Research Paper Recommender Systems: A Literature Survey" (<http://nbn-resolving.de/urn:nbn:de:bsz:352-0-311312>). *International Journal on Digital Libraries*. **17** (4): 305–338. doi:10.1007/s00799-015-0156-0 (<https://doi.org/10.1007%2Fs00799-015-0156-0>).
26. Waila, P.; Singh, V.; Singh, M. (26 April 2016). "A Scientometric Analysis of Research in Recommender Systems" (<http://www.jscires.org/sites/default/files/10.5530jscires.5.1.10.pdf>) (PDF). *Journal of Scientometric Research*. **5**: 71–84. doi:10.5530/jscires.5.1.10 (<https://doi.org/10.5530%2Fjscires.5.1.10>).
27. Stack, Charles. "System and method for providing recommendation of goods and services based on recorded purchasing history (<https://patentimages.storage.googleapis.com/pdfs/US6782370.pdf>)." U.S. Patent 7,222,085, issued May 22, 2007.
28. Herz, Frederick SM. "Customized electronic newspapers and advertisements." U.S. Patent 7,483,871, issued January 27, 2009.
29. Herz, Frederick, Lyle Ungar, Jian Zhang, and David Wachob. "System and method for providing access to data using customer profiles (<https://patentimages.storage.googleapis.com/3c/1b/54/5c6688e454a63a/US8056100.pdf>)." U.S. Patent 8,056,100, issued November 8, 2011.
30. Harbick, Andrew V., Ryan J. Snodgrass, and Joel R. Spiegel. "Playlist-based detection of similar digital works and work creators (<https://patents.google.com/patent/US8468046B2/en>)." U.S. Patent 8,468,046, issued June 18, 2013.
31. Linden, Gregory D., Brent Russell Smith, and Nida K. Zada. "Automated detection and exposure of behavior-based relationships between browsable items (<https://patentimages.storage.googleapis.com/3c/f6/13/d5f70fcdcf1b6d/US9070156.pdf>)." U.S. Patent 9,070,156, issued June 30, 2015.
32. John S. Breese; David Heckerman & Carl Kadie (1998). *Empirical analysis of predictive algorithms for collaborative filtering*. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI'98). arXiv:1301.7363 (<https://arxiv.org/abs/1301.7363>).

33. Dietmar Jannach; Markus Zanker; Alexander Felfernig; Gerhard Friedrich (2010). *Recommender Systems: An Introduction* (<https://web.archive.org/web/20150831032309/http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=9780521493369>). CUP. ISBN 978-0-521-49336-9. Archived from the original (<http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=9780521493369>) on 2015-08-31.
34. Breese, John S.; Heckerman, David; Kadie, Carl (1998). *Empirical Analysis of Predictive Algorithms for Collaborative Filtering* (<http://research.microsoft.com/pubs/69656/tr-98-12.pdf>) (PDF) (Report). Microsoft Research.
35. Ghazanfar, Mustansar Ali; Prügel-Bennett, Adam; Szedmak, Sandor (2012-11-15). "Kernel-Mapping Recommender system algorithms". *Information Sciences*. **208**: 81–104. CiteSeerX 10.1.1.701.7729 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.701.7729>). doi:10.1016/j.ins.2012.04.012 (<https://doi.org/10.1016%2Fj.ins.2012.04.012>).
36. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. (2000). "Application of Dimensionality Reduction in Recommender System A Case Study" (<http://glaros.dtc.umn.edu/gkhome/node/122>).,
37. Allen, R.B. (1990). "User Models: Theory, Method, Practice". International J. Man-Machine Studies.
38. Parsons, J.; Ralph, P.; Gallagher, K. (July 2004). "Using viewing time to infer user preference in recommender systems". AAAI Workshop in Semantic Web Personalization, San Jose, California. .
39. Sanghack Lee and Jihoon Yang and Sung-Yong Park, *Discovery of Hidden Similarity on Collaborative Filtering to Overcome Sparsity Problem* (https://books.google.com/books?id=u4qzLZAEjegC&pg=PA396&lpg=PA396&dq=sparsity+problem+content-based&source=bl&ots=ZEI50fmXTh&sig=jcwEba0LykRiLnYVbLYUwdrHYEo&hl=en&ei=-W7CTvboCleHsgKf9NHDBA&sa=X&oi=book_result&ct=result&resnum=1&ved=0CC0Q6AEwAA#v=onepage&q=sparsity%20problem%20content-based&f=false), Discovery Science, 2007.
40. Collaborative Recommendations Using Item-to-Item Similarity Mappings (<http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=/netahtml/PTO/search-bool.html&r=1&f=G&l=50&co1=AND&d=PTXT&s1=6,266,649.PN.&OS=PN/6,266,649&RS=PN/6,266,649>) Archived (<https://web.archive.org/web/20150316185024/http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fmetahtml%2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&co1=AND&d=PTXT&s1=6%2C266%2C649.PN.&OS=PN%2F6%2C266%2C649&RS=PN%2F6%2C266%2C649>) 2015-03-16 at the *Wayback Machine*
41. Aggarwal, Charu C. (2016). *Recommender Systems: The Textbook*. Springer. ISBN 9783319296579.
42. Peter Brusilovsky (2007). *The Adaptive Web*. p. 325. ISBN 978-3-540-72078-2.
43. D.H. Wang, Y.C. Liang, D.Xu, X.Y. Feng, R.C. Guan(2018), "A content-based recommender system for computer science publications (<https://www.sciencedirect.com/science/article/pii/S0950705118302107>)", *Knowledge-Based Systems*, 157: 1-9
44. Blanda, Stephanie (May 25, 2015). "Online Recommender Systems – How Does a Website Know What I Want?" (<http://blogs.ams.org/mathgradblog/2015/05/25/online-recommender-systems-website-want/>). *American Mathematical Society*. Retrieved October 31, 2016.
45. X.Y. Feng, H. Zhang, Y.J. Ren, P.H. Shang, Y. Zhu, Y.C. Liang, R.C. Guan, D. Xu, (2019), "The Deep Learning–Based Recommender System “Pubmender” for Choosing a Biomedical Publication Venue: Development and Validation Study (<https://www.jmir.org/2019/5/e12957/>)", *Journal of medical Internet research*, 21 (5): e12957
46. Lakiotaki, K.; Matsatsinis; Tsoukias, A (March 2011). "Multicriteria User Modeling in Recommender Systems". *IEEE Intelligent Systems*. **26** (2): 64–76. CiteSeerX 10.1.1.476.6726 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.476.6726>). doi:10.1109/mis.2011.33 (<https://doi.org/10.1109%2Fmis.2011.33>).
47. Gediminas Adomavicius, Nikos Manouselis, YoungOk Kwon. "Multi-Criteria Recommender Systems" (<https://web.archive.org/web/20140630021251/http://ids.csom.umn.edu/faculty/gedas/NSFCareer/MCRS-chapter-2010.pdf>) (PDF). Archived from the original (<http://ids.csom.umn.edu/faculty/gedas/NSFCareer/MCRS-chapter-2010.pdf>) (PDF) on 2014-06-30.
48. Bouneffouf, Djallel (2013), *DRARS, A Dynamic Risk-Aware Recommender System* (<http://tel.archives-ouvertes.fr/tel-01026136/fr/>) (Ph.D.), Institut National des Télécommunications
49. Yong Ge; Hui Xiong; Alexander Tuzhilin; Keli Xiao; Marco Gruteser; Michael J. Pazzani (2010). *An Energy-Efficient Mobile Recommender System* (<http://www.winlab.rutgers.edu/~gruteser/papers/KDD10.pdf>) (PDF). Proceedings of the 16th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York City, New York: ACM. pp. 899–908. Retrieved 2011-11-17.

50. Pimenidis, Elias; Polatidis, Nikolaos; Mouratidis, Haralambos (3 August 2018). "Mobile recommender systems: Identifying the major concepts". *Journal of Information Science*. **45** (3): 387–397. arXiv:1805.02276 (<https://arxiv.org/abs/1805.02276>). doi:10.1177/0165551518792213 (<https://doi.org/10.1177%2F0165551518792213>).
51. Tuukka Ruotsalo; Krister Haav; Antony Stoyanov; Sylvain Roche; Elena Fani; Romina Deliai; Eetu Mäkelä; Tomi Kauppinen; Eero Hyvönen (2013). "SMARTMUSEUM: A Mobile Recommender System for the Web of Data". *Web Semantics: Science, Services and Agents on the World Wide Web*. **20**: 657–662. CiteSeerX 10.1.1.676.7109 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.676.7109>). doi:10.1016/j.websem.2013.03.001 (<https://doi.org/10.1016%2Fj.websem.2013.03.001>).
52. Rinke Hoekstra, *The Knowledge Reengineering Bottleneck* (<http://www.semantic-web-journal.net/sites/default/files/swj32.pdf>), Semantic Web – Interoperability, Usability, Applicability 1 (2010) 1, IOS Press
53. Gomez-Urbe, Carlos A.; Hunt, Neil (28 December 2015). "The Netflix Recommender System". *ACM Transactions on Management Information Systems*. **6** (4): 1–19. doi:10.1145/2843948 (<https://doi.org/10.1145%2F2843948>).
54. Robin Burke, *Hybrid Web Recommender Systems* (<http://www.dcs.warwick.ac.uk/~acristea/courses/CS411/2010/Book%20-%20The%20Adaptive%20Web/HybridWebRecommenderSystems.pdf>) Archived (<https://web.archive.org/web/20140912085014/http://www.dcs.warwick.ac.uk/~acristea/courses/CS411/2010/Book%20-%20The%20Adaptive%20Web/HybridWebRecommenderSystems.pdf>) 2014-09-12 at the Wayback Machine, pp. 377-408, The Adaptive Web, Peter Brusilovsky, Alfred Kobsa, Wolfgang Nejdl (Ed.), Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, Lecture Notes in Computer Science, Vol. 4321, May 2007, 978-3-540-72078-2.
55. Lohr, Steve. "A \$1 Million Research Bargain for Netflix, and Maybe a Model for Others" (<https://www.nytimes.com/2009/09/22/technology/internet/22netflix.html>). *The New York Times*.
56. R. Bell; Y. Koren; C. Volinsky (2007). "The BellKor solution to the Netflix Prize" (http://www.netflixprize.com/assets/ProgressPrize2007_KorBell.pdf) (PDF).
57. Bodoky, Thomas (2009-08-06). "Mátrixfaktorizáció one million dollars" (http://index.hu/tech/net/2009/08/07/matrixfaktorizacio_egymillio_dollarert/). *Index*.
58. Rise of the Netflix Hackers (<https://www.wired.com/science/discoveries/news/2007/03/72963>) Archived (<https://web.archive.org/web/20120124011808/http://www.wired.com/science/discoveries/news/2007/03/72963>) January 24, 2012, at the Wayback Machine
59. "Netflix Spilled Your Brokeback Mountain Secret, Lawsuit Claims" (<https://www.wired.com/threatlevel/2009/12/netflix-privacy-lawsuit/>). *WIRED*. 17 December 2009. Retrieved 1 June 2015.
60. "Netflix Prize Update" (<http://blog.netflix.com/2010/03/this-is-neil-hunt-chief-product-officer.html>). Netflix Prize Forum. 2010-03-12.
61. Lathia, N., Hailes, S., Capra, L., Amatriain, X.: *Temporal diversity in recommender systems* (http://www.academia.edu/download/46585553/lathia_sigir10.pdf). In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010*, pp. 210–217. ACM, New York
62. Turpin, Andrew H, Hersh, William (2001). "Why batch and user evaluations do not give the same results". *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 225–231.
63. "MovieLens dataset" (<https://grouplens.org/datasets/movielens/>). 2013-09-06.
64. Jannach, Dietmar; Lerche, Lukas; Gedikli, Fatih; Bonnin, Geoffray (2013-06-10). Carberry, Sandra; Weibelzahl, Stephan; Micarelli, Alessandro; Semeraro, Giovanni (eds.). *User Modeling, Adaptation, and Personalization*. Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp. 25–37. CiteSeerX 10.1.1.465.96 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.465.96>). doi:10.1007/978-3-642-38844-6_3 (https://doi.org/10.1007%2F978-3-642-38844-6_3). ISBN 9783642388439.
65. Turpin, Andrew H.; Hersh, William (2001-01-01). *Why Batch and User Evaluations Do Not Give the Same Results* (<https://archive.org/details/proceedingsof24t0000inte>). *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '01. New York, NY, USA: ACM. pp. 225–231. CiteSeerX 10.1.1.165.5800 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.165.5800>). doi:10.1145/383952.383992 (<https://doi.org/10.1145%2F383952.383992>). ISBN 978-1581133318.

66. Langer, Stefan (2015-09-14). "A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems". In Kapidakis, Sarantos; Mazurek, Cezary; Werla, Marcin (eds.). *Research and Advanced Technology for Digital Libraries*. Lecture Notes in Computer Science. **9316**. Springer International Publishing. pp. 153–168. doi:[10.1007/978-3-319-24592-8_12](https://doi.org/10.1007/978-3-319-24592-8_12) (https://doi.org/10.1007/978-3-319-24592-8_12). ISBN 9783319245911.
67. Basaran, Daniel; Ntoutsis, Eirini; Zimek, Arthur (2017). *Proceedings of the 2017 SIAM International Conference on Data Mining*. pp. 390–398. doi:[10.1137/1.9781611974973.44](https://doi.org/10.1137/1.9781611974973.44) (<https://doi.org/10.1137/1.9781611974973.44>). ISBN 978-1-61197-497-3.
68. Beel, Joeran; Genzmehr, Marcel; Langer, Stefan; Nürnberger, Andreas; Gipp, Bela (2013-01-01). *A Comparative Analysis of Offline and Online Evaluations and Discussion of Research Paper Recommender System Evaluation*. *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*. RepSys '13. New York, NY, USA: ACM. pp. 7–14. CiteSeerX [10.1.1.1031.973](https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1031.973) (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1031.973>). doi:[10.1145/2532508.2532511](https://doi.org/10.1145/2532508.2532511) (<https://doi.org/10.1145/2532508.2532511>). ISBN 9781450324656.
69. Ziegler, C.N., McNeel, S.M., Konstan, J.A. and Lausen, G. (2005). "Improving recommendation lists through topic diversification". *Proceedings of the 14th international conference on World Wide Web*. pp. 22–32.
70. Joeran Beel; Stefan Langer; Marcel Genzmehr; Andreas Nürnberger (September 2013). "Persistence in Recommender Systems: Giving the Same Recommendations to the Same Users Multiple Times" (http://docear.org/papers/persistence_in_recommender_systems_-_giving_the_same_recommendations_to_the_same_users_multiple_times.pdf) (PDF). In Trond Aalberg; Milena Dobrev; Christos Papatheodorou; Giannis Tsakonas; Charles Farrugia (eds.). *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013)*. Lecture Notes of Computer Science (LNCS). **8092**. Springer. pp. 390–394. Retrieved 1 November 2013.
71. Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., Riedl, J. (2003). "Is seeing believing?: how recommender system interfaces affect users' opinions" (<https://pdfs.semanticscholar.org/d7d5/47012091d11ba0b0bf4a6630c5689789c22e.pdf>) (PDF). *Proceedings of the SIGCHI conference on Human factors in computing systems*. pp. 585–592.
72. {P}u, {P}., {C}hen, {L}., {H}u, {R}. (2012). "Evaluating recommender systems from the user's perspective: survey of the state of the art" (http://doc.rero.ch/record/317166/files/11257_2011_Article_9115.pdf) (PDF). *User Modeling and User-Adapted Interaction*: 1–39.
73. Naren Ramakrishnan; Benjamin J. Keller; Batul J. Mirza; Ananth Y. Grama; George Karypis (2001). *Privacy Risks in Recommender Systems* (<https://archive.org/details/sigir2002proceed0000inte>). *IEEE Internet Computing*. **5**. Piscataway, NJ: IEEE Educational Activities Department. pp. 54–62. CiteSeerX [10.1.1.2.2932](https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.2932) (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.2932>). doi:[10.1109/4236.968832](https://doi.org/10.1109/4236.968832) (<https://doi.org/10.1109/4236.968832>). ISBN 978-1-58113-561-9.
74. Joeran Beel; Stefan Langer; Andreas Nürnberger; Marcel Genzmehr (September 2013). "The Impact of Demographics (Age and Gender) and Other User Characteristics on Evaluating Recommender Systems" (http://docear.org/papers/the_impact_of_users_demographics_age_and_gender_and_other_characteristics_on_evaluating_recommender_systems.pdf) (PDF). In Trond Aalberg; Milena Dobrev; Christos Papatheodorou; Giannis Tsakonas; Charles Farrugia (eds.). *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013)*. Springer. pp. 400–404. Retrieved 1 November 2013.
75. {K}onstan, {J}., {A}., {R}iedl, {J}. (2012). "Recommender systems: from algorithms to user experience" (<https://link.springer.com/content/pdf/10.1007/s11257-011-9112-x.pdf>) (PDF). *User Modeling and User-Adapted Interaction*. **22** (1–2): 1–23. doi:[10.1007/s11257-011-9112-x](https://doi.org/10.1007/s11257-011-9112-x) (<https://doi.org/10.1007/s11257-011-9112-x>).
76. {R}icci, {F}., {R}okach, {L}., {S}hapira, {B}., {K}antor {B}. {P}. (2011). "Recommender systems handbook". *Recommender Systems Handbook*: 1–35.
77. Montaner, Miquel, Lloppez, Beatriz, de la Rosa, Josep Lluís (2002). "Developing trust in recommender agents" (<https://www.researchgate.net/publication/221454720>). *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. pp. 304–305.

78. Beel, Joeran, Langer, Stefan, Genzmehr, Marcel (September 2013). "Sponsored vs. Organic (Research Paper) Recommendations and the Impact of Labeling" ([http://docear.org/papers/sponsored_vs._organic_\(research_paper\)_recommendations_and_the_impact_of_labeling.pdf](http://docear.org/papers/sponsored_vs._organic_(research_paper)_recommendations_and_the_impact_of_labeling.pdf)) (PDF). In Trond Aalberg; Milena Dobрева; Christos Papatheodorou; Giannis Tsakonas; Charles Farrugia (eds.). *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013)*. pp. 395–399. Retrieved 2 December 2013.
79. Ekstrand, Michael D.; Ludwig, Michael; Konstan, Joseph A.; Riedl, John T. (2011-01-01). *Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit*. *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys '11. New York, NY, USA: ACM. pp. 133–140. doi:10.1145/2043932.2043958 (<https://doi.org/10.1145%2F2043932.2043958>). ISBN 9781450306836.
80. Konstan, Joseph A.; Adomavicius, Gediminas (2013-01-01). *Toward Identification and Adoption of Best Practices in Algorithmic Recommender Systems Research*. *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*. RepSys '13. New York, NY, USA: ACM. pp. 23–28. doi:10.1145/2532508.2532513 (<https://doi.org/10.1145%2F2532508.2532513>). ISBN 9781450324656.
81. Breiteringer, Corinna; Langer, Stefan; Lommatzsch, Andreas; Gipp, Bela (2016-03-12). "Towards reproducibility in recommender-systems research" (<http://nbn-resolving.de/urn:nbn:de:bsz:352-0-324818>). *User Modeling and User-Adapted Interaction*. 26 (1): 69–101. doi:10.1007/s11257-016-9174-x (<https://doi.org/10.1007%2Fs11257-016-9174-x>). ISSN 0924-1868 (<https://www.worldcat.org/issn/0924-1868>).
82. Said, Alan; Bellogín, Alejandro (2014-10-01). *Comparative recommender system evaluation: benchmarking recommendation frameworks*. *Proceedings of the 8th ACM Conference on Recommender Systems*. RecSys '14. New York, NY, USA: ACM. pp. 129–136. doi:10.1145/2645710.2645746 (<https://doi.org/10.1145%2F2645710.2645746>). hdl:10486/665450 (<https://hdl.handle.net/10486%2F665450>). ISBN 9781450326681.

Further reading

Books

Kim Falk (January 2019), *Practical Recommender Systems*, Manning Publications, ISBN 9781617292705

- Bharat Bhasker; K. Srikumar (2010). *Recommender Systems in E-Commerce* (<https://web.archive.org/web/20100901164550/http://www.tatamcgrawhill.com/html/9780070680678.html>). CUP. ISBN 978-0-07-068067-8. Archived from the original (<http://www.tatamcgrawhill.com/html/9780070680678.html>) on 2010-09-01.
- Francesco Ricci; Lior Rokach; Bracha Shapira; Paul B. Kantor, eds. (2011). *Recommender Systems Handbook* (<http://www.springer.com/computer/ai/book/978-0-387-85819-7>). Springer. ISBN 978-0-387-85819-7.
- Bracha Shapira; Lior Rokach (June 2012). *Building Effective Recommender Systems* (<https://archive.is/20140501234448/http://www.springer.com/computer/ai/book/978-1-4419-0048-7>). ISBN 978-1-4419-0047-0. Archived from the original (<https://www.springer.com/computer/ai/book/978-1-4419-0048-7>) on 2014-05-01.
- Dietmar Jannach; Markus Zanker; Alexander Felfernig; Gerhard Friedrich (2010). *Recommender Systems: An Introduction* (<https://web.archive.org/web/20150831032309/http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=9780521493369>). CUP. ISBN 978-0-521-49336-9. Archived from the original (<http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=9780521493369>) on 2015-08-31.

Scientific articles

- Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. (2002) *Content-Boosted Collaborative Filtering for Improved Recommendations*. (<http://www.cs.utexas.edu/users/ml/papers/cbcbf-aaai-02.pdf>) *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, pp. 187–192, Edmonton, Canada, July 2002.
- Frank Meyer. (2012), Recommender systems in industrial contexts. ArXiv e-prints. (<https://arxiv.org/abs/1203.4487>)
- Bouneffouf, Djallel (2012), "Following the User's Interests in Mobile Context-Aware Recommender Systems: The Hybrid-e-greedy Algorithm", *Proceedings of the 2012 26th International Conference on Advanced Information Networking and Applications Workshops* (https://web.archive.org/web/20140514042848/http://www.inf.int-evry.fr:80/~bouneff_d/index_fichiers/Hybrid-e-greedy%20for%20Mobile%20Context-aware%20Recommender%20System.pdf) (PDF), Lecture Notes in Computer Science, IEEE Computer Society, pp. 657–662, ISBN 978-0-7695-4652-0, archived from the original (http://www.inf.int-evry.fr/~bouneff_d/index_fichiers/Hybrid-e-greedy%20for%20Mobile%20Context-aware%20Recommender%20System.pdf) (PDF) on 2014-05-14.

- Bouneffouf, Djallel (2013), *DRARS, A Dynamic Risk-Aware Recommender System* (<http://tel.archives-ouvertes.fr/tel-01026136/fr/>) (Ph.D.), Institut National des Télécommunications.

External links

- Robert M. Bell; Jim Bennett; Yehuda Koren & Chris Volinsky (May 2009). "The Million Dollar Programming Prize" (<http://web.archive.org/web/20090511144610/http://www.spectrum.ieee.org/may09/8788>). *IEEE Spectrum*. Archived from the original (<http://www.spectrum.ieee.org/may09/8788>) on 2009-05-11. Retrieved 2018-12-10.
 - Hangartner, Rick, "What is the Recommender Industry?" (<https://web.archive.org/web/20071219144013/http://www.msearchgroove.com/2007/12/17/guest-column-what-is-the-recommender-industry/>), MSearchGroove, December 17, 2007.
 - ACM Conference on Recommender Systems (<http://recsys.acm.org/>)
 - Recsys group at Politecnico di Milano (<http://recsys.deib.polimi.it/>)
 - Data Science: Data to Insights from MIT (recommendation systems) (https://web.archive.org/web/20170118041300/https://mitprofessionalx.mit.edu/courses/course-v1:MITProfessionalX+DSx+2016_T1/about)
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Recommender_system&oldid=919588155"

This page was last edited on 4 October 2019, at 14:44 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.