

Collaborative filtering

Collaborative filtering (CF) is a technique used by recommender systems.^[1] Collaborative filtering has two senses, a narrow one and a more general one.^[2]

In the newer, narrower sense, collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). The underlying assumption of the collaborative filtering approach is that if a person *A* has the same opinion as a person *B* on an issue, *A* is more likely to have *B*'s opinion on a different issue than that of a randomly chosen person. For example, a collaborative filtering recommendation system for preferences in television programming could make predictions about which television show a user should like given a partial list of that user's tastes (likes or dislikes).^[3] Note that these predictions are specific to the user, but use information gleaned from many users. This differs from the simpler approach of giving an average (non-specific) score for each item of interest, for example based on its number of votes.

In the more general sense, collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc.^[2] Applications of collaborative filtering typically involve very large data sets. Collaborative filtering methods have been applied to many different kinds of data including: sensing and monitoring data, such as in mineral exploration, environmental sensing over large areas or multiple sensors; financial data, such as financial service institutions that integrate many financial sources; or in electronic commerce and web applications where the focus is on user data, etc. The remainder of this discussion focuses on collaborative filtering for user data, although some of the methods and approaches may apply to the other major applications as well.



This image shows an example of predicting of the user's rating using collaborative filtering. At first, people rate different items (like videos, images, games). After that, the system is making predictions about user's rating for an item, which the user hasn't rated yet. These predictions are built upon the existing ratings of other users, who have similar ratings with the active user. For instance, in our case the system has made a prediction, that the active user won't like the video.

Contents

Overview

Methodology

Types

Memory-based

Model-based

Hybrid

Deep-Learning

Context-aware collaborative filtering

Application on social web

Problems

Challenges

Data sparsity

Scalability

Synonyms

Gray sheep

Shilling attacks

Diversity and the long tail

Innovations

Auxiliary information

See also

References

External links

Overview

The growth of the Internet has made it much more difficult to effectively extract useful information from all the available online information. The overwhelming amount of data necessitates mechanisms for efficient information filtering. Collaborative filtering is one of the techniques used for dealing with this problem.

The motivation for collaborative filtering comes from the idea that people often get the best recommendations from someone with tastes similar to themselves. Collaborative filtering encompasses techniques for matching people with similar interests and making recommendations on this basis.

Collaborative filtering algorithms often require (1) users' active participation, (2) an easy way to represent users' interests, and (3) algorithms that are able to match people with similar interests.

Typically, the workflow of a collaborative filtering system is:

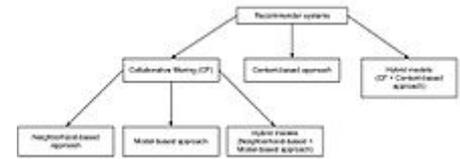
1. A user expresses his or her preferences by rating items (e.g. books, movies, or music recordings) of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
2. The system matches this user's ratings against other users' and finds the people with most "similar" tastes.
3. With similar users, the system recommends items that the similar users have rated highly but not yet being rated by this user (presumably the absence of rating is often considered as the unfamiliarity of an item)

A key problem of collaborative filtering is how to combine and weight the preferences of user neighbors. Sometimes, users can immediately rate the recommended items. As a result, the system gains an increasingly accurate representation of user preferences over time.

Methodology

Collaborative filtering systems have many forms, but many common systems can be reduced to two steps:

1. Look for users who share the same rating patterns with the active user (the user whom the prediction is for).
2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user



Collaborative Filtering in Recommender Systems

This falls under the category of user-based collaborative filtering. A specific application of this is the user-based Nearest Neighbor algorithm.

Alternatively, item-based collaborative filtering (users who bought x also bought y), proceeds in an item-centric manner:

1. Build an item-item matrix determining relationships between pairs of items
2. Infer the tastes of the current user by examining the matrix and matching that user's data

See, for example, the Slope One item-based collaborative filtering family.

Another form of collaborative filtering can be based on implicit observations of normal user behavior (as opposed to the artificial behavior imposed by a rating task). These systems observe what a user has done together with what all users have done (what music they have listened to, what items they have bought) and use that data to predict the user's behavior in the future, or to predict how a user might like to behave given the chance. These predictions then have to be filtered through business logic to determine how they might affect the actions of a business system. For example, it is not useful to offer to sell somebody a particular album of music if they already have demonstrated that they own that music.

Relying on a scoring or rating system which is averaged across all users ignores specific demands of a user, and is particularly poor in tasks where there is large variation in interest (as in the recommendation of music). However, there are other methods to combat information explosion, such as web search and data clustering.

Types

Memory-based

The memory-based approach uses user rating data to compute the similarity between users or items. Typical examples of this approach are neighbourhood-based CF and item-based/user-based top-N recommendations. For example, in user based approaches, the value of ratings user u gives to item i is calculated as an aggregation of some similar users' rating of the item:

$$r_{u,i} = \text{aggr}_{u' \in U} r_{u',i}$$

where U denotes the set of top N users that are most similar to user u who rated item i . Some examples of the aggregation function include:

$$r_{u,i} = \frac{1}{N} \sum_{u' \in U} r_{u',i}$$

$$r_{u,i} = k \sum_{u' \in U} \text{simil}(u, u') r_{u',i}$$

where k is a normalizing factor defined as $k = 1 / \sum_{u' \in U} |\text{simil}(u, u')|$, and

$$r_{u,i} = \bar{r}_u + k \sum_{u' \in U} \text{simil}(u, u') (r_{u',i} - \bar{r}_{u'})$$

where \bar{r}_u is the average rating of user u for all the items rated by u .

The neighborhood-based algorithm calculates the similarity between two users or items, and produces a prediction for the user by taking the weighted average of all the ratings. Similarity computation between items or users is an important part of this approach. Multiple measures, such as Pearson correlation and vector cosine based similarity are used for this.

The Pearson correlation similarity of two users x, y is defined as

$$\text{simil}(x, y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)(r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r}_x)^2} \sqrt{\sum_{i \in I_{xy}} (r_{y,i} - \bar{r}_y)^2}}$$

where I_{xy} is the set of items rated by both user x and user y .

The cosine-based approach defines the cosine-similarity between two users x and y as:^[4]

$$\text{simil}(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \times \|\vec{y}\|} = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_x} r_{x,i}^2} \sqrt{\sum_{i \in I_y} r_{y,i}^2}}$$

The user based top-N recommendation algorithm uses a similarity-based vector model to identify the k most similar users to an active user. After the k most similar users are found, their corresponding user-item matrices are aggregated to identify the set of items to be recommended. A popular method to find the similar users is the Locality-sensitive hashing, which implements the nearest neighbor mechanism in linear time.

The advantages with this approach include: the explainability of the results, which is an important aspect of recommendation systems; easy creation and use; easy facilitation of new data; content-independence of the items being recommended; good scaling with co-rated items.

There are also several disadvantages with this approach. Its performance decreases when data gets sparse, which occurs frequently with web-related items. This hinders the scalability of this approach and creates problems with large datasets. Although it can efficiently handle new users because it relies on a data structure, adding new items becomes more complicated since that representation usually relies on a specific vector space. Adding new items requires inclusion of the new item and the re-insertion of all the elements in the structure.

Model-based

In this approach, models are developed using different data mining, machine learning algorithms to predict users' rating of unrated items. There are many model-based CF algorithms. Bayesian networks, clustering models, latent semantic models such as singular value decomposition, probabilistic latent semantic analysis,

multiple multiplicative factor, latent Dirichlet allocation and Markov decision process based models.^[5]

Through this approach, dimensionality reduction methods are mostly being used as complementary technique to improve robustness and accuracy of memory-based approach. In this sense, methods like singular value decomposition, principal component analysis, known as latent factor models, compress user-item matrix into a low-dimensional representation in terms of latent factors. One advantage of using this approach is that instead of having a high dimensional matrix containing abundant number of missing values we will be dealing with a much smaller matrix in lower-dimensional space. A reduced presentation could be utilized for either user-based or item-based neighborhood algorithms that are presented in the previous section. There are several advantages with this paradigm. It handles the sparsity of the original matrix better than memory based ones. Also comparing similarity on the resulting matrix is much more scalable especially in dealing with large sparse datasets.^[6]

Hybrid

A number of applications combine the memory-based and the model-based CF algorithms. These overcome the limitations of native CF approaches and improve prediction performance. Importantly, they overcome the CF problems such as sparsity and loss of information. However, they have increased complexity and are expensive to implement.^[7] Usually most commercial recommender systems are hybrid, for example, the Google news recommender system.^[8]

Deep-Learning

In recent years a number of neural and deep-learning techniques have been proposed. Some generalize traditional Matrix factorization algorithms via a non-linear neural architecture,^[9] or leverage new model types like Variational Autoencoders.^[10] While deep learning has been applied to many different scenarios: context-aware, sequence-aware, social tagging etc. its real effectiveness when used in a simple collaborative recommendation scenario has been put into question. A systematic analysis of publications applying deep learning or neural methods to the top-k recommendation problem, published in top conferences (SIGIR, KDD, WWW, RecSys), has shown that on average less than 40% of articles are reproducible, with as little as 14% in some conferences. Overall the study identifies 18 articles, only 7 of them could be reproduced and 6 of them could be outperformed by much older and simpler properly tuned baselines. The article also highlights a number of potential problems in today's research scholarship and calls for improved scientific practices in that area.^[11] Similar issues have been spotted also in sequence-aware recommender systems.^[12]

Context-aware collaborative filtering

Many recommender systems simply ignore other contextual information existing alongside user's rating in providing item recommendation.^[13] However, by pervasive availability of contextual information such as time, location, social information, and type of the device that user is using, it is becoming more important than ever for a successful recommender system to provide a context-sensitive recommendation. According to Charu Aggrawal, "Context-sensitive recommender systems tailor their recommendations to additional information that defines the specific situation under which recommendations are made. This additional information is referred to as the context."^[6]

Taking contextual information into consideration, we will have additional dimension to the existing user-item rating matrix. As an instance, assume a music recommender system which provide different recommendations in corresponding to time of the day. In this case, it is possible a user have different

preferences for a music in different time of a day. Thus, instead of using user-item matrix, we may use tensor of order 3 (or higher for considering other contexts) to represent context-sensitive users' preferences.^{[14][15][16]}

In order to take advantage of collaborative filtering and particularly neighborhood-based methods, approaches can be extended from a two-dimensional rating matrix into a tensor of higher order. For this purpose, the approach is to find the most similar/like-minded users to a target user; one can extract and compute similarity of slices (e.g. item-time matrix) corresponding to each user. Unlike the context-insensitive case for which similarity of two rating vectors are calculated, in the context-aware approaches, the similarity of rating matrices corresponding to each user is calculated by using Pearson coefficients.^[6] After the most like-minded users are found, their corresponding ratings are aggregated to identify the set of items to be recommended to the target user.

The most important disadvantage of taking context into recommendation model is to be able to deal with larger dataset that contains much more missing values in comparison to user-item rating matrix. Therefore, similar to matrix factorization methods, tensor factorization techniques can be used to reduce dimensionality of original data before using any neighborhood-based methods.

Application on social web

Unlike the traditional model of mainstream media, in which there are few editors who set guidelines, collaboratively filtered social media can have a very large number of editors, and content improves as the number of participants increases. Services like Reddit, YouTube, and Last.fm are typical examples of collaborative filtering based media.^[17]

One scenario of collaborative filtering application is to recommend interesting or popular information as judged by the community. As a typical example, stories appear in the front page of Reddit as they are "voted up" (rated positively) by the community. As the community becomes larger and more diverse, the promoted stories can better reflect the average interest of the community members.

Wikipedia is another application of collaborative filtering. Volunteers contribute to the encyclopedia by filtering out facts from falsehoods.^[18]

Another aspect of collaborative filtering systems is the ability to generate more personalized recommendations by analyzing information from the past activity of a specific user, or the history of other users deemed to be of similar taste to a given user. These resources are used as user profiling and helps the site recommend content on a user-by-user basis. The more a given user makes use of the system, the better the recommendations become, as the system gains data to improve its model of that user.

Problems

A collaborative filtering system does not necessarily succeed in automatically matching content to one's preferences. Unless the platform achieves unusually good diversity and independence of opinions, one point of view will always dominate another in a particular community. As in the personalized recommendation scenario, the introduction of new users or new items can cause the cold start problem, as there will be insufficient data on these new entries for the collaborative filtering to work accurately. In order to make appropriate recommendations for a new user, the system must first learn the user's preferences by analysing past voting or rating activities. The collaborative filtering system requires a substantial number of users to rate a new item before that item can be recommended.

Challenges

Data sparsity

In practice, many commercial recommender systems are based on large datasets. As a result, the user-item matrix used for collaborative filtering could be extremely large and sparse, which brings about challenges in the performance of the recommendation.

One typical problem caused by the data sparsity is the cold start problem. As collaborative filtering methods recommend items based on users' past preferences, new users will need to rate a sufficient number of items to enable the system to capture their preferences accurately and thus provides reliable recommendations.

Similarly, new items also have the same problem. When new items are added to the system, they need to be rated by a substantial number of users before they could be recommended to users who have similar tastes to the ones who rated them. The new item problem does not affect content-based recommendations, because the recommendation of an item is based on its discrete set of descriptive qualities rather than its ratings.

Scalability

As the numbers of users and items grow, traditional CF algorithms will suffer serious scalability problems. For example, with tens of millions of customers $O(M)$ and millions of items $O(N)$, a CF algorithm with the complexity of n is already too large. As well, many systems need to react immediately to online requirements and make recommendations for all users regardless of their millions of users, with most computations happening in very large memory machines.^[19]

Synonyms

Synonyms refers to the tendency of a number of the same or very similar items to have different names or entries. Most recommender systems are unable to discover this latent association and thus treat these products differently.

For example, the seemingly different items "children's movie" and "children's film" are actually referring to the same item. Indeed, the degree of variability in descriptive term usage is greater than commonly suspected. The prevalence of synonyms decreases the recommendation performance of CF systems. Topic Modeling (like the Latent Dirichlet Allocation technique) could solve this by grouping different words belonging to the same topic.

Gray sheep

Gray sheep refers to the users whose opinions do not consistently agree or disagree with any group of people and thus do not benefit from collaborative filtering. Black sheep are a group whose idiosyncratic tastes make recommendations nearly impossible. Although this is a failure of the recommender system, non-electronic recommenders also have great problems in these cases, so having black sheep is an acceptable failure.

Shilling attacks

In a recommendation system where everyone can give the ratings, people may give many positive ratings for their own items and negative ratings for their competitors'. It is often necessary for the collaborative filtering systems to introduce precautions to discourage such manipulations.

Diversity and the long tail

Collaborative filters are expected to increase diversity because they help us discover new products. Some algorithms, however, may unintentionally do the opposite. Because collaborative filters recommend products based on past sales or ratings, they cannot usually recommend products with limited historical data. This can create a rich-get-richer effect for popular products, akin to positive feedback. This bias toward popularity can prevent what are otherwise better consumer-product matches. A Wharton study details this phenomenon along with several ideas that may promote diversity and the "long tail".^[20] Several collaborative filtering algorithms have been developed to promote diversity and the "long tail"^[21] by recommending novel, unexpected,^[22] and serendipitous items.^[23]

Innovations

- New algorithms have been developed for CF as a result of the Netflix prize.
- Cross-System Collaborative Filtering where user profiles across multiple recommender systems are combined in a multitask manner; this way, preference pattern sharing is achieved across models..^[24]
- Robust collaborative filtering, where recommendation is stable towards efforts of manipulation. This research area is still active and not completely solved.^[25]

Auxiliary information

User-item matrix is a basic foundation of traditional collaborative filtering techniques, and it suffers from data sparsity problem (i.e. cold start). As a consequence, except for user-item matrix, researchers are trying to gather more auxiliary information to help boost recommendation performance and develop personalized recommender systems.^[26] Generally, there are two popular auxiliary information: attribute information and interaction information. Attribute information describes a user's or an item's properties. For example, user attribute might include general profile (e.g. gender and age) and social contacts (e.g. followers or friends in social networks); Item attribute means properties like category, brand or content. In addition, interaction information refers to the implicit data showing how users interplay with the item. Widely used interaction information contains tags, comments or reviews and browsing history etc. Auxiliary information plays a significant role in a variety of aspects. Explicit social links, as a reliable representative of trust or friendship, is always employed in similarity calculation to find similar persons who share interest with the target user.^{[27][28]} The interaction-associated information - tags - is taken as a third dimension (in addition to user and item) in advanced collaborative filtering to construct a 3-dimensional tensor structure for exploration of recommendation.^[29]

See also

- Attention Profiling Mark-up Language (APML)
- Cold start
- Collaborative model
- Collaborative search engine
- Collective intelligence

- Customer engagement
- Delegative Democracy, the same principle applied to voting rather than filtering
- Enterprise bookmarking
- Firefly (website), a defunct website which was based on collaborative filtering
- Filter bubble
- Page rank
- Preference elicitation
- Psychographic filtering
- Recommendation system
- Relevance (information retrieval)
- Reputation system
- Robust collaborative filtering
- Similarity search
- Slope One
- Social translucence

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>), Recommender Systems Handbook, Springer, 2011, pp. 1-35
2. Terveen, Loren; Hill, Will (2001). "Beyond Recommender Systems: Helping People Help Each Other" (<http://www.grouplens.org/papers/pdf/rec-sys-overview.pdf>) (PDF). Addison-Wesley. p. 6. Retrieved 16 January 2012.
3. An integrated approach to TV & VOD Recommendations (<http://www.redbeemedia.com/insights/integrated-approach-tv-vod-recommendations>) Archived (<https://web.archive.org/web/20120606225352/http://www.redbeemedia.com/insights/integrated-approach-tv-vod-recommendations>) 6 June 2012 at the Wayback Machine
4. John S. Breese, David Heckerman, and Carl Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering (http://uai.sis.pitt.edu/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=231&proceeding_id=14), 1998 Archived (https://web.archive.org/web/20131019134152/http://uai.sis.pitt.edu/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=231&proceeding_id=14) 19 October 2013 at the Wayback Machine
5. Xiaoyuan Su, Taghi M. Khoshgoftaar, A survey of collaborative filtering techniques (<http://www.hindawi.com/journals/aai/2009/421425/>), Advances in Artificial Intelligence archive, 2009.
6. Recommender Systems - The Textbook | Charu C. Aggarwal | Springer (<https://www.springer.com/us/book/9783319296579>). Springer. 2016. ISBN 9783319296579.
7. Ghazanfar, Mustansar Ali; Prügel-Bennett, Adam; Szedmak, Sandor (2012). "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>).
8. Das, Abhinandan S.; Datar, Mayur; Garg, Ashutosh; Rajaram, Shyam (2007). "Google news personalization". *Proceedings of the 16th international conference on World Wide Web - WWW '07*. p. 271. doi:10.1145/1242572.1242610 (<https://doi.org/10.1145%2F1242572.1242610>). ISBN 9781595936547. S2CID 207163129 (<https://api.semanticscholar.org/CorpusID:207163129>).

9. He, Xiangnan; Liao, Lizi; Zhang, Hanwang; Nie, Liqiang; Hu, Xia; Chua, Tat-Seng (2017). "Neural Collaborative Filtering" (<https://dl.acm.org/citation.cfm?id=3052569>). *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee: 173–182. arXiv:1708.05031 (<https://arxiv.org/abs/1708.05031>). doi:10.1145/3038912.3052569 (<https://doi.org/10.1145%2F3038912.3052569>). ISBN 9781450349130. S2CID 13907106 (<https://api.semanticscholar.org/CorpusID:13907106>). Retrieved 16 October 2019.
10. Liang, Dawen; Krishnan, Rahul G.; Hoffman, Matthew D.; Jebara, Tony (2018). "Variational Autoencoders for Collaborative Filtering" (<https://dl.acm.org/citation.cfm?id=3186150>). *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee: 689–698. arXiv:1802.05814 (<https://arxiv.org/abs/1802.05814>). doi:10.1145/3178876.3186150 (<https://doi.org/10.1145%2F3178876.3186150>). ISBN 9781450356398.
11. Ferrari Dacrema, Maurizio; Cremonesi, Paolo; Jannach, Dietmar (2019). "Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches" (<https://dl.acm.org/authorize?N684126>). *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM: 101–109. arXiv:1907.06902 (<https://arxiv.org/abs/1907.06902>). doi:10.1145/3298689.3347058 (<https://doi.org/10.1145%2F3298689.3347058>). hdl:11311/1108996 (<https://hdl.handle.net/11311%2F1108996>). ISBN 9781450362436. S2CID 196831663 (<https://api.semanticscholar.org/CorpusID:196831663>). Retrieved 16 October 2019.
12. Ludewig, Malte; Mauro, Noemi; Latifi, Sara; Jannach, Dietmar (2019). "Performance Comparison of Neural and Non-neural Approaches to Session-based Recommendation" (<https://dl.acm.org/citation.cfm?id=3347041>). *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM: 462–466. doi:10.1145/3298689.3347041 (<https://doi.org/10.1145%2F3298689.3347041>). ISBN 9781450362436. Retrieved 16 October 2019.
13. Adomavicius, Gediminas; Tuzhilin, Alexander (1 January 2015). Ricci, Francesco; Rokach, Lior; Shapira, Bracha (eds.). *Recommender Systems Handbook* (<https://repositorio.unal.edu.co/handle/unal/79598>). Springer US. pp. 191–226. doi:10.1007/978-1-4899-7637-6_6 (https://doi.org/10.1007%2F978-1-4899-7637-6_6). ISBN 9781489976369.
14. Bi, Xuan; Qu, Annie; Shen, Xiaotong (2018). "Multilayer tensor factorization with applications to recommender systems" (<https://projecteuclid.org/euclid.aos/1536631275>). *Annals of Statistics*. **46** (6B): 3303–3333. arXiv:1711.01598 (<https://arxiv.org/abs/1711.01598>). doi:10.1214/17-AOS1659 (<https://doi.org/10.1214%2F17-AOS1659>). S2CID 13677707 (<https://api.semanticscholar.org/CorpusID:13677707>).
15. Zhang, Yanqing; Bi, Xuan; Tang, Niansheng; Qu, Annie (2020). "Dynamic tensor recommender systems". arXiv:2003.05568v1 (<https://arxiv.org/abs/2003.05568v1>) [stat.ME (<https://arxiv.org/archive/stat/ME>)].
16. Bi, Xuan; Tang, Xiwei; Yuan, Yubai; Zhang, Yanqing; Qu, Annie (2021). "Tensors in Statistics" (<https://www.annualreviews.org/doi/abs/10.1146/annurev-statistics-042720-020816#:~:text=Abstract,are%20useful%20data%20representation%20architectures.>). *Annual Review of Statistics and Its Application*. **8** (1): annurev. Bibcode:2021AnRSA...842720B (<https://ui.adsabs.harvard.edu/abs/2021AnRSA...842720B>). doi:10.1146/annurev-statistics-042720-020816 (<https://doi.org/10.1146%2Fannurev-statistics-042720-020816>). S2CID 224956567 (<https://api.semanticscholar.org/CorpusID:224956567>).
17. Collaborative Filtering: Lifeblood of The Social Web (http://www.readwriteweb.com/archives/collaborative_filtering_social_web.php) Archived (https://web.archive.org/web/20120422120844/http://www.readwriteweb.com/archives/collaborative_filtering_social_web.php) 22 April 2012 at the [Wayback Machine](http://www.waybackmachine.org/)

18. Gleick, James (2012). *The information : a history, a theory, a flood* (<https://www.worldcat.org/oclc/745979816>) (1st Vintage books ed., 2012 ed.). New York: Vintage Books. p. 410. ISBN 978-1-4000-9623-7. OCLC 745979816 (<https://www.worldcat.org/oclc/745979816>).
19. 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>), Proceedings of the 22nd international conference on World Wide Web
20. Fleder, Daniel; Hosanagar, Kartik (May 2009). "Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity". *Management Science*. **55** (5): 697–712. doi:10.1287/mnsc.1080.0974 (<https://doi.org/10.1287%2Fmnsc.1080.0974>). SSRN 955984 (<https://ssrn.com/abstract=955984>).
21. Castells, Pablo; Hurley, Neil J.; Vargas, Saúl (2015). "Novelty and Diversity in Recommender Systems" (https://link.springer.com/chapter/10.1007/978-1-4899-7637-6_26). In Ricci, Francesco; Rokach, Lior; Shapira, Bracha (eds.). *Recommender Systems Handbook* (2 ed.). Springer US. pp. 881–918. doi:10.1007/978-1-4899-7637-6_26 (https://doi.org/10.1007%2F978-1-4899-7637-6_26). ISBN 978-1-4899-7637-6.
22. Adamopoulos, Panagiotis; Tuzhilin, Alexander (January 2015). "On Unexpectedness in Recommender Systems: Or How to Better Expect the Unexpected". *ACM Transactions on Intelligent Systems and Technology*. **5** (4): 1–32. doi:10.1145/2559952 ([https://doi.org/10.1145/2559952](https://doi.org/10.1145%2F2559952)). S2CID 15282396 (<https://api.semanticscholar.org/CorpusID:15282396>).
23. Adamopoulos, Panagiotis (October 2013). *Beyond rating prediction accuracy: on new perspectives in recommender systems*. *Proceedings of the 7th ACM Conference on Recommender Systems*. pp. 459–462. doi:10.1145/2507157.2508073 ([https://doi.org/10.1145/2507157.2508073](https://doi.org/10.1145%2F2507157.2508073)). ISBN 9781450324090. S2CID 1526264 (<https://api.semanticscholar.org/CorpusID:1526264>).
24. Chatzis, Sotirios (October 2013). "Nonparametric Bayesian multitask collaborative filtering". *CIKM '13: Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. Portal.acm.org. pp. 2149–2158. doi:10.1145/2505515.2505517 ([https://doi.org/10.1145/2505515.2505517](https://doi.org/10.1145%2F2505515.2505517)). ISBN 9781450322638. S2CID 10515301 (<https://api.semanticscholar.org/CorpusID:10515301>).
25. Mehta, Bhaskar; Hofmann, Thomas; Nejdl, Wolfgang (19 October 2007). *Proceedings of the 2007 ACM conference on Recommender systems - Rec Sys '07*. Portal.acm.org. p. 49. CiteSeerX 10.1.1.695.1712 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.695.1712>). doi:10.1145/1297231.1297240 (<https://doi.org/10.1145%2F1297231.1297240>). ISBN 9781595937308. S2CID 5640125 (<https://api.semanticscholar.org/CorpusID:5640125>).
26. Shi, Yue; Larson, Martha; Hanjalic, Alan (2014). "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges". *ACM Computing Surveys*. **47**: 1–45. doi:10.1145/2556270 ([https://doi.org/10.1145/2556270](https://doi.org/10.1145%2F2556270)). S2CID 5493334 (<https://api.semanticscholar.org/CorpusID:5493334>).
27. Massa, Paolo; Avesani, Paolo (2009). *Computing with social trust*. London: Springer. pp. 259–285.
28. Groh Georg; Ehmig Christian. *Recommendations in taste related domains: collaborative filtering vs. social filtering*. Proceedings of the 2007 international ACM conference on Supporting group work. pp. 127–136. CiteSeerX 10.1.1.165.3679 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.165.3679>).
29. Symeonidis, Panagiotis; Nanopoulos, Alexandros; Manolopoulos, Yannis (2008). *Tag recommendations based on tensor dimensionality reduction*. *Proceedings of the 2008 ACM Conference on Recommender Systems*. pp. 43–50. CiteSeerX 10.1.1.217.1437 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.217.1437>). doi:10.1145/1454008.1454017 ([https://doi.org/10.1145/1454008.1454017](https://doi.org/10.1145%2F1454008.1454017)). ISBN 9781605580937. S2CID 17911131 (<https://api.semanticscholar.org/CorpusID:17911131>).

External links

- *Beyond Recommender Systems: Helping People Help Each Other* (<http://www.grouplens.org/papers/pdf/rec-sys-overview.pdf>), page 12, 2001
- Recommender Systems. (<http://www.prem-melville.com/publications/recommender-systems-eml2010.pdf>) Prem Melville and Vikas Sindhwani. In *Encyclopedia of Machine Learning*, Claude Sammut and Geoffrey Webb (Eds), Springer, 2010.
- Recommender Systems in industrial contexts - PHD thesis (2012) including a comprehensive overview of many collaborative recommender systems (<https://arxiv.org/abs/1203.4487>)
- Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions (http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1423975). Adomavicius, G. and Tuzhilin, A. *IEEE Transactions on Knowledge and Data Engineering* 06.2005
- Evaluating collaborative filtering recommender systems (https://web.archive.org/web/20060527214435/http://ectrl.itc.it/home/laboratory/meeting/download/p5-l_herlocker.pdf) (DOI (<http://www.doi.org/>): [10.1145/963770.963772](https://doi.org/10.1145/963770.963772) (<https://dx.doi.org/10.1145/963770.963772>))
- GroupLens research papers (<http://www.grouplens.org/publications.html>).
- Content-Boosted Collaborative Filtering for Improved Recommendations. (<http://www.cs.utexas.edu/users/ml/papers/cbcf-aaai-02.pdf>) Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, pp. 187–192, Edmonton, Canada, July 2002.
- A collection of past and present "information filtering" projects (including collaborative filtering) at MIT Media Lab (<http://agents.media.mit.edu/projects.html>)
- Eigentaste: A Constant Time Collaborative Filtering Algorithm. Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. *Information Retrieval*, 4(2), 133-151. July 2001. (<http://www.ieor.berkeley.edu/~goldberg/pubs/eigentaste.pdf>)
- A Survey of Collaborative Filtering Techniques (<http://downloads.hindawi.com/journals/aaai/2009/421425.pdf>) Su, Xiaoyuan and Khoshgortaar, Taghi. M
- Google News Personalization: Scalable Online Collaborative Filtering (<http://dl.acm.org/citation.cfm?id=1242610>) Abhinandan Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. *International World Wide Web Conference, Proceedings of the 16th international conference on World Wide Web*
- Factor in the Neighbors: Scalable and Accurate Collaborative Filtering (<https://web.archive.org/web/20101023032716/http://research.yahoo.com/pub/2435>) Yehuda Koren, *Transactions on Knowledge Discovery from Data (TKDD)* (2009)
- Rating Prediction Using Collaborative Filtering (<https://web.archive.org/web/20110224164938/http://webpages.uncc.edu/~asaric/ISMIS09.pdf>)
- Recommender Systems (<http://www.cis.upenn.edu/~ungar/CF/>)
- Berkeley Collaborative Filtering (<http://www2.sims.berkeley.edu/resources/collab/>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Collaborative_filtering&oldid=1092795851"

This page was last edited on 12 June 2022, at 17:07 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License 3.0; 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.