Recommender system

From Wikipedia, the free encyclopedia

Recommender systems or **recommendation systems** (sometimes replacing "system" with a synonym such as platform or engine) are a subclass of information filtering system that seek to predict the "rating" or "preference" that a user would give to an item.[1][2]

Recommender systems have become extremely common in recent years, and are utilized in a variety of areas: some popular applications include movies, music, news, books, research articles, search queries, social tags, and products in general.[3] However, there are also recommender systems for experts,[4][5] collaborators,[6] jokes, restaurants, financial services,[7] life insurance, romantic partners (online dating), and Twitter pages.[8]

Contents

Overview

Recommender systems typically produce a list of recommendations in one of two ways – through collaborative and content-based filtering or the personality-based approach.[3][9] Collaborative filtering approaches building 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.[10] Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties.[11] These approaches are often combined (see Hybrid Recommender Systems). The personality-based approach derives product and service preferences from a user's personality.[3]

The differences between collaborative and content-based filtering can be demonstrated by comparing two popular 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) in order 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 own strengths and weaknesses. In the above example, Last.fm requires a large amount of information on a user in order to make accurate recommendations. This is an example of the cold start problem, and is common in collaborative filtering systems.[12][13][14] While Pandora needs very little information to get started, 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 by themselves. Interestingly enough, recommender systems are often implemented using search engines indexing non-traditional data.

Montaner provided the first overview of recommender systems from an intelligent agent perspective.[15] Adomavicius provided a new, alternate overview of recommender systems.[16] Herlocker provides an additional overview of evaluation techniques for recommender systems,[17] and Beel et al. discussed the problems of offline evaluations.[18] Beel et al. have also provided literature surveys on available research paper recommender systems and existing challenges.[19][20]

Recommender systems are an active research topic in the data mining and machine learning fields.[3] Conferences that address recommender system research include RecSys, SIGIR, and KDD.

Approaches

Collaborative filtering

One approach to the design of recommender systems that has wide use is collaborative filtering.[21] Collaborative filtering methods are based on collecting and analyzing a large amount of information on users’ behaviors, activities or preferences and predicting what users will like based on their similarity to other users. 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[22] and the Pearson Correlation as first implemented by Allen.[23]

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.

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.

Examples of implicit data collection include the following:

- Observing the items that a user views in an online store.
- Analyzing item/user viewing times[24]
- 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

The recommender system compares the collected data to similar and dissimilar data collected from others and calculates a list of recommended items for the user. Several commercial and non-commercial examples are listed in the article on collaborative filtering systems.

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.[25] Other examples include:

- As previously detailed, Last.fm recommends music based on a comparison of the listening habits of similar users.
- Facebook, MySpace, LinkedIn, and other social networks use collaborative filtering to recommend new friends, groups, and other social connections (by examining the network of connections between a user and their friends).[1] Twitter uses many signals and in-memory computations for recommending who to follow to its users.[8]

Collaborative filtering approaches often suffer from three problems: cold start, scalability, and sparsity.[26]

- Cold start: These systems often require a large amount of existing data on a user in order to make accurate recommendations. [12][13]
- 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.

A particular type of collaborative filtering algorithm uses matrix factorization, a low-rank matrix approximation technique.[27][28][29]

Collaborative filtering methods are classified as memory-based and model based collaborative filtering. A well-known example of memory-based approaches is user-based algorithm[30] and that of model-based approaches is Kernel-Mapping Recommender.[31]

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 preference.[32] In a content-based recommender 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). 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 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).

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. 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.

Direct feedback from a user, usually in the form of a like or dislike button, can be used to assign higher or lower weights on the importance of certain attributes (using Rocchio classification or other similar techniques).

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.

As previously detailed, Pandora Radio is a popular example of a content-based recommender system that plays music with similar characteristics to that of a song provided by the user as an initial seed. There are also a large number of content-based recommender systems aimed at providing movie recommendations, a few such examples include Rotten Tomatoes, Internet Movie Database, Jinni, Rovi Corporation, Jaman and See This Next (http://www.seethisnext.com/). Document related recommender systems aim at providing document recommendations to knowledge workers, for example Noggle (https://www.noggle.online/knowledge-base/document-recommendation/) and Google Springboard.

Hybrid recommender systems

Recent research has demonstrated that a hybrid approach, combining collaborative filtering and content-based filtering could be more effective in some cases. 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[16] for a complete review of recommender systems). Several studies empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrate 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.

Netflix is a good example of the use of hybrid recommender systems. They make 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).

A variety of techniques have been proposed as the basis for recommender systems: collaborative, content-based, knowledge-based, and demographic techniques. Each of these techniques has known shortcomings, such as the well known cold-start problem for collaborative and content-based systems (what to do with new users with few ratings) and the knowledge engineering bottleneck[33] in knowledge-based approaches. A hybrid recommender system is one that combines multiple techniques together to achieve some synergy between them.

- Collaborative: The system generates recommendations using only information about rating profiles for different users. Collaborative systems locate peer users with a rating history similar to the current user and generate recommendations using this neighborhood.
- Content-based: The system generates recommendations from two sources: the features associated with products and the ratings that a user has given them. 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.
- Demographic: A demographic recommender provides recommendations based on a demographic profile of the user. Recommended products can be produced for different demographic niches, by combining the ratings of users in those niches.
- Knowledge-based: A knowledge-based recommender suggests products based on inferences about a user’s needs and preferences. This knowledge will sometimes contain explicit functional knowledge about how certain product features meet user needs.[34][35]

The term hybrid recommender system is used here to describe any recommender system that combines multiple recommendation techniques together to produce its output. There is no reason why several different techniques of the same type could not be hybridized, for example, two different content-based recommenders could work together, and a number of projects have investigated this type of hybrid: NewsDude, which uses both naive Bayes and kNN classifiers in its news recommendations is just one example.[34]

Seven hybridization techniques:

- Weighted: The score of different recommendation components are combined numerically.
- Switching: The system chooses among recommendation components and applies the selected one.
- Mixed: Recommendations from different recommenders are presented together.
- Feature Combination: Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.
- Feature Augmentation: One recommendation technique is used to compute 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.[34]

Personality-based recommender systems

A new approach stems from Buettner who proposed a personality-based product recommender (PBPR) framework analyzing social media data in order to predict a user's personality and to subsequently derive its personality-based product preferences.[3]

Beyond accuracy

Typically, research on recommender systems is concerned about finding the most accurate recommendation algorithms. However, there is 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.[36]
- **Recommender persistence** – In some situations it is more effective to re-show recommendations,[37] or let users re-rate items,[38] 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[39] 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. 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.[40] 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.[41] This led in part to the cancellation of a second Netflix Prize competition in 2010.[42] Much research has been conducted on ongoing privacy issues in this space. 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.[43]
- **User demographics** – Beel et al. found that user demographics may influence how satisfied users are with recommendations.[44] 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.[45]
- **Serendipity** – Serendipity is a measure of "how surprising the recommendations are".[46] 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.[47] 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.[48] 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%). Interestingly, recommendations with no label performed best (CTR=9.87%) in that study.

Mobile recommender systems

One growing area of research in the area of recommender systems is mobile recommender systems. With the increasing ubiquity of internet-accessing smart phones, it is now possible 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]). 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 is one that offers potentially profitable driving routes for taxi drivers in a city.[49] This system takes input data in the form of GPS traces of the routes that taxi drivers took while working, which include 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. This type of system is obviously location-dependent, and since it must operate on a handheld or embedded device, the computation and energy requirements must remain low.

Another example of mobile recommendation is what (Bouneffouf et al., 2012) developed for professional users. Using GPS traces of the user and his agenda, it suggests suitable information depending on his situation and interests. The system uses machine learning techniques and reasoning processes in order to dynamically adapt the mobile recommender system to the evolution of the user's interest. The author called his algorithm hybrid-ε-greedy.[50]

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.

Risk-aware recommender systems

The majority of existing approaches to recommender systems focus on recommending the most relevant content to users using contextual information and do not take into account the risk of disturbing the user in specific situation. However, in many applications, such as recommending personalized content, it is also important to consider the risk of upsetting the user so as not to push recommendations in certain circumstances, for instance, during a professional meeting, early morning, late-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.

Risk definition

"The risk in recommender systems is the possibility to disturb or to upset the user which leads to a bad answer of the user".[52]

In response to these challenges, the authors in[52] have developed a dynamic risk sensitive recommendation system called DRARS (Dynamic Risk-Aware Recommender System), which models the context-aware recommendation as a bandit problem. This system combines a content-based technique and a contextual bandit algorithm. They have shown that DRARS improves the Upper Condence Bound (UCB) policy, the currently available best algorithm, by calculating the most optimal exploration value to maintain a trade-off between exploration and exploitation based on the risk level of the current user's situation. The authors conducted experiments in an industrial context with real data and real users and have shown that taking into account the risk level of users' situations significantly increased the performance of the recommender systems.

The Netflix Prize

One of the key 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.[53]

The most accurate algorithm in 2007 used an ensemble method of 107 different algorithmic approaches, blended into a single prediction:[54]

> 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.[53][55] 4-Tell, Inc. created a Netflix project-derived solution for ecommerce websites.

A second contest was planned, but was ultimately canceled in response to an ongoing lawsuit and concerns from the Federal Trade Commission.[42]

Performance measures

Evaluation is important in assessing the effectiveness of recommendation algorithms. 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. Recently, diversity, novelty, and coverage are also considered as important aspects in evaluation.[56] However, many of the classic evaluation measures are highly criticized.[57] Often, results of so-called offline evaluations do not correlate with actually assessed user-satisfaction.[58] The authors conclude "we would suggest treating results of offline evaluations [i.e. classic performance measures] with skepticism".

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.[59] See this chapter[60] for an extended introduction.

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), Recommender Systems Handbook, Springer, 2011, pp. 1-35
2. "Facebook, Pandora Lead Rise of Recommendation Engines - TIME". *TIME.com*. 27 May 2010. Retrieved 1 June 2015.
3. Buettner, Ricardo (2016). "Predicting user behavior in electronic markets based on personality-mining in large online social networks: A personality-based product recommender framework". *Electronic Markets: The International Journal on Networked Business*. Springer: 1–19. doi:10.1007/s12525-016-0228-z.
4. Buettner, Ricardo (2014). *A Framework for Recommender Systems in Online Social Network Recruiting: An Interdisciplinary Call to Arms*. 47th Annual Hawaii International Conference on System Sciences. Big Island, Hawaii: IEEE. pp. 1415–1424. doi:10.13140/RG.2.1.2127.3048.
5. 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), in arXiv preprint 2015
6. 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), in ACM/IEEE Joint Conference on Digital Libraries (JCDL) 2011
7. 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), in AAAI/IAAI 2007, pp. 1692-1699, Vancouver, Canada, 2007.
8. 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
9. 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), International Journal of Information and Education Technology, June 2012
10. Prem Melville and Vikas Sindhwani, Recommender Systems (http://www.prem-melville.com/publications/recommender-systems-eml2010.pdf), Encyclopedia of Machine Learning, 2010.
11. R. J. Mooney & L. Roy (1999). *Content-based book recommendation using learning for text categorization*. In Workshop Recom. Sys.: Algo. and Evaluation.
12. Rubens, Neil; Elahi, Mehdi; Sugiyama, Masashi; Kaplan, Dain (2016). "Active Learning in Recommender Systems". In Ricci, Francesco; Rokach, Lior; Shapira, Bracha. *Recommender Systems Handbook* (2 ed.). Springer US. ISBN 978-1-4899-7637-6.
13. Elahi, Mehdi; Ricci, Francesco; Rubens, Neil. *A survey of active learning in collaborative filtering recommender systems*. Computer Science Review, 2016, Elsevier.
14. Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, David M. Pennock (2002). *Methods and Metrics for Cold-Start Recommendations*. 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.
15. 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.
16. Adomavicius, G.; Tuzhilin, A. (June 2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". *IEEE Transactions on Knowledge and Data Engineering*. **17** (6): 734–749. doi:10.1109/TKDE.2005.99.
17. 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. doi:10.1145/963770.963772.
18. Beel, J.; Langer, S.; Genzmehr, M.; Gipp, B. (October 2013). "A Comparative Analysis of Offline and Online Evaluations and Discussion of Research Paper Recommender System Evaluation" (PDF). *Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM Recommender System Conference (RecSys)*.
19. Beel, J.; Langer, S.; Genzmehr, M.; Gipp, B.; Breitinger, C. (October 2013). "Research Paper Recommender System Evaluation: A Quantitative Literature Survey" (PDF). *Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM Recommender System Conference (RecSys)*.
20. Beel, J.; Gipp, B.; Langer, S.; Breitinger, C. (26 July 2015). "Research Paper Recommender Systems: A Literature Survey". *International Journal on Digital Libraries*: 1–34. doi:10.1007/s00799-015-0156-0.
21. 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).
22. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. (2000). "Application of Dimensionality Reduction in Recommender System A Case Study",
23. Allen, R.B. (1990). "User Models: Theory, Method, Practice". International J. Man-Machine Studies.

24. 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.
25. 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)
26. Sanghack Lee and Jihoon Yang and Sung-Yong Park, Discovery of Hidden Similarity on Collaborative Filtering to Overcome Sparsity Problem (http://books.google.com/books?id=u4qzlZAEjegC&pg=PA396&lpg=PA396&dq=sparsity+problem+content-based&source=bl&ots=ZEI50fmXTh&sig=jcwEba0LykRiLnYVbLYUwdrHYEo&hl=en&ei=-W7CTvboCIeHsgKf9NHDBA&sa=X&oi=book_result&ct=result&resnum=1&ved=0CC0Q6AEwAA#v=onepage&q=sparsity%20problem%20content-based&f=false), Discovery Science, 2007.
27. I. Markovsky, Low-Rank Approximation: Algorithms, Implementation, Applications, Springer, 2012, ISBN 978-1-4471-2226-5
28. Takács, G.; Pilászy, I.; Németh, B.; Tikk, D. (March 2009). "Scalable Collaborative Filtering Approaches for Large Recommender Systems" (PDF). *Journal of Machine Learning Research*. **10**: 623–656
29. Rennie, J.; Srebro, N. (2005). Luc De Raedt, Stefan Wrobel, ed. *Fast Maximum Margin Matrix Factorization for Collaborative Prediction* (PDF). Proceedings of the 22nd Annual International Conference on Machine Learning. ACM Press.
30. Breese, John S.; Heckerman, David; Kadie, Carl (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering (PDF) (Report). Microsoft Research.
31. "Kernel-Mapping Recommender system algorithms". *Information Sciences*. **208**: 81–104. doi:10.1016/j.ins.2012.04.012. Retrieved 1 June 2015.
32. Peter, Brusilovsky (2007). *The Adaptive Web*. p. 325. ISBN 978-3-540-72078-2.
33. 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
34. Robin Burke , Hybrid Web Recommender Systems (http://www.dcs.warwick.ac.uk/~acristea/courses/CS411/2010/Book%20-%20The%20Adaptive%20Web/HybridWebRecommenderSystems.pdf), 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.
35. Alexander Felfernig and Robin Burke. Constraint-based Recommender Systems: Technologies and Research Issues (http://www.ist.tugraz.at/felfernig/images/constraint_based_recommendation.pdf), Proceedings of the ACM International Conference on Electronic Commerce (ICEC'08), Innsbruck, Austria, Aug. 19-22, pp. 17-26, 2008.
36. Ziegler, C.N., McNee, 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.
37. Joeran Beel; Stefan Langer; Marcel Genzmehr; Andreas Nürnberger (September 2013). "Persistence in Recommender Systems: Giving the Same Recommendations to the Same Users Multiple Times". In Trond Aalberg; Milena Dobreva; Christos Papatheodorou; Giannis Tsakonas; Charles Farrugia. *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013)* (PDF). Lecture Notes of Computer Science (LNCS). **8092**. Springer. pp. 390–394. Retrieved 1 November 2013.
38. Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., Riedl, {J}. (2003). "Is seeing believing?: how recommender system interfaces affect users' opinions". *Proceedings of the SIGCHI conference on Human factors in computing systems*. pp. 585–592.
39. {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". *User Modeling and User-Adapted Interaction*. Springer: 1–39.
40. Rise of the Netflix Hackers (http://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.
41. "Netflix Spilled Your Brokeback Mountain Secret, Lawsuit Claims". *WIRED*. 17 December 2009. Retrieved 1 June 2015.
42. "Netflix Prize Update". Netflix Prize Forum. 2010-03-12.
43. Naren Ramakrishnan; Benjamin J. Keller; Batul J. Mirza; Ananth Y. Grama; George Karypis (2001). "Privacy Risks in Recommender Systems". *IEEE Internet Computing*. Piscataway, NJ: IEEE Educational Activities Department. **5** (6): 54–62. doi:10.1109/4236.968832. ISBN 1-58113-561-0.

44. 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". In Trond Aalberg; Milena Dobreva; Christos Papatheodorou; Giannis Tsakonas; Charles Farrugia. *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013)* (PDF). Springer. pp. 400–404. Retrieved 1 November 2013.
45. {K}onstan, {J}.{A}., {R}iedl, {J}. (2012). "Recommender systems: from algorithms to user experience". *User Modeling and User-Adapted Interaction*. Springer: 1–23.
46. {R}icci, {F}., {R}okach, {L}., {S}hapira, {B}., {K}antor {B}. {P}. (2011). "Recommender systems handbook". *Recommender Systems Handbook*. Springer: 1–35.
47. Montaner, Miquel, L{\'o}pez, Beatriz, de la Rosa, Josep Llu{\'\i}s (2002). "Developing trust in recommender agents". *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. pp. 304–305.
48. Beel, Joeran, Langer, Stefan, Genzmehr, Marcel (September 2013). "Sponsored vs. Organic (Research Paper) Recommendations and the Impact of Labeling". In Trond Aalberg; Milena Dobreva; Christos Papatheodorou; Giannis Tsakonas; Charles Farrugia. *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013)* (PDF). pp. 395–399. Retrieved 2 December 2013.
49. Yong Ge; Hui Xiong; Alexander Tuzhilin; Keli Xiao; Marco Gruteser; Michael J. Pazzani (2010). *An Energy-Efficient Mobile Recommender System* (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. 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* (PDF), Lecture Notes in Computer Science, IEEE Computer Society, pp. 657–662, ISBN 978-0-7695-4652-0
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*. Elsevier. **20**: 657–662. doi:10.1016/j.websem.2013.03.001.
52. Bouneffouf, Djallel (2013), *DRARS, A Dynamic Risk-Aware Recommender System* (Ph.D.), Institut National des Télécommunications
53. Lohr, Steve. "A $1 Million Research Bargain for Netflix, and Maybe a Model for Others". *The New York Times*.
54. R. Bell; Y. Koren; C. Volinsky (2007). "The BellKor solution to the Netflix Prize" (PDF).
55. Bodoky, Thomas. "Mátrixfaktorizáció one million dollars". *Index*.
56. Lathia, N., Hailes, S., Capra, L., Amatriain, X.: Temporal diversity in recommender systems. In: Proceeding of the 33rd International ACMSIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 210–217. ACM, New York
57. 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.
58. 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: 7–14. doi:10.1145/2532508.2532511. ISBN 9781450324656.
59. Lakiotaki, K.; Matsatsinis; Tsoukias, A. "Multicriteria User Modeling in Recommender Systems". *IEEE Intelligent Systems*. **26** (2): 64–76. doi:10.1109/mis.2011.33.
60. Gediminas Adomavicius, Nikos Manouselis, YoungOk Kwon. (PDF) Recommender Systems http://ids.csom.umn.edu/faculty/gedas/NSFCareer/MCRS-chapter-2010.pdfttiel=Multi-Criteria Recommender Systems Check `|url=` value (help). Missing or empty `|title=` (help)

Further reading

Books

Kim Falk (2015). *Practical Recommender Systems*. ISBN 9781617292705.

- Bharat Bhasker; K. Srikumar (2010). *Recommender Systems in E-Commerce*. CUP. ISBN 978-0-07-068067-8.

- Francesco Ricci; Lior Rokach; Bracha Shapira; Paul B. Kantor, eds. (2011). *Recommender Systems Handbook*. ISBN 978-0-387-85819-7.
- Bracha Shapira; Lior Rokach (June 2012). *Building Effective Recommender Systems*. ISBN 978-1-4419-0047-0.
- Dietmar Jannach; Markus Zanker; Alexander Felfernig; Gerhard Friedrich (2010). *Recommender Systems:An Introduction*. CUP. ISBN 978-0-521-49336-9.

Scientific articles

- Hennig-Thurau, Thorsten; Marchand, André; Marx, Paul (2012). "Can Automated Group Recommender Systems Help Consumers Make Better Choices?" (PDF). *Journal of Marketing*. **76** (5): 89–109. doi:10.1509/jm.10.0537.
- Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. (2002) Content-Boosted Collaborative Filtering for Improved Recommendations. (http://www.cs.utexas.edu/users/ml/papers/cbcf-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. (http://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* (PDF), Lecture Notes in Computer Science, IEEE Computer Society, pp. 657–662, ISBN 978-0-7695-4652-0.
- Bouneffouf, Djallel (2013), *DRARS, A Dynamic Risk-Aware Recommender System* (Ph.D.), Institut National des Télécommunications.

External links

- Recommender Systems Wiki (http://www.recsyswiki.com/)
- Robert M. Bell; Jim Bennett; Yehuda Koren & Chris Volinsky (May 2009). "The Million Dollar Programming Prize". *IEEE Spectrum*.
- 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.
- Recommender Systems Website (http://recommender-systems.org/)
- seethisnext – Film Similarity Search (http://www.seethisnext.com/)
- ACM Conference on Recommender Systems (http://recsys.acm.org/)
- Coursebooks on Recommender Systems (http://www.recommenderbook.net/)
- Rival toolkit for recommender system evaluation (http://rival.recommenders.net/)
- Recsys group at Politecnico di Milano (http://recsys.deib.polimi.it/)
- Recommendation as a service - API (http://www.programmableweb.com/api/recombee)
- Document recommendation dekstop tool (https://www.noggle.online/knowledge-base/document-recommendation/)
- Data Science: Data to Insights from MIT (recommendation systems) (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=738287685"

Categories: Recommender systems | Applied statistics | Behavioral and social facets of systemic risk

- This page was last modified on 8 September 2016, at 01:28.
-