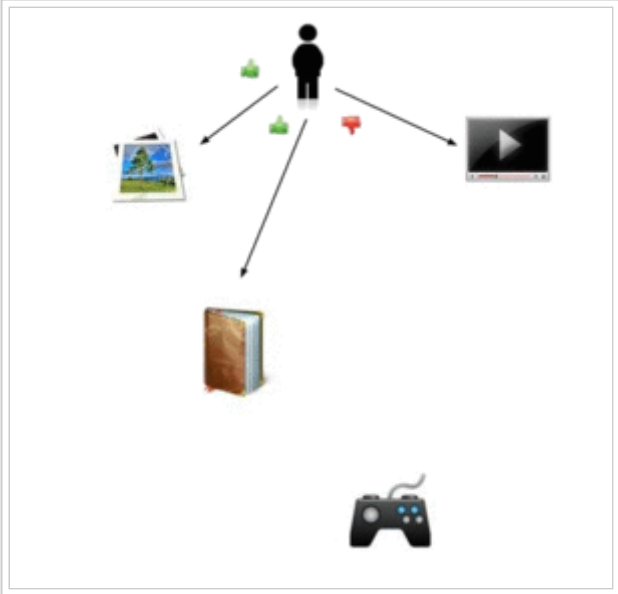
Collaborative filtering

From Wikipedia, the free encyclopedia

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 *x* than to have the opinion on x of a person chosen randomly. For example, a collaborative filtering recommendation system for television tastes 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.



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.

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.

Contents

Introduction

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:

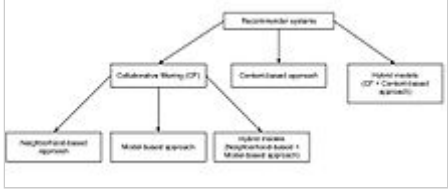
1. A user expresses his or her preferences by rating items (e.g. books, movies or CDs) of the system. These ratings can be viewed as an approximate representation of the user's interest in the corresponding domain.
2. The system matches this user's ratings against other users' and finds the people with 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

This approach uses user rating data to compute the similarity between users or items. This is used for making recommendations. This was an early approach used in many commercial systems. It's effective and easy to implement. 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:

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

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 includes:

$$r_{u,i} = \frac{1}{N} \sum_{u' \in U} r_{u',i}$$
$$r_{u,i} = k \sum_{u' \in U} \mathbf{simil}(u,u') r_{u',i}$$
$$r_{u,i} = \bar{r_u} + k \sum_{u' \in U} \mathbf{simil}(u,u') (r_{u',i} - \bar{r_{u'}})$$

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

$$\operatorname{simil}(x,y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2 \sum_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

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]

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

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

Models are developed using data mining, machine learning algorithms to find patterns based on training data. These are used to make predictions for real data. There are many model-based CF algorithms. These include Bayesian networks, neural embedding models,[5] 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.[6]

This approach has a more holistic goal to uncover latent factors that explain observed ratings.[7] Most of the models are based on creating a classification or clustering technique to identify the user based on the training set. The number of the parameters can be reduced based on types of principal component analysis.

There are several advantages with this paradigm. It handles the sparsity better than memory based ones. This helps with scalability with large data sets. It improves the prediction performance. It gives an intuitive rationale for the recommendations.

The disadvantages with this approach are in the expensive model building. One needs to have a tradeoff between prediction performance and scalability. One can lose useful information due to reduction models. A number of models have difficulty explaining the predictions.

Hybrid

A number of applications combines the memory-based and the model-based CF algorithms. These overcome the limitations of native CF approaches. It improves the prediction performance. Importantly, it overcomes the CF problems such as sparsity and loss of information. However, they have increased complexity and are expensive to implement.[8] Usually most of the commercial recommender systems are hybrid, for example, Google news recommender system.[9]

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 example of collaborative filtering based media.[10]

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.

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 the challenges in the performances 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 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 system, they need to be rated by substantial number of users before they could be recommended to users who have similar tastes with the ones rated them. The new item problem does not limit the content-based recommendation, 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 purchases and ratings history, which demands a higher scalability of a CF system. Large web companies such as Twitter use clusters of machines to scale recommendations for their millions of users, with most computations happening in very large memory machines.[11]

Recently, a method named Item2Vec[5] was introduced for a scalable item-based Collaborative Filtering. Item2Vec produces embedding for items in a latent space and is capable of inferring item-to-item relations even when user information is not available.

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 movie" and "children 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 the opposite 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 black sheep is an acceptable failure.

Shilling attacks

In a recommendation system where everyone can give the ratings, people may give lots of 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 kind of 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."[12] Several collaborative filtering algorithms have been developed to promote diversity and the "long tail" by recommending novel, unexpected,[13] and serendipitous items.[14]

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 privacy preserving manner.
- Robust collaborative filtering, where recommendation is stable towards efforts of manipulation. This research area is still active and not completely solved.[15]

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
- Preference elicitation
- 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" (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. Barkan, O; Koenigstein, N (14 March 2016). "Item2Vec: Neural Item Embedding for Collaborative Filtering" (http://arxiv.org/abs/1603.04259). arXiv:1603.04259.
6. 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.
7. Factor in the Neighbors: Scalable and Accurate Collaborative Filtering (http://research.yahoo.com/pub/2435) Archived (https://web.archive.org/web/20101023032716/http://research.yahoo.com/pub/2435) 23 October 2010 at the Wayback Machine.
8. "Kernel-Mapping Recommender system algorithms". *Information Sciences*. **208**: 81–104. doi:10.1016/j.ins.2012.04.012.
9. "Google news personalization".
10. Collaborative Filtering: Lifeblood of The Social Web (http://www.readwriteweb.com/archives/collaborative_filtering_social_web.php)
11. 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
12. Fleder, Daniel; Hosanagar, Kartik (May 2009). "Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity". *Management Science*. doi:10.1287/mnsc.1080.0974.
13. 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*. doi:10.1145/2559952.
14. Adamopoulos, Panagiotis (October 2013). "Beyond rating prediction accuracy: on new perspectives in recommender systems". *Proceedings of the 7th ACM conference on Recommender systems*. doi:10.1145/2507157.2508073.

15. "Robust collaborative filtering". Portal.acm.org. 19 October 2007. doi:10.1145/1297231.1297240. Retrieved 2012-05-15.

External links

- Item2Vec: Neural Item Embedding for Collaborative Filtering (http://arxiv.org/abs/1603.04259), Barkan, O; Koenigstein, N (14 March 2016) arXiv:1603.04259.
- *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 (http://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 (http://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/aai/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 (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=739063143"

Categories: Collaboration | Collaborative software | Collective intelligence | Information retrieval techniques | Recommender systems | Social information processing | Behavioral and social facets of systemic risk

---

- This page was last modified on 12 September 2016, at 15:46.
- 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.