Literature Survey: Recommendation Systems

Karan Vijay Singh
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1
kv3singh@uwaterloo.ca
Student ID : 20745105
CS 698: Intro to ML Fall 2017 Project Report

Abstract

Given the large number of products and items available on the ecommerce websites, or videos in you tube or movies in netflix, users are interested in personalized recommendations for these items.So, recommender systems have become a viable component especially for companies dealing with end users for better user experience.Through this literature survey, I will measure the performance of basic machine learning algorithms used in building of recommendation systems along with their pros and cons and also how they are impacting the internet world, making it more interactive and suggestive for people to easily make choices.

1 Introduction

In recent years, there has been an exponential increase in the number of smart phone users which has resulted in humongous wireless data traffic generation daily and it is estimated that more than 1 million new mobile broadband subscribers will be added per day for the next 6 years[5].The massive amount of data requires systems that can efficiently filter this huge data/information . Also modern day users are flooded with a lot more choices be it in online shopping, watching videos or news articles. Due to limitation of screen of these devices, people are interested in seeing what they want to see, be it personalised news feed, watching you tube videos or buying an item from an ecommerce website. So, in order for companies especially dealing with end users to woo customers/end users, they need to be specific in showing personalised products/items according to ones interest and likings. So, recommender systems have become a viable component for these companies as giving personalized recommendations adds another dimension to users experience and helps in gaining customers confidence.

Companies like amazon for product recommendation, youtube for video recommendation, Netflix for movie recommendations, linkedin for job recommendations, Spotify for music recommendations etc have made these recommendation systems an essential part of their website.For instance, Twitter's user recommendation service - WTF(Who to Follow) helps both existing and new users to make new connections[7].

This paper is an attempt to survey recommender systems in general, collaborative filtering based algorithms that power recommendation systems, how they were started and where have they reached.This paper is organized as follows: section 2 tells about the history of recommender systems while section 3 explains the definition and main type of these recommender systems. Section 4 explains the basic algorithms based on collaborative filtering that are most widely used in building recommender systems and section 5 analyses some variations of these algorithms and discusses the results. Section 6 briefs about Netflix prize Challenge and twenty years of recommendation systems in amazon.com. This paper concludes in the last section with a conclusion section.

2 Background

This research in this area started in mid 90s when Tapestry (an email filtering system)[6], the first commercial recommender system, was developed at the Xerox Palo Alto Research Center by the group, led by David Goldberg. Increasing use of electronic mail and to prevent users from getting flooded by a large number of streaming documents was the motivation for the development of Tapestry [6].In Tapestry, for every email message the user received, he could write a comment (annotations) for it and these annotations could be shared with other group users, so that other users could benefit. One of the shortcomings of it was that it required human involvement to write annotations and also there was no way to analyze whether users were receiving the content they actually wanted to see.Then the group lense system [Resnick et al. 1994; Konstan et al. 1997] was the first to provide users with personalised automated recommendations on usenet news articles[16]. The recommendations for articles for a user were based on users with similar tastes and interests [16].From then onwards, there is no looking back and we can see how these recommendation systems from the past three decades are making the online world more close to us by giving us personalised recommendations.

3 Definition And Types

Recommendation systems are used to identify the liking that a user will have for a specific item[17].It is a filtering system that filters information and uses that filtered information to predict items that will be of interest to a user or group of users with similar interest. Or we can say it makes easier for users to decide which products to focus on (this focus may mean reading, buying, watching, listening etc).Most recommendation system follow one of these two techniques i.e. collaborative filtering or content based filtering but combination of these two approaches leads to hybrid approaches also occur [2].

3.1 Collaborative filtering

The collaborative filtering technique models recommendations based on users past action and history. This can be based on a single users action or a group of users with similar action or behaviour. When the other users with similar actions are taken into consideration, the recommendations by collaborative filtering are given based on group of alike users. So the basic concept is that people with similar interest in certain things will have similar interests in other things too. For instance if two customers liked items X and Y in the past and one person also likes item Z, it is likely that the other person will also like Z.

3.2 Content Based Filtering

The content based filtering technique models recommendations by creating a profile for each user/ product using the distinct features of the user/product to characterize its nature. For example a product profile could include its price, company, type etc whereas user profile could include age, gender, demographic information etc. The system then learns the user choices over characteristics, and uses these computed choices to recommend new items with similar characteristics.

3.3 Hybrid approach

The fusion of the above two leads to hybrid approaches. Making collaborative filtering and Content based filtering predictions separately and then using the results of the two to make a final recommendation to the consumer is an example of hybrid approach.Hence, sometimes using Hybrid approaches is advantageous as it is able to avoid certain shortcomings of both content based and collaborative filtering[1].

4 Collaborative Filtering Techniques & Algorithms

There are primarily two approaches to Collaborative filtering:

1. the neighbourhood approach
2. the latent factor approach

4.1 Neighbourhood models

The prediction of neighbourhood models is based on the similarity relationships among either users or items. For example, two users are similar if they rate an item or set of items identically[3].The user oriented approach evaluates users choice for a product based on the ratings of other similar users for that product. On the other hand, the item oriented approach evaluates the users choice for a product based on his/her own ratings on similar products. The item oriented approach is usually preferred [18],because of obvious reasons below

1. In real life applications, the number of users are much larger than the number of products/items and each user rates only a fraction of products(usually which he/she really likes or dislikes), so there are often situations where no recommendations can be made, also computing all pair wise correlation between users is expensive due to large number.
2. Also item-item similarity is fairly stable as compared to users as user profile could change rapidly and we need to compute it in real time as users dont want to wait for even a second to get updated recommendations.
3. Sometimes some users dont have anything similar with any other users.So, it becomes difficult to give recommendations.
4. The computational complexity of the user based grows linearly with the increase in the number of customers and in commercial systems, users can grow in millions in no time[4].So, scalability is a big issue in these.

So, the following steps are followed while building an item-item based neighbourhood CF model:

1. In the first step, the similarity between pairs of items is calculated. The similarity between two items i and j is defined as the likelihood of users to rate item i and j similarly. This similarity is calculated on common raters. This is done basically to identify the neighbourhood of items similar to a particular item i. The similarity is also to referred to as weights i.e. $w_{i,j}$ The similarity between two items 'i' and 'j' is usually high if both have been rated by a large number of users.

   Different algorithms use different similarity measures and there are no appropriate ground reasons for choosing a particular similarity function but each one of these measures has its pros and cons. For instance, in computing Cosine based similarity, one cannot take into account negative ratings and the items for which there are no ratings are assumed to have zero ratings[14]. This similarity between two items can be calculated using cosine function , the adjusted cosine , the Pearson correlation coefficient , Spearman rank correlation or mean squared differences etc[8].Sometimes researchers try and compare different similarity measures on their problem and choose the one which better fits the data.A good empirical comparison of some of the similarity measures can be seen in [19].

   As item-item similarities are stable, to speed up final computation, these item-item similarities are precomputed and then updated periodically. There is an another way by which we can compute the similarity between each pair of items i and j and it is based on the conditional probability i.e. Probability of a user buying an item given that he has already bought some other item.

2. In the second step, we predict the user-item rating.In this step, while making final prediction $r_{u,i}$ i.e. rating r for user u and item i, we use only k most similar items to item i that have been rated by user u i.e. using only k items that are in the neighbourhood of item i.So, the final prediction is made by taking the weighted average of all the ratings by the user u on the k items. So, neighbourhood models get strengthen by kNN (k nearest neighbourhood)

approach which assists in disregarding the items that are poorly correlated to item i, i.e. target item, thereby improving the quality of recommendations by reducing the noise. If value of k is chosen to be too small, then chances of inaccurate prediction increases and if the value of k is too large, then it leads to noise. So we need to find optimal value of k according to our data.

4.2 Latent factor Models

These models follow a different approach to make predictions for the ratings i.e. they characterize items and users on some factors derived from the ratings pattern [10].These factors are not obvious in general i.e. they are hidden factors .Our motto is to estimate the latent factors from the observed rating data by using mathematical techniques.

4.2.1 Matrix Factorisation Technique

Matrix factorisation techniques are one of the strong realisations of latent factor models(widely used).Matrix factorization techniques are usually efficient than item-based CF in some cases as they allow us to discover the hidden factors underlying the communication between items and users.In this, latent factors inferred from the ratings pattern are used to characterise both items and users[10] i.e. each user and item is mapped to latent factor space (K). Each user u is associated with a vector $\mathrm{p_u}$ in latent feature space(K) and each item i is associated with a vector $\mathrm{q_i}$ in latent feature space(K).

$$\mathrm{p_u}, \mathrm{q_i} \in \mathbb{R}^{\mathrm{K}}$$

where (K $\ll$ n,m) n being the number of users and m being the number of items.The dot product of the two vectors i.e. user vector with item vector assuming user-items effects have been removed tells us the users u liking for that particular item i.

$$\widehat{r}_{ui} = p_{\mathrm{u}} \cdot q_{\mathrm{i}}$$

So, this way we predict user u's rating for item i. The main difficulty comes in estimating the user and item vectors in latent factor space. Once they are estimated, we can easily predict the rating for any item i by a user.

So more formally, we have a user-item rating matrix R containing ratings of all users for items they have rated. It will be a sparse matrix as each user rates only a small set of items, i.e. the items which he usually likes or dislikes. So , we need to estimate the ratings for the items which user has never rated using the ratings of the items he has rated.So its like doing SVD but its not actually SVD. We need to estimate the user feature vector and item feature vector. We estimate these feature vectors in such a way that it minimizes the squared error for the ratings we know.So the model learns from the observed user-item ratings. Since our aim is to predict unknown ratings, we avoid overfitting the observed ratings by adding a regularisation term. Also user and item effects are removed prior to doing these estimations. It is possible that an items popularity may decrease with time, also users interest for an item may change over a period of time, So, these item and user effects are taken into account or are removed before making the final prediction rating and these temporal effects help in noteworthy performance and accuracy improvements.

We can minimise the regularised squared error using stochastic gradient descent and also alternating least squares method.

$$\min_{p_{\mathrm{u}}, q_{\mathrm{i}}} \sum_{(u,i)\in T} (r_{ui} - \widehat{r}_{ui}) + \lambda[\sum_{u} \|p_{\mathrm{u}}\|^2 + \sum_{i} \|q_{\mathrm{i}}\|^2]$$

where, T is the set of the user-item pairs for which rating $\mathrm{r_{ui}}$ is known. While stochastic gradient is faster yet alternating least squares is preferred in some cases like where system can parallelise[10]. Since both the user and item feature vectors are unknown, hence the overall problem is non-convex. If we fix one and optimise the other and vice-versa, then the each step becomes convex i.e. simply a ridge regression and this can be repeated until convergence.

5 Analysis

While these are the basic approaches/ algorithms that are being used in the building of a recommendation system, certain variations to these basic algorithms or a combination of these approaches are often used depending upon the data and problem in hand. Usually developers while making recommender systems only take into account the numeric ratings which users give to different items and ignore the review text/suggestions that users write along with the numeric rating. In [13], author incorporates these review texts along with numeric rating for making a final prediction for the items user has not reviewed yet, by developing a statistical model that combines Latent factors in rating data along with topics in review text.

This model helps in eliminating the major shortcoming of collaborative filtering approach i.e. cold start problem. Cold Start problem occurs when we dont have enough past information of a user or an item, so it becomes hard for recommender system to predict ratings for them. In those scenarios, where we dont have enough data, this becomes useful as significant information can be provided by even a single review text i.e. we can get a great benefit of review text as it can tell many characteristics of an item or a person[13]. While with the combined model significant improvement can be seen in terms of Mean Squared Error in case of sparse data as compared to Latent Factor model alone, but when the data is abundantly available, both the models perform similarly. So in cases where there is more sparsity in data set (as in case of Amazon movie data set where the combined models performance was greatest in [13]), we can do this extra review text along with latent factor model but where the data sparsity problem is not there, it does not seem efficient to do this extra computations of review text.

From the starting days of Collaborative filtering recommendation systems, variations of Item-based top N recommendation algorithms are in use. In [23], to reduce the sparsity problem in collaborative filtering an item oriented algorithm i.e. Random Walk Recommender model has been used. It can be seen as an advancement to item-based top N recommendation Algorithm in which user makes random walks in the item space according to similarity of items. The model used in the algorithm is a Markov Chain model in which the probability of going to the next state depends only on the probability of current state i.e. a user viewing a particular item i will probably view the next item j in kth step of random walk will depends on the probability of user viewing item i.

Basically the algorithm consists of three steps. The initial step involves the computation of transition probabilities which depends on the similarity measures between the items.While computing these transition probabilities, the model also takes into account a small probability in case a user steps to an arbitrary item. Then, the next step involves the computation of ranks of each item for all users and finally scaling of ranks is done in a way such that the highest ranked item of each user is given the highest rating and each users ranks are scaled independently of each other[23]. On comparing item based top N recommendation model, random walk method and default voting method (an extension of item based top-N recommendation model which results in dense similarity matrix, irrespective of the size of training set as it computes similarity matrix, filling half of the users unrated items with average rating of the user) using different similarity measures, the results on Movie Lens Data set were found to be best for Random walk method in terms of error metrics i.e. using Root mean Square Error (RMSE), Mean Absolute Error (MAE).So, depending upon the type of problem we are trying to solve and the data in hand, we need to make appropriate choice of the algorithm to use for the problem.

6 How Companies are using these systems

6.1 Netflix Prize Challenge

In 2006, Netflix, an online DVD-rental and video streaming service company announced a machine learning competition for prediction of movie ratings. The company announced a 1 million $ prize to be awarded if anyone is able to improve the accuracy of their current system "Cinematch" by 10%. The company provided a training data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies. The company also gave a progress prize of $50,000 each year to the best result in that year until the winning solution came. "However, in order to win this prize, an algorithm had to improve the RMSE on the test set by at least 1% over the previous progress prize winner"[22].The company conducted this competition as providing recommendations to users is a essential part of their business. The RMSE was used to measure the accuracy of the algorithms. The winning algorithm was actually a blend of several different algorithms that were developed independently and published description of their algorithms can be found in the papers[21],[9],[15].The winning algorithm had a RMSE of 0.8567 on the test subset which was 10.6% improvement over Cinematchs score at the start of competition. After 3 years, the price was finally won by BellKor's Pragmatic Chaos team in September 2009.

However the company never implemented the solution as it transitioned from renting DVDs to video streaming[11] but it was an important driver of research in the field of recommendation systems.

6.2 Amazon

Nearly twenty years ago, Amazon.com launched recommender system in its website to provide recommendations to users. Amazon using recommendation systems, customises the online store according to users interest and liking, for example showing medical books to a doctor, baby products to a new mother."Amazon.com uses recommendations as a targeted marketing tool in many email campaigns and on most of its Web sites pages, including the high traffic Amazon.com homepage"[12].Amazon is using item-item collaborating filtering technique since past 20 years for recommendations for obvious reasons. One, most of the computation are done "offline as a batch build of identical items and the final recommendation computation is done in real time as a series of look up"[20]. Also, the algorithm scales very well to millions of users and millions of items without requiring any other technique. Though the algorithm is getting adapted and updated to improve diversity, recency and other problems along with the increasing variety of products and increasing number of customers, but we can realise how powerful the technique is, as it is being used by the worlds largest ecommerce website and we have already seen some of the advantages above.

So, we can imagine the impact of recommender systems on Amazon.com website from the past two decades in providing better end user experience and can realise the importance of these systems in making the Amazon.com as one of the largest online retail seller in the world.

7 Conclusion

We can see how recommender systems are changing end user websites to personalised feeds for each of their customers, be it ecommerce websites like Amazon or video streaming websites like youtube, Netflix etc. The internet world cannot imagine watching a latest youtube video, buying latest fashion shoes or listening to latest songs without the power of these recommendation systems. This is not only helping companies in increasing their customer base and hence increasing their revenues but also making them to dig deeper and further research in these techniques to further enhance the end consumer experience.

We can see the growth of Amazon using item based collaborative filtering recommendation system from the past twenty years and can very well see how powerful these systems are. "Simplicity, explainability, scalability, adaptability and relatively high-quality recommendations"[20], makes item-based collaborative filtering one of the popular recommendation algorithms today.With a devoted ACM conference, recommender system continue to be a dynamic field of research in machine learning, data mining and information retrieval. Thus, we can imagine these recommendation systems providing us with better interactive experience.

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.

[2] Marko Balabanović and Yoav Shoham. Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, March 1997.

[3] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 39–46, New York, NY, USA, 2010. ACM.

[4] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, January 2004.

[5] Ericsson. Ericsson mobility report. https://www.ericsson.com/en/mobility-report, June 2017.

[6] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, December 1992.

[7] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. Wtf: The who to follow service at twitter. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 505–514, New York, NY, USA, 2013. ACM.

[8] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 230–237, New York, NY, USA, 1999. ACM.

[9] Yehuda Koren. 1 the bellkor solution to the netflix grand prize, 2009.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[11] Jake Kovoor. Netflix prize. http://saintlad.com/why-netflix-never-used-algorithm-won-netflix-1-million-challenge/, December 2016.

[12] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[13] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 165–172, New York, NY, USA, 2013. ACM.

[14] Prem Melville and Vikas Sindhwani. Recommender systems. In *Encyclopedia of Machine Learning*, pages 829–838. 2010.

[15] Martin Piotte and Martin Chabbert. The pragmatic theory solution to the netflix grand prize, in: Netflix prize documentation, 2009.

[16] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM.

[17] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, pages 1–35. Springer US, oct 2010.

[18] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. pages 285–295, 2001.

[19] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[20] Brent Smith and Greg Linden. Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 21(3):12–18, May 2017.

[21] Andreas Tscher, Michael Jahrer, and Robert M. Bell. The bigchaos solution to the netflix grand prize, 2009.

[22] Wiki. Netflix prize. https://en.wikipedia.org/wiki/Netflix_Prize.

[23] Hilmi Yildirim and Mukkai S. Krishnamoorthy. A random walk method for alleviating the sparsity problem in collaborative filtering. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 131–138, New York, NY, USA, 2008. ACM.