Evaluating Recommender Strategies

SHIKHAR JULKA
Research Scholar: Computer Science and Engineering
Maharaja Surajmal Institute of Technology (GGSIPU), Delhi, India
julka.shikhar@gmail.com

AAKASH VERMA
Research Scholar: Computer Science and Engineering
Maharaja Surajmal Institute of Technology (GGSIPU), Delhi, India
aakashverma@gmail.com

Abstract—Recommender systems are a subclass of information filtering systems that seek to generate meaningful recommendations to users for products or items that might interest them. In recent times, it has become common to collect large amounts of data that allows for a deeper analysis of how a user interacts with the products being offered. Recommender Systems have evolved to fulfill the dual need of buyers and sellers by automating the generation of recommendations based on data analysis. This paper will focus on the extent to which recommender systems are helpful, will compare the various methods used to implement them, problems in collaborative filtering and content based filtering methods and illustrate on some open problems that are common to any recommender system.

Keywords- Recommender systems, Collaborative filtering, Item based filtering

I. INTRODUCTION

The way people find products, information and even other people has been changed by Recommender systems. It studies patterns of behavior to know what someone will prefer from among a collection of things he has never experienced. The technology behind recommender systems has evolved over the past years into a rich collection of tools that enable the practitioner or researcher to develop recommendations [1]. Although many different approaches to recommender systems have been developed in the past few years, the interest in this area still remains high due to growing demand on practical applications, which are able to provide personalized recommendations. These growing demands pose some key challenges to recommender systems and to deal with these problems many advanced techniques are proposed like item based filtering, collaborative filtering, spanning tree traversal, bipartite projection and many more. Despite of these advances, recommender systems still require improvement and thus becoming a rich research area. In this paper, before discussing the major limitations of recommendation methods, the comprehensive survey of recommendation approaches is provided. The discussion of various approaches and their limitations in a proper manner thereby provides the future research possibilities in recommendation systems.

II. RECOMMENDER SYSTEMS

Recommender systems are a type of information filtering that is trying to present products (movies, books, music) or social elements (users, groups) that are likely of interest to a user [2]. In other words, recommendation systems are trying to filter the information that is going to be shown to the user. Collaborative filtering or user-based is an approach to recommendation systems and it’s widely used by many web sites. Its aim is to find similar users and then recommend to one what the other similar user likes and hence provides very good recommendation results. It is based on his recent behavior like purchasing a book, watching a movie or listening to music. A common recommendation from Amazon is “Customers who bought items in your basket also bought…” To achieve this kind of information filtering, an effective algorithm is required. Today, many web sites are using recommendation systems such as Amazon, IMDB, Google, Last.fm etc. For instance, a user registered to a movies web site declares that he likes horror movies, either by searching horror movies regularly or indicating that he has watched many of them. Then the recommendation system is able to understand this trend and will recommend him other horror movies that he hadn’t watched yet. This can bring many benefits for a company and can boost its sales and that’s because many users may love things which they didn’t know existed. That’s why recommender systems are trying to find similar things that a user likes and brings them closer by recommending them.

III. COLLABORATIVE FILTERING

Collaborative filtering is a method of making automatic predictions (filtering) [2] about the interests of a user by collecting preferences or taste information from many users (collaborating).The underlying principle of the collaborative filtering approach is that if a person Q has the same opinion as a person P on an issue, Q is more likely to have P'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 clothes tastes could make predictions about which clothes show a user should like given a partial list of that user's tastes (likes or dislikes).Collaborative filtering can be measured by using the following:

A. Manhattan Distance Measure (Taxicab)

Taxicab geometry, considered by Hermann Minkowski in Germany, is a form of measure in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates [3]. An advantage of this measure is that it is fast to compute. In a 2D case if a recommendation is given using x and the y coordinate and is represented by (x, y) a point, then the Manhattan distance is calculated by

$$| X1 – X2| + | Y1 – Y2|$$

(The absolute value of the difference between the *x* values plus the absolute value of the difference between the *y* values).

B. Euclidean Distance Measure

Euclidean distance is the ordinary (i.e. straight-line) distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space [4]. The associated norm is called the Euclidean norm. In a 2D case if a recommendation is given using x and the y coordinate and is represented by (x, y) a point, then the Euclidean distance is calculated by

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Both the above measures do not perform well when missing values are present in data.

C. Pearson Correlation Coefficient Measure

The Pearson Correlation Coefficient is a measure of correlation between two variables used in case of grade inflation when ratings given by two users though on same product vary differently [5]. It ranges between -1 and 1 inclusive. 1 indicates perfect agreement. -1 indicates perfect disagreement. In a 2D case if a recommendation is given using x and the y coordinate and is represented by (x, y) a point, then the Pearson Correlation Coefficient is calculated by

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

D. Merits and Demerits

Collaborative filtering or user-based is an approach to recommendation systems and it’s widely used by many web sites. Its aim is to find similar users and then recommend to one what the other similar user likes provides very good recommendation results. On the other hand, the disadvantages of using this technique besides its high complexity and expensiveness is that it is not efficient to map every user to similar users within sites with many users and a new user cannot get good recommendations. Also the systems performed poorly when they had many items but comparatively few ratings. When the user profile changed all the system model had to be recomputed [6].

IV. ITEM-BASED FILTERING

Item-item collaborative filtering, or item-based is a form of collaborative filtering based on the similarity between items calculated using people's ratings of those items [7]. In item-based filtering, ahead of time we find the most similar items, and combine that with a user's rating of items to generate a recommendation. Item-based filtering is also called model-based collaborative filtering because it doesn’t need to store all the ratings. A model is built representing how close every item is to every other item. Item based filtering can be measured by using the following:

A. *Adjusted Cosine Similarity Measure*

This similarity measurement is a modified form of vector-based similarity where the fact that different users have different ratings schemes; in other words, some users might rate items highly in general, and others might give items lowers ratings as a preference [8]. To remove this drawback from vector-based similarity, subtract average ratings for each user from each user's rating for the pair of items in question:

$$s(i,j) = \frac{\sum_{u \in U} (R_{u,i} - \overline{R}_u)(R_{u,j} - \overline{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \overline{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \overline{R}_u)^2}}$$

B. *Slope One*

A major advantage of Slope One is that it is simple and hence easy to implement [9]. This measure works in two parts- compute deviations between every pair of items and then use those deviations to make predictions. The average deviation of an item *i* with respect to item *j* am:

$$dev_{i,j} = \sum_{u \in S_{i,j}(X)} \frac{u_i - u_j}{card(S_{i,j}(X))}$$

Where *card*(*S*) is how many elements are in *S* and *X* is the entire set of all ratings. The predictions formula is given by

$$P^{wS1}(u)_j = \frac{\sum_{i \in S(u)-\{j\}} (dev_{j,i} + u_i) c_{j,i}}{\sum_{i \in S(u)-\{j\}} c_{j,i}}$$

Where,

$$c_{j,i} = card(S_{j,i}(\chi))$$

PwS1 (u) j means our prediction using Weighted Slope One of user *u*'s rating for item *j*.

C. *Merits and Demerits*

Item-based models use rating distributions *per item*, not *per user*. With more users than items, each item tends to have more ratings than each user, so an item's average rating usually doesn't change quickly [10]. This leads to more stable rating distributions in the model, so the model doesn't have to be rebuilt as often. When users consume and then rate an item, that item's similar items are picked from the existing system model and added to the user's recommendations [2]. Item-based collaborative filtering had less error than user-user collaborative filtering. In addition, its less-dynamic model was computed less often and stored in a smaller matrix, so item-item system performance was better than user-user systems.

V. SPANNING TREE

It is an algorithm that makes use of the graph structure. The algorithm explores a way of designing similarities between products and customers. In this algorithm, the product-customer relationship is modeled as a bipartite graph. One set of the node is the product nodes and the other set is the customer nodes. An edge connecting a product to a customer indicates a review [11]. The edge also carries a rating associated with that review. Now for example, we want to predict the rating between customer u and product x. The algorithm will create a spanning tree from the graph, rooted at product x. The spanning tree will also have two kinds of node, product node and customer node. Each node may have several children. For a product node, the children are the customers that have reviewed the product; for a customer node, the children are the product that he/she has reviewed this edge carries the rating that the customer gave to the product. So finally, with U denotes the set of customers, the predicted rate that customer u gives product x would be given by:

$$rate(x) = \frac{\sum_{u \in U} rate_u(parent(u)) \cdot w(x,u)}{\sum_{u \in U} w(x,u)}$$

VI. BIPARTITE PROJECTION

Besides calculating similarities through collaborative filtering, we can also calculate the user-user coefficient by projecting the bipartite graph to the customer set [11]. The projection will generate a graph only with customer nodes and also show the structure among them. Naive way of doing projection is to simply draw an edge between a pair of customers if they have reviews on the same product. However, the resulting un- weighted graph may not retain enough information in a method based on resource allocation for calculating weights between a pair of nodes when doing bipartite graph projection the resource allocation treats weights as a format of resource. It takes two steps. At the first step, the projected set (customer set) evenly distribute their ratings to the non-projected set (product set). At the second step, the ratings are projected back to the customer set.

From the two-step process, the weight is given by

$$w_{ij} = \frac{1}{k(u_j)} \sum_l \frac{a_{il} a_{jl}}{k(p_l)}$$

Then, an algorithm goes through the weights and builds the recommendation list. For each product l, let a, be the predicted rating of that product given by customer i. We sum over for each customer j, for any rating on product l that j would like to contribute to i. In order to get a rating back in the same range, again we calculate the weighted sum of these ratings.

$$\tilde{a}_{il} = \frac{\sum_j w_{ij} a_{jl}}{\sum_j w_{ij}}$$

In practice, the recommendation system calculates the ‘a’ for all products and recommends the ones with highest score to the customer.

VII. OPEN PROBLEMS WITH RECOMMENDATION SYSTEMS

A. Cross Domain Recommendations

Current systems are really good at learning preferences in one domain (say music), [21] but the same algorithms do not work as well in other domains. E.g. if you like rock and pop in music, what does it say about your movie tastes? It would really be nice to see a unified model of preference for an individual that explains how different domains interact and inform our preferences [12].

B. Constraint -Based Recommendations

Most of the research has focused on virtual goods such as movies and music, where an item can be recommended unlimited number of times. In the real world, that's often not the [21] case. Consider restaurants. If a great restaurant is recommended to more people than it can handle, it struggles to cope up with the load and its service declines (the *crowd avoidance* problem). [19] How do you recommend in domains where the items are limited? Here recommendation becomes a relaxed version of the classic matching problem.

C. Group Recommendation

Here, the basic premise is to recommend an item to a group of people, e.g. going to a movie together. This problem has been explored for some time, but we are still excelling in only a part of the problem. Typical models compute individual recommendations and then use a smart way to combine them. But often there will be disagreements [21], and different groups may have different dynamics. There is a vast literature on consensus and voting strategies and mechanisms that could be explored, as well as multiple-staged user-interaction paradigms [13].

D. Impact of Recommendation

Recommendations do have an effect on the rating tendencies of individuals (Is seeing believing?), but till date there has been little research on understanding how recommendations may impact our preferences. E.g. if I am told that 99% of people like me buy this item, will I be more willing to buy it? What effects can it have on the long-term distribution of sales in the inventory, and how can you control for that? This gets more interesting when you think of a group of individuals in a community [14] [21]. A particular recommendation algorithm may not only restrict people's access to information, it may also influence their opinions. It would be great if there could be community-wide models for understanding and controlling for these effects, rather than just item-based or user-based ones.

E. Recommendation and Social Networks

"Social" recommendation seems to have caught on in a big way. Many companies use social features in their recommendations based on your friends' preferences and other network characteristics [21]. E.g. X and 20 others like this item. However, little is understood about how the social forces underlying a social network interact and influence people's preferences, both towards an item and towards their friends. Social psychologists have vast literature describing theories such as social influence, identity, social proof, conformity etc. but it is unclear how they play out in online social networks. Further, social networks themselves act as vehicles of information, diffusing ideas and information across the network [15]. This means that user preferences in a network are not only dynamic; they have interdependency with the network.

F. Context-Aware Recommendation

With increasing use of mobile devices, users self-disclose a lot of contextual information, e.g. location [21], current activity etc. These data sources give real-time information, which are both an opportunity and challenge for a recommender system [16].

G. Recommendation and Privacy

Having access to large quantities of data about user’s raises big privacy questions. Privacy in recommendation is a major concern, and it would be great to see some theoretical and empirical work. On the theory side, the need is for models that can give guarantees of privacy (such as differential privacy) for certain design of recommender. On the empirical side, it is important to understand the tradeoff between usability and privacy concerns of recommendations, and how to design for more privacy-consistent outcomes [17] [21].

H. Recommendation as Intelligent Task Routing

This is a rather new field which talks about the growth of online communities. A good example is Wikipedia, which is currently trying hard to attract new editors [18]. If you think of each article as a task, then the recommendation task is to suggest new articles to users so that individual interests are well-served, *and* there is a favorable productivity in Wikipedia. More generally, the goal is to understand the growth and evolution of human organizations, and how recommendation (of people or tasks) can help. [20] Or thinking of online reviews community, and how recommendation of previous similar reviews may help you write a better and more useful review. This may also be a useful construct for recommendation in the enterprise [21].

REFERENCES

[1] Wikipedia, 'Recommender system', 2015. [Online]. Available: https://en.wikipedia.org/wiki/Recommender_system.
[2] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-Based Collaborative Filtering Recommendation Algorithms. In Proceedings of WWW ’01, pp. 285–295, New York, NY, USA, 2001. ACM
[3] A. Singla and M. Karambir, 'Comparative Analysis & Evaluation of Euclidean Distance Function and Manhattan Distance Function Using K-means Algorithm', International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 7, pp. 298-300, 2012.
[4] D. Sinwar and R. Kaushik, 'Study of Euclidean and Manhattan Distance Metrics using Simple K-Means Clustering', INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENC E AND ENGINEERING TECHNOLOGY (IJRASET), vol. 2, no. 5, pp. 270-274, 2014.
[5] J. Hauke and T. Kossowski, COMPARISON OF VALUES OF PEARSON’S AND SPEARMAN’S CORRELATION COEFFICIENTS ON THE SAME SETS OF DATA, 1st ed. poland: Adam Mickiewicz University, Institute of Socio-Economic Geography and Spatial Management, Poznań, Poland, 2011
[6] O. Krasnoshchok, 'Collaborative Filtering Recommender Systems – Benefits and Disadvantages', 2015. [Online]. Available: http://recommender.no/info/collaborative-filtering-approach/.
[7] R. J. Mooney and L. Roy. Content-Based Book Recommending Using Learning for Text Categorization. In Proceedings of DL ’00, pp. 195–204, New York, NY, 2000. ACM Press.
[8] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, John, and T. Riedl. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22:5–53, 2004.
[9] D. Lemire and A. Maclachlan, 'Slope One Predictors for Online Rating-Based Collaborative Filtering', 2005.
[10] P. Symeonidis, M. Ruxanda, A. Nanopoulos, and Y. Manolopoulos. Ternary Semantic Analysis of Social Tags for Personalized Music Recommendation. In Proceedings of ISMIR ’08, pp. 219–224, 2008
[11] J. Hu and B. Zhang, 'Product Recommendation System', CS224W Project Report, 2012.
[12] P. Winoto and T. Tang, 'If You Like the Devil Wears Prada the Book, Will You also Enjoy the Devil Wears Prada the Movie? A Study of Cross-Domain Recommendations', New Generation Computing, vol. 26, no. 3, pp. 209-225, 2008
[13] S. -Yahia, S. Roy, A. Chawlat, G. Das and C. Yu, 'semantics and efficiency', Proceedings of the VLDB Endowment, vol. 2, no. 1, pp. 754-765, 2009.
[14] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan and J. Riedl, 'Is seeing believing?: how recommender system interfaces affect users' opinions', CHI '03 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 585-592, 2003.
[15] J. Leskovec, A. Singh and J. Kleinberg, 'Patterns of influence in a recommendation network',PAKDD'06 Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining, pp. 380-389, 2006.
[16] G. Adomavicius and L. Baltrunas, http://cars-workshop.org/, 2012. .
[17] F. McSherry and I. Mironov, 'Differentially private recommender systems: building privacy into the net', KDD '09 Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 627-636, 2009.
[18] D. Cosley, D. Frankowski, L. Terveen and J. Riedl, 'using intelligent task routing to help people find work in wikipedia', IUI '07 Proceedings of the 12th international conference on Intelligent user interfaces, no. 32-41, 2007.
[19] G. Adomavicius and A. Tuzhilin, '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, vol. 17, no. 6, pp. 734-749, 2005.
[20] Recommendation Systems, 1st ed. california: infolab, 2009, p. chapter 9.
[21] Sharma, Amit. "What Are Some Of The Most Interesting Research Problems In Recommender Systems? - Quora". Quora.com. N.p., 2012. Web. 20 Jan. 2016.