A Context Based Cross Domain Collaborative Filtering Approach in Folksonomies

Harshit Sharan
IDD (B.Tech+M.Tech)
Dept. of Computer Engg.
Indian Institute of Technology (BHU)
Varanasi, India

Abhinay Agrawal
Integrated Masters Degree
Dept. of Mathematical Sciences
Indian Institute of Technology (BHU)
Varanasi, India

Dr. Rajeev Srivastava
Associate Professor
Dept. of Computer Engg.
Indian Institute of Technology (BHU)
Varanasi, India

***Abstract*—Recommender Systems have the task of analyzing already available Content based data of the items or the socially available rating data about the items available from the social network, e commerce websites, etc. This data after analysis lead to the prediction of ratings of items which are not already rated by a user. Crossdomain approaches can be applied to the collaborative filtering recommender systems so as to tackle the cold start problem of the approach. Also, it is good to acquire the current context of the user before recommending items like music. To deal with scalability problems, a distributed approach is required for implementing the recommender system. Here we propose a context based cross domain approach, that can be implemented in a distributed architecture using the mapreduce model on systems like Apache Hadoop.**

Keywords—Recommender System, Collaborative Filtering, Mapreduce, Cross domain collaborative filtering, Context Based Recommendation

I. INTRODUCTION

Recommender or Recommendation Systems [1] are a children class of information filtering system that intend to anticipate the 'rating' or 'preference' that a user might give to a product.

In the recent age of explosive information, recommender systems have become extremely important and common, both in terms of usage and in the fields of research areas. The most famous recommendation systems are built to predict the choice in the field of movies, music, books, news, search queries, social tags, research articles, and other marketing products in general. However, there also exist recommender systems in less common areas of jokes, financial services, experts, life insurance, persons (dating), restaurants, and social network followers/friends.

In academic research areas, multiple issues concerning the recommendation systems have been discussed, such as the recommendation algorithms [2] and [3], evaluation metrics [4] and [5], context-awareness recommendation [6], human interaction, etc. As part of of actual business applications, recommendation systems have also been found to be very successful.

Recommender systems generally generate a list of recommendations in one of the two ways – through collaborative or content-based filtering. Collaborative filtering [7] techniques create a model from a user's past behavior as well as alike decisions made by other users; then use that model to predict items, or rate items that the user may have interest in. Content-based filtering [8] approaches utilize a series of distinct properties of an item in order to recommend additional items with similar characteristics. These approaches are often combined to give Hybrid approaches.

The collaborative filtering approaches suffer from the challenges like that of Data Sparsity [9], Scalability [10], Grey Sheep [11], Long Tail [12], etc. Here in this paper we try to address three major concerns which are that of data sparsity, scalability, and context awareness. Although, context awareness of a recommender system is not a challenge pertaining directly to the collaborative filtering approach, it is an issue to keep in mind while designing recommender systems in general. For example, the next generation recommender systems for music, books, movies, etc should be able to grasp the context in which the user is currently in, and then predict something for him using the traditional approaches.

Traditional approaches predict by learning from a user-item dualistic relationship. These systems neglect the user's behavior and current mood and activity that he is performing. The current mobile devices are getting smarter and thus acquiring information concerning the context of the user can be easily and portably obtained. The portable devices now have a platform that can be provided with many Application Programmer Interfaces or APIs that can track this information [13]. The user, moreover, equipped with such devices would like to have access to music or books that would suit to his or her current context.

Most of the efforts and researches only focus on within-one domain recommendation until now. This leads to a major problem which is of Data Sparsity in the respective domain. For cross-domain recommendation problem, there still exists great potential both in research areas and in business fields.

Cross domain recommendation [14] can bring a lot of gains to both the users and recommendation engine websites. In the traditional recommender systems, when user is going through resources in one domain, the recommended item list is only formed from this domain. So, why not recommend a

classic movie like "Pursuit of Happiness" when the user is looking for inspirational books? Why not recommend a relaxing song when the user has read philosophical books? In this way, the user experience improves by giving more diversified and serendipitous recommendations. Moreover, as websites already have users' preference rating data in original domains, cross domain recommendation can be used to quickly open up new fields in business, saving precious money and time. In addition, cold-start problem and/or data sparsity [15] problem in the target domain can be also mitigated by the cross-domain recommendations.

Although many models have been proposed for traditional single domain recommendations, most of these models can't be directly used for solving cross domain recommendation problem. Traditional recommendation models derive user's preferences depending on the previous preference information from the same domain. On the other hand, preference information in the target domain is completely unknown or little known for cross-domain recommendation.

For cross-domain recommendation, if we are able to utilize preference data in the source domain to predict the user's preference data in the target domain, then the present information and the deduced information are from the same domain, and the cross-domain problems are shifted to single domain problems, and thus the various models in traditional single domain can be directly used. Thus, the main issue is how to make the connection between the different domains.

In general, the domains are mutually exclusive, each dealing with a certain type of item (e.g, movies, music, books), and hence it is troublesome to get common properties from data to make the bridging connection among the different domains. The user generated tags can be used instead of the domain data to link domains in this paper. Systems that utilize use user-created-tags are called folksonomy.

Folksonomies are systems that collaboratively create and manage tags to comment on the resources' properties [16]. In place of choosing particular resources as features, tags in folksonomies have greater advantages for solving cross-domain recommendation problem:

1. Separate domains might have different resources, but can share many tags which have similar meanings. For example, "action" can be used as a tag for both a thriller novel, and an action movie. Hence, its easy to make the bridge between the domain using the tags.
2. Tags have a comparatively better understanding of the user preferences. If we know user's favorite tagswe can obtain what factors have an influence on user's preferences.
3. Tags can be used to end the sparsity problem. There can be lots of resources in one domain in ecommerce websites so that the resulting user-rating matrix is sparse. However, the number of tags in a domain is not so much. The conversion from resources to tags thus alleviates the problem of sparsity.

Inspired by these ideas, an algorithm has already been proposed for Cross Domain recommendation in folksonomies: CRF [17]. The idea of CRF is to generate user's tag profile in the target domains.

We propose an extension to the CRF algorithm, a CCRF algorithm, which takes into account the Context behavior of the user and then utilize the cross domain potential of the recommender systems. The algorithms in the recommender system can then be implemented in a distributed way using the mapreduce models and libraries.

MapReduce [18] is a programming paradigm and arelated implementation for analyzing and generating very large data sets. Programmers write a map function that receives key/value pairsand thencreate a set of intermediate key/value pairs, and a reduce function that combines all intermediate values associated with the same intermediate key. Many real world tasks are representable in this model.

Programs written in this functional style are automatically parallelized and run on a large cluster ofgeneral purpose machines. The run-time system takes care of the details of the scheduling the program's execution across a set of machines,partitioning the input data,mitigating machine failures, and managing the required inter-system communication. This allows programmers who have no experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

II. RELATED WORK

In traditional Recommendation System with collaborative filtering technique, rating data by users is used to recommend an item (a set of items). Here, active user's context and his preferences are not considered during the process of recommendation.

The ubiquitous Recommendation System also deal with context information of users. For example, nn case of music data, the evaluation is done according to the user's current context to determine the type of user and favorable music to be recommended.

In comparison to the single domain recommendations, cross domain recommendations are more helpful in real world. Traditional single-domain recommendation methods are adopted by Amit [16] to recommend items from other domain, to evaluate impacts of dissimilar source domains on recommendation outcomes. Cross-system user modeling [19] combines tag-based user profiles from dissimilar domains. However, all these works do not put forward a cross-domain recommendation algorithm.

As exemplified above, detecting linkage among domains is the key challenge for cross-domain recommendation, there are several papers that try to solve it from different prospects. Multi-domain Collaborative Filtering [20] unfolds probabilistic matrix factorization to learn a correlation. Both rating pattern and correlation matrix mentioned above are unquestioning, [15] and [21] use the implicit data to transfer knowledge between different domains. On the contrary, TagCDCF [22] follow similar tags among different domains, and experimentation proves that this explicit information is more reliable and efficient for cross-domain recommendation. However, if information of common tags is sparse, TagCDCF can hardly be applied.

III. PROBLEM DEFINITION

The problem's structure can be stated as follows: there is one source domain S and one target domain T. The subscripts $_S$ and $_T$ are used to separate the source domain and the target domain terms. Moreover, tags are used instead of resources in the folksonomies. Therefore, consider the following definitions.

Definition 1: Source Domain and Target Domain can be shown by S / T = {U,R,T,Y}, where U, R and T are finite sets and called users, resources and tags respectively. Here, Y denotes a ternary tag assignment relation between them, Y⊆ U x R x T . For ∀ u ∈U , Y^u represents u x R x T .

*Definition 2:*User profile for user u is denoted as $profile_u = (n_1^u , n_2^u ,\ldots, n_m^u)$, m is the number of tags in this domain. For ∀ i ∈ [1,m], n_i^u measures the times user u uses tag i.

Definition 3: Context log consists of the log of history of users who previously used the item in the target domain in a particular context. For example, the data stored can consist of songs listened by the user, alongwith the context in which the user was in.

*Definition 4 :*Contextually similar users [23] are the ones which are similar to a given user with respect to current context of that user. They can be found out using a reduction based approach [24]. This approach converts a multidimensional recommendation problem to a 2 dimensional User x Item space.

For example, a three dimensional rating function can be defined as

$$R^{D}_{User\ X\ Content\ \ X\ Time} : UXCXT \rightarrow rating$$

where D consists of records as *user, time, content, rating* for the user-specified ratings. This 3-Dimensional function can be reduced to a 2-Dimensional function as follows:

$$R^{D[Time=t](User,Content,rating)}_{User\ X\ Content}\ (u,c)$$

where, D[*Time=t*](*User, Content, Rating*) represents a rating set found from D by selecting only those records where *Time* dimension has value t. Thus, if D is a 3-Dimensional relation, then D[*Time=t*](*User, Content, Rating*) is simply another relation obtained from D by performing 2 relational operations : selection followed by projection.

Context Type	**Abbreviation**	**Possible Values**
Location	LN	Indoor, Outdoor
Motion	M	Stop, Slow, Middle, Fast
Calender	C	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
Time	T	Morning, Afternoon Evening
Light	LT	Bright, Moderate, Dim, Dark
Humidity	H	High, Moderate, Low
Air Temperature	AT	High, Moderate, Low

Table 1. The various contexts and their possible values

IV. THE PROPOSED CONTEXT BASED CROSS DOMAIN RECOMMENDATION IN FOLKSONOMIES (CCRF)

The initial step of the proposed framework involves the search of contextually similar users. The reduction based approach [24] is used for this. The rating can be written as

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

where*i* is the item rated by user *u*.

Different approaches have been used to calculate similarity measure *sim(u, u')* between users. Usually, similarity depends

upon the ratings of items that has been rated by both the users *u* and *u'*.

Among most popular approaches correlation based approach,

$$sim(x,y) = \frac{\sum_{s \in Sxy}(r(x,s) - r'(x))(r(y,s) - r'(y))}{\sqrt{\sum_{s \in Sxy}(r(x,s) - r'(x))^2 \sum_{s \in Sxy}(r(y,s) - r'(y))^2}}$$

and cosine distance based approach,

$$\cos(X,Y) = \frac{X \,.\, Y}{||X|| * ||Y||}$$

$$= \frac{\sum_{s \in Sxy} r(x,s) r(y,s)}{\sqrt{\sum_{s \in Sxy} \left(r(x,s)\right)^2} \sqrt{\sum_{s \in Sxy} \left(r(y,s)\right)^2}}$$

where $r_{x,s}$ and $r_{y,s}$ are the ratings of item s given by users x and y respectively. $S_{xy} = \{s \in Items \mid r_{x,s} \_ = \varepsilon \wedge r_{y,s} \_ = \varepsilon\}$ is the set of all items co-rated by both users and X·Y denotes the dot product of rating vectors of corresponding users.

Using above stated approach, it is possible to reduce multidimensional context information of user to a 2-dimensional matrix with users, items and ratings only and contextually similar users with maximum relevancy to the current users' context can be extracted.

After the contextually similar users are extracted, three more steps are followed which take the problem of rating prediction to the next level of Cross Domain recommendation.

The Profile Generation Algorithm involves generating the user profile of the contextually similar users, in the source domain. This happens on the basis of two information: User-resource information and the resource-tag information. The algorithm works under the following assumptions:

- Tags of a resource describe the resource's characteristics well;
- The most frequently used tags of a resource can represent all the tags of the resource.
- If user u has used resource r , the typical tags of r show the preference of u .

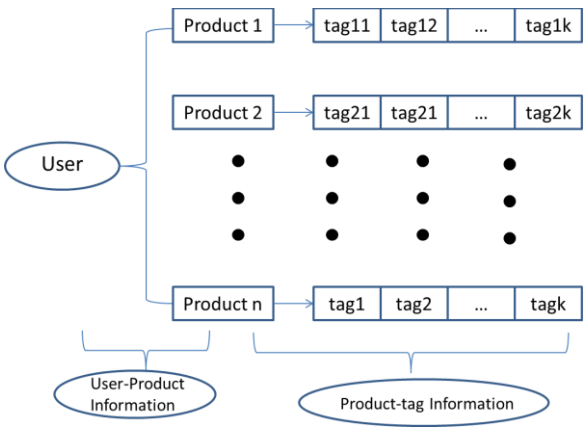

Figure 1.Representation of User's behavior information

The idea of this step is to use "resource" as a linking bridge to build the relationship between users and the tags. Each resource experienced by the user is traversed in one domain, and for each tag the resource has, 1 is added to the corresponding number in the user's tag profile.\

The next step is the Profile Mapping Algorithm, which is key to generate the user's profile in the target domain. The main challenge of this algorithm is to find the correlation between the two domains' tags. If we know that, we can find the most similar tag in the target domain for each tag from the user's given profile.To measure the similarity between the two domains' tags, the idea of collaborative filtering is used [25] and [26]. For the train set of users, their profiles are created using the Profile Generation Algorithm and then the user tag matrices are generated from these profiles. For ex, to find out the tag most similar to a tag 't', evaluate and rank the similarities of each column vector in the target domain matrix and the column vector of t.

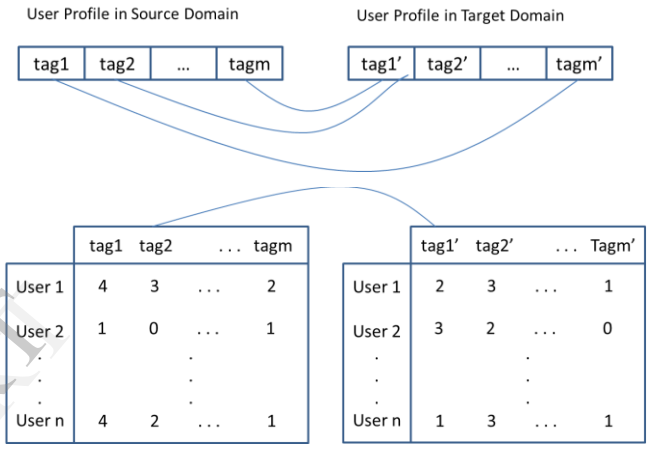

Figure 2. Tag Mapping

Also, as discussed in the paper [17], there can be various methods of feature selection. As depicted in the figure, [tag1, tag2,…,tagm] and [tag1', tag2',…, tagm'] are used to make preference space in source domain and target domain, respectively. Three ways are given in the algorithm CRF to choose the tags from the sets [tag1, tag2,…,tagm] and [tag1', tag2', tag3',…, tagm'].

- *CRF-all*: all the tags in the domain are used as features
- *CRF-common*: only the common tags shared by the two domains are used as features. In this method, there is no need to calculate the similarity between the tags of the 2 domains
- *CRF-typical*: the algorithm's efficiency is effected by the scale of features. Thus, if the number of features can be deduced, algorithm's performance can be improved.

The complete workflow of the proposed CCRF algorithm is shown in the following block diagram:

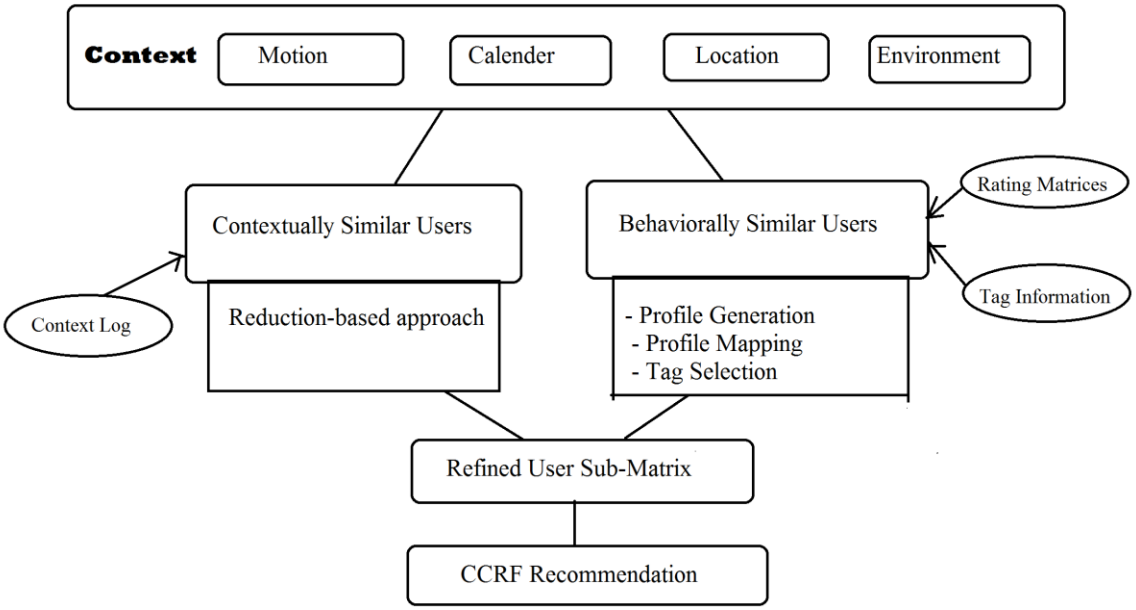

Figure 3. Block Diagram describing the full framework

Scalability is another issue that needs to be tackled and the above proposed framework can be easily made scalable. The framework consists of individual steps that involve the implementation of collaborative filtering approaches. The user based collaborative filtering algorithm can be implemented in a distributed fashion using the mapreduce model [27], on Apache Hadoop. Hadoop's machine learning library, Mahout [28] provides easy and direct platform for running such algorithm.

V. CONCLUSION

The area of recommender system has been deeply studied and the ideas of single domain collaborative filtering and content based approaches have been exhaustively used. As such, to improve the quality of recommendations, and mitigate other problems of collaborative filtering approaches, context based and cross domain approaches have been respectively studied. Here we presented a combined approach, involving both the ideas, i.e. context based and the cross domain models of recommendation systems. Moreover, the algorithmic part of the proposed framework majorly consists of usage of collaborative filtering techniques of the recommender systems, which can be easily implemented in a distributed manner, using the Mapreduce model in frameworks like Apache Hadoop.

VI. FUTURE WORK

The development of portable operating systems and devices can lead to easy capturing of context information of a user. Thus, the proposed framework can be deployed easily for systems like music recommender system.The social networking websites and other platforms are a good source of cross domain data. This system, in future, will be implemented using the Mapreduce model, so that it is scalable and can be run on commodity distributed environments like Apache Hadoop.

VII. REFERENCES

[1] Francesco Ricci, Lior Rokach and Bracha Shapira."Introduction to Recommender Systems Handbook". Springer, 2011

[2] M. J. Pazzani. "A framework for collaborative, content-based anddemographic filtering". *Artificial Intelligence Review*, vol. 13, no.5-6, pp. 393-408, 1999.

[3] Joseph A. Konstan . "Introduction to recommender systems: Algorithms and Evaluation"

[4] A. Gunawardana and G. Shani. "A survey of accuracy evaluation metrics of recommendation tasks".*The Journal of Machine Learning Research*, vol. 10, pp. 2935-2962, 2009.

[5] H. Steck, A. Lucent, and M. Hill. "Item popularity and recommendation accuracy".*Proc. fifth ACM Conference on Recommender Systems*, New York, USA, 2011, pp. 125-132.

[6] G. Adomavicius and A. Tuzhilin. Chapter title: Context-Aware Recommender Systems. Book title: *Recommender Systems Handbook*, Springer, 2011, pp. 217-253.

[7] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen."Collaborative Filtering Recommender Systems".

[8] Michael J. Pazzani and Daniel Billsus. "Content-Based Recommendation Systems".

[9] Miha Grčar, Dunja Mladenič, Blaž Fortuna, Marko Grobelnik. "Data Sparsity Issues in the Collaborative Filtering Framework".Advances in Web Mining and Web Usage Analysis Lecture Notes in Computer Science Volume 4198, 2006, pp 58-76

[10] G´abor Tak´acs , Istv´an Pil´aszy, Botty´an N´emeth, Domonkos Tikk. "Scalable Collaborative Filtering Approachesfor Large Recommender Systems"

[11] Mustansar Ali Ghazanfar, Adam Prugel-Bennett. "Fulfilling the Needs of Gray-Sheep Users inRecommender Systems, A Clustering Solution".

[12] Yoon-Joo Park , Alexander Tuzhilin."The Long Tail of Recommender Systems and Howto Leverage It"

[13] "Android.com: Application programming interfaces". Website: http://developer.android.com/index.html

[14] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, Francesco Ricci. "Cross-domain recommender systems: A survey of the State of the Art"

[15] Bin Li, Qiang Yang and Xiangyang Xue. "Can Movies and Books Collaborate?Cross-Domain Collaborative Filtering for Sparsity Reduction"

[16] A. Tiroshi and T. Kuflik. "Domain ranking for cross domain collaborative filtering".*Proc. 20th International Conference on User Modeling, Adaptation, and Personaliztion*, 2012, pp. 328-333.

[17] Ying Guo and Xi Chen."A Framework for Cross-domain Recommendation in Folksonomies".

[18] Jeffrey Dean and Sanjay Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters".

[19] F. Abel, S. Araujo, Q. Gao, and G. J. Houben. "Analyzing cross-system user modeling on the social web".*ICWE*, vol. 6757, pp. 28-43, 2011.

[20] Y. Zhang, B. Cao, and D. Y. Yeung, (2012). "Multi-domain collaborative filtering". arXiv preprint arXiv:1203.3535.

[21] B. Li, Q. Yang, and X. Xue. "Transfer learning for collaborative filtering via a rating-matrix generative model".*Proc. 26th International Conference on Machine Learning*, 2009, pp. 617-624.

[22] Y. Shi, M. Larson, and A. Hanjalic. "Tags as bridges between domains: improving recommendation with tag-induced cross-domain collaborative filtering".*Proc. 19th International Conference on User Modeling, Adaption, and Personalization*, 2011, pp. 306-316.

[23] Reena Pagare, Intekhab Naser, Vinod Pingale, Nayankumar Wathap. "Enhancing collaborative filtering in music recommender system by using context based approach"

[24] Gediminas Adomavicius , Ramesh Sankaranarayanan, Shahana Sen, Alexander Tuzhilin. "Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach"

[25] G. Linden, B. Smith, and J. York. "Amazon.com recommendations: item-to-item collaborative filtering".*Internet Computing, IEEE*, vol. 7, no. 1, pp. 76-80, Jan/Feb 2003.

[26] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. "GroupLens: An open architecture for collaborative filtering of netnews".*Proc. ACM Conference on Computer Supported Cooperative Work*, New York, USA, 1994, pp. 175-186.

[27] Zhi-Dan Zhao , Ming-Sheng Shang."User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop". Knowledge Discovery and Data Mining, 2010. WKDD '10

[28] "Apache Mahout: a scalable machine learning library". Website: https://mahout.apache.org/