International Journal of Allied Practice, Research and Review
Website: www.ijaprr.com (ISSN 2350-1294)

Pre Filtering Approach with Association Rule Mining for Context Based Recommendation

Arati R. Deshpande and Dr. Emmnuel M.
[1]Research Scholar, Computer Engineering,
Pacific Academy of Higher Education and Research University, Udaipur, India
[2]Department of IT, Pune Institute of Computer Technology, Pune, India

Abstract - Recommendation systems are part of personalized web applications which suggest the users about the items to explore more which can be of more interest to them. The context based recommendation has been used currently to extract and incorporate the context as information to improve the quality of recommendations. The context can be the user or item attributes or the situation of the use's interaction with the system. A context based recommendation system uses pre filtering or post filtering or contextual modeling approach to generate recommendation. The pre filtering approach uses any one of the existing recommendation algorithms with the addition of filtering process in recommendation. A pre filtering approach with association rule mining is proposed which can use the collaborative filtering method for recommendation which uses the rating matrix. The proposed method extends the reduction based approach with class association rule mining to form the contextual segments of rating matrix for recommendation. The contextual segments are evaluated with the collaborative recommendation which shows the effect of context in recommendation. This method can be applied in pre filtering approach of context based recommendation for improvement in quality of recommendation.

Keywords-- ***Context based/aware recommendation system; Pre filtering; Class association rule mining; Apriori algorithm; Collaborative filtering.***

I. Introduction

The Recommendation systems have become a very important aspect of any application which gives the relevant information out of more available data or options. The items recommended by recommendation depend on the applications which vary from different products to people recommendation [1]. The context based recommendation systems have been developed to improve the recommendation quality which also use the context of the interaction of the user with the system. The context of interaction can be user properties or item properties or the environment of interaction. The context can be current situation, time or location. The context with other data such as user and/item profile can be used to increase the relevance and accuracy of recommendation. The quality of recommendation will increase the user's trust in the application/service. This also benefits the service provider to get more clients and profit.

The user's interest differs in different situations. The context plays an important role in making the decision. It will influence the user to make the choice related to the context. The context based recommendation system makes use of the available context as the information in addition to the user, item and rating information to give recommendation. The context can be applied with existing recommendation algorithm using any one of the approaches namely pre filtering, post filtering and contextual modeling [2].

The pre filtering approach includes the context in recommendation to reduce the history of information. This makes the approach to use the existing recommendation algorithm without context, after filtering the data to be used for recommendation. Post filtering approach filters the recommendation list generated by a recommendation algorithm to provide the recommendation according to active user context. In this approach context filter is applied on recommendation list. The contextual modeling approach incorporates the context of active user in prediction engine. This will change the existing recommendation algorithm to make use of context for providing the recommendation.

The quality of recommendation depends on how much relevant these items are to the user. Though the relevance of recommendations can be increased with the use of, user or item profiles, it is difficult to make accurate recommendations as the predictions are done with the available user and item data. The challenges of context based recommendation include context representation, quality, sparsity, and scalability. In this paper a pre filtering approach is proposed to produce the reduced user item matrix of recommendation with association rule mining.

II. Related Work

The traditional recommendation methods incorporate the data about users, items and implicit or explicit ratings to predict the ratings for items to recommend top N items which have higher predictions and are not seen by the user. The three methods of recommendations are collaborative filtering, content filtering and hybrid filtering [3]. But the items preferred by any user also depend on the context at the time of user interaction with the system.

A. Pre Filtering Methods for Recommendation

A multidimensional view of recommendation was proposed in [4] by incorporating context as a third dimension in addition to user and item dimensions of user item matrix. The traditional recommendation system is taken as two dimensional one with user and item. The recommendation space *R* with user, item and context dimensions as *U*, *I*, and *C* correspondingly can be given as,

$$R: U \times I \times C \ \rightarrow Rating \tag{1}$$

$$R: U \ \times I \rightarrow Rating \tag{2}$$

The reduction based method reduces the user item matrix to contain the ratings of items given in the specific context. For user *U*, item *I* and context *T*(time) dimensions, $\forall\,(u,i,t) \in (UxIxT)$ the rating prediction function $\acute{R}$ is,

$$\acute{R}(u,i,t)^{D}_{U\text{x}I\text{x}T} \ = \acute{R}(u,i)^{D(Time=t)(u,i,r)}_{U\text{x}I} \qquad (3)$$

where $D(Time=t)(u,i,r)$ denotes the rating set obtained by selecting only those records where time dimension has value t which is taken as the contextual segment with time as context and with value t. This uses a context segment which is the generalization of any context like Monday is the specific context and weekday is the generalization. The users and items are selected from generalized context segment for prediction to which the given context belongs to. The evaluation is done for a movie recommender system with MAE, precision, recall and F1 measure. It is shown that the reduction based method outperforms the user based collaborative filtering for some contextual segments. The contextual segment on which the reduction based method outperforms depends on the application.

Item splitting method is given in [5] for the pre filtering approach uses context to split items and applies the traditional collaborative filtering on the *User* X *Item* matrix for recommendation. In this, item with different experience in two different contexts is split in to two items with same rating like item in summer and item in winter. If the ratings for item i, are different under context $c=c_j$ and $c\neq c_j,$ then the item is split in to two with one having ratings of item i when $c=c_j$ and other having ratings of item i, when $c\neq c_j$. The rating predictions for all items not rated by target user are computed with modified *User X Item* matrix. Top K items with highest predicted ratings are recommended. The context based approach with reduction and item splitting have better accuracy than context free approach.

The time is used as the context dimension in many context based recommendations. A pre filtering context based recommendation with time context is used in [6]. Three 'x' month's duration from current date (time) is used to define the contextual segments for prediction. Collaborative filtering is applied for rating prediction in three contextual segments to obtain the three lists of recommendations. The lists are combined with the weights given by Fuzzy Inference System (FIS) to obtain the recommendation for popularity. The pre filtering method improves the recommendation process evaluated with Movie lens dataset, as only relevant data is taken for recommendation.

A scalable context aware recommendation system [7] uses pre filtering with clustering. The clustering of users is done before collaborative filtering to reduce the size of user item matrix. The users are clustered with hierarchies according to their demographic values as context. The run time performance of collaborative filtering increases by a factor of k if k equal partitions of users are created. The quality may reduce as the numbers of ratings belong to one cluster only.

Thus the pre filtering approach uses the filter which can be a data mining algorithm to reduce the user and rating matrix depending on the context of the current user. The reduced data of user, item and rating according to context are called contextual segments. The recommendation is generated for current user using one of the contextual segments generated. In the proposed approach the reduction based method is modified to use association rule mining to generate contextual segments.

III. Context Modeling

The context is defined as "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between the user and an application, including the user and applications themselves" in [8]. The context acquired from the interaction of users with the system may not be in the form required by the recommendation process. The information can be the time stamp which has to be converted into day, date and time or season for the recommendation. This information of context is represented as a hierarchical form in [2], [4]. The modeling of context can be done with different methods which are taken from pervasive and ubiquitous computing. The model used in the proposed approach is the object oriented model given in [9].

A representational view of context is used in many context based recommendation systems [10]. The context information to be used is identified for the application at the design time. The context is stored as an attribute associated with the rating or the interaction of a user with an item. Each type of context like time, location and intent of purchase is considered as a dimension. Each dimension consists of a set of contextual elements which can take values from a particular domain. In a context dimension the contextual elements are related in a hierarchical way. This hierarchy can be stored as a predefined form for each context dimension. One context dimension can have different hierarchical structure as depicted in Fig.1.

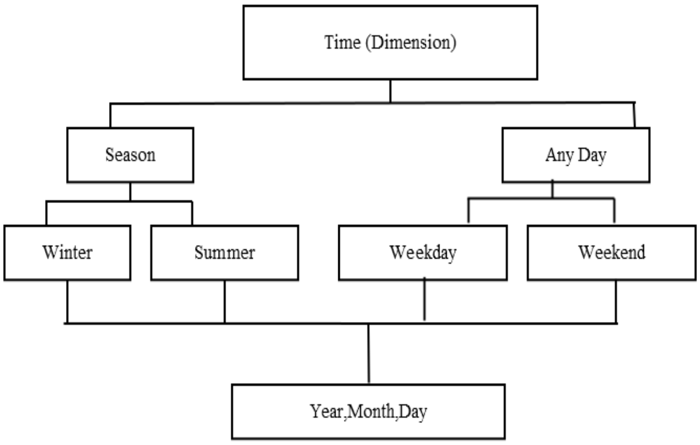

Fig.1 Time context hierarchy

A. Context Storage with Rating

The context information is processed and stored in the data repository. The user interaction data like rating, user profile, item profile are also stored in the data repository. The context is stored with each action of the user while interacting with the application. The actions to be considered for recommendation are decided at analysis and design before implementation. The example of actions are rating and tagging. For each dimension of context, the context elements and their values are stored with each action. For example, the context elements of time dimension can be Season and Time of the day. The Season can have values of summer, winter autumn and spring. The Time of the day can have values morning or evening. The association rules are generated with the action as the class of the rule. These are called class association rules

The context dimensions with context elements are arranged from left to right in table according to order of preference. Context elements of context dimensions are selected from the context hierarchy of each dimension at design time which will be considered in recommendation system. For example, if Time and Age are considered as context dimension, then Time: season, Time: weekday, Time: daytime, Age: adult can be the context elements. Each context element has values defined for that attribute. For example, Time: season = summer and Age: adult= young. In movie dataset, time dimension can have Time: season, Time: weekday, Time: time (time of day) as context elements of Time dimension The rating matrix is stored with context for each of the user action like rating with context element values in rating table as shown in Table I. The values of attributes are explained in section V. The class attribute 'ratingout' is added to generate rules. The 'ratingout' values can be pos or less depending on the rating value is more or less respectively. The association rule is applied on modified rating table to generate rules which are used to generate the contextual segments.

Table I

User Rating Table for User KNN with Context

userID	movieID	rating	season	weekday	time	ratingout
75	3	1	Autumn	we	E	less
75	32	4.5	Autumn	we	E	pos
75	160	2	Autumn	we	E	less
78	17	4	Autumn	we	N	pos
78	29	4.5	Spring	wd	E	pos
78	32	5	Spring	wd	E	pos
127	1707	3	Autumn	wd	E	pos
127	1911	2.5	Summer	wd	E	pos
127	2013	3.5	Summer	wd	E	pos
175	1	5	Winter	we	M	pos

IV. Pre Filtering with Class Association Rule Mining

The pre filtering is applied to user item rating table with context to generate the contextual segments. The context rules generation system generates the rules for pre filtering which are stored in the data repository. The rules are generated to extract the relation of context to user action like rating. The influence of context on user action is modeled as class association rules [12]. The class association rule given by [13] uses the apriori algorithm to generate rules but the rules with only one consequent are generated. The consequent is the class attribute chosen for rule mining. The proposed context rule generation uses the class attribute as action like rating and values of action are the classes for the rules generated. The context dimension used for testing is Time and the class attribute is 'ratingout' which depends on rating values. The values defined for class are Pos and less which shows the positive and negative rating. Each rule is of the form,

$$contextvariable_1 = value_1, \dots\ contextvariable_k = value_k => action_i = value_i$$

(4)

Where k is the number of context elements in the rule and $k<=n$ if n is the total number of rules. The antecedent of the rule has context elements with values.

B. Algorithm for Context Rules Generation and storage

The class association rule mining is used for generation of rules of context with rating. The rules generated are stored in the rule repository for recommendation system. The main steps of rule generation are,

1) Extract the user, item, context and action (rating) data.
2) Apply class association rule mining.
3) Generate the rules of the form as given in equation 4.
4) Store the top K rules.

The consequent of the rule is the class action with its values. The action can be like rating, downloading or tagging. The action is the last element in the table. Define the action values which will decide the class of the rule. The context elements and action values are extracted from the user rating table shown in Table I. The table is input to rule generation algorithm with context elements and action (rating) as shown in Table II.

Table II

Table of Input to Context Rule Generation

Context elements of User Rating Table			
season	**weekday**	**time**	**ratingout**
Autumn	we	E	less
Autumn	we	E	pos
Autumn	we	E	less
Autumn	we	N	pos
Spring	wd	E	pos
Spring	wd	E	pos
Autumn	wd	E	pos
Summer	wd	E	pos
Summer	wd	E	pos
Winter	we	M	pos

Algorithm Listing for Context Rule Generation:

1. Arrange the important context dimension attributes from left to right.

2. Arrange the action as the right most element in the table

3. Set the class attribute as rightmost one(rating out)

4. Generate the Class Association rules with minimum support and confidence

5. Find the number of ratings in segments with these rules.

6. If any segment will not have enough number of ratings or actions discard those rules.

7. Create and use these segments for rating prediction or preference prediction in online recommendation. (Segment is the users, items and rating values in that context)

The rules generated are stored in the rule repository. The stored rule table is in the same form as Table II but if any attribute is missing in the rule, its value is replaced with null value. The rules are pruned according to minimum support and minimum confidence. The contextual segments are created with the rules and also stored in the recommendation system repository.

V. Experimental Results

The data set of Hitrec 2011[14] is used for the experiment to generate rules. The dataset is sampled and context elements are identified and generalized for the experiments. The items to be recommended are movies. The action of users on the movies is considered as rating. The dataset contains 2113 users and 10197 movies and 47957 tag assignments and 855598 rating assignments. The rating has the scale of 0.5 to 5 in intervals of 0.5 with 5 as highest rating. The sampling is done for 150 users and 200 items such that the users who have both rated movies and tagged are selected and the reduced dataset contain 5109 ratings.

The context rules for rating action are generated. The context dimension selected from the dataset is ‘Time’ dimension. The ‘Time’ dimension context elements are generalized in to season, weekday (day is weekend or weekday) and time. The context elements have values,

season = {summer, winter, autumn, spring}

weekday = {weekend(we), weekday(wd)}

time={ morning(M), noon(N) evening(E)}

ratingout ={ positive(pos) , negative(less) }

The context elements of ‘Time’ dimension are generalized in to three context elements from the date and hour: minutes; second data of the dataset for rating action. The generalization of context from specific date and time is to reduce sparsity and also include context to increase accuracy. The context elements are decided depending on the influence of these on the rating action on movies as time decides the watching of a particular movie. The context rule generation extracts the rules from each action which is rating. The attributes of table for generation of rules are selected as season, weekday, time and 'ratingout' for rating action as shown in Table II. The 'ratingout' attribute is derived from numerical attribute of rating having scale of 0.5 to 5 which is discredited into positive (pos) and negative(less) values according to rating is more or less.

Table III

Rule Table for Rating

Ruleid	season	weekday	time	ratingout
1	summer	wd	E	pos

The generated rules are stored in rule table with the context elements and action. The rule table has context elements and action as attributes. The action is the classes attribute which is rating. The structure is shown in Table III.

In order to generate the rules, WEKA software [15], [16] is used. It provides machine learning algorithms to implement many data mining tasks. It is an open source software. An i5 processor with 6 GB RAM, Windows 7 and Java 1.8 with Eclipse IDE is used to implement the proposed algorithm and to write the associated functions. The class association rule mining is used to generate the context rules for each of the actions. The actions used for the dataset are rating. In the experiment, the rules are generated with different minimum support and confidence. The number of rules generated is varied to find the rules with more number of ratings in each context segment.

The rules are generated with the ratings action. The positive rules generated are taken for pre filtering. The rules are pruned with positive actions and also number of ratings associated with the each rule. A threshold of 1000 ratings is kept for pruning rules before storing the rules in rule table. The sample rules generated are shown in Fig.2.

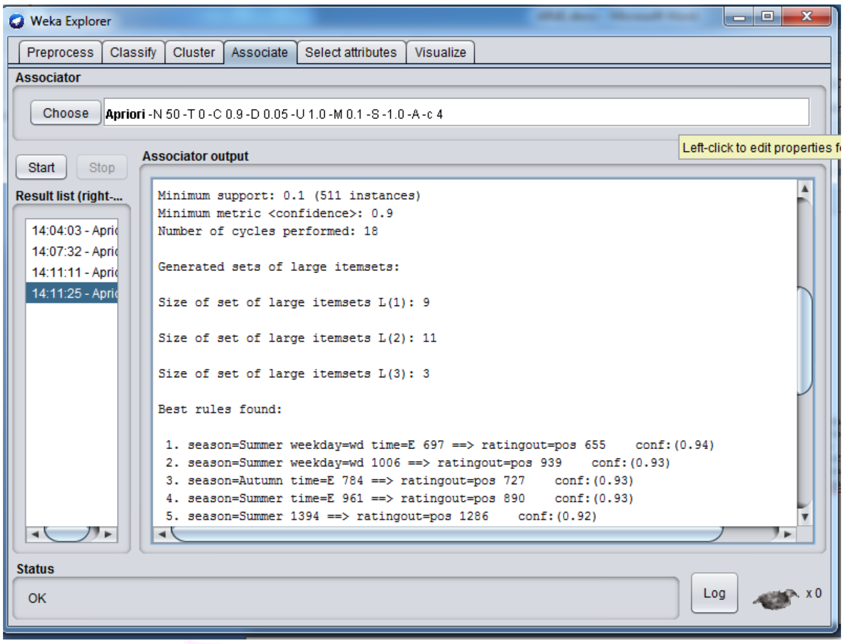

Fig.2 Sample rules generated for the dataset

The minimum support of 0.2 and confidence of 0.9 gives the optimal rules with number of rules as nine. These nine rules are taken as nine segments. The contextual segments are created for these nine rules. The context segments created with number of ratings is given in Table IV.

Table IV

Context Segments with Number of Ratings

Segment Number	Number of Ratings	MAE (KNN=20)
Seg1	1394	0.773
Seg2	1172	0.789
Seg3	2440	0.710
Seg4	3388	0.671
Seg5	3643	0.640
Seg6	1466	0.669
Seg7	1221	0.768
Seg8	1138	0.821
Seg9	1322	0.816

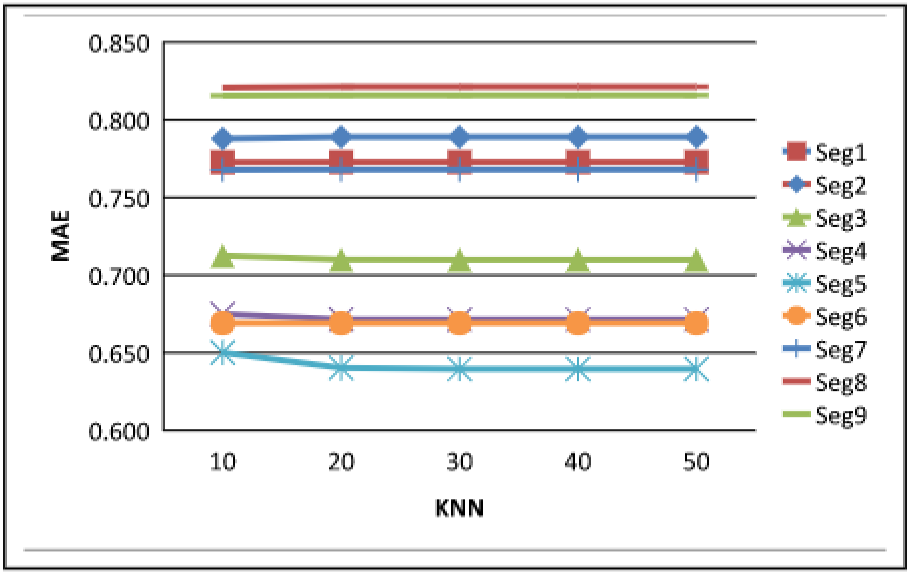

Fig.3 MAE for context segments with User KNN

The collaborative filtering is applied on each contextual segment generated to find the accuracy of the segment which will be used for context based recommendation. The accuracy measure used is Mean Absolute Error (MAE) [17] for rating prediction. The Librec API is used with user based k nearest neighbor (UserKNN) as the collaborative filtering algorithm [18], [19]. The UserKNN is the most general and baseline algorithm for recommendation. The dataset is divided in to 80:20 ratio for training and testing. The similarity measure used is cosine similarity. The results are given with number of neighbours varying from KNN=10 to 50 in Fig.3. The analysis shows that the accuracy is more with segments having more ratings. The MAE varies from 0.640 to 0.821. This shows that the segments will have influence if used in recommendation according to context.

VI. Conclusion

The context segments are created in context based recommendation system for incorporating context in recommendation. A pre filtering approach with association rule mining for creating context segments is proposed for context based recommendation system. The context segments are created with class association rule mining for reduction based method of pre filtering approach. The context segments reduce the data used for recommendation by filtering the data according to context. The accuracy of context segments for Hitrec 2011 dataset of movie recommendation shows that it can be used with collaborative filtering for context based recommendation system. The future work is to apply the rules with context based recommendation system with collaborative filtering.

VII. References

[1] **Ricci, Francesco, Lior Rokach, and Bracha Shapiro "Introduction to recommender systems" in Recommender Systems handbook, pp.1-35, Springer US, 2011.**

[2] **Adomavicius, Gediminas, and Alexander Tuzhilin, "Context-aware recommender systems," in Recommender Systems handbook, pp.217-253, Springer US, 2011.**

[3] **Adomavicius, Gediminas, and Alexander Tuzhilin, "Towards next generation of recommender systems: A survey of state of the art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol.17, no.6, pp. 34-739, June 2005.**

[4] **Adomavicius, Gediminas, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," ACM Transactions on Information Systems (TOIS), vol. 23, no.1, pp. 103-145, ACM, 2005.**

[5] **Baltrunas, Linas, and Francesco Ricci. "Experimental evaluation of context-dependent collaborative filtering using item splitting," User Modeling and User-Adapted Interaction, vol.24, no.1-2 pp. 7-34, 2014.**

[6] **Ramirez-Garcia, Xochilt, and Mario Garcia-Valdez. "A Pre-filtering Based Context-Aware Recommender System using Fuzzy Rules," Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization, pp. 497-505, Springer International Publishing, 2015.**

[7] **Datta, Suman, Joydeep Das, Prosenjit Gupta, and Subhashis Majumder. "SCARS: A scalable context-aware recommendation system," in 3rd International Conference on Computer, Communication, Control and Information Technology *(C3IT)*, pp. 1-6. IEEE, 2015.**

[8] **Abowd, G., Dey, A., Brown, P., Davies, N., Smith, M. and Steggles, P., "Towards a better understanding of context and context-awareness", In Handheld and ubiquitous computing (pp.304-307). Springer Berlin /Heidelberg 1999.**

[9] **Arati R Deshpande and Emmanuel M..."Conceptual Modeling of Context based Recommendation System", International Journal of Computer Applications 180(12):42-47, January 2018.**

[10] **Dourish, P.," What we talk about when we talk about context" Personal and ubiquitous computing, 8(1), pp.19-30, 2004.**

[11] **M. Emmanuel, S. M. Khatri, and D. R. R. Babu, "A Novel Scheme for Term Weighting in Text Categorization: Positive Impact Factor", *IEEE International Conference System Man and Cybernetics*, pp.2292–2297, Manchester, UK, 2013.**

[12] **R. Agrawal, R. Srikant: Fast Algorithms for Mining Association Rules in Large Databases. In 20th International Conference on Very Large Data Bases, 478-499, 1994.**

[13] **Bing Liu, Wynne Hsu, Yiming Ma: Integrating Classification and Association Rule Mining. In 4th International Conference on Knowledge Discovery and Data Mining, 80-86, 1998.**

[14] **Cantador, and Brusilovsky, Peter and Kuflik, 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011), Proceedings of the 5th ACM conference on Recommender systems, 2011.**

[15] **Eibe Frank, Mark A. Hall, and Ian H. Witten, the WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.**

[16] **Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, "The WEKA Data Mining Software: An Update", SIGKDD Explorations, Volume 11, Issue 1, 2009.**

[17] **Herlocker, Jonathan L., Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. "Evaluating collaborative filtering recommender systems," ACM Transactions on Information Systems (TOIS) vol.22, no.1, pp. 5-53, ACM, 2004.**

[18] **Guibing Guo, Jie Zhang, Zhu Sun and Neil Yorke-Smith, "LibRec: A Java Library for Recommender Systems", in Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization (UMAP), 2015.**

[19] **G. Guo, J. Zhang and N. Yorke-Smith, "TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings", in Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI), 123-129, 2015.**