Comparison of pre and post-filtering algorithms for Conditional Recommendation

GIANG NGUYEN TRUNG
Electrical and Information Engineering Department
Seoul National University of Science and Technology
Seoul, Nowon-gu, Gongneung-dong 232
KOREA
ntgttm@gmail.com http://icomlab.seoultech.ac.kr/

HEEJUNE AHN
Electrical and Information Engineering Department
Seoul National University of Science and Technology
Seoul, Nowon-gu, Gongneung-dong 232
KOREA
heejune@seoultech.ac.kr http://icomlab.seoultech.ac.kr/

Abstract: - The demand for recommendation systems gets higher and more popular these days. There have been several studies on recommendation systems, but the most recent ones give focus on the product as a whole and do not give much attention to the user's preferences such as price, type, and color. This paper proposes pre-filtering methods and compares the benefits and performance between pre- and post-filtering methods. The pre-filtering method ignores ratings of items that are not relevant to the user's preferences, then reduces the size of target data set to process, saving processing time. The experimental result with MovieLens dataset shows that pre-filtering can provide the recommendation with 8.5 times less computations than post-filtering by restricting item set, and shows 2% improvement in F measurement. Moreover, rating estimation performance can vary from 1% improvement in the ML-1M dataset to 1% decrease in the ML-100K dataset in the RMSE.

Key-Words: - Information Retrieval; Conditional Recommendation; Matrix Factorization; Hierarchical System; Recommendation System; Pre-filtering

1 Introduction

Recommendation systems are essential in the information and e-commerce ecosystem. Over the past 20 years, there have been several studies and movements focusing on these systems, aiming to discover more efficient solutions. The Netflix Prize [1] is a big event that motivated research on recommendation system, makes it more practical and popular, particularly those based on user ratings. Many recent studies focus on rating matrix and incorporate additional user-item information such as context [2], social relationships [3], reviews [4], and genre [5] to boost recommendation system, but these additional data are not always available and difficult to obtain. Moreover, incorporation approach could increase processing time which we don't really want to happen in a live application. In common situations, the user often chooses a product depending on its attributes. For example, in the movie recommendation, sometimes we like to watch an Action movie and sometimes Drama is taken priority.

This paper mainly focuses on providing conditional recommendations based on a dditional attributes on items. The conditional recommendation could be obtained through pre- or post-filtering and rating history data. The paper proposes the pre-filtering methods and compares the benefits and performance between pre- and post-filtering methods. These two-dimensional rating matrices are generated separately with item attribute. Our proposal can filter out irrelevant items that don't belong to user requirement, reducing the number of ratings to consider and matrix size. Then recommendation system can make predictions quickly at the option of the user. This paper also proposes two rating estimation methods by combining these rating matrices.

2 Related work and background

2.1 Collaborative Filtering

Recommendation methods are typically classified into two: content-based (CB) and collaborative filtering (CF). Content-based recommendation builds user and item profile, which is often difficult to build and cannot predict the items outside the user profiles [6]. CF analyzes relationships between users and interdependencies among products to estimate new user-item associations [7], thus providing better accuracy than CB. Matrix factorization (MF), a collaborative filtering model, is the most often used approach in recommender systems because it combines good scalability with predictive accuracy [8].

2.2 Bias Matrix Factorization

In a real life situation, some people tend to give higher ratings than others [8]. Some famous movies with known director and cast members can receive higher ratings. Using this kind of bias play an important role in reducing the effects caused by the differences among the users and the items.

The equation for the prediction model is as follows:

$$\tilde{r}_{ui} = \mu + b_u + b_i + p_u q_i^T \tag{1}$$

The overall average rating is denoted by μ; the parameters b_u and b_i indicate the observed deviations of user u and item i. The system needs to minimize new squared error function:

$$\begin{aligned} &\min L_{BL}(P,Q) \\ &\quad = \min_{P,Q} \{ (\left\| r - \mu - b_u - b_i - p_u q_i^T \right\|^2 \\ &\quad + \lambda \left(\|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2 \right) \} \end{aligned} \tag{2}$$

Using SGD, the gradient at current values is computed by differentiating above equation with 4 variables separately. The updated equation for p_{uk}, q_{ki}, b_u, b_i are:

$$\begin{aligned} \acute{p}_{uk} &= p_{uk} + \eta \left(e_{ij} q_{ki} - \lambda p_{uk} \right) \\ \acute{q}_{ki} &= q_{ki} + \eta \left(e_{ij} p_{uk} - \lambda q_{ki} \right) \\ \acute{b}_u &= b_u + \eta_u \left(e_{ij} - \lambda b_u \right) \\ \acute{b}_i &= b_i + \eta_i \left(e_{ij} - \lambda b_i \right) \end{aligned} \tag{3}$$

2.3 Post-Filtering Algorithms

As shown in Figure 1, the post-filtering method ignores item information and does matrix factorization for whole rating matrix in the first step. Then, candidate list is generated and ordered by rating of each user. In order to provide the recommendation list, the system will filter and retain the items matching the user requirement.

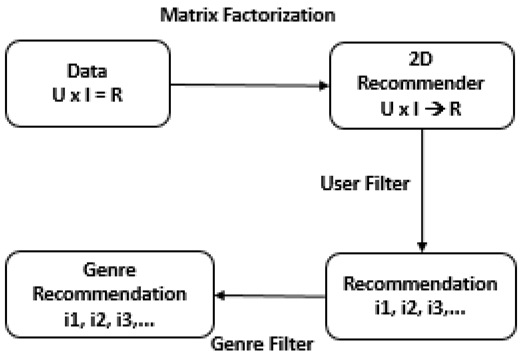

Fig.1 Post-filtering Recommender

3 Pre-filtering algorithms

In order to make a system capable of quickly supporting the user in choosing products with the attributes meeting their requirements, we decided to split the items using their attributes, and then estimate the user's assessment of the subcomponents. This approach is shown in figure 2. The system is suitable in case the user provides the necessary attributes they want such as the genre of the movie.

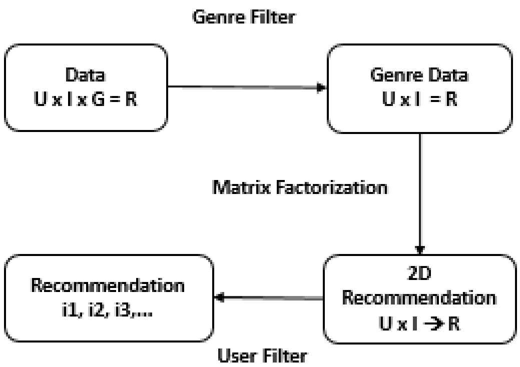

Fig.2 Recommender for each genre in pre-filtering

For unconditional recommendation result, the proposed method, named "Hierarchical Method", sums up the results from the subcomponents. The recommendation list will be processed in figure 3. The rating user u give to the item i is combined the rating user u has given to the item i in each component. Two intuitive and traditional methods are proposed and considered in this paper, more elaborate ones can be developed and compared in the future.

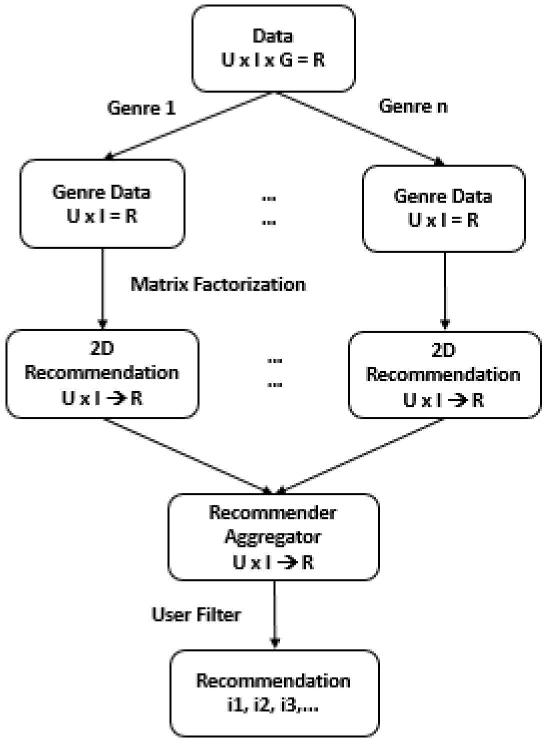

Fig 3. Combining Multiple Genre-recommender

3.1 Hierarchical Content Matrix Factorization (HMF)

In this approach, the final rate is retrieved from the combination of multiple genre-recommenders as follows:

$$r_{u,i} = \frac{1}{N_{G(i)}} \sum\nolimits_{g \in G(i)} r_{u,i,g} \qquad (4)$$

where G(i) is the list of genres item i belongs to, $N_{G(i)}$ is number of genres item i belongs to and $r_{u,i,g}$ is the rating of user u for item i in genre g's recommender.

3.2 Weighted Hierarchical Content Matrix Factorization (HMF)

Considering a particular user, we assume that the rating is more reliable with the genre recommender that the user likes. The combined rating is calculated as follows:

$$r_{u,i} = \frac{1}{\sum_{g \in G(i)} p_{u,g}} \sum\nolimits_{g \in G(i)} r_{u,i,g} * p_{u,g} \qquad (5)$$

where G(i) is the list of genres item i belongs to, $p_{u,g}$ is how much user u likes genre g and $r_{u,i,g}$ is the rating of user u for item i in genre g's recommender. The user-genre matrix, $p_{u,g}$ is calculated as follows:

$$p(u,g) = \sum_{r=1}^{5} \frac{n_{u,g,r} * r}{n_{u,g}} \qquad (6)$$

where $n_{u,g,r}$ denotes the number of items genre g which user u rated with rating r and $n_{u,g}$ denotes number of items belonging to genre g that the user rated.

4 Evaluation and experiments

4.1 Evaluation metrics

There are several metrics that are traditionally used to evaluate the performance of recommender systems such as mean absolute error (MAE), root mean squared error (RMSE), precision, recall and F-measure [9]. They are classified into statistical accuracy and decision-support accuracy metrics.

Statistical accuracy metrics (RMSE, MAE) compare the predicted ratings against the actual user ratings on the test data. RMSE is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{T}} \sum_{\mathrm{i,j}} \left(\mathrm{r_{ij}} - \tilde{\mathrm{r}}_{\mathrm{ij}}\right)^2} \qquad (7)$$

where r_{ij} denotes the rating user i gave to item j, and $\tilde{r}_{ij}$ denotes the rating user i gave to item j as predicted by a certain method, and T denotes the number of tested ratings.

F-measure is defined as a harmonic mean of the precision and recall.

$$\mathrm{F-measure} = \frac{2 \text{ x Precision x Recall}}{\text{Precision + Recall}} \qquad (8)$$

The decision-support accuracy metrics (precision, recall, F-measure) measure how well a recommender system can predict which of the unknown items will be highly rated. Here, we considered a movie is "good" if its rating is higher than 3 and "bad" otherwise.

In order to measure how well an algorithm can order the item list for each user by interest level, we use Spearman's rank correlation as the third metric. In order to calculate Spearman's rank correlation, we need to rank two variables first. Assume x_i and y_i are the ranking of item i in 2 item lists of a user, $\bar{x}$ and $\bar{y}$ is mean of all item ratings. The Spearman's correlation is defined as:

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \quad (9)$$

4.2 Experiment Dataset

We employ two real-life data sets of MovieLens 100K and MovieLens 1M. The MovieLens 100K dataset includes 943 users, 1,682 items, and 100,000 ratings and the 1M dataset includes 3,900 movies, 6,040 user and more than 1 million ratings. The basic information about movies and users are also available. For the movie, the datasets provide the title, release year, IMDB link and genres. There are 18 different genres such as "Action", "Drama" and "Comedy" with each movie belonging to one or more genres. User information includes age, genre, occupation and zip code.

Table 1. Raw Dataset

UserId	MovieId	Rating	Time
196	242	3	881250949
186	302	4	891717742

u.data.txt

UserId	Age	Gender	Occupation
196	24	M	Technician
186	30	F	Writer

u.user.txt
21 occupation

MovieId	Title	Release	IMDB URL
242	A	2000	http://
302	B	2004	http://

MovieId	Action	…	Western
242	1	.	0
302	0	.	1

u.item.txt
18 genres

The scale being used to give ratings to movies ranges from 1 (worst) to 5 (best). For a fair evaluation, we ran 5-fold cross validation. The dataset is randomly split into two subsets, the training set containing 80% of the dataset and the testing set with the remaining 20%.

4.3 Experiment

We implemented three recommendation methods, namely BiasMF, HMF, WHMF and checked whether our approach improved evaluation metrics or not. The performance of recommendation algorithms highly depends on the parameter selection. In order to be confident with the results, we used the same initialization procedures and parameter in all of the methods when applicable.

Table 2. Parameter Values

	MovieLens 100K	MovieLens 1M
No of factors (k)	10	40
Max iteration	100	100
Learning rate (λ)	0.07	0.07
Bias reg. (η)	0.1	0.1
User reg. (λ_u)	0.1	0.1
Item reg.(λ_i)	0.1	0.1

4.4 Discussion

We compared three recommendation methods under the same circumstances and parameters. Table 3 depicts RMSE, F-measure and Spearman correlation across each genre matrix factorization with MovieLens 1M. In some genres, matrix factorization achieves much better result than others. For example, "Action" and "Adventure" genre gain low result in RMSE, high in F-measure and Spearman correlation, much better than "Documentary" and "Fantasy". One reason for this difference is the number of items that belong to each genre. A genre contains more items (Action – 503, Adventure - 283, Documentary – 127, Fantasy - 68) have greater chances of getting better RMSE and F-measure than others. Comparing the two approaches, pre-filtering and post-filtering (BiasMF), we can see pre-filtering achieves better results in F-measure, but slightly worse in RMSE and Spearman correlation.

Table 3. RMSE, F-measure and Spearman correlation for each Genre Matrix (one fold) – ML 1M

	Pre-filtering		
	RMSE	F-measure	Spearman correlation
1.Action	0.8481*	0.5538*	0.4914
2.Adventure	0.8685	0.5333*	0.4735
3.Animation	0.8741	0.6166*	0.3953
4.Children's	0.8667	0.5271*	0.4383
5.Comedy	0.8793*	0.5252*	0.4405
6.Crime	0.8767	0.6228*	0.4393
7.Documentary	0.9187	0.6976*	0.1633
8.Drama	0.8618	0.6352*	0.3963
9.Fantasy	0.9323	0.4695*	0.4315
10.Film-Noir	0.8569	0.7861*	0.3142
11.Horror	0.9444	0.4221*	0.4624
12.Musical	0.9193	0.5821*	0.3513
13.Mystery	0.9194	0.5764	0.3979
14.Romance	0.8964	0.5373*	0.3803
15.Sci-Fi	0.8868*	0.5438*	0.4911
16.Thriller	0.8601	0.5765*	0.4597
17.War	0.8777	0.7269*	0.3985
18.Western	0.9023	0.5832	0.4207

(*) Pre-filtering is better than post-filtering

	Post-Filtering (BiasMF)		
	RMSE	RMSE	RMSE
1.Action	0.8565	0.8565	0.8565
2.Adventure	0.8614	0.8614	0.8614
3.Animation	0.8588	0.8588	0.8588
4.Children's	0.8656	0.8656	0.8656
5.Comedy	0.8799	0.8799	0.8799
6.Crime	0.8431	0.8431	0.8431
7.Documentary	0.8942	0.8942	0.8942
8.Drama	0.8553	0.8553	0.8553
9.Fantasy	0.8753	0.8753	0.8753
10.Film-Noir	0.8120	0.8120	0.8120
11.Horror	0.9289	0.9289	0.9289
12.Musical	0.8938	0.8938	0.8938
13.Mystery	0.8606	0.8606	0.8606
14.Romance	0.8665	0.8665	0.8665
15.Sci-Fi	0.8902	0.8902	0.8902
16.Thriller	0.8487	0.8487	0.8487
17.War	0.8343	0.8343	0.8343
18.Western	0.8446	0.8446	0.8446

Table 4. Number of items, ratings and processing time for each Genre Matrix and full matrix – ML 1M

	No of items	No of ratings	Processing time (ms)
1.Action	503	81,499	5,692
2.Adventure	283	48,523	2,729
3.Animation	105	19,604	842
4.Children's	251	43,203	1,580
5.Comedy	1,200	189,137	7,087
6.Crime	211	36,471	1,639
7.Documentary	127	21,791	270
8.Drama	1,603	267,505	8,320
9.Fantasy	68	10,576	815
10.Film-Noir	44	9,608	406
11.Horror	343	55,342	1,667
12.Musical	114	22,099	932
13.Mystery	106	16,016	1,005
14.Romance	471	85,473	2,975
15.Sci-Fi	276	48,351	3,304
16.Thriller	492	76,476	3,788
17.War	143	25,221	1,397
18.Western	68	11,081	514
Sum (18 genres)	--	--	44,962
Full matrix (Post-Filtering)	3,900	1,000,209	21,343

Our experiment is developed using a Java library, LibRec [10], and run on a computer with Intel Core i5 2.4GHz processors, Windows 7- 64 bit, 4 GB memory and HDD drive. Table 4 shows the number of items, number of ratings and processing time across each genre matrix factorization with MovieLens 1M. The processing time is influenced by the number of ratings in the dataset.

Experiment results show that single genre matrix provides the recommendation 8.5 times faster than post-filtering in average. We used BiasMF results as a baseline in order to compare the two algorithms. For a fair evaluation, we ran a 5-fold cross validation.

Table 5. Comparison of combination methods and BiasMF (average)

MovieLens 100K			
	BiasMF	HMF	WHMF
RMSE	0.9182	0.9286	0.92914
F-measure	0.52488	0.53114*	0.53566*
Spearman correlation	0.38816	0.37576	0.37406
MovieLens 1M			
	BiasMF	HMF	WHMF
RMSE	0.86608	0.85948*	0.8599*
F-measure	0.56422	0.57646*	0.57968*
Spearman correlation	0.46234	0.45586	0.45576

The average of 5 fold cross validation for the three algorithms is presented in table 5. The result shows that both HMF and WHMF algorithms give better F-measure result when compared to the post-filtering method (BiasMF). WHMF provides the largest F-measure in comparison to other methods. For RMSE comparison, the two proposed methods provided worse result than traditional MF in ML 100K dataset but better in ML 1M dataset. Although both types of measures are important, decision-support metrics, such as F-measure, are better suited for recommender systems than statistical accuracy metrics, RMSE since recommender systems mainly focus on recommending high-quality items [9]. The experiment proved that recommender system can recommend movie list with better quality than traditional post-filtering MF. BiasMF achieves better Spearman correlation than these unified methods, thus it can provide the best ordering the item list for each user by interest level. Definite interpretations of this experiment results needs future in-depth studies, but we consider there are some attributes that affect the overall rate of the item but not affects strongly in the conditional environments. For example, one likes an actor in Comedy movies whereas he/she does not like the actor in romantic movies.

5 Conclusion

In this paper, we have compared the performance of 'pre' and 'post'- filtering methods for item recommendation with addition user constraints. Our approach splits overall matrix into sub-matrices using item attributes, then estimates the user's rating for each sub-matrix. Pre-filtering can provide the recommendation 8.5 times less computation than post-filtering by restricting item set. For unconditional recommendation, we designed 2 methods HMF, WHMF to combine these sub results. The experimental results demonstrate that the proposed algorithms slightly outperform the state-of-the-art algorithms (BiasMF), improving 2% in the decision-support metric (F-measure) on MovieLens 100K and MovieLens 1M dataset. In RMSE comparison, our proposed algorithms are 1% better on ML-1M dataset and 1% worse on M L-100K dataset.

Item genre is used in our experiment because it is the key information accurately distinguish each movie. However, there may be some other significant attributes which can be used in the current model. It would be interesting to study a combination of these attributes in the hierarchy model. These different contents are used can generate divergent prediction results.

Acknowledgements

This study was supported by the Research Program funded by the Seoul National University of Science and Technology.

References:

[1] J. Bennett, S. Lanning, The netflix prize, in: *Proceedings of KDD cup and workshop*, 2007, p. 35.

[2] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, *Recommender systems handbook Springer* US (2015) 191-226.

[3] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, L. Getoor, HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems, in: *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, pp. 99-106.

[4] J. McAuley, J. Leskovec, Hidden factors and hidden topics: Understanding rating dimensions with review text, in: *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 165-172

[5] J. Nguyen, M. Zhu, Content-boosted matrix factorization techniques for recommender systems, *Statistical Analysis and Data Mining* 6 (4) (2013) 286-301.

[6] G. Adomavicius, 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* 17 (6) (2005) 734–749.

[7] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor , *Recommender Systems Handbook*, Springer, 2010.

[8] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, *Computer* 42 (8) (2009) 30-37.

[9] G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin, Incorporating contextual information in recommender systems using a multidimensional approach, *ACM Transactions on Information Systems* 23 (1) (2005) 103-145.

[10] G. Guo, J. Zhang, Z. Sun, N. Yorke-Smith, LibRec: A Java Library for Recommender Systems, in: *Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization*, 2015.