Slope One Meets Neighborhood: Revisiting Slope One Predictor in Collaborative Filtering

Rabi Shaw and Bidyut Kumar Patra

Department of Computer Science of Engineering
National Institute of Technology, Rourkela, India-769008
{518cs6017,patrabk}@nitrkl.ac.in

Abstract. : Collaborative filtering (CF) framework in recommendation is a very popular technique for providing personalized recommendation. Slope one predictor is a model based CF which has received good attention from researchers and practitioners. In this paper, we revisit the slope one predictor to incorporate strong features of neighbourhood based CF into it for providing personalized recommendation to users. Preliminary results with two real world datasets are very promising. Proposed technique outperforms original slope one and its performance is at par with a variant of slope one introduced recently.

Keywords: Collaborative framework(CF), slope one revisited, personalize recommendation

1 INTRODUCTION

Very popular and widely used framework in recommender system is collaborative filtering (CF) deployed in many industries [12, 10, 11, 2, 1, 14]. These CF techniques are broadly classified into two categories namely, neighbourhood based CF and model based CF. The neighbourhood based CF method exploits association among the neighbours of an active user (item). However, it is unable to capture global information (total structure) of the data. On the other hand, model based CF is capable in capturing overall structure of the data [7]. It creates a model out of the rating data-set using machine learning or other techniques. Lemire et al [8] introduced a model based approach called slope one which works on principle of differential popularity between pair of items. This determines how much one item is likely to be compared to other item in the pair. A deviation matrix for all pairs of items is computed and these matrix entries are used for predict rating of an unknown item. This algorithm is very intuitive, simple and accuracy is high compare to with many complex model based approach. Therefore, it has been gaining attention across the research community since its introduction [8]. It has already received more than 700 citations. Menezes et al. [9] proposed to improve the performance of weighted slope one (WSO) (a variant of slope one algorithm) by introducing personalized weighting scheme for a user. The slope one predictor preserves the total structure of the data. However, it does not give importance to localized information of an active user unlike popular neighbourhood based CF. The deviation between a pair of items remains unchanged across the users. We argue that deviation between a pair of items (likeness of one item over other) can not be the same for all users in a system.

Slope One predictor [8] does not consider the personalized deviation value for an active user. So we propose Weighted Slope One algorithm, where our main contribution is to calculate the deviation value between a pair of an item for providing personalized recommendation using neighborhood concept.

The rest of the paper is organized as follows: In section 2, we discussed about the related work, In section 3, we discussed about the background of work, in section 4 our proposed methodology, Dataset and metric evaluation methods are discussed. In section 5, preliminary result and discussion is done. In section 6 we concluded our paper with future work.

2 RELATED WORK

In recent development collaborative recommendation is the most well know approach in recommendation system. In traditional collaborative recommendation algorithms predictions are performed using user similarity [3, 6]. However, the scalability problem with user based collaborative algorithms exists, when the rapid increase of number of users and items. Item similarity based collaborative approach is proposed by Sarwar et al. [11] in contrast of the traditional algorithms. In Sarwar et al. [11] approach, they propose an algorithm to recommend an item to a user based on similar items rated by the same user. It overcome the scalability problem to some extend and generate good recommendations compared to traditional algorithms.

Gao et al. [5] claims in their work about importance of user's recommendation than others. They also claims that user must be given some weightage with item-based collaborative filtering , including Slope One recommendations. They achieved this by computing relative weights depend on ratings for each users. Different literature exists for other variants of the Slope One algorithm. An algorithm propose by Wang et al. [13] depends on Slope One and user-based collaborative filtering to improve performance of recommending items. In their approach they handled the missing ratings using Slope One. User based collaborative filtering is applied after filling missing ratings to produce better recommendations . On contrary to Wang et al. [13] Zhang [15] uses item-based collaborative algorithm for recommendation keeping Slope One algorithm in the same way for missing rating calculation.

3 BACKGROUND

Lemire et al. [8] introduce a concept to predict an item rating using differential popularity. Consider A_1, A_2, I_1 and I_2 are two users and two items, respectively. A_1 gave rating 1.0 to I_1 and rating 1.5 to I_2 , whereas A_2 gave rating 2 to I_1. Differential popularity is applied to calculate the difference between the ratings given for two items I_1 and I_2 by user A_1 as shown in figure 1, difference is 0.5 (i.e. 1.5(I_1) - 1.0(I_2) of user A_1). This value is used to predict rating for I_2 of user A_2. From the figure it is concluded that 2.5 is the rating for I_2 of user A_2 to preserve the same difference.

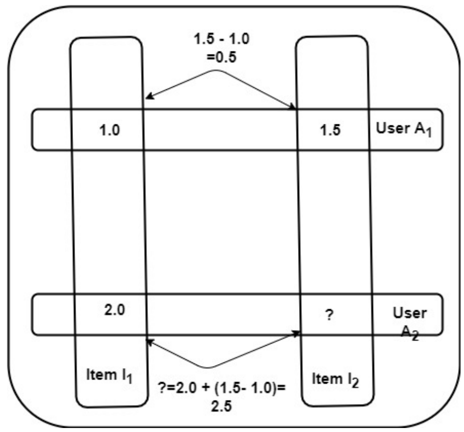

Fig. 1. Differential popularity to predict item rating

In general, assume the group of all users denoted by A and S be the group of all items in the system. Compute differential popularity matrix (deviation matrix) as follows :

1. Consider for items $I_i, I_j \in$ S user, $A^{'} \in A$ has rated R_{A',I_i} and R_{A',I_j} ratings. This shows A$^{'}\in S_{I_i,I_j}$, where S_{I_i,I_j} denotes group of all users who rated I_i and I_j both items.
2. Calculate deviation matrix *dev* for all items of S :

$$dev_{I_i,I_j} = \frac{\sum_{\acute{A}\in S_{I_i,I_j}} (R_{\acute{A}I_i} - R_{\acute{A}I_j})}{\mid S_{I_i,I_j} \mid} \tag{1}$$

3. Using calculated matrix in step 2 and the group of ratings for all users, for item i of user $A^{'}$, $P_{\acute{A},i_i}$ is predicted as:

$$P_{\acute{A},I_i} = \frac{\sum_{I_{i,j}\in R_{\acute{A}}} (dev_{I_i,I_j} + R_{\acute{A}I_j}) \times \mid S_{I_i,I_j} \mid}{\sum_{I_{i,j}\in R_{\acute{A}}} \mid S_{I_i,I_j} \mid} \tag{2}$$

4 Proposed Methodology

In these section, we describe revise Slope One, dataset and metric for evaluation.

Table 1. Datasets propertics

DataSet	# User U	# Item I	Ratings	Density (in %)
MovieLens 1M	6040	3706	1000209	4.1%
Netflix	8141	9318	196656	0.25%

4.1 Revisiting Slope One Prediction For Personalizing

In this paper, we revisit the slope one predictor [8] and propose to combine the neighbourhood concept (retaining local structure) with slope one for providing personalized recommendation. We propose to modify the way the deviation between a pair of item I_1 and I_2 is computed in the slope one paper [8]. To provide personalized recommendation, we associate the active user's neighbourhood information while computing deviation between a pair of items.

Let I_1 be the target item for an active user A and R_A be the set of rated items by the user A. We compute deviation between item I_1 and a rated item $I_2 \in R_A$ as follows.

$$dev(I_i, I_j) = \frac{\sum_{\acute{A} \in A_{I_i,I_j}} (R_{\acute{A},I_i} - R_{\acute{A},I_j}) \times exp^{sim(A,\acute{A})}}{\sum_{\acute{A} \in A_{I_i,I_j}} exp^{sim(A,\acute{A})} \times \mid A_{i,j} \mid} \tag{3}$$

Let A_{I_1,I_2} be the set of users rated both the items I_1 and I_2. We compute Pearson correlation coefficient (PCC) [4] ($sim(A, \acute{A})$) between active user A and $\acute{A} \in A_{I_1 I_2}$. We give more weightage to the ratings obtained from the users whose similarity value is greater than 0 while less weights for other users $\in A_{I_1,I_2}$ using Equation 3

It can be noted here that dev_{I_i,I_j} varies across the users and this is personalized deviation value.

4.2 Dataset

This section explains the dataset used, evaluation metrics and experimental result. In this paper, MovieLens 1M and Netflix datasets are used to evaluate our approach. The dataset description is provided in Tabe 1.

4.3 Metric For Evaluation

Mean Absolute Error (MAE) is calculated using equation 4, which is average absolute error over the all redictions and smaller value indicates a better accuracy. In equation 4, Max denotes the quantity of rating instances within the check set. R_i and $\bar{R}_i$ are the actual rating value and predicted rating of an active user on an item.

$$MAE = \frac{\sum_{i=1}^{MAX} |R_i - \tilde{R}_i|}{MAX} \tag{4}$$

Root Mean Square Error(RMSE) is found using equation 5, this is another metric to evaluate the accuracy and compare between predicted and actual rating. Where, Occurrences of ratings in the test set is denoted by Max, R_i is the original rating value and $\tilde{R_i}$ is the predicted rating. RMSE is root of squared sum of error value.

$$RMSE = \sqrt{\frac{(\sum_{i=1}^{MAX} |R_i - \tilde{R_i}|)^2}{MAX}} \tag{5}$$

Precision is calculated using equation 6 which is defined as the fraction of the number of relevant items by the total number of recommended items.

$$Precision = \frac{|L_R \cap L_{rev}|}{L_R} \tag{6}$$

Where $|L_R \cap L_{rev}|$ denotes the common items which are recommended and relevant both. L_R denoted the Number of recommended items. Whereas, the ratio between the number of recommended relevant items to the total number of relevant items in the system is called as Recall (equation 7). But, with respect to recommender system, recall value is prejudiced and highly depends on the total number of relevant items rated by each user.

$$Recall = \frac{|L_R \cap L_{rev}|}{L_{rev}} \tag{7}$$

F-measure is defined (equation 8) as the ratio of the precision with multiply by recall and addition of recall and precision

$$F_{measure} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{8}$$

If Number of Relevant item and recommend item in test set be equal then all three value (precision, recall, F-measure) will always be same. Generally it does not happen because of large dataset.

5 Experimental Results and Comparison

We discuss the experimental setup, results and comparison with the existing model Weighted Slop One(WSO)[8] and Linear Weighted Slope One Function (LIUSO) proposed by Danilo Menezes et al [9]. The Datasets are divided into parts 80% in training set and 20% in test set respectively. Linear Weighted Slope One Function(LIUSO) which was calculated using equation 9 , where MaxMAE value is 5, because 5 is the maximum rating value and MAE_{A,I_i} is computed using equation 10. This equation used only in training dataset not use in test dataset. We implemented the proposed approach and tested on MovieLens and Netflix dataset.

$$ItemUsefulness_{A,I_i} = MaxMAE - MAE_{A,I_i} \tag{9}$$

$$MAE_{A,I} = \frac{\sum_{I \in \acute{R}_A}(|(R_{A,I} - dev_{\acute{I},I}) - R_{A,\acute{I}}|)}{| R_A |} \tag{10}$$

In the experiments for each user the relevant thresholds are considered in two ways. In the first way, the item ratings whose values are greater than the user mean rating are considered as relevant because some user are biased to rate the item with low ratings and aggregate item prediction is likely to be relevant item for this kind of user considered in Table 3 and 5. In the second approach, the rating instances whose actual rating value is greater than 4 are considered as relevant items shown in Table 2 and 4. Generally, in recommender system rating value 4 or above 4 considered as a good rating in scale 1-5. The results shows that the proposed approach outperforms for slope one predictor in terms of precision, recall and F1-measure.

From Table 2, we can observe that MAE and RMSE of the proposed approach are 0.6984 and 0.8888 respectively, which are lesser than WSO (0.7081 and 0.9008), LIUSO (0.7045 and 0.9016). F1-measure of the Proposed approach is better then WSO and LIUSO by 3.78% and 1.23% respectively. For the threshold value is equal to user mean rating, F1-measure is better then 2.13%, LIUSO 0.52% shown in table 3.

Table 2. Results on Netflix Dataset,threshold_rating=4

Metric	WSO	LIUSO	Proposed Approach
MAE	0.7081	0.7045	0.6984
RMSE	0.9008	0.9016	0.8888
Precision	85.43%	84.29%	85.77%
Recall	36.46%	39.14%.	40.15%
F1-measure	51.11%	53.46%	54.69%

Table 3. Results on Netflix Dataset,threshold_rating=user mean rating

Metric	WSO	LIUSO	Proposed Approach
MAE	0.7081	0.7045	0.6984
RMSE	0.9008	0.9016	0.8888
Precision	66.86%	66.40%	67.69%
Recall	61.41%	64.87%.	64.68%
F1-measure	64.02%	65.63%	66.15%

From table 4 we can see, the MAE and RMSE values on MovieLens dataset with different threshold values are 0.6990 and 0.8840 respectively, which are lesser than WSO (0.7033 and 0.8892) and LIUSO (0.7003 and 0.8898). The proposed approach outperforms in terms of F1-measure over WSO and LIUSO by 3.22% and 0.35% as shown in table 4, and 1.92% and 0.85% as shown in table 5.

Table 4. Results on ML Dataset,threshold_rating=4

Metric	WSO	LIUSO	Proposed Approach
MAE	0.7033	0.7003	0.6990
RMSE	0.8892	0.8898	0.8840
Precision	86.49%	85.89%	86.81%
Recall	41.29%	43.65%.	43.81%
F1-measure	55.89%	57.88%	58.23%

Table 5. Results on ML Dataset,threshold_rating=user mean rating

Metric	WSO	LIUSO	Proposed Approach
MAE	0.7033	0.7003	0.6990
RMSE	0.8892	0.8898	0.8840
Precision	69.92%	69.55%	70.33%
Recall	67.92%	70.42%.	71.35%
F1-measure	68.91%	69.98%	70.83%

6 CONCLUSION AND FUTUREWORK

The traditional slope one predictor approach is modified to accommodate user neighbourhood features while computing the deviation matrix to provide the personalized recommendations. Proposed approach shows better result for threshold values - user mean rating, 4 respectively than LIUSO and WSO. The promising results encourage us to extend this work by incorporating effective neighbourhood computation strategies in the future.

References

1. Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.
2. Fidel Cacheda, Víctor Carneiro, Diego Fernández, and Vreixo Formoso. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web (TWEB)*, 5(1):2, 2011.

3. Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007.
4. Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, 4(2):81–173, 2011.
5. Min Gao, Zhongfu Wu, and Feng Jiang. Userrank for item-based collaborative filtering recommendation. *Information Processing Letters*, 111(9):440–446, 2011.
6. Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
7. Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.
8. Daniel Lemire and Anna Maclachlan. Slope one predictors for online rating-based collaborative filtering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 471–475. SIAM, 2005.
9. Danilo Menezes, Anisio Lacerda, Leila Silva, Adriano Veloso, and Nivio Ziviani. Weighted slope one predictors revisited. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 967–972. ACM, 2013.
10. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.
11. Badrul Munir Sarwar, George Karypis, Joseph A Konstan, John Riedl, et al. Item-based collaborative filtering recommendation algorithms. *Www*, 1:285–295, 2001.
12. Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 267–274. ACM, 2008.
13. Pu Wang and HongWu Ye. A personalized recommendation algorithm combining slope one scheme and user based collaborative filtering. In *2009 International Conference on Industrial and Information Systems*, pages 152–154. IEEE, 2009.
14. Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 153–162. ACM, 2016.
15. DeJia Zhang. An item-based collaborative filtering recommendation algorithm using slope one scheme smoothing. In *2009 Second International Symposium on Electronic Commerce and Security*, volume 2, pages 215–217. IEEE, 2009.