Learning to Recommend with Negative Ratings Based on Factorization Machine

Caihong Sun, Xizi Zhang

***Abstract*—**Rating prediction is an important problem for recommender systems. The task is to predict the rating for an item that a user would give. Most of the existing algorithms for the task ignore the effect of negative ratings rated by users on items, but the negative ratings have a significant impact on users' purchasing decisions in practice. In this paper, we present a rating prediction algorithm based on factorization machines that consider the effect of negative ratings inspired by Loss Aversion theory. The aim of this paper is to develop a concave and a convex negative disgust function to evaluate the negative ratings respectively. Experiments are conducted on MovieLens dataset. The experimental results demonstrate the effectiveness of the proposed methods by comparing with other four the state-of-the-art approaches. The negative ratings showed much importance in the accuracy of ratings predictions.

***Keywords*—**Factorization machines, feature engineering, negative ratings, recommendation systems.

I. Introduction

IN parallel with the web, recommender systems (RS) have developed rapidly and have drawn a high degree of attention [1] for solving information redundancy problems. Rating prediction is to predict the rating a user would assign to an item that the user has not rated, which is a widely studied problem in the domain of rating-based RS.

One of the most widely used frameworks for the rating prediction problem is the collaborative filtering (CF) approach [12], which is also a commonly used recommendation technique for RS [2]. Since CF algorithms use the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users [3], the sparsity problems are generated as a major problem limiting the usefulness of CF which refers to a situation where transactional and feedback data is sparse and insufficient to identify similarities in consumer interests [4]. Factorization machines (FM) attract a lot of attention both in research and in industrial areas for its high-prediction accuracy with the notion of matrix factorization involved [5], which has been shown to be a useful decomposition for multivariate data to alleviate the sparsity problem in CF [6]. Another advantage of FM is that it combines the high prediction accuracy of factorization models with the flexibility of feature engineering [5].

The features in data are important to the recommendation predictions and will greatly influence the results. Meanwhile, the quality and quantity of the features have great influence on whether the model is good or not [11]. Researchers are committed to exploring latent features to model the behaviours of users and items to improve the prediction performance. For example, a novel preference model distinguishing different rating patterns as user preference is developed [7]. Ratings are considered to have different recommended weights in different circumstances such as based on timeline [8]. The domain knowledge of trust and distrust are used for better rating prediction performance [9], [10].

Many features have already been studied and introduced in RS for better performance in many aspects. To our best knowledge, the effect of negative ratings is rarely considered into rating prediction until now. Here, negative ratings are those ratings with low points. For instance, if a user rates an item as 1 point, or 2 points, we consider that the user gives this item negative feedback. In practice, in an online shopping scenario, people pay more attention to those negative reviews than positive ones. Since from the aspect of behavioural psychology, most people have instinct to avoid loss, an item with more negative ratings could have tendency to get more negative feedback. Thus, we argue that negative ratings are more important than those positive ratings in rating prediction. Equally weighting negative and positive ratings can be biased. In this paper, we would create negative effect features based on the ratio of negative ratings and introduce these features into FM models to improve the performance of rating prediction.

The main contributions of our paper are drawn as following:

- We model the negative ratings as item preference features with concave and convex functions inspired by Loss Aversion theory. And then introduce negative effect features into an FM model to improve the performance of rating prediction.
- Via many experiments, we find that the negative effects do have more influence on rating prediction, providing a new perspective for investigating negative effects as features in recommendation.

The rest of this paper is organized as follows. Section II models the negative ratings and Section III describes the FM model and expounds how the features are introduced to the FM model. Section IV presents the data sets, experiment design and analysis of the experimental results. Section V draws the conclusions and discusses the limitation and future work.

II. Measuring Negative Effect

In RS, the ratings which users give to items play an important role in item recommendation. The rating usually is a five-point

Caihong Sun is with the Renming University of China (corresponding author, e-mail: chsun@ruc.edu.cn).

Xizi Zhang is with the Renming University of China (e-mail: zxzding@ruc.edu.cn).

Financial support from the Science and Technology Plan Project of Beijing (No.Z171100000117009) is acknowledged.

integer scale to express the degree of favourability to each item (normally, 1, 2, 3, 4 and 5 represent “hate”, “do not like”, “neutral”, “like”, and “love”, respectively). While 5-points means that the user presents a very positive attitude and 1-point means a negative attitude, here we regard 1- or 2-point ratings as negative ratings. In rating prediction, most researchers consider the ratings of items equally, while in the real-world different kinds of rating scores can have a distinguishing effect on the impression of items. The drawbacks of using absolute ratings have been identified [12]. In practice, many people will pay more attention on those negative ratings and reviews when buying some items. Based on the theory of Loss Aversion in behaviouristic psychology [13], people are inclined to avoid loss and the effect of the loss is greater that the loss itself. For instance, when you try to purchase an item, and find there are 50% negative ratings, then you will consider the 50% negative reviews more important than the other 50% positives and the 50% negative ratings itself. Negative ratings play a big role in e-commerce; they can affect people’s buying decision and their assessment of an item. Sometimes, the losses and disadvantages have a greater impact on preference than gains and advantages. In this paper, we assume that negative ratings of an item contribute more than the positive ones, and weigh negative effect more on the rating prediction algorithms. In this section, we propose two functions to measure the effect of negative ratings.

The absolute number of negative ratings has some bias for measuring the perception of negative degree. For example, there are two items both having five negative ratings, but they have 1000 ratings and 10 ratings respectively in total. Of course the second item has more negative assessment. Hence, we use the ratio of negative ratings to measure the negative degree instead. Let *np* represent the ratio of negative ratings. For example, np_1 denotes the ratio of 1 point ratings while np_2 denotes the ratio of point 2. As we assumed, *np* has made a significant impact on a user’s impression on an item. If the overall *np* of the item is pretty low, people may think it worth buying. As *np* increases, so too does the probability that people feel an item is worth buying. Aside from the negative rating ratio, we propose two more simple functions inspired by the value function of Loss Aversion Theory to measure the perceived negative effect that people often value negative things more with the increase in negative rating ratio.

A. A Concave Growth Negative Disgust Effect Measurement

We define a concept of Concave Growth Negative Disgust in which case people value the growth of a negative rating ratio at the beginning the most. That is, with the increase in *np*, the sensitivity of users’ purchase disgust for an item is less. Thus, a concave function is built to measure the perceived negative effect where the disgust of negative ratings grows a little faster than the linear function but a little lower than the convex one. The concave and convex functions measure the extent that affect people’s perceived negative effect.

Given the *np* of an item i, the negative effect of *np* is weighted as:

$$concave_{np}(i)=np(i)^{1/2}+np(i) \tag{1}$$

B. A Convex Growth Negative Disgust Effect Measurement

The definition of Convex Growth Negative Disgust is explained where people pay more attention to the larger negative rating ratio compared with Concave Growth Negative Disgust. As *np* rises to the top, the degree of users’ purchase disgust for an item also reaches a peak. The Convex Growth Negative Disgust effect function is defined as:

$$convex_{np}(i)=np(i)^2+np(i) \tag{2}$$

Equations (1) and (2) ensure the weighted negative effect is greater than *np* itself to fit the Loss Aversion theory.

C. Negative Controversiality Level

Inspired by the idea of Controversial User proposed [14], we put forward the concept of Negative Controversial Item. A Negative Controversial Item means that in some extent, the item has more negative ratings that are worth us to pay more attention, especially on the negative aspect. We classify items into 10 different levels based on the ratio of negative ratings. Then, we will test how the negative effects work in rating prediction on items with different negative controversial levels. We define the Negative Controversiality Level (*ncl*) of an item is related to the percentage of the negative ratings which disagree with the positive ones in rating an item by all users. The Negative Controversiality Level is set as:

$$ncl(i)=floor(np(i)\times 10) \tag{3}$$

When *np* has the value ranging from 0 to 1, the *ncl* could be transformed from 1 to 10 utilizing (3).

III. A FM Model with Negative Effect

FMis a generic approach that allows mimicking most factorization models by feature engineering [5]. Since the negative effect is an important feature in rating prediction, we integrate the negative effect features into an FM algorithm to improve the accuracy of rating prediction.

A. Factorization Machines

FM has a very good ability to process the data of them user-item rating matrix, the user features, and item features. With its power of processing data, dealing with the sparsity problem and fast speed, FM is well performed in rating prediction problem. FM is able to perform a regression model whose task is to estimate a function $y:R^n \rightarrow T$ from a real valued vector $x \in \mathbb{R}^n$ to a target domain T. In supervised settings, it is assumed that there is a training dataset $D=\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),\ldots\}$ of examples for the target function y given. What is more, FM could model all single and pairwise interactions between the input variables by using factorized interaction parameters. The FM model can be defined as follows:

$$\hat{y}(x):=\omega_0+\sum_{i=1}^{n}\omega_i x_i+\sum_{i=1}^{n}\sum_{j=j+1}^{n}<V_i,V_j>x_i x_j \tag{4}$$

where the model parameters that have to be estimated are:

$$\omega_0 \in R, w \in R^p, V \in R^{p\times k} \tag{5}$$

and <·,·> is the dot product of two vectors of size k:

$$<V_i, V_j> := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \tag{6}$$

A row V_i within V describes the i-th variable with k factors. $k \in N_o^+$ is a hyperparameter that defines the dimensionality of the factorization. ω_0 is the global bias. ω_i models the strength of the i-th variable. Here, the interaction between the i-th and j-th variable is modelled with a factorized parameter $\widehat{\omega}_{i,j} := <V_i, V_j>$ which is referred to the matrix factorization. Moreover, it is the reason that allows FM to estimate reliable parameters even in highly sparse data where standard models fail.

B. A FM Model with Negative Effect

Table I presents an example data and we use it to illustrate how the FM model deals with data. There are two users and three items in the example data. Each user has a categorical feature age ["1-9", "10-29"] and favourite colour ["red", "blue", "green"]. Each item has a numerical feature price. The rating records list as in Table I.

TABLE I
RATING RECORDS OF TWO USERS AND THREE ITEMS

User	Item	Age	Colour	Price	Rating
user1	item2	9	red&blue	100	3
user1	item3	9	red&blue	200	4
user2	item1	23	green	1000	5
user2	item3	23	green	200	1

Then features for an FM model could be transformed as follows:

TABLE II
REAL VALUED FEATURE VECTORS X FOR A FM MODEL

	Feature vector x												Target y
	User		item			age		colour			price	ratings	
$x^{(1)}$	1	0	0	1	0	0	1	1	0	1	100	3	$y^{(1)}$
$x^{(2)}$	1	0	0	0	1	0	1	1	0	1	200	4	$y^{(2)}$
$x^{(3)}$	0	1	1	0	0	1	1	0	1	0	1000	5	$y^{(3)}$
$x^{(4)}$	0	1	0	0	1	1	1	0	1	0	200	1	$y^{(4)}$

As shown in Table II, every row represents a feature vector $x^{(i)}$ with its corresponding target, $y^{(i)}$. We use n columns to indicate n users, and m columns to indicate m items. For example, $x^{(1)}$ means that user 1 rates item 2 with 3 points, while user 1 is aged 1-9 years, and his favourite colours are red and blue; the price of item 2 is 100.

We introduce negative effect as item features and add these features into the FM model. Here, we measure negative rating effects as item features in four different sets: N, CN, CN+P and 2CN+3P. And each time we introduce one feature set into the FM model to test its effectiveness.

- **N**: By considering both the 1-point and 2-point ratings as negative effect, we add the $np_1(i)$ and $np_2(i)$ together as N. And the N will be introduced as a numerical feature of the item. This is the simple linear negative effect measurement.
- **CN**: Input N as the ratio of negative rating, and then we compute the Concave Negative Disgust and the Convex Negative Disgust negative effect features based on (3) and (4). CN consists of two numerical features.
- **CN+P**: Besides the negative effect feature sets CN, we add an additional positive effect feature P. We compute the positive rating ratio P in the same way for the computation of N. P is the ratio of positive ratings (i.e. point 3, 4, and 5). CN+P consists of two numerical features in two different ways based on the process of negative effect: the concave growth disgust negative effect and P, and the convex growth disgust negative effect and P. Note, the values of these two features are normalized by making their summation into 1.
- **2CN+3P**: We measure negative effect with five features. The negative effect feature sets include np_1, np_2, pp_1, pp_2, and pp_3. Where pp_1, pp_2, and pp_3 are the ratio of positive ratings with 3-points, 4-points and 5-points, respectively. Note: there are two different ways to process np_1 , np_2 with concave and convex growth disgust functions. Note that the values of these five features are normalized by making their summation into 1.

IV. EXPERIMENTAL DESIGN AND ANALYSIS

In this section, we conduct several experiments to validate the effect of negative ratings based on an FM model. Next, four state-of-the-art algorithms are used to compare with the proposed method.

A. Data Description

We use MovieLens 1M dataset to test the performance of our proposed method. The dataset contains four features: users, items (movies here), personal ratings that a user gives to the item and timestamps. It includes 1,000,209 ratings evaluated by 6,040 users on 3,706 items of the online movie recommender service MovieLens. The statistics of the dataset are summarized below:

TABLE III
STATISTICS OF THE DATASETS

Numbers of Ratings	1,000,209
Users	6,040
Items	3,706
Sparsity	95.53%
Minimum Number of Ratings by any User	20
Maximum Number of Ratings by any User	2,314
Average Number of Ratings by any User	166
Minimum Number of Ratings for any Item	1
Maximum Number of Ratings for any Item	3,428
Average Number of Ratings for any Item	270

In addition, all items are divided into 10 groups by item's Negative Controversiality Level (*ncl*). The results are shown in Table IV. The groups are used to test the effectiveness of negative features in different negative rating scenarios.

TABLE IV
STATISTICS OF THE NCL

ncl	Number of Items	Number of Ratings
1	1,010	448,621
2	896	275,025
3	563	127,238
4	381	64,869
5	294	37,353
6	226	23,558
7	132	13,787
8	86	7,308
9	35	2,128
10	83	322

B. Evaluation Metrics

We choose the widely used Root Mean Square Error (RMSE) [15] as the evaluation metric in our experiments. The RMSE is defined as:

$$RMSE=\sqrt{\frac{\sum_{i,j}(r_{i,j}-\hat{r}_{i,j})^2}{N}} \tag{7}$$

where the $r_{i,j}$ denotes the rating of item i_j rated by user u_i , $\hat{r}_{i,j}$ elaborates the corresponding predicted rating, and N represents the number of ratings.

C. Four Baseline Algorithms

To illustrate the effectiveness of our proposed method, we choose the following four state-of-the-art recommendation baseline algorithms for comparisons.

- **MF** (Matrix Factorization): Factorizing the observed rating values using a factor matrix for users and one for items [6].
- **Biased MF** (Biased Matrix Factorization): Matrix factorization with explicit user and item bias [16].
- **SVD++**: A recommender algorithm which successfully addresses different aspects of the data by exploiting both explicit and implicit feedback from users as user preference [17].
- **ItemKNN**: Weighted item-based neighbourhood algorithm [18].

The MF and Biased MF methods are relevant to FMs as latent models. The SVD++ algorithm considers the user preference, while our negative ratings model reveals the preference information of items. In addition, the ItemKNN approach is based on item feature and our algorithm is focusing on the items. Thus, we choose these four state-of-art algorithms for comparisons with the proposed method.

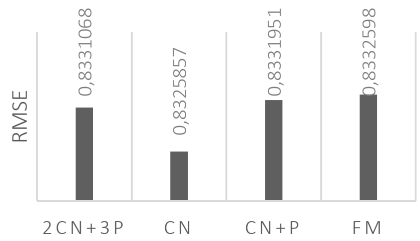
D. Experimental Design and Analysis

Our experiments are designed to answer the following questions:

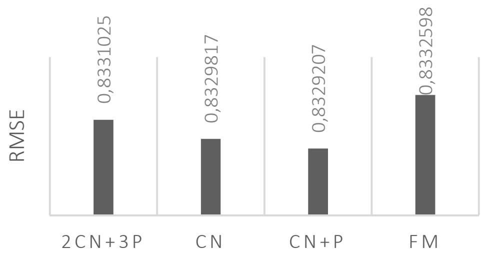
1) Does our FM model with negative effects outperform the traditional FM model?
2) Does the Loss Aversion Theory work in rating prediction?
3) Does our proposed method outperform the four state-of-the-art recommendation baseline algorithms we chose?
4) How do the negative ratings affect the rating prediction accuracies with different negative rating ratio?

1. Does our FM Model with Negative Effects Outperform the Traditional FM Model?

We compare the performance of FM model with and without negative effects. Ten-fold cross validation is performed in our experiments [19], where 90% of the rating data is randomly selected as the training set, while the remaining 10% is as the test set. We conduct the experiments at iterations of 1,000 in which case all errors of the predictions can achieve the convergence, and finally, the average error after the 10-fold cross validation will be used to estimate prediction accuracy.



(a) ConCave negative disgust

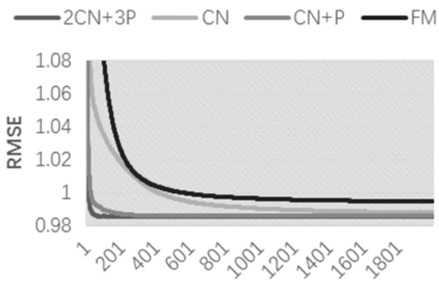


(b) Convex Negative Disgust

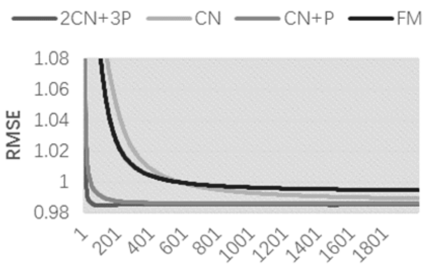
Fig. 1 Compared with FM in Concave Negative Disgust and Convex Negative Disgust

The prediction accuracies of both Concave Negative Disgust and Convex Negative Disgust evaluated by RMSE are shown in Fig. 1. Although the data is very sparse, all methods are doing a good job on prediction quality due to the matrix FM nested. We can observe that whether in Figs. 1 (a) or (b), all of our approached, 2CN+3P, CN and CN+P, outperforms the FM algorithm alone. This indicates that the introduction of the negative effect features works in rating prediction. The negative effect feature set CN has shown a relatively stable good performance in comparison with 2CN+3P and CN+P. Furthermore, we conducted another experiment in which the original data is ordered by user IDs. It is divided into 10 parts by order and is used to do a 10-fold cross validation. Such dataset partitioning could have the probability of the emergence of cold users who are new to the system and the test dataset becomes larger. A cold-start problem, which gives recommendations to novel users who have no preference on any items or recommends items that no user of the community has seen yet [20], is another main block for CF, except the sparsity problem that FM can ease, we set our approaches in cold-start troubles to test its robustness.

To complete the iteration, 2,000 steps of RMSE updated are plotted in Fig. 2.



(a) Concave Negative Disgust



(b) Convex Negative Disgust

Fig. 2 Compared with FM under the cold-start predicament

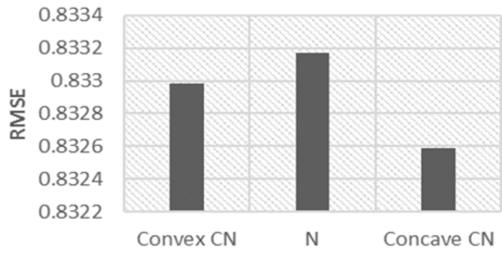
As Fig. 2 illustrates, each RMSE of our approaches is lower than FM itself. What is more, the gap between the proposed model and the FM in the cold-start dataset is bigger than in the randomly selected dataset, which proves the robustness of our methods. Even in the case of an unfavourable situation, the proposed methods can also maintain good performance with the effective information of negative ratings.

Additionally, we observe an interesting phenomenon: CN+P and 2CN+3P all perform better than CN, whether in Figs. 2 (a) or (b) with cold users. This observation is reasonable since in the absence of information, the appropriate and correct information supplement will play a good role. Another obvious phenomenon is the Concave Negative Disgust behaves better than the Convex Negative Disgust with new users probably because new users are more cautious about the goods, in which case even slightly negative comments will be cause for concern.

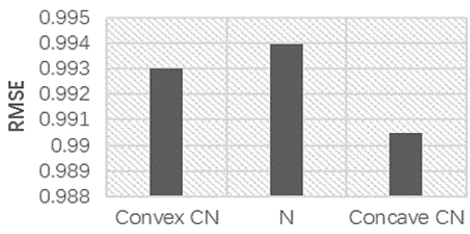
2. The Loss Aversion Effect of Negative Ratings

Moreover, in order to check the Loss Aversion effect of the negative ratings, an experiment was designed comparing the features including CNs (i.e. CN, CN+P and 2CN+3P) with N. Having confirmed the effectiveness of the features, including CNs based on CN, we just abandon the extra information here to give an essential comparison between CN and N.

Obviously, both of the CN in Concave Negative Disgust and Convex Negative Disgust contribute a lower RMSE than the unweighted N from Fig. 3, which interprets the Loss Aversion effect of negative ratings.



(a) Experiment on randomly selected data



(b) Experiment on cold users data

Fig. 3 Comparison between CN and N

3. Does the Proposed Method Outperform the Four State-of-the-art Recommendation Baseline Algorithms Selected for Comparison?

In order to show the effectiveness of the proposed recommendation approaches, a comparison was conducted with the baseline methods mentioned above. In the following experiments, we set the dimension of the feature vector in latent feature models to the 16 and the number of neighbours of the ItemKNN algorithm to 80. The iteration step is set to 1,000 and the negative feature set CN is used as the representative method.

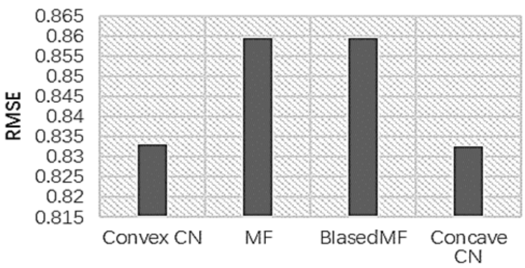
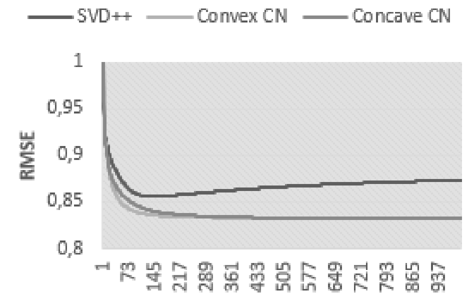


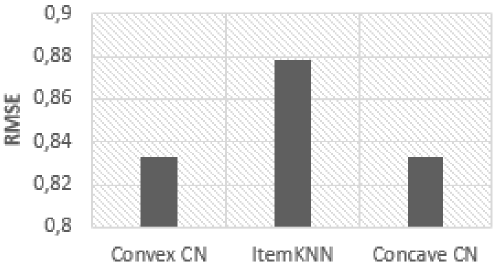
Fig. 4 Compared with MF and Biased MF algorithms

As shown in Fig. 4, the CN in Concave Negative Disgust and Convex Negative Disgust features outperform the MF and Biased MF algorithms in prediction accuracies. It demonstrates the advantage of in the proposed approaches in matrix factorization based algorithms.

In Fig. 5 (a), the x-axis is the simulation iterations. We find that after a few steps of iterations, the SVD++ model begins to overfit, while our Concave CN and Convex CN do not suffer the overfitting problem. The RMSE of SVD++ is higher than ours. SVD++ is time consuming, and FMs can be calculated in linear time. Meanwhile, we can see from Fig. 5 (b) that our proposed models have done a better performance in comparison with the ItemKNN algorithm in RMSE. The experimental results show that the negative effect features play an important role in item preference and rating prediction.

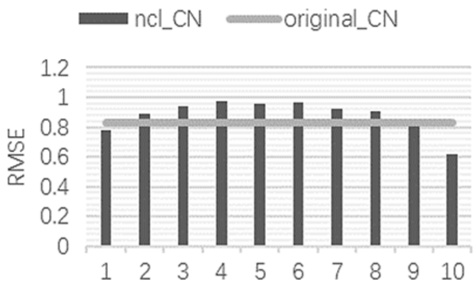


(a) Comparison with SVD++

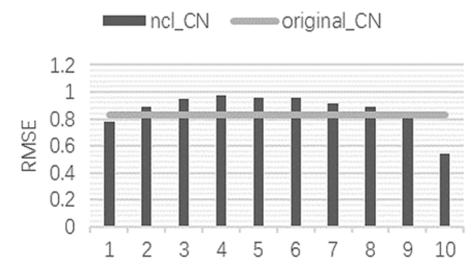


(b) Comparison with ItemKNN

Fig. 5 Compared with SVD++ and ItemKNN algorithms



(a) Concave negative disgust



(b) Convex negative disgust

Fig. 6 CN in different *ncl* datasets compared with original average CN

4. How Do the Negative Ratings Affect the Rating Prediction Accuracies with Different Negative Rating Ratio?

In this experiment, we divide the MovieLens dataset into ten datasets according to the Negative Controversiality Level (*ncl*) of items (i.e. 1 to 10) and implement our algorithms on the created 10 datasets, respectively. Moreover, each dataset consists of the training set that is randomly extracted with the 90% of the rating data and the remaining 10% as the test set. We compare the prediction accuracy of 10 groups with the average CN value on randomly selected 10-fold cross validation on all data. The iteration steps for the experiment are set to1,000.

From Fig. 6, we can see that the items with extremely low and high negative ratios perform better than that of all dataset, but the other eight groups do not. The interesting observation is the RMSE of CN in Convex Negative Disgust is a bit lower than in Concave Negative Disgust, while *ncl* is equal to 10, because the negative ratings are weighted more by Convex function on this dataset. It gives us some hints that the distribution of negative ratings matters in rating prediction, and thus, requires further investigation.

V. Conclusions

In this paper, we propose negative rating features for rating prediction by using an FM model. We build the negative effect measurement by the ratio of negative ratings based on Loss Aversion Theory. Our experiments show the improvement of rating prediction accuracy by introducing the negative effect features. In a comparison with other four state-of-art recommendation algorithms, the proposed method was shown to outperform them. Considering negative rating into rating prediction is reasonable and possible both in theory and practices. The experimental results also demonstrate that the proposed method works well on cold-start problems. These research findings can offer some hints in feature engineering on the negative effects on recommendation.

We build negative effect measurements only with two simple functions in this paper. The negative effect feature engineering deserves further studies. Moreover, we find that our approach works very well on those items having extremely low or high negative ratings, for other proportion negative rating groups, further study is need. Here, we only consider the negative rating from the aspect of items, from the user aspect, different users have different preferences towards the ratings, and taking the user's preference of negative rating into consideration, could have the chance to improve the prediction performance. Future works will also include testing the proposed method on more datasets.

References

[1] Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. Knowledge-Based Systems, 46, 109-2.

[2] Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: a survey. Decision Support Systems, 74, 12-32.

[3] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009, 4.

[4] Huang, Z., Chen, H., &Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Transactions on Information Systems (TOIS), 22(1), 116-142.

[5] Rendle, S. (2010, December). Factorization machines. In 2010 IEEE International Conference on Data Mining (pp. 995-1000). IEEE.

[6] Lee, D. D., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In Advances in neural information processing systems (pp. 556-562).

[7] Lee, J., Lee, D., Lee, Y. C., Hwang, W. S., & Kim, S. W. (2016). Improving the accuracy of top-N recommendation using a preference

model. Information Sciences, 348, 290-304.
[8] Haihong, E., Li, Y., Zhao, X., Song, M., & Song, J. (2016, January). A General Rating Recommended Weight-Aware Model for Recommendation System. In International.
[9] Ma, H., Lyu, M. R., & King, I. (2009, October). Learning to recommend with trust and distrust relationships. In Proceedings of the third ACM conference on Recommender systems (pp. 189-196). ACM.
[10] Forsati, R., Barjasteh, I., Masrour, F., Esfahanian, A. H., & Radha, H. (2015, September). Pushtrust: An efficient recommendation algorithm by leveraging trust and distrust relations. In Proceedings of the 9th ACM Conference on Recommender Systems (pp. 51-58). ACM.
[11] "Feature Engineering: How to transform variables and create new ones?" Retrieved 2016-01-10. https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/
[12] Sankar, D. M. and Sarkar, S. (2012) Rating Prediction using Preference Relations Based on Matrix FactMod Workshop in UMAP July 2012.
[13] Tversky, A., &Kahneman, D. (1991). Loss aversion in riskless choice: A reference-dependent model. The quarterly journal of economics, 1039-1061.
[14] Massa, P., & Avesani, P. (2007). Trust metrics on controversial users: Balancing between tyranny of the majority. International Journal on Semantic Web and Information Systems (IJSWIS), 3(1), 39-64.
[15] Steck, H. (2013, October). Evaluation of recommendations: rating-prediction and ranking. In Proceedings of the 7th ACM Conference on Recommender Systems (pp. 213-220). ACM.
[16] Rendle, S., & Schmidt-Thieme, L. (2008, October). Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In Proceedings of the 2008 ACM conference on Recommender systems (pp. 251-258). ACM.
[17] Koren, Y. (2008, August). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 426-434). ACM.
[18] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.
[19] Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145).
[20] Lam, X. N., Vu, T., Le, T. D., & Duong, A. D. (2008, January). Addressing cold-start problem in recommendation systems. In Proceedings of the 2nd international conference on Ubiquitous information management and communication (pp. 208-211). ACM.