Dynamic Selection Debiasing Recommendations Based On Sentiment Analysis

Fan Zhang , Wenjie Luo * , Xiudan Yang

Posted Date: 8 January 2025

doi: 10.20944/preprints202501.0607.v1

Keywords: dynamic debiasing recommendations; selected bias;sentiment analysis; IPS

Article

Dynamic Selection Debiasing Recommendations Based On Sentiment Analysis

Fan Zhang [1], Wenjie Luo [1,*] and Xiudan Yang [2]

[1] School of Cyber Security and Computer, Hebei University, Baoding 071000, China
[2] School of Management, Hebei University, Baoding 071000, China
* Correspondence: lwj12111@hbu.edu.cn

Abstract: Debiased recommendation has received significant attention from the academic community in recent years. Currently, our research on selection bias indicates that the existing literature ignores the effect of sentiment factors on selection bias. In the recommendation task, the existence of sentiment bias due to sentiment factors may lead to the recommendation of low-quality products to important users and unfair recommendation to niche(products designed for specific target markets and purposes) items. Eliminating sentiment bias and improving the quality of recommendations to important users can help balance research on selection bias. Sentiment bias is hidden in user ratings and reviews. Consequently, an effective approach to mitigate this bias is to analyze user ratings and comments in order to ascertain their genuine opinions. We have developed a sentiment analysis module with the objective of eliminating the bias between reviews and ratings, providing accurate sentiment ratings, mining users' real sentiment opinions, and eliminating the sentiment bias. Currently, a combinatorial function is constructed that performs the corresponding computations based on three different scenarios. Furthermore, the concept of dynamic debiasing is introduced, where the modeling time is dynamic. A selection debiasing recommendation method based on time-dynamic scenarios is also proposed. This paper demonstrates how the three aforementioned approaches can effectively mitigate selection bias by addressing data sparsity (sentiment analysis), optimizing data at the level of the data set (combinatorial functions), and considering dynamic scenarios and applying inverse propensity weighting methods (time-dynamic modeling). Our experiments on several real datasets demonstrate that our model can effectively improve the performance of recommendations.

Keywords: dynamic debiasing recommendations; selected bias;sentiment analysis; IPS

1. Introduction

Recommender systems are of vital importance in addressing the issue of information overload. Traditional models are susceptible to exposure or selection bias, which can result in suboptimal outcomes. This is because users are typically only able to evaluate a limited number of items and lack comprehensive knowledge of their preferences. In the context of recommender systems, selection bias can be defined as the phenomenon in which users select items that do not accurately reflect their true interests or preferences. It is essential to recognize that user recommendations may not always be impartial and may contain subjective evaluations. It is important to note that users may exhibit bias towards certain items for a variety of reasons. Consequently, the recommendations generated by the recommender system may be influenced by selection bias, resulting in recommendations that do not accurately reflect the user's actual preferences [1–4].

The presence of selection bias in recommender systems can give rise to a number of problematic outcomes. The presence of selection bias can have a detrimental impact on the performance of recommender systems. This can result in a reduction in accuracy and an overspecialisation of recommendations [5] , as they may reflect users' selection preferences rather than their actual interests. In order to solve the above problems, a reliable method has been used in the past, i.e., conduct of online randomised experiments. However, this may have a detrimental impact on the user experience

 doi:10.20944/preprints202501.0607.v1

due to the extended time required to occupy the business traffic. In contrast to online randomised experiments, several offline debiasing methods have been developed in recent years, including the popular inverse propensity scoring (IPS) [6] strategy. While existing IPS-based debiasing methods improve recommendations over methods that ignore the effects of bias, two significant limitations were identified. The application of IPS-based debiasing to recommendations assumes that (1) the effect of selection bias is static over time and (2) user preferences remain constant as items are used. When selection bias and user preferences are dynamic, that is to say, when they change over time, the current IPS-based approach is unable to remove recommendation bias [7].

Concurrently, the concepts pertaining to sentiment bias as elucidated by the most recent research can also be employed to offset the consequences of selection bias [8,9]. At present, the utilisation of sentiment analysis to remove sentiment bias and alleviate data sparsity represents a promising approach that is unlikely to give rise to modelling difficulties due to differences in data structures. Some researchers argue that affective bias can be removed by representing users' opinions on different items as ratings [10,11]. Others attempt to rate comments as positive or negative through sentiment analysis methods to help recommend [12,13]. It is desirable that user ratings and reviews of the goods consumed at the same time be obtained, as this will enable more accurate recommendations to be made based on comprehensive and authentic opinions.

Therefore, to address the issues of selection bias and sentiment bias,we propose to design a debiased recommendation framework (sentiment classification and temporal dynamic debiased recommendation module ,SCTD) based on sentiment analysis and temporal dynamic debiasing. Specifically, SCTD consists of a sentiment classification scoring module (sentiment classification scoring module,SCS) and a temporal dynamic debiased recommendation module (temporal dynamic debiased recommendation module,TDR).

The main contributions of this paper are as follows:

(1) We propose a recommendation framework based on affective bias and temporal dynamic debiasing: the SCTD, which exploits the affective scores of reviews and ratings to obtain the user's true opinion for recommendation.
(2) Capturing dynamic user interests and better capturing real-world user behaviours by alleviating data imbalance and sparsity issues through two modules, SCS and TDR.
(3) We conducted extensive experiments on two real datasets (yelp and Amazon), and according to the experimental results, our proposed SCTD recommendation model makes good progress and outperforms traditional debiased models.

2. Related Work

Selection debiased recommendation has become a hot research topic. In this section, we review some of the recent literature on selection biased recommendation methods. This includes how dynamic modelling can mitigate the effects of selection bias, and how mitigating data sparsity for studying sentiment bias can in turn help mitigate selection bias.

2.1. Dynamic Debiased Recommender System

To address dynamic scene debiasing, three common methods are typically employed [1]: Time Window,Time Decay Function,and Integration-Based Learning.The use of time windows and time decay function for the analysis of recent operations may result in the loss of pertinent information and the introduction of noise. Integration-Based Learning employs a multitude of models, each of which considers a mere fraction of the total behavioural spectrum. To address these issues, one paper proposes a definition of dynamic debiasing [1]. It is essential to recognise that data is subject to change over time and therefore requires constant updating with the latest modelling techniques in order to reflect its current nature. The article above is predicated on the assumption that selection bias is static and that only dynamic user preference scenarios are modelled, which results in the debiasing approach exhibiting suboptimal performance. In response to the aforementioned issue, some researchers have

 doi:10.20944/preprints202501.0607.v1

sought to enhance factor-based collaborative filtering algorithms by incorporating temporal features [14]. They have proposed that the temporal model is more effective than the static model. A challenge persists, as existing collaborative filtering algorithms are static models, wherein relationships are assumed to be fixed at different times. However, it is often the case that relational data evolves over time, exhibiting strong temporal patterns. A paper presented at WSDM '22 [7] made the theoretical assumption and proved that existing debiasing methods are no longer unbiased when both selection bias and user preferences are dynamic.

Thus, the current problems that need to be solved in this research direction can be split into: problem of selection bias, and debiased in dynamic scenes. Currently how to solve the problem of selection bias mainstream are data filling (Data imputation) and reverse tendency weighting (Propensity score). The mainstream method of data imputation suffers from empirical errors due to improper designation of the missing data model or inaccurate estimation of the rating value.The performance of IPS-based models depends on the accuracy of propensities.Therefore, previous studies have used heuristic optimisation schemes such as directly populating missing values. Such methods are subject to empirical errors due to improperly specified missing data model or inaccurate score estimation.Currently, the use of propensity scores in training recommendation models is a more common method to mitigate selection bias. The problem faced by propensity scores is that they directly target IPS-based unbiased estimators and optimise specific Iosses, but formulating accurate propensity scores is very rigorous and the performance of IPS-based models depends on the accuracy of the propensity. Propensity score based methods have large variance and can lead to non-optimal results.

2.2. Sentiment Bias

Sentiment bias refers to the bias that exists directly between comments and evaluations [15]. Basically, rating-based models attempt to measure the relationships between different users and items with the user–item rating matrix. When the dataset is huge, sparse and unbalanced, the rating-based models perform limited by cold-start issues and data sparsity in most cases [16].In review-based models, researchers utilize different methods (text analysis [17], sentiment classification [18]) to infer users' preferences through their reviews and make recommendations.These papers summarize some of the latest methods and theories that are often used in review-based models [13,19]. Although both review-based and rating-based models can achieve satisfying results in some scenarios, they are not designed for utilizing both reviews and ratings on unbalanced data in real-world [12,13].The researchers failed to consider the potential of sentiment analysis to mitigate selection bias. By removing bias between reviews and ratings and mining users' true sentiment opinions, sentiment analysis could help to replace data stuffing methods in selection bias data preprocessing.

3. The Framework for SCTD

In this section,we focus on a framework for SCTD based on sentiment analysis and causality.SCTD consists of two modules:SCS and TDR.SCS is a convolutional neural network for sentiment analysis. The SCS output from user U is a sentiment vector whose entries are the scores of each item in U 's review. At the same time, the columns of the user-item rating matrix are selected to form the rating vector V_r . Using the combinatorial function, an augmented rating matrix R_c is obtained, which we then provide to the TDR for accurate recommendation.The frame diagram for SCTD is shown in Figure 1.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 January 2025 doi:10.20944/preprints202501.0607.v1

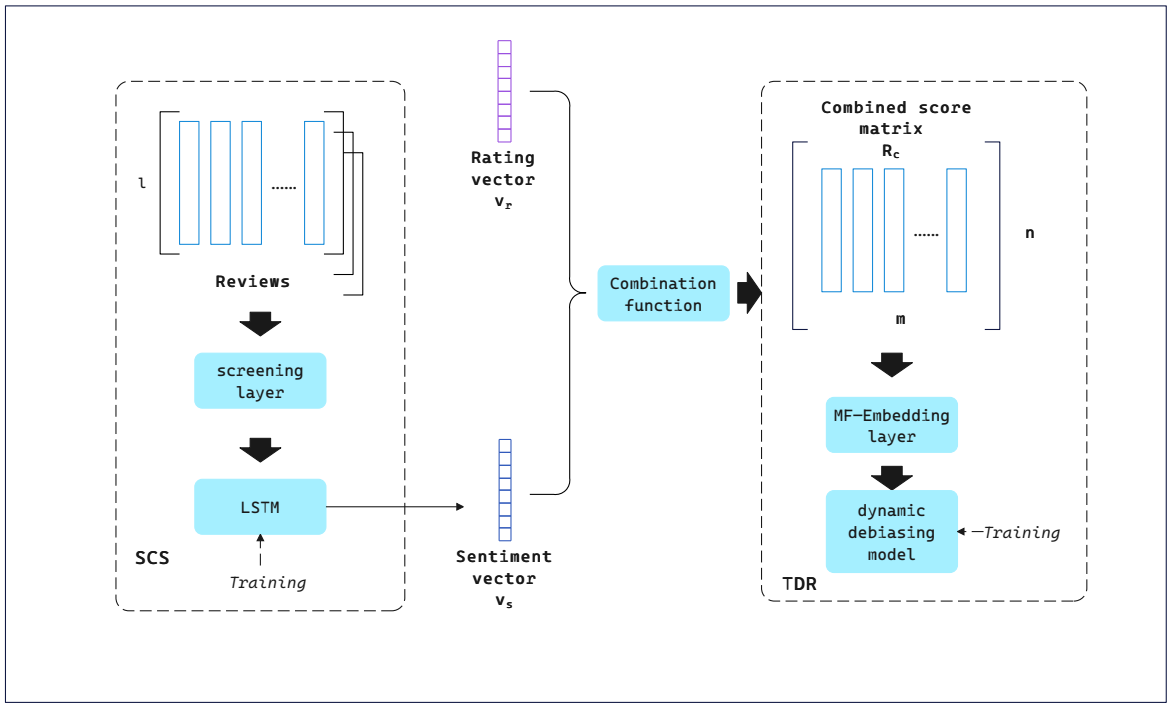

Figure 1. The Framework for SCTD.

3.1. SCS

Unlike previous dichotomous sentiment analysis tasks, we have optimised the output of the sentiment score ratings to fit the scores, ranging from 0 to 5 to fit the corresponding score ratings.In our proposed recommender system, let U, I be the set of users and the set of items, respectively, $|U| = m$, $|I| = n$. The user-item matrix $R \in R_{n\times m}$ consists of m users, and n items. r_{ui} represents the evaluation of item i by user u, and V represents the entire review text.The input to SCS is V and the output is the set of vectors V_s .We propose that there are four hidden layers in the SCS model.The first layer is the embedding layer, which aims to convert user comments V into a dense set of feature vectors. We arrange all the comments of user $u, v_u \in V$ into a matrix $R_s \in R^{l\times u_s}$. For each R_s, the embedding layer can be represented as:

$$f^1 : V \to R_s \to R^{l\times q_s\times k} \tag{1}$$

Where q_s is equal to the number of items rated by user u,l represents the length of the review, and k represents the latent dimension of the embedded word.Turning each word in a review into a K-dimensional vector, the latent description of user u's reviews is as follows:

$$LD_u = d_1 \odot d_1 \odot d_2 \odot ... \odot d_{q_s} \tag{2}$$

Where d_i is an I-dimensional vector and the term of the inner word is a K-dimensional potential vector.d_i represents u's review of i. $\odot$ represents the concatenation operator that combines these vectors into a $q_s \times l \times k$ description matrix D_u.And then we input D_u into the second layer, the convolution layer. We use attention theory to adjust different weights for each word of the review and each review of the review description matrix. Where g represents the K-dimensional embedding of each word.

W_g and $W - d$ represent the attention vector for each word in the review and the attention vector for each review in the review matrix. Next, we input the output vector obtained above into the convolution layer to extract contextual features between words. It can be expressed as:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 January 2025 doi:10.20944/preprints202501.0607.v1

$$f^2 : C_u = f^2(W^2(W^d(W^g d)) + b^2) \tag{3}$$

Where, W^2 and b^2 are the weights of the filter function in the convolutional neural network. f^2 is a nonlinear activation function. The third layer is the maximum pooling layer, which extracts the most important information from many contextual features. The max-pooling function is defined as follows:

$$f^3 : C_u^m = max(C_{u1}, C_{u2}, ..., C_{upl}) \tag{4}$$

The above vectors will be resolved to the final layer: the fully connected layer:

$$f^4 : v_s = f^4(W^4 C_u^m + b^4) \tag{5}$$

Where W^4 and b^4 are the weights in the fully connected neural network. f^4 can be a nonlinear activation function or a linear activation function.

Because we want to train the network with ratings rather than binary sentiments, we use softmax as the output function. The SCS process can be described as follows:

$$v_s = SCS(V|\alpha) \tag{6}$$

Where α stands for $(W^g, W^d, W^2, W^4, b^2, b^4)$.

We use user ratings to train this convolutional neural network. Unlike traditional methods, we did not convert the scores in the training data (0-5) into six ranges of 0-1 for bi-affective analysis. Instead, we design the output of SCS into six classes, which can be treated as a regularization of the output layer. Our model naturally matches a user's rating (from 0 to 5) with an sentiment score. Finally, we calculate the sentiment score vector v_s, whose entry is sentiment score s_{ui}, with a range of (0-5).The specific flow of SCS is shown in Figure 2.

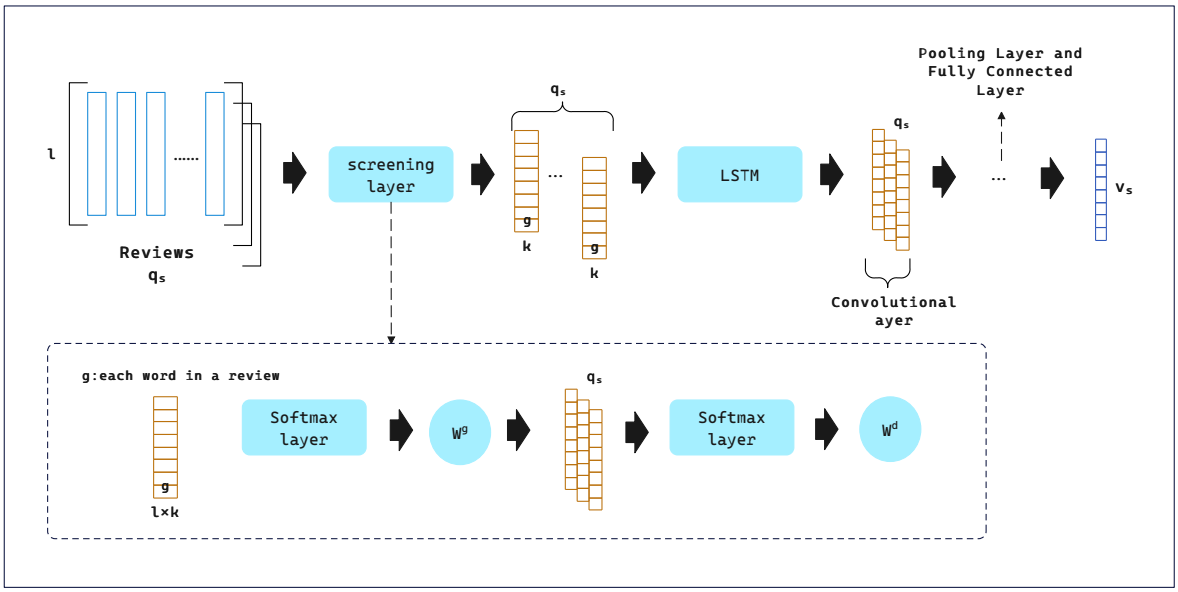

Figure 2. SCS Module.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 January 2025 doi:10.20944/preprints202501.0607.v1

3.2. Combination Function

In this section, we introduce the combination function which captures the opinion bias of reviews on ratings and augments the original rating matrix.The specific flow of the combined function is shown in Figure 3.

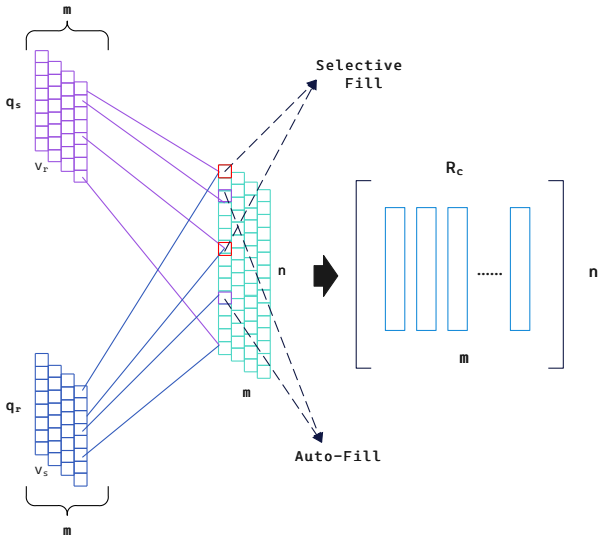

Figure 3. Combinational Function Module.

In the combination function, we make use of the sentiment vector v_s for each user in the sentiment rating module. we also extract each column from the user-item rating matrix $R \in \mathbb{R}^{n \times m}$ as a rating vector v_r. We then use a simple but efficient function to build a joint vector v_e with v_r and v_s:

$$v_e = v_s \oplus v_r \tag{7}$$

where $\oplus$ is a combinatorial function of SCTD. v_e is an n-dimensional vector, v_s is a q_s-dimensional vector with entry s_{ui}, and v_r is a q_r-dimensional vector with entry r_{ui}, where q_r is equal to the number of the item evaluated by the user. In practice, $q_r \geq q_s$. We assume that $q_r = q_s$.Whether we use review-based recommender system or rating-based recommender system alone, some important information of the other party is inaccessible, so it is difficult for us to solve the problem of data imbalance. Meanwhile, some users may make some useless reviews for some reasons (malicious reviews, hurried reviews, etc.). If a user rates an item and writes a review, we can compute the sentiment rating s_{ui} from SCS and get the rating r_{ui} from r. The difference between s_{ui} and r_{ui} is exactly the opinion bias we want to detect, Ob. If $Ob_{ui} = |s_{ui} - r_{ui}|$ is too large, we treat the review as a bad review and delete it. If the user's Ob is too large and too frequent, we treat it as a bad user and discard it. Basically, we want to enhance expressiveness through reviews and ratings while filtering out useless reviews. Thus the combinatorial function $\oplus$ can be generalised to the following three cases.

- Auto-Fill : If user u only rates or reviews on item i, $\oplus$ will populate it with r_{ui} or s_{ui}.
- Selective Fill : If user u rates and reviews on item i and does not discard it, then $\oplus$ will fill v_e with a fixed weighted linear function $v_{ei} = \varepsilon s_{ui} + (1 - \varepsilon) r_{ui}, \varepsilon \in (0, 1)$.

 doi:10.20944/preprints202501.0607.v1

- Drop : If $Ob_{ui} \geq$, $\oplus$ reduces the user's sentiment score s_{ui} for item i; if $(drop_u/q_s) \geq \iota$, $\oplus$ drops user u as a bad user. μ, ι are predefined thresholds and $drop_u$ is the number of deleted reviews.

According to the processing corresponding to the above situation, the enhanced rating matrix R_c is constructed, which is expressed by the formula, where β represents ε, μ, ι:

$$R_c = CF(v_s, v_r|\beta) \tag{8}$$

3.3. TDR

Traditional debiased recommendation methods are based on static scenarios.

$$\mathcal{L}_{\text{Naive}} = \frac{1}{|\{u,i,t : o_{u,i,t} = 1\}|} \sum_{u,i,t:o_{u,i,t}=1} L(\hat{y}_{u,i,t}, y_{u,i,t}) \tag{9}$$

However,traditional recommendation methods with static bias are not unbiased for the current treatment of selection bias in dynamic scenarios.The current widely used debiasing method uses IPS estimation to correct the probability that a user will rate an item. It uses static propensity p_u , which is the probability that user u observes a particular rating in any time period.

But this approach this ignores the dynamics of selection bias, i.e., these probabilities will vary t in each time period, causing the static IPS estimator to become:

$$\mathcal{L}_{\text{staticIPS}} = \frac{1}{|\mathcal{U}| \cdot |\mathcal{I}| \cdot |\mathcal{T}|} \sum_{u,i,t:o_{u,i,t}=1} \frac{L(\hat{y}_{u,i,t}, y_{u,i,t})}{p_{u,i}} \tag{10}$$

Items in collection $\mathcal{I} = \{i_1, ..., i_M\}$ are recommended to users in collection $\mathcal{U} = \{u_1, ..., u_M\}$.Users have preferences with respect to items, which are generally modelled by a label $y_{u,i,t}$ per user $u \in \mathcal{U}$ and per item $i \in \mathcal{I}$.$\mathcal{T} = \{t_1, ..., t_T\}$ is a set of time periods; we allow the user's preferences $y_{u,i,t}$ to change at different time periods $t \in \mathcal{T}$.As Eq. (10) shows, $p_{u,i}$ refers to the static propensity, which refers to the probability that user u observes a particular rating of item i in any time period.In reality, user interaction data is very sparse, so it is unrealistic to have all users provide ratings for all items. In the Eq. (10), the observation metrics matrix $O \in \{0,1\}^{|\mathcal{U}| \cdot |\mathcal{I}| \cdot |\mathcal{T}|}$ indicates which ratings were recorded in the recorded interaction data and which ratings were recorded at that time period. When $o_{u,i,t} = 1, o_{u,i,t} \in O$ it means that ratings of user u for item i in time period t have been recorded in the logged data, and if $o_{u,i,t} = 0$ it means that they are missing.Compared to the Eq. (10), the conventional RS debiasing losses are as follows:

$$\mathcal{L} = \frac{1}{|\mathcal{U}| \cdot |\mathcal{I}| \cdot |\mathcal{T}|} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}} L(\hat{y}_{u,i,t}, y_{u,i,t}) \tag{11}$$

Where the function L can be chosen according to different RS evaluation metrics, if it is a common Mean Square Error (MSE) metric, the formula is as follows:

$$L(\hat{y}_{u,i,t}, y_{u,i,t}) = (\hat{y}_{u,i,t} - y_{u,i,t})^2 \tag{12}$$

In the above we mentioned the plain recommendation loss function and the static IPS-based loss function, next we focus on the dynamic selection bias.We set the exact propensity$p_{u,i,t}$ such that the dynamic selection bias can be fully corrected by applying the Eq. (13) to the evaluation of the predicted scores with reverse weighting:

$$\mathcal{L}_{\text{TDR}} = \frac{1}{|\mathcal{U}| \cdot |\mathcal{I}| \cdot |\mathcal{T}|} \sum_{u,i,t:o_{u,i,t}=1} \frac{L(\hat{y}_{u,i,t}, y_{u,i,t})}{p_{u,i,t}} \tag{13}$$

Apply the Eq. (13) to the matrix factorization(MF) model. We refer to this method, which combines MF and time-dynamic debiasing methods, as TDR. Given an observed rating $y_{u,i,t}$ from user u on item i at time t,MF computes the predicted rating $\hat{y}_{u,i,t}$ as:$\hat{y}_{u,i,t} = \mathbf{p}_u^T \mathbf{q}_i + b + b_i + b_u + b_t$,where

 doi:10.20944/preprints202501.0607.v1

the $\mathbf{p}_u \in R_c$ and $\mathbf{q}_i \in R_c$ are embedding vectors of user u and item i, and $b_u \in R$, $b_i \in R$, and $b \in R$ are user, item and global offsets, respectively. Crucially, b_i is a time-dependent offset and models the impact of time in rating prediction.Under this model, the proposed TDR is optimised by minimising the loss of:

$$\arg\min_{P,Q,B}\left[\sum_{u,i,t:o_{u,i,t}=1}\frac{L(\hat{y}_{u,i,t},y_{u,i,t})}{p_{u,i,t}} +\lambda\left(\|P\|_F^2+\|Q\|_F^2+\|B\|_F^2\right)\right] \tag{14}$$

where P,Q and B denote the embedding of all users, all items, and all offset items, respectively; δ is the MSE loss function.

4. Experiments

4.1. Datasets

To validate the effectiveness of our model, we conducted extensive experiments on datasets from Amazon.com and Yelp's RecSys. The Amazon and Yelp datasets are well known public recommendation datasets with textual information. However, both datasets are pre-filtered, so they cannot verify the robustness to imbalance and opinion bias. Therefore, we collect two real-world datasets from Taobao and Jingdong on our own as a supplement to validate our method. All datasets contain a rating range of 0 to 5. We use 5-fold cross-validation to partition the datasets, with 80% as the training set, 10% as the testing set, and 10% as the validation set. The details of the datasets are shown in Table 1.

Table 1. The datasets' characteristics.

Dataset	Amazon	Yelp	Taobao	Jingdong
user	30,759	45,980	10,121	8031
item	16,515	11,537	9892	3025
review	285,644	229,900	10,791	8310
rating	285,644	229,900	49,053	25,152
Sparsity	0.051%	0.043%	0.049%	0.12%
Avg words/s	10.1	9.9	12.7	13.2
Avg words/r	10.4	130	65	70
Avg sentences/r	9.7	11.9	4.9	5.1
Avg reviews/u	9.29	5.00	1.06	1.03

4.2. Baselines

The following is a description of the models compared in this paper:

(1) Static Average Item Rating (Avg): The average observed rating across all item-ages.
(2) Time-aware Average Item Rating (T-Avg): The average observed rating per item-age.
(3) Static matrix factorization (MF): A standard MF model that assumes selection bias is static.
(4) Time-aware matrix factorization (TMF) [16]: TMF captures the drift in popularity as items get older by adding an agedependent bias term.
(5) Time-aware tensor factorization (TTF) [17]: TTF extends MF by modelling the effect of item-age via element-wise multiplication.
(6) MF-StaticIPS: MF combined with static IPS methods.
(7) TMF-StaticIPS: TMF combined with static IPS methods.
(8) MF-DANCER: MF combined with dynamic IPS methods.

 doi:10.20944/preprints202501.0607.v1

4.3. Experiment Results

The main results of our comparison are shown in Table 2, from which we can easily observe that the event-based approach outperforms the static approach by a large margin. This shows that assuming a user preference is static when it is actually dynamic can seriously compromise method performance. Among them, SCTD is the best performing method, which starts from the data and establishes the sentiment score module, alleviating the problem of data sparsity and imbalance. Meanwhile, it combines IPS and MF based on dynamic scene time correlation, alleviating the influence of selection bias in many ways.

Table 2. Compared with different methods.

MEthod	MSE↓	MAE↓	ACC↑
Avg	0.3155	0.4321	0.3623
T-Avg	0.328	0.4326	0.3614
MF	0.1811	0.3314	0.468
TMF	**0.1338**	**0.2818**	**0.5396**
MF-StaticIPS	0.1879	0.3377	0.4598
TMF-StaticIPS	0.1086	0.2491	0.6065
MF-DANCER	0.1533	0.3032	0.5074
SCTD	**0.1032**	**0.2527**	**0.6202**

5. Conclusions

In this paper, we consider selection bias in dynamic scenarios and the impact of sentiment factors on bias. Based on the bias caused by sentiment factors, we built the SCS module for processing, which solves the problem of sentiment bias between review and ratings by filtering bad reviews through a screening layer, and then deriving the sentiment score matrix through a training layer. At the same time, We applied a debiased recommendation method based on time dynamic scenarios to process the enhanced rating matrix obtained from the previous optimization of the sentiment score module to mitigate the influence of selection bias.The experimental results on Amazon,Yelp and Taobao datasets showed that our approach outperforms these baselines in terms of valuation metrics by analogy with.

In future work, we should consider other aspects of time, such as seasonal effects and the difference between weekdays and days off, while the possibility of multi-modal debiasing of recommendation system can be investigated through problems related to sentiment bias.

Author Contributions: Conceptualization, F.Z.; methodology, F.Z.; software, F.Z.; validation, F.Z.; formal analysis, F.Z. and W.L.; investigation, F.Z.; resources, F.Z.; data curation, F.Z.; writing—original draft preparation, F.Z.; writing—review and editing, W.L.; visualization, F.Z.; supervision, X.Y.; project administration, W.L.; funding acquisition, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was inspired by the National Social Science Foundation of China (NSFC) project 'Research on Data Bias Identification and Governance Mechanisms Based on Digital Twins', Project No. 22BTQ058.

Institutional Review Board Statement: Not aplicable.

Informed Consent Statement: Not aplicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: We are grateful to Hebei University for their cooperation and assistance.

Conflicts of Interest: The authors declare no conflicts of interest.The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

 doi:10.20944/preprints202501.0607.v1

Abbreviations

The following abbreviations are used in this manuscript:

IPS	Inverse Propensity Scoring
MF	Matrix Factorization
SCTD	Sentiment Classification And Temporal Dynamic Debiased Recommendation Module
SCS	Sentiment Classification Scoring Module
TDR	Temporal Dynamic Debiased Recommendation Module
LSTM	Long Short-Term Memory

References

1. Marlin, B.M.; Zemel, R.S. Collaborative prediction and ranking with non-random missing data. In Proceedings of the Proceedings of the third ACM conference on Recommender systems, 2009, pp. 5–12.
2. Ovaisi, Z.; Ahsan, R.; Zhang, Y.; Vasilaky, K.; Zheleva, E. Correcting for selection bias in learning-to-rank systems. In Proceedings of the Proceedings of The Web Conference 2020, 2020, pp. 1863–1873.
3. Pradel, B.; Usunier, N.; Gallinari, P. Ranking with non-random missing ratings: influence of popularity and positivity on evaluation metrics. In Proceedings of the Proceedings of the sixth ACM conference on Recommender systems, 2012, pp. 147–154.
4. Schnabel, T.; Swaminathan, A.; Singh, A.; Chandak, N.; Joachims, T. Recommendations as treatments: Debiasing learning and evaluation. In Proceedings of the international conference on machine learning. PMLR, 2016, pp. 1670–1679.
5. Adamopoulos, P.; Tuzhilin, A. On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems. In Proceedings of the Proceedings of the 8th ACM Conference on Recommender systems, 2014, pp. 153–160.
6. Imbens, G.W.; Rubin, D.B. *Causal inference in statistics, social, and biomedical sciences*; Cambridge university press, 2015.
7. Huang, J.; Oosterhuis, H.; De Rijke, M. It is different when items are older: Debiasing recommendations when selection bias and user preferences are dynamic. In Proceedings of the Proceedings of the fifteenth ACM international conference on web search and data mining, 2022, pp. 381–389.
8. Yang, X.; Guo, Y.; Liu, Y.; Steck, H. A survey of collaborative filtering based social recommender systems. *Computer communications* **2014**, *41*, 1–10.
9. Beel, J.; Gipp, B.; Langer, S.; Breitinger, C. Paper recommender systems: a literature survey. *International Journal on Digital Libraries* **2016**, *17*, 305–338.
10. Rubens, N.; Elahi, M.; Sugiyama, M.; Kaplan, D. Active learning in recommender systems. *Recommender systems handbook* **2015**, pp. 809–846.
11. Zhang, F.; Yuan, N.J.; Lian, D.; Xie, X.; Ma, W.Y. Collaborative knowledge base embedding for recommender systems. In Proceedings of the Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 353–362.
12. Panniello, U.; Tuzhilin, A.; Gorgoglione, M. Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction* **2014**, *24*, 35–65.
13. Champiri, Z.D.; Shahamiri, S.R.; Salim, S.S.B. A systematic review of scholar context-aware recommender systems. *Expert Systems with Applications* **2015**, *42*, 1743–1758.
14. Koren, Y. Collaborative filtering with temporal dynamics. In Proceedings of the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 447–456.
15. Xu, Y.; Yang, Y.; Han, J.; Wang, E.; Zhuang, F.; Yang, J.; Xiong, H. NeuO: Exploiting the sentimental bias between ratings and reviews with neural networks. *Neural Networks* **2019**, *111*, 77–88.
16. Koren, Y.; Rendle, S.; Bell, R. Advances in collaborative filtering. *Recommender systems handbook* **2021**, pp. 91–142.
17. Rosenthal, S.; Farra, N.; Nakov, P. SemEval-2017 task 4: Sentiment analysis in Twitter. *arXiv preprint arXiv:1912.00741* **2019**.

 doi:10.20944/preprints202501.0607.v1

18. Han, J.; Zuo, W.; Liu, L.; Xu, Y.; Peng, T. Building text classifiers using positive, unlabeled and 'outdated'examples. *Concurrency and Computation: Practice and Experience* **2016**, *28*, 3691–3706.
19. Liu, Q.; Wu, S.; Wang, L. Cot: Contextual operating tensor for context-aware recommender systems. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2015, Vol. 29.