SAM: A SELF-ADAPTIVE ATTENTION MODULE FOR CONTEXT-AWARE RECOMMENDATION SYSTEM

Jiabin Liu\(^1\), Zheng Wei\(^1\), Zhengpin Li\(^1\), Xiaojun Mao\(^2\), Jian Wang\(^1\)\*, Zhongyu Wei\(^1\) and Qi Zhang\(^3\)

\(^1\)School of Data Science, Fudan University, Shanghai, China
\(^2\)School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai, China
\(^3\)School of Computer Science, Fudan University, Shanghai, China

ABSTRACT

Recently, textual information has been proven to positively affect recommendation systems. However, most of the existing methods only focus on representation learning of textual information in ratings, while potential selection bias induced by the textual information is ignored. In this work, we propose a novel and general self-adaptive module, the self-adaptive attention module (SAM), which adjusts the selection bias by capturing contextual information based on its representation. This module can be embedded into recommendation systems that contain learning components of contextual information. Experimental results on three real-world datasets demonstrate the effectiveness of our proposal, and the state-of-the-art models with SAM significantly outperform the original ones.

Index Terms— Selection bias, text representation learning, recommendation systems, self-adaptive attention.

1. INTRODUCTION

With the explosive growth of data volume, recommendation systems (RS) have been a powerful tool helping people to alleviate information overload. As the most commonly used collaborative filtering technique for RS, matrix factorization (MF) \([1, 2, 3]\) models the interactions of users and items behind the historical ratings by learning a shared latent space for their representations. To date, many state-of-the-art methods have been devised based on MF. Examples include probabilistic matrix factorization, deep matrix factorization, neural collaborative filtering, etc. However, data sparsity and intrinsic bias of observations can affect validity of latent presentations, thereby deteriorating the prediction accuracy.

To build more effective MF models, one mainstream solution is to utilize the auxiliary textual information, such as user profiles, item descriptions, and reviews of users to items. Wang and Blei \([4]\) proposed the collaborative topic regression (CTR) that adopts a latent Dirichlet allocation (LDA) technique to improve the traditional collaborative filtering under a probabilistic framework. Afterwards, several variants of CTR were presented, which also employ the LDA technique to discover the valuable aspects from textual reviews \([5]\). In addition, Bao et al. \([6]\) proposed a novel MF recommendation based on topic modelling, which utilizes the non-negative matrix factorization to derive topics from textual reviews.

It should be mentioned that the aforementioned models mainly adopt the bag-of-words model, while ignoring the contextual information of documents (e.g., the surrounding words and word orders). To address this issue, Kim et al. \([7]\) firstly utilized deep learning models to capture the contextual understanding of textual information. They proposed a novel document context-aware recommendation model called convolutional matrix factorization (ConvMF). Zhang et al. \([8]\) developed a new hybrid model that jointly models content information as representations of effectiveness and compactness, and leverages the implicit user feedback to make accurate recommendations. Lu et al. \([9]\) presented an innovative recommendation model, which utilizes attention-based recurrent neural networks to extract topical information from review documents. Although these models have achieved outstanding performance, they ignore the selection bias problem caused by the potential distortion of available textual information, and thus cannot accurately reflect the target population.

In this paper, we propose a novel and general self-adaptive module called the self-adaptive attention module (SAM), which can self-adaptively learn attention by utilizing the representation of textual information to offset the selection bias. Our module can be seamlessly integrated into any model containing textual information learning components. In particular, the integrated model effectively takes advantage of the attention learned from the representation of textual information to enhance prediction accuracy by altering the objective function of MF. We evaluate the effectiveness of the SAM module on three real-world datasets. Experimental results demonstrate that the integrated models can significantly outperform the state-of-the-art models. Moreover, extensive experiments verify that our proposal achieves performance improvement constantly over a series of sparse scenarios.

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61971146. J. Liu, Z. Wei and Z. Li contribute equally to this work. Corresponding authors: X. Mao and J. Wang (e-mail: maoxj@sjtu.edu.cn, jian_wang@fudan.edu.cn).

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
2. PRELIMINARY

2.1. Matrix Factorization

Matrix factorization generates the latent features matrices for users and items and obtains the predicted rating using the inner product of the user matrix and item matrix. We denote the observed rating matrix as \( \mathbf{R}_O \in \mathbb{R}^{m \times n} \) where \( m \) and \( n \) indicate the number of users and items, and \( \Omega \) refers to the indices set of observation. Suppose the true hidden rating matrix \( \mathbf{R} \) can be decomposed into two latent feature matrices \( \mathbf{U} \in \mathbb{R}^{m \times d} \) and \( \mathbf{V} \in \mathbb{R}^{n \times d} \) of rank \( d \). The rating \( r_{ij} \) for user \( i \) and item \( j \) can be predicted by \( \hat{r}_{ij} = \mathbf{u}_i^\top \mathbf{v}_j = \sum_{k=1}^{d} u_{ik} v_{kj} \). To obtain the latent features \( \mathbf{U} \) and \( \mathbf{V} \), a general way is to minimize the loss function \( \mathcal{L} \) which represents the sum of square loss between the observed ratings and corresponding predicted ratings. To avoid the overfitting issue, the \( L_2 \) regularization term tends to be considered into loss function as follows:

\[
\mathcal{L} = \sum_{i,j} I_{ij} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2 + \lambda_u \sum_i ||\mathbf{u}_i||^2 + \lambda_v \sum_j ||\mathbf{v}_j||^2, \tag{1}
\]

where \( I \) is indicator matrix that \( I_{ij} \) equals 1 if \((i, j) \in \Omega \) and equals 0 otherwise, and \( ||\cdot|| \) denotes the \( L_2 \) norm.

2.2. Text Representation Learning

In this part, we briefly review the deep learning (DL) models for text representation. Feed-forward networks are among the simplest DL models to achieve text representation. For example, Iyyer et al. [10] introduced the deep averaging network, which feeds an unweighted average of word vectors through multiple hidden layers. Le and Mikolov [11] proposed an unsupervised learning algorithm, which learns vector representations for variable-length pieces of text. These models essentially considered the text as a bag of words and learned a vector representation for each word while ignoring the word dependencies and text structures.

To address the above issue, methods based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been developed to capture a deeper understanding of textual information. Kalchbrenner et al. [12] first proposed a CNN-based model for text representation. Afterwards, Kim [13] proposed a simpler CNN-based model that only has one layer of convolution and performs remarkably well. CNN-based models focus on patterns of words across space, whereas RNN-based ones can better capture the time-level features of words. Tai et al. [14] introduced a generalization of the standard Long Short-Term Memory (LSTM) architecture to tree-structured network topologies, which achieves high performance for representing sentence meaning over a sequential LSTM. Considering the huge success attention mechanism achieved in computer vision, it has attracted increasing research in text representation. Recently, there have been many works on incorporating attention modules into various DL models such as [15]. In contrast to our proposal, these models focused on alleviating the bias from the perspective of model structure.

2.3. Selection Bias

It has been well known that if the observations do not represent the distribution of the underlying data, selection bias will appear [16]. The reason may be that the research is adulterated with artificial selection criteria. Many studies are trying to correct selection bias. Heckman [17] discussed the bias that results from the usage of non-randomly selected samples. Smith and Elkan [18] used Bayesian networks to formalize different types of selection bias.

3. METHODOLOGY

3.1. Overview

The sketch of the integrated models is shown in Figure 1. Specifically, we decompose the observed matrix \( \mathbf{R}_O \) into the completed matrix \( \mathbf{R} \) and the bias-corrected weights matrix \( \mathbf{W} \). We denote textual information by \( \mathbf{X} \), the internal parameters in the DL models of context-aware RS and SAM by \( \mathbf{P}_1 \) and \( \mathbf{P}_2 \), respectively. The training of SAM is independent of \( \mathbf{U} \) and \( \mathbf{V} \) whereas \( \mathbf{W} \) is indispensable to obtain \( \mathbf{U} \) and \( \mathbf{V} \). The natural idea is to model the probabilities of the selection process and adopt a bias-corrected risk function:

\[
\mathcal{L}_{\mathbf{R}} = \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} w_{ij} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2, \tag{2}
\]

where \( w_{ij} \) is the corresponding bias-corrected weights. With the consideration of auxiliary textual information, a naive modelling of the bias-corrected weights would be \( w_{ij} = 1/\mathbb{P}(I_{ij} = 1|x_{ij}) \), where \( x_{ij} \) denotes the context information of item \( j \). This leads to an unbiased risk function due to the observation

\[
E(\mathcal{L}_{\mathbf{R}}|\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{m} \sum_{j=1}^{n} (r_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2.
\]

As the probability \( \mathbb{P}(I_{ij} = 1|x_{ij}) \) is usually unknown in practice, most works provided several parametric ways to model it and plugged the estimators of propensity scores into different rating estimation procedures respectively. However, these works suffer from the prior knowledge of specific selection models and usually require additional parameter tuning processes to avoid extreme propensities phenomena [19]. We observe that the textual information influences the target ratings and, more importantly, can interact with the rating process in most context-aware RS. To include the textual information in the selection model without the requirement of specific modelling, we adopt a DL-based method to obtain the bias-corrected weightings motivated by [7].
3.2. DL-based Bias-corrected Weightings

More specifically, we aim to choose the desired DL-based bias-corrected weights \( \{w_{ij}\} \) for a range of \( U \in \mathbb{R}^{m \times d} \) and \( V \in \mathbb{R}^{n \times d} \) such that

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} (I_{ij}w_{ij} - 1) (r_{ij} - u_i^T v_j)^2 \approx 0. \tag{3}
\]

To allow our weight to be adaptively embedded into different recommendation systems, we need to remove the specific rating information and one direct way is to control \( \sum_{i=1}^{m} \sum_{j=1}^{n} (I_{ij}w_{ij} - 1)^2 \) under the constraint that all \( w_{ij} \geq 1 \). To adopt DL-based bias-corrected weight, we propose the following optimization:

\[
\min \| I \odot W - J \|_S, \quad \text{s.t.} \quad w_{ij}(x_j) \geq 1 \tag{4}
\]

where \( I = \{I_{ij}\} \) and \( J \) represents indicator matrix and matrix with all entries 1 respectively, and \( \odot \) denotes the elementwise Hadamard product of two matrices. We adopt the objective function as \( \| \cdot \|_S \) to be the largest singular value of a matrix to evaluate all the bias-corrected weights simultaneously. Note that the bias-corrected rates are restricted to be greater than or equal to 1 due to the observation that \( 1/P(I_{ij} = 1 | x_j) \) should be greater than or equal to 1. As for each weight \( w_{ij} \), we adopt a CNN architecture to learn \( w_{ij} = CNN(P_2, x_j) \). By considering the regularization terms, we can obtain the complete loss function by extending \( \mathcal{L}_R \):

\[
\mathcal{L} = \mathcal{L}_R + \lambda_u \sum_{i=1}^{m} ||u_i||^2 + \lambda_v \sum_{j=1}^{n} ||v_j - dL(P_1, X)||^2, \tag{5}
\]

where \( DL(P_1, X) \) represents latent vectors of items via one specific DL model, detailed implementations of which can be obtained by referring to the following section.

4. EMPIRICAL STUDY

4.1. Experimental Setting

We evaluate our proposal on two movie rating datasets from MovieLens [20] and the news reading-time dataset created by ourselves. For MovieLens datasets, users rate items on a scale of 1 to 5, which is similar to other homogeneous datasets. To verify the performance of our proposal over various recommendation scenarios, we build a new dataset that comes from Sogou News and consists of the reading time of users on the news. The Sogou News dataset consists of the reading time of users on the news, ranging from [0, 60]. To avoid problems caused by the scale of reading time, we utilize the bi-scaling technique [6] to standardize the interaction matrix before training. We summarize the datasets in Table 1.

| Model          | # User | #Item | #Interaction | Density  |
|----------------|--------|-------|--------------|----------|
| ML-100K        | 943    | 1483  | 88673        | 6.341%   |
| ML-1M          | 6040   | 3544  | 993482       | 4.641%   |
| Sogou News     | 1728   | 2573  | 52912        | 1.190%   |

Note that MovieLens does not contain the textual information, we extract the movie descriptions from IMDB and perform preprocessing on it as [7] did. For Sogou News, we set the length of news to 500 words by truncating the longer news and padding the shorter news. In addition, we select the top 36000 words as a vocabulary and remove all non-vocabulary words from the news.

We employ the classical MF and ConvMF, a representative model combining MF with DL models for textual information.
Table 2: Overall test RMSE. Models with ‘+’ in their names represent combinations with SAM.

| Model       | Dataset       | MovieLens-100K | MovieLens-1M | Sougou News |
|-------------|---------------|----------------|--------------|-------------|
| MF          |               | 0.959          | 0.902        | 0.940       |
| ConvMF      |               | 0.910          | 0.853        | 0.954       |
| FTMF        |               | 0.928          | 0.845        | 0.961       |
| RCNNMF      |               | 0.931          | 0.846        | 0.966       |
| MF+         |               | 0.958          | 0.880        | 0.938       |
| ConvMF+     |               | 0.907          | 0.844        | 0.939       |
| FTMF+       |               | 0.922          | 0.840        | 0.948       |
| RCNNMF+     |               | 0.915          | 0.843        | 0.961       |
| Improvement |               | 0.33%          | 0.59%        | 0.17%       |

Table 3: Test RMSE over various sparseness of training data on MovieLens 100K dataset. Models with ‘+’ in their names represent combinations with SAM.

| Model       | Training ratio of dataset (density) |
|-------------|-------------------------------------|
| MF          | 20%(1.27%) 40%(2.54%) 60%(3.80%)     |
| ConvMF      | 1.248     1.070  1.001               |
| FTMF        | 1.044     0.962  0.928               |
| RCNNMF      | 1.115     0.989  0.972               |
| MF+         | 1.163     1.011  0.971               |
| ConvMF+     | 1.223     1.063  1.000               |
| FTMF+       | 1.058     0.957  0.927               |
| RCNNMF+     | 1.021     0.951  0.930               |
| Improvement | 2.30%     1.14%  0.11%               |

As baselines. Considering that ConvMF provides a framework for the integration of MF and DL models, we design two new models RCNNMF that replaces “Conv” part with recurrent convolutional neural networks (RCNN) [21], and FTMF that substitutes FastText [22] for “Conv” part for baselines. We evaluate the performance improvement produced by the integration of these models and the SAM.

For each dataset, we randomly split observations into 80% and 20% as train/test sets. To make MF work on all users and items, we ensure at least one interaction exists on each user and item. We evaluate the performance of prediction using the root mean square error (RMSE), which is computed by

\[
\text{RMSE} = \sqrt{\frac{\sum_{(i,j) \in \Omega} (r_{ij} - \hat{r}_{ij})^2}{|\Omega|}},
\]

where \(\Omega\) indicates the indexes of the test set. To avoid overfitting, we set the maximum iteration to 200 with early stopping.

4.2. Performance Comparison

Table 2 shows the performance of baselines and models with SAM. For enhanced clarity, we rescaled the results of Sougou News through the RMSE multiplying by \(10^{-2}\). We observe that models with SAM significantly outperform the corresponding baselines on all datasets. Specifically, for two MovieLens datasets, the best performances are achieved by integrating ML and DL models. The improvements demonstrate the effectiveness of SAM when textual information facilitates the rating prediction. For Sougou News, ML with SAM also obtains considerable improvement over the vanilla MF, which further proves the potency of SAM.

Table 3 reveals the comparison results of all methods under a series of missing scenarios. Overall, we observe that improvements caused by SAM get more significant when the train set becomes more sparse. It implies that SAM has excellent potential in extremely sparse scenarios.

4.3. Parameter Analysis

We further conduct experiments on the regularization parameters in Fig. 2, which illustrates the impacts of \(\lambda_u\) and \(\lambda_v\) on MovieLens-100K and MovieLens-1M respectively. We choose ConvMF+SAM as an example and employ grid search to select optimal parameter pairs to serve as the final regularization parameters. We set the parameter gap by ten times from the start value of one.

5. CONCLUSIONS

In this paper, we have proposed a general attention-based module SAM to alleviate selection bias derived from the utilization of textual information in recommendation systems. The proposed module can be seamlessly integrated into models containing learning components of the textual information. Empirical studies on three real-world datasets have demonstrated the effectiveness of SAM. Moreover, extensive experiments have implied the great potential of the SAM module under extremely sparse scenarios.
6. REFERENCES

[1] J. Wei, C. Tong, B. Wu, Q. He, S. Qi, Y. Yao, and Y. Teng, “An entropy weighted nonnegative matrix factorization algorithm for feature representation,” IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 9, pp. 5381–5391, 2022.

[2] H. Zhang, F. Luo, J. Wu, X. He, and Y. Li, “Lightfr: Lightweight federated recommendation with privacy-preserving matrix factorization,” ACM Transactions on Information Systems, vol. 41, no. 4, pp. 1–28, 2023.

[3] C. Tong, J. Wei, S. Qi, Y. Yao, T. Zhang, and Y. Teng, “A majorization–minimization based solution to penalized nonnegative matrix factorization with orthogonal regularization,” Journal of Computational and Applied Mathematics, vol. 421, p. 114877, 2023.

[4] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, pp. 448–456.

[5] G. Ling, M. R. Lyu, and I. King, “Ratings meet reviews, a combined approach to recommend,” in Proceedings of the 8th ACM Conference on Recommender Systems, 2014, pp. 105–112.

[6] Y. Bao, H. Fang, and J. Zhang, “Topicmf: Simultaneously exploiting ratings and reviews for recommendation,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28, no. 1, 2014, pp. 2–8.

[7] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, “Convolutional matrix factorization for document context-aware recommendation,” in Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 233–240.

[8] S. Zhang, L. Yao, and X. Xu, “Autosvd++ an efficient hybrid collaborative filtering model via contractive auto-encoders,” in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 957–960.

[9] Y. Lu, R. Dong, and B. Smyth, “Coevolutionary recommendation model: Mutual learning between ratings and reviews,” in Proceedings of the 2018 World Wide Web Conference, 2018, pp. 773–782.

[10] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, “Deep unordered composition rivals syntactic methods for text classification,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015, pp. 1681–1691.

[11] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in Proceedings of the 31st International Conference on Machine Learning, 2014, pp. 1188–1196.

[12] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” arXiv preprint arXiv:1404.2188, 2014.

[13] Y. Kim, “Convolutional neural networks for sentence classification,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746–1751.

[14] K. S. Tai, R. Socher, and C. D. Manning, “Improved semantic representations from tree-structured long short-term memory networks,” arXiv preprint arXiv:1503.00075, 2015.

[15] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “Hierarchical attention networks for document classification,” in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics, 2016, pp. 1480–1489.

[16] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, “Bias and debias in recommender system: A survey and future directions,” ACM Transactions on Information Systems, vol. 41, no. 3, pp. 1–39, 2023.

[17] J. J. Heckman, “Sample selection bias as a specification error,” Econometrica: Journal of the Econometric Society, pp. 153–161, 1979.

[18] A. Smith and C. Elkan, “A bayesian network framework for reject inference,” in Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 286–295.

[19] J. D. Kang and J. L. Schafer, “Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data,” Statistical Science, vol. 22, no. 4, pp. 523–539, 2007.

[20] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” ACM Transactions on Interactive Intelligent Systems, vol. 5, no. 4, pp. 1–19, 2015.

[21] S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent convolutional neural networks for text classification,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, no. 1, 2015.

[22] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” arXiv preprint arXiv:1607.01759, 2016.