Attention Mechanism indicating Item Novelty for Sequential Recommendation

Li-Chia Wang
Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan
q36074285@gs.ncku.edu.tw

Hao-Shang Ma
Institute of Computer and
Communication Engineer
Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan
ablove904@gmail.com

Jen-Wei Huang
Department of Electrical Engineering
National Cheng Kung University
Tainan, Taiwan
jwhuang@mail.ncku.edu.tw

***Abstract*—Most sequential recommendation systems, including those that employ a variety of features and state-of-the-art network models, tend to favor items that are the most popular or of greatest relevance to the historic behavior of the user. Recommendations made under these conditions tend to be repetitive; i.e., many options that might be of interest to users are entirely disregarded. This paper presents a novel algorithm that assigns a novelty score to potential recommendation items. We also present an architecture by which to incorporate this functionality in existing recommendation systems. In experiments, the proposed NASM system outperformed state-of-the-art sequential recommender systems, thereby verifying that the inclusion of novelty score can indeed improve recommendation performance.**

***Index Terms*—Sequential recommendation, self-attention, Novelty Embedding**

I. Introduction

Recent successes in the use of deep learning for word embedding and document embedding have prompted their application in recommender systems. Some recommendation systems use historical information of the user as contextual information; however, such information can be difficult to obtain. Markov Chain based recommendation [3, 4] successfully capture short-term item transitions. Recurrent Neural Networks (RNNs) memorize the user preferences compiled over a long period; [5] however, that approach requires large amounts of rich data. Self-attention mechanisms [7] inspired by Transformer [14] have been used to improve Markov Chain models and RNN-based models. Sequential recommendation schemes favor items that are related to recently viewed items; however, those items do not necessarily reflect the preferences of the user. Some recommendation systems consider the novelty of items [11]; however, this generally reduces overall accuracy. One such system [16] combined novelty-seeking with LDA for cross-domain recommendation; however, the accuracy was far lower than that of recent sequential recommendation systems.

In this paper, we developed an attention mechanism used to determine the novelty seeking behavior of users. Note that the proposed system is generalizable to existing recommendation systems. The proposed Novel Attention Mechanisms with item for Sequential Recommendation (NAMSR) algorithm uses a sliding windows to divide the historic sequence of user behaviors into multiple segments, quantifies the novelty of items in that sequence, and then injects novelty embeddings for the attention mechanism. The proposed system also uses popular temporal features based on MEANTIME [1] to improve recommendation results. In experiments, NAMSR outperformed state-of-the-art algorithms on a variety of real-world datasets. The contributions of this paper are as follows:

- ✓ We developed a sequential recommendation architecture that combines a self-attention architecture, temporal features, and a metric indicating the degree to which the user favors novelty.
- ✓ The proposed system outperformed state-of-the-art recommendation systems on real-world datasets and the proposed novelty feature proved highly generalizable to existing systems.

II. Related Works

A. Temporal Recommendation

Note that the methods described above focus entirely on the sequence, thereby disregarding other important features, such as contextual information (e.g., type, tags, categories) and temporal information (i.e., behaviors that vary over time). TiSASRec [10] considered temporal information using an attention mechanism for next-item recommendation. MEANTIME [1] is an improvement on TiSASRec using multiple types of temporal embeddings. TGSRec [2] adopted a graph structure to unify sequential patterns and temporal collaborative signals.

B. Novelty-Seeking behavior

Novelty seeking behavior refers to the disposition of the user toward unexpected or surprising outcomes. The fact that existing recommendation systems disregard novelty seeking behavior limits the scope of items included in the recommendations. A failure to account for novelty inevitably leads to recommendation lists made up entirely of items that are similar to those that the user has already seen. PPNW [11] sought to overcome this limitation by performing a re-ranking of base model outputs [6]. This single-stage approach uses the popularity of items to adjust loss weighting during item

IEEE/ACM ASONAM 2022, November 10-13, 2022
978-1-6654-5661-6/22/$31.00 É 2022 IEEE

selection. CDNST [16] is a computational framework designed to identify instances of novelty-seeking behavior for use in a cross-domain recommendation model. In the current study, we adopted a similar approach in our development of our novelty attention mechanism.

III. Novelty Attention Mechanisms for Sequential Recommendation

Personalized sequential recommendation systems are widely used for movies, music, and retail products. Recommendation systems based on a self-attention mechanism have also proven highly effective. MEANTIME [1] applies an attention mechanism to a variety of temporal features. In the current study, we developed an algorithmic model by which to quantify the novelty of a given items for a given user for implementation using state-of-the-art sequential recommendation systems. In Section III-A below, we define the problem addressed in this study. In Section III-B, we outline the architecture of the proposed recommendation system.

A. Problem Definition

Let U denote a set of users and I be a set of items. We sort the items with which the user has previously interacted based on timestamps in order to derive a sequence for a given user, as follows: $I^u = [i^u_1, ..., i^u_k, ..., i^u_{|I^u|}]$. Our objective is to predict the next item of relevance to that user for a given time based on patterns observed in that sequence. The historic sequence $S = [i^u_{k-1}, i^u_k, ..., i^u_{|S|-1}]$ is fed into the model to predict item $i^u_{|S|}$, which should be of interest to the user.

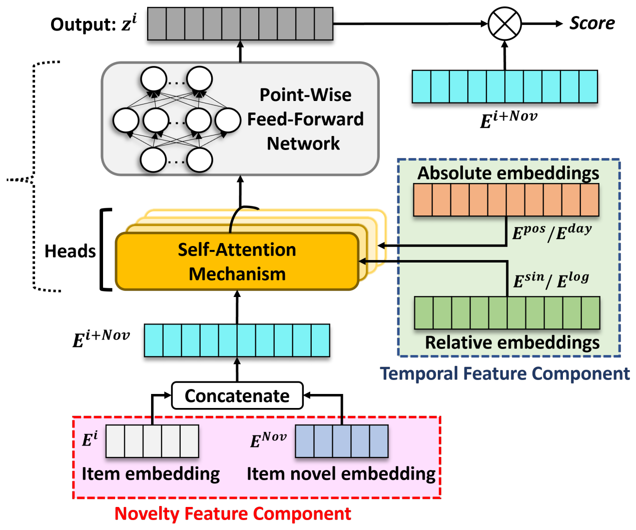

Fig. 1. Framework of proposed NAMSR model.

B. Framework

As shown in Fig. 1, the NAMSR performs four main tasks:

1. Assign a novelty score for a given item for a given user.
2. Concatenate item embeddings with novelty embeddings.
3. Calculate weights for relevant features using the proposed attention mechanism.
4. Assign a prediction score for a given item based on output embeddings from the NLP layer.

C. Item Embedding

Note that items can also be ranked according to contextual information (e.g., type, tags, categories) in accordance with the preferences of the user. This information can be used to calculate a novelty score for a given item for a given user for a given period of time. For example, when dealing with a user who has recently watched multiple horror-tagged movies, similar movies would earn a lower novelty score for that individual in the subsequent period.

1) Embedding Layer: Here, we deal with three embeddings: Item, temporal features, and novelty. The embedded information includes one-hot encoding and the feature representation of the item, which are combined for injection into the attention layer.

- **Input embedding:** Item nodes are encoded using a vector table as follows: $E^i \in \mathbb{R}^{|I|\times(h-|P|)}$. Temporal features include the absolute position and relative position within an item sequence. The absolute position is encoded to two absolute embeddings, referred to as $E^{pos} \in \mathbb{R}^{|I|\times h}$ and $E^{day} \in \mathbb{R}^{|D|\times h}$, where E^{pos} is the learnable positional embedding matrix used in [1, 7, 10], considering that the self-attention model is unaware of the positions of previous items within a historic sequence; E^{day} indicates the learnable positional embedding matrix in which the day of each timestamp is embedded within a vector; and $|D|$ is the span of the dataset measured in days. The relative position describing the relationship between each item pair in the sequence is encoded to two relative embeddings, referred to as E^{sin} and $E^{log} \in \mathbb{R}^{|I|\times|I|\times h}$. Note that E^{sin} and E^{log} respectively apply sin and log functions [1] to temporal difference information to obtain relative embedding values for the various items.
- **Item novelty embedding:** The $softmax$ function is used to normalize $NAL^{i\in I}$ for each item resulting in an output probability distribution for use as item novel embedding $E^{Nov} \in \mathbb{R}^{|I|\times|P|}$. Finally, each item embedding E^i is concatenated with E^{Nov} as a new item embedding $E^{i+Nov} \in \mathbb{R}^{|I|\times h}$.

D. Self-Attention

1) Attention Architecture: In the current study, we adopted the multi-head self-attention architecture outlined in previous studies [1, 7, 10, 13]. The scaled dot-product attention proposed in Transformer [14] is defined as

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{\delta}})V \qquad (1)$$

where Q refers to queries, K refers to keys, V refers to Values, $\sqrt{\delta}$ is the scale factor.

When feeding in absolute embeddings, the score $x_{Abs} \in \mathbb{R}^{|I|\times h/4}$ is computed as follows:

$$x_{Abs} = \frac{(Q_I + Q_{Abs})(K_I + K_{Abs})^T}{\sqrt{\delta}} \tag{2}$$

where Q_{Abs} refers to queries of absolute embedding E^{pos}, E^{day}, and K_{Abs} refers to the keys of the absolute embedding.

Thus, when feeding in relative embeddings, the score of $x_{Rel} \in \mathbb{R}^{|I| \times h/4}$ is computed as follows:

$$x_{Rel} = \frac{(Q_I + b_{Rel})K_{Rel}^T + Q_I K_I^T}{\sqrt{\delta}} \tag{3}$$

Where Q_I refers to the queries of item embedding E^{i+Nov} and K_I is the key of item embedding E^{i+Nov}, whereas K_{Rel} is the key of relative embedding E^{sin}, E^{log}. Note that b_{Rel} indicates global bias.

In the multi-head self-attention layer, all outputs are concatenated from heads, before applying the $softmax$ function and performing an inner product operation with the item embedding, as follows:

$$x = softmax\left([x_{pos}; x_{day}; x_{sin}; x_{log}]\right) V_I \tag{4}$$

where x_{pos} and x_{day} is x_{Abs}, then x_{sin} and x_{log} is x_{Rel}. The V_I is values of item embedding E^{i+Nov}.

where x_{pos} and x_{day} belong to x_{Abs} (calculated using e.g., (2)); and x_{sin} and x_{log} belong to x_{Rel} (calculated using e.g., (3)). V_I indicates the values of item embedding E^{i+Nov}.

2) Point-Wise Feed-Forward Network: We obtain output y from the Attention Layer using the Add and Norm functions. We then apply the Position-wise Feed Forward Network (FFN) and GELU to the self-attention result, as follows:

$$FFN(y) = GELU\left(yW^1 + b^1\right) W^2 + b^2 \tag{5}$$

where $W^1 \in \mathbb{R}^{h \times 4h}, W^2 \in \mathbb{R}^{4h \times h}$ is a learnable weight and $b^1 \in \mathbb{R}^{4h}, b^2 \in \mathbb{R}^h$ is the bias.

Layer normalization, residual connections, and dropout regularization are used to overcome overfitting and instabilities during training. The sublayer is stacked as follows:

$$y = LayerNorm(E + Dropout(Attention(E))) \tag{6}$$

$$z = LayerNorm(y + Dropout(FFN(y))) \tag{7}$$

E. Prediction Layer

Given output $z \in \mathbb{R}^{|I| \times h}$ from the FFN Layer, we then multiply z^i with item embedding $E^{i+Nov} \in \mathbb{R}^{|I| \times h}$ to obtain the item score indicating the probability of that item being viewed during the subsequent period:

$$P(I|i) = softmax(GELU(z^i W^u + b^u)E^i + b^z) \tag{8}$$

where $W^u \in \mathbb{R}^{h \times h}$, $b^u \in \mathbb{R}^h$ and $b^z \in \mathbb{R}^{|I|}$ are learnable parameter.

F. Model Optimization

The final step involves model convergence using the Adam optimizer [8]. We sample a subarray from the item sequence $S = [i_{k-1}^u, i_k^u, ..., i_{|S|-1}^u]$ using parameter σ to adjust the ratio of masked items in order to obtain training sequence S^{train}, which is fed into the model to obtain the prediction score $P(I|S^{train})$, the loss formulation of which is as follows:

$$Loss = -\sum_{i_k \text{ is masked}} \log(P(i_k|S^{train})) \tag{9}$$

IV. Experiment

In this chapter, we outline the setup and results of the experiments. The efficacy of the proposed system was evaluated within the context of the following topics:

- **RQ 1:** Did the proposed model outperform the baseline recommender algorithm?
- **RQ 2:** Did the novelty feature make it possible to determine whether a given user would like to see novel items in their recommendation list? How does the performance of NAMSR vary when applied to users who display novelty seeking behavior versus those who do not?

A. Dataset

We used four real-world datasets in evaluating the proposed model. Note that due to the nature of this model, we were limited to datasets with timestamps and item categories:

- **MovieLens**: This dataset is widely used in recommender systems. Note that we used Movieslen1M, which includes 6440 users (with at least 20 ratings), 3706 movies, and 18 tags, for a total of roughly 1 million ratings.
- **MovieLens + IMDb/Rotten Tomatoes:** This dataset is an extension of MovieLens10M dataset based on IMDb and Rotten Tomatoes movie review systems. This dataset includes 2113 users (at least 20 ratings), 10,197 movies, and 20 tags for a total of roughly 80,000 ratings.
- **Ciao:** This dataset is from popular shopping product review sites. Here, the number of products is large while the number of users is low; therefore, we used only the users with at least 10 reviews. This resulted in roughly 1000 users and 15,000 items with 6 tags.
- **Epinions:** This dataset is from popular shopping product review sites. Here, the number of products is large; therefore, we used only items that appeared at least five times and users with at least 20 reviews. This resulted in roughly 7000 users and 30,000 items with 26 tags.

B. Experiment Setting

The experiments were implemented in two stages. We first compared the proposed model with other basic methods. We then evaluated the efficacy of proposed novelty feature when applied to users who do or do not display novelty seeking behavior. The hyper-parameter settings for the other methods in this comparison were those suggested by the respective authors. When applied to datasets to which the

TABLE I
PERFORMANCE COMPARISON OF BASELINE ALGORITHMS FOR ALL USERS. THE BEST SCORES IN EACH ROW ARE IN BOLD, AND THE SECOND-BEST SCORES ARE UNDERLINED. THE IMPROVEMENT SCORE INDICATES THE MARGIN BETWEEN THE BEST SCORE AND THE SECOND-BEST SCORE.

Dataset	Metric	MF[9]	BPR[12]	BPR+ PPNW-G[11]	SASRec[7]	MEANTIME[1]	NAMSR	Improvement Percentage(%)
ML-1M	NDCG@10	0.20005	0.27681	0.20987	0.56142	0.60206	**0.63881**	6.10404
	HT@10	0.37963	0.51390	0.39288	0.79188	0.81573	**0.84120**	3.12236
ML+IMDb/ Rotten Tomatoes	NDCG@10	0.33688	0.33896	0.32939	0.50055	0.55721	**0.58051**	4.18155
	HT@10	0.55513	0.58353	0.54661	0.74585	0.78284	**0.80995**	3.46303
Ciao	NDCG@10	0.16722	0.17567	0.09858	0.17745	0.20527	**0.21655**	5.49520
	HT@10	0.27861	0.28611	0.16041	0.28517	0.33878	**0.35015**	3.35616
Epinions	NDCG@10	0.18279	0.30637	0.19281	0.34709	0.40555	**0.42323**	4.35951
	HT@10	0.31263	0.51505	0.33387	0.53629	0.61745	**0.62950**	1.95158

TABLE II
PERFORMANCE COMPARISON OF BASELINE ALGORITHMS FOR CLUSTER1 ($r > 0.5$). THE BEST SCORES IN EACH ROW ARE IN BOLD AND THE SECOND-BEST SCORES ARE UNDERLINED. THE IMPROVEMENT SCORE INDICATES THE MARGIN BETWEEN THE BEST SCORE AND THE SECOND-BEST SCORE.

Dataset	Metric	MF[9]	BPR[12]	BPR+ PPNW-G[11]	SASRec[7]	MEANTIME[1]	NAMSR	Improvement Percentage(%)
ML-1M	NDCG_c1@10	0.06783	0.13218	0.07595	0.47456	0.53416	**0.59684**	11.73431
	HT_c1@10	0.13209	0.27309	0.14181	0.70583	0.73761	**0.79515**	7.80087
ML+IMDb/ Rotten Tomatoes	NDCG_c1@10	0.11378	0.11806	0.09163	0.39227	0.46880	**0.47455**	1.22654
	HT_c1@10	0.19572	0.25622	0.16725	0.61921	0.64979	**0.70322**	8.22266
Ciao	NDCG_c1@10	0.03078	0.04972	0.02817	0.03994	0.04432	**0.05168**	3.94208
	HT_c1@10	0.05925	0.08998	0.05432	0.07654	0.08831	**0.10039**	11.56923
Epinions	NDCG_c1@10	0.06376	0.21832	0.05497	0.28262	0.33570	**0.36018**	7.29223
	HT_c1@10	0.12688	0.39764	0.10754	0.45471	0.55866	**0.57417**	2.77629

TABLE III
PERFORMANCE COMPARISON OF BASELINE ALGORITHMS FOR CLUSTER2 ($r <= 0.5$). THE BEST SCORES IN EACH ROW ARE IN BOLD AND THE SECOND BEST SCORES ARE UNDERLINED. THE IMPROVEMENT SCORE INDICATES THE MARGIN BETWEEN THE BEST SCORE AND THE SECOND-BEST SCORE.

Dataset	Metric	MF[9]	BPR[12]	BPR+ PPNW-G[11]	SASRec[7]	MEANTIME[1]	NAMSR	Improvement Percentage(%)
ML-1M	NDCG_c2@10	0.23399	0.31395	0.24426	0.58372	0.61904	**0.64879**	4.80583
	HT_c2@10	0.44319	0.57573	0.45734	0.81398	0.83456	**0.85249**	2.14844
ML+ IMDb/ Rotten Tomatoes	NDCG_c2@10	0.37110	0.37285	0.36586	0.51716	0.57101	**0.59731**	4.60587
	HT_c2@10	0.61026	0.63373	0.60480	0.76528	0.80330	**0.82725**	2.98145
Ciao	NDCG_c2@10	0.25083	0.25283	0.14172	0.26170	0.26416	**0.27077**	2.50227
	HT_c2@10	0.41301	0.40695	0.22541	0.41301	0.43818	**0.44154**	0.76681
Epinions	NDCG_c2@10	0.23022	0.34146	0.24772	0.37278	0.43397	**0.44851**	3.35046
	HT_c2@10	0.38665	0.56184	0.42406	0.56879	0.64172	**0.65178**	1.56766

other methods have not previously been applied, a grid-search of hyperparameters was used to derive parameter settings. We set hidden dimension $h \in \{32, 64, 128\}$, drop ratio $\in \{0.1, 0.2, 0.3, 0.5\}$, number of heads $\in \{1, 2\}$, and length of sequence $\in \{50, 100, 200\}$. For BPR+PPNW-G [11], we set the specific parameter in accordance with the original paper: α to 2.5,$\lambda = 2$, and $\gamma = \{40, 125\}$.

C. Evaluation Metric

We adopted the leave-one-out strategy in evaluating the proposed model. We adopted as a positive item the last item with which the user interacted as well as a sample of 99 negative items(i.e., those with which the user has not interacted). In the Test set, the model ranked the 100 items for each user using two Top-K metric, NDCG, and Hit ratio(HR) as performance metrics.

D. Performance Evaluation

*1) **Performance versus baseline (RQ1)**:* We first compared the performance of the proposed model with the baseline values obtained using the methods listed in Table I. The proposed NAMSR model achieved excellent performance when applied to all four datasets and particularly Ciao. NAMSR greatly improved NDCG@K, thereby verifying its efficacy in ranking recommended items. Note that combining the BPR method (using the parameters from the original paper) with PPNW-G led to the inclusion of novel items as well as notable decrease in prediction accuracy.

*2) **Performance versus user group (RQ2)**:* We then divided the users into two groups according to whether they wanted to see novel items (novelty ratio > 0.5; Cluster 1) or did not want to see novel items (novelty ratio $r <= 0.5$; Cluster2). Note that we set the novelty ratio $r = 0.5$ for Table II,Table III. In accordance with the Pareto Principle outlined in [11], novel items were identified as those belonging to the long-tail portion of the sample [15]. Note that a long tail here refers to the majority of items (80%) that are not popular, whereas 20% of the items are popular. From among the last ten items with which the user interacted, we calculated the proportion that fell within the long tail.

As shown in Table II and Table III, NAMSR achieved the best results for Cluster1 when applied to the ML and ML+ IMDb/Rotten Tomatoes datasets, while MEANTIME achieved the second-best results for all datasets except Ciao, in which it was outperformed by BPR. The unexpected poor performance of MEANTIME can perhaps be attributed to insufficient data for training the neural network. Overall, NAMSR proved highly effective in tailoring recommendations based on the desire (or lack of desire) to see novel items; i.e., in accordance with the predisposition of the user toward novelty.

V. Conclusions

In this work, we capture changes in the novelty of items for that user as a function of time in order to characterize users in terms of novelty seeking behavior. The proposed novelty attention mechanism was then implemented in the NAMSR recommendation system. In experiments on well-known databases, we determined that the novelty of items is an important issue for many users. Our results also demonstrated the generalizability of the proposed framework. In the future, it should be possible to focus on other contextual or feature-related information in deriving the novelty level.

References

[1] S. M. Cho, E. Park, and S. Yoo. Meantime: Mixture of attention mechanisms with multi-temporal embeddings for sequential recommendation. pages 515–520, 2020.

[2] Z. Fan, Z. Liu, J. Zhang, Y. Xiong, L. Zheng, and P. S. Yu. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*, pages 433–442, 2021.

[3] R. He, W. Kang, and J. J. McAuley. Translation-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys*, pages 161–169, 2017.

[4] R. He and J. McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining, ICDM*, pages 191–200, 2016.

[5] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR*, 2016.

[6] M. Jugovac, D. Jannach, and L. Lerche. Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Systems with Applications*, 81:321–331, 2017.

[7] W.-C. Kang and J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining, ICDM*, pages 197–206, 2018.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.

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

[10] J. Li, Y. Wang, and J. McAuley. Time interval aware self-attention for sequential recommendation. pages 322–330, 2020.

[11] K. Lo and T. Ishigaki. Matching novelty while training: Novel recommendation based on personalized pairwise loss weighting. In *2019 IEEE International Conference on Data Mining, ICDM*, pages 468–477, 2019.

[12] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[13] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM*, pages 1441–1450, 2019.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[15] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen. Challenging the long tail recommendation. *Proc. VLDB Endow.*, 5(9):896–907, 2012.

[16] F. Zhuang, Y. Zhou, F. Zhang, X. Ao, X. Xie, and Q. He. Sequential transfer learning: Cross-domain novelty seeking trait mining for recommendation. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 881–882, 2017.