TUR: Utilizing Temporal Information to Make Unexpected E-Commerce Recommendations

Yongxin Ni[1], Ningxia Wang[1(✉)], Li Chen[1], Rui Chen[2], and Lei Li[1]

[1] Hong Kong Baptist University, Hong Kong, China
{csyxni,nxwang,lichen,csleili}@comp.hkbu.edu.hk
[2] Harbin Engineering University, Harbin, China
ruichen@hrbeu.edu.cn

Abstract. *Unexpectedness recommendations* are getting more attention as a solution to the over-specialization of traditional accuracy-oriented recommender systems. However, most of the existing works make limited use of available interaction information to compute distance and neglect the fact that varying time intervals for recommendations would lead to different perceptions of unexpectedness from users. In this work, we propose a novel **Temporal Unexpected Recommendation (TUR)** approach to improve e-commerce recommendations' unexpectedness. Specifically, we consider the complementarity of both implicit and explicit distances, modeling unexpectedness from the latent space (i.e., embedding vectors) and the side information (i.e., item taxonomy) respectively. Meanwhile, we import a module based on the time-aware GRU to leverage the impact of timeliness on recommendation unexpectedness. Experiments on a large-scale e-commerce dataset containing real users' feedback show that TUR significantly outperforms the baselines in enhancing unexpectedness while maintaining a comparable accuracy level.

Keywords: Recommender systems · Unexpectedness · Timeliness · E-commerce

1 Introduction

Over-specialization caused by accuracy-oriented recommendation approaches may isolate users from potential items that may better match their hidden preferences [7]. To counteract such limitations, various beyond-accuracy objectives have been taken into consideration, among which serendipity has been proven to be more effective in affecting user satisfaction [2]. By definition, *serendipity* aims to achieve a desired trade-off between *accuracy* (also call relevance) and *unexpectedness*. The latter usually refers to the degree of items being different from the user's profile and in nature involves the user's emotional response [7]. How to enhance recommendation unexpectedness while still ensuring accuracy comparable to that of classical recommendation algorithms hence becomes a challenging issue.

Either by modifying accuracy-oriented algorithms [1,5,15] or through employing some advanced deep learning techniques (e.g., neural networks) [8–11], existing approaches manage to enhance unexpectedness and alleviate the

R. Chbeir et al. (Eds.): WISE 2022, LNCS 13724, pp. 553–563, 2022.
https://doi.org/10.1007/978-3-031-20891-1_39

over-specialization, but to a limited extent. For example, most of them capture unexpectedness either by implicit distance (e.g., the difference between item embeddings) [8] or by explicit distance with side information (e.g., co-rated users and/or co-occurrence of items) [6,16]. But few works attempt to leverage their complementarity so as to better model unexpectedness. Moreover, few works consider *timeliness* (i.e., how good the time is to recommend a certain item to the target user) in model design, though its significant relationship with users' perceived unexpectedness was demonstrated in a large-scale user survey [2].

To address the above limitations, we propose a novel Temporal Unexpected Recommendation (TUR) approach for e-commerce recommendations in this work. Specifically, we utilize the complementarity of implicit and explicit distances, as well as capturing recommendation timeliness via a time-aware GRU-based module. As for algorithm evaluation, we employ a user survey dataset containing users' real feedback on recommendation unexpectedness [2,13] to evaluate the performance of our method TUR in comparison with several baselines. Results show that TUR significantly outperforms the baselines in terms of unexpectedness, while still maintaining a comparable level of recommendation accuracy.

2 Related Work

The first type of unexpectedness-oriented recommendation approaches modify conventional accuracy-oriented recommendation algorithms using techniques like pre-filtering [5] or re-ranking [1,15]. However, approaches of this type might be constrained by the prediction ability of the underlying algorithm employed. Therefore, recently, some machine learning methods are proposed. For example, Onuma *et al.* [11] employ the graph-based mining technique to recommend items that are close enough to the target user's preferences but have a high potential to reach other nodes. Li *et al.* [9] identify the target user's short-term preferences for movie genres through a RNN with Gated Recurrent Units (GRU), and calculate the elastic relevance between the target movie's user diversity and the target user's in-profile movie diversity. In their follow-up work [10], they further capture not only users' short-term preferences but also long-term preferences, through Gaussian Mixture Model and Capsule Networks respectively. Li *et al.* [8] define unexpectedness as the distance between the target item and the target user's preference closure in the latent vector space and use Self-Attentive GRU to predict item ratings.

However, those approaches are limited in the following four aspects. *First*, their utilization of available information is limited because they mainly rely on a single type of distance (i.e., either implicit distance or explicit distance) to define item unexpectedness. *Second*, existing approaches for unexpectedness mostly neglect recommendation timeliness. That is, an item would bring the same unexpectedness to the user no matter when it is recommended, which may result in low user satisfaction [2]. *Third*, they mostly evaluate recommendation unexpectedness through self-designed approximation metrics, but lack validation from users' real feedback. *Fourth*, the works purely emphasizing unexpectedness may compromise recommendation accuracy (relevance) to a certain extent [11]. Thus, in this work, we have been engaged in overcoming these limitations.

3 Temporal Unexpected Recommendation (TUR)

Adopting the utility theory that combining relevance and unexpectedness into a hybrid utility function may jointly learn the two for producing the overall utility [1,7,16], we extend the utility function of Personalized Unexpected Recommender System (PURS) [8], which, to the best of our knowledge, is the most recent representative utility-based unexpectedness-oriented work. Formally, given the target user u and the target item i, PURS computes the overall recommendation utility as

$$Utility(u,i) = Rel(u,i) + f(Unexp(u,i)) * Unexp\_factor(u,i) \tag{1}$$

where $Rel(u,i)$ is the estimated relevance score of the target item i to the target user u; $Unexp(u,i)$ is the predicted unexpectedness of i to u, generalized via the activation function $f(x) = x * e^{-x}$; and $Unexp\_factor(u,i)$ is to measure the user u's propensity to accepting the item i's unexpectedness, which can be learned through a local activation unit.

To overcome the limitations mentioned previously, our proposed method TUR extends the computation of unexpectedness by adding the explicit distance instead of purely computing the implicit distance as in PURS and considering timeliness. The utility function is as follows:

$$Utility^*(u,i,t_r) = Rel(u,i) + f(Unexp^*(u,i,t_r)) * Unexp\_factor(u,i) \tag{2}$$

where t_r is the time to provide the recommendation, and the unexpectedness score is computed as

$$Unexp^*(u,i,t_r) = \sigma(\alpha \cdot Implicit(u,i) + \beta \cdot Explicit(u,i) + \gamma \cdot Timeliness(u,i,t_r)) \tag{3}$$

where σ is the sigmoid function that turns the integrated impact into the predicted unexpectedness score, and α, β, and γ are hyperparameters integrating the different scales.

We then predict the user's purchase probability by adding a sigmoid function to the utility, i.e., $Prob(u,i) = \sigma(Utility^*(u,i,t_r))$. With the ground truth labels that indicate whether i is really bought by the target user u, we formally train the framework via a cross-entropy loss function. To be noted, our unexpectedness module can be easily integrated with other accuracy-oriented prediction approaches under this framework, by changing the computation method of $Rel(u,i)$ in Eq. (1).

3.1 Implicit Distance and Explicit Distance

The computation of *implicit distance* is mainly inspired by PURS [8] that calculates the difference between the embedding vector of the target item i and user interest clusters. Concretely, the average Euclidean distance is employed:

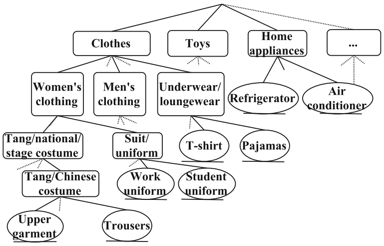

Fig. 1. The item category taxonomy from Mobile Taobao [13].

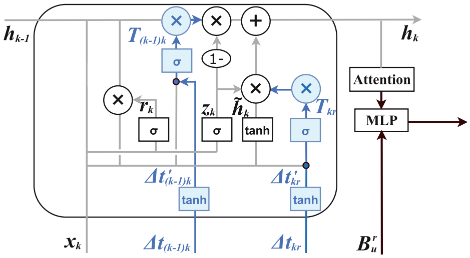

Fig. 2. The time-aware GRU in our framework.

$$Implicit(u,i) = \frac{\sum_{C_u^j \in C_u} d(\vec{i}, \vec{C_u^j})}{|C_u|} \tag{4}$$

where C_u denotes the set of user interest clusters through an unsupervised clustering algorithm [8] over all the embeddings of items that belong to the user's profile P_u, $\vec{i}$ and $\vec{C_u^j}$ are respectively the embedding vectors of the target item i and one interest cluster C_u^j, and $d(\vec{i}, \vec{C_u^j})$ is the Euclidean distance.

We further consider *explicit distance* based on side information. Because hierarchical item taxonomy (see example in Fig. 1) is typically adopted by popular e-commerce platforms (e.g., Amazon, e-Bay, and Taobao) to categorize products, we calculate **taxonomy-based category distance**. A common approach is to count the least hops between two items' leaf categories (hop-based distance). However, this method cannot well distinguish the category differences between pairs of items when their leaf categories lie at different levels. For example, in Fig. 1, the hop-based distance between an item under the leaf category "Work uniform" and that under "Student uniform" is 2, and that between items under "T-shirt" and "Pajamas" is also 2, but the former two nodes turn different at the 4th-level, while the latter two turn different at the 3rd level, so the distances should be different.

Therefore, we employ a computation method that can identify the level at which two items' category paths start to turn different [13]. Formally, the unexpectedness based on explicit distance is defined as:

$$Explicit(u,i) = \sum_{j \in P_u} \frac{1}{|P_u|}(L + 1 - \mathcal{L}_{ij}) \tag{5}$$

where P_u is the user u's profile, L is the total number of levels in the employed taxonomy (e.g., $L = 5$ in Fig. 1), and $\mathcal{L}_{ij}$ is the category distance of the target item i and an item j as represented by the length of their common category path (e.g., $\mathcal{L}_{ij} = 3$, if i is under "Work uniform" and j is under "Student uniform" in Fig. 1, while it is 2 if i is under "T-shirt" and j is under "Pajamas").

In a short summary, compared to previous works that purely consider either implicit distance or explicit distance, our framework leverages both of them for taking advantage of their complementarity.

3.2 Time-Aware GRU for Unexpectedness

As mentioned before, although the significant effect of timeliness on user perception of unexpectedness has been revealed [2], little algorithm work takes timeliness into consideration when predicting an item's unexpectedness score. To fill this gap, we design a time-aware GRU module (see Fig. 2) to learn the impact of the interaction time of each historical behavior on the unexpectedness degree of a certain item that the target user might perceive.

First, for each item k that belongs to u's profile P_u where all visited items are sorted by their interaction timestamps, we obtain its action embedding $x_k = \mathcal{L}_{ki} \cdot B_u^k$, where $\mathcal{L}_{ki}$ is the length of common category path between k and i, B_u^k is the concatenation of the embedding vectors of the user $\vec{u}$ and the item k's category $\vec{c_k}$, both of which are updated via back-propagation through the utility function optimization. With the dot product, we make x_k a weighted representation of u's historical action at the category level, thus larger category distance $\mathcal{L}_{ki}$ would empower the historical action with greater impact during the learning process since it may imply higher unexpectedness.

Next, we consider the actual temporal distance between the target item and the historical interaction. As users with high-frequency behaviors might be more sensitive to a long time interval from the recent interaction till the current recommendation than those with low-frequency, we encode two kinds of time information as the additional inputs, i.e., the time interval $\Delta t_{(k-1)k} = log(t_{uk} - t_{u(k-1)} + 1) + 1$ between $t_{u(k-1)}$ (when the user u visited the $k-1$-th item) and t_{uk} (when the user visited the k-th item), and the time interval $\Delta t_{kr} = log(t_r - t_{uk} + 1) + 1$ between t_{uk} and the time of providing the current recommendation t_r.

Inspired by the enhancement on LSTM in [14], we introduce two additional time gates for GRU by assigning more trainable dense matrices to the three input variables (i.e., x_k, $\Delta t_{(k-1)k}$, and Δt_{kr}) for linear transformation, so that the historical behaviors and temporal features could be learned in a joint manner (see Eqs. (8) and (9)). To be more specific, we introduce weight matrices $W^{'}$ for $t_{(k-1)k}$ and t_{kr} to integrate the two time features. By adding more matrices W/U and establishing interactions with x_k, the two features are converted into the time gates respectively. The governing equations of the modified part are:

$$\Delta t^{'}_{(k-1)k} = \sigma_h(W^{'}_{(k-1)k}\Delta t_{(k-1)k} + b^{'}_{(k-1)k}) \quad (6)$$

$$\Delta t^{'}_{kr} = \sigma_h(W^{'}_{kr}\Delta t_{kr} + b^{'}_{kr}) \quad (7)$$

$$T_{(k-1)k} = \sigma_g(x_k W_{(k-1)k} + \Delta t^{'}_{(k-1)k} U_{(k-1)k} + b_{(k-1)k}) \quad (8)$$

$$T_{kr} = \sigma_g(x_k W_{kr} + \Delta t^{'}_{kr} U_{kr} + b_{kr}) \quad (9)$$

From Fig. 2 we can see that the current state is controlled not only by the original update gate z_k and the reset gate r_k, but also by the two time gates $T_{(k-1)k}$ and T_{kr}.

To further differentiate the impact of each historical interaction on the item i's unexpectedness to user u, we apply a self-attention block to the output of the

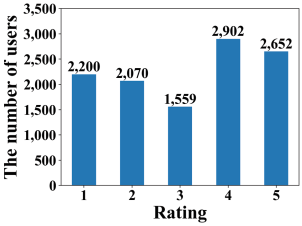

Fig. 3. The distribution of users' ratings on unexpectedness.

Table 1. Statistics of the taobao serendipity dataset

Data	User number	Item number	Interaction number	Sparsity
Historical records	11,383	7,717,420	21,405,555	0.024%
Survey data	11,383	9,985	11,383	0.010%
All	11,383	7,719,403	21,416,938	0.024%

time-aware GRU. The weighted output is subsequently presented to an MLP, incorporated with B_u^r (i.e., the concatenation of user vector $\vec{u}$ and embedding vector of item category $\vec{c_r}$) to forecast how good the time is to surprise the user (i.e., $Timeliness(u, i, t_r)$ in Eq. (3)).

4 Experiment

To measure the performance of our proposed method, we employed *Taobao Serendipity Dataset*[1] that contains users' real feedback on the recommendation's unexpectedness [2,13].

4.1 Taobao Serendipity Dataset

This dataset was collected from *Mobile Taobao*, a popular e-commerce platform in China, from Dec. 21, 2017 to March 17, 2018. Concretely, 11,383 users' feedback on the unexpectedness of recommendation was acquired through an online survey (w.r.t. the question "*The item recommended to me is unexpected*"), as well as their purchase intention (w.r.t. "*I would buy the item recommended, given the opportunity*"), both rated on a 5-point Likert scale from 1 - "strongly disagree" to 5 - "strongly agree". From Fig. 3 we can see that users' unexpectedness ratings are distributed over all the five points. In addition, we have every user's historical records (clicks/purchases) in the past three months before s/he took part in the survey. Each record can be denoted as (u, i, c_i, t_{ui}, p), indicating that the user u clicked ($p = 0$) or purchased ($p = 1$) the item i at the timestamp t_{ui}, the category of which is c_i. In total, there are 21,405,555 historical records, 4.21% of which are purchasing records (see statistics in Table 1). What's more, the category c_i of each item i denotes its leaf category along the path over the hierarchical item taxonomy (see Fig. 1).

To train the time-aware GRU, we followed the idea of session-based recommendation by extracting items during a session window of length $K = 10$ as one input sample. For example, if a user u has an interaction sequence $P_u = [i_{u1}, i_{u2}, i_{u3}, ...i_{u11}]$, there will be 2 data samples $(u, i_{u10}, [i_{u1}, i_{u2}, i_{u3}, ...i_{u9}])$ and

[1] https://github.com/greenblue96/Taobao-Serendipity-Dataset.

$(u, i_{u11}, [i_{u2}, i_{u3}, i_{u4}, ...i_{u10}])$, the dimension of which respectively denote the target user, the target item to be predicted, and the most recent nine interactions before the recommendation time. Then we followed the popular 80/20 rule to split historical data samples into the training/testing dataset, and all the survey data were treated as testing data. The goal was to recommend items users are likely to purchase, so records with $p = 1$ (for historical records) or purchase intention > 3 (for survey data) are labeled as positive samples.

4.2 Compared Methods

There are in total nine compared methods, and eight variations by respectively integrating PURS and TUR with the four accuracy-oriented approaches (e.g., "DIN+TUR" refers to the integration of TUR with the accuracy-oriented approach DIN). Concretely, we compared our method TUR with 5 unexpectedness-oriented methods:

- **Full-Auralist** [15], a personalized algorithm injecting novelty, diversity and serendipity into the learning process;
- **HOM-LIN** [1], a utility-based model to estimate the overall preference a user holds for an item;
- **UNEXP-AUG** [16], a modification of PureSVD by including the unexpectedness as a penalty factor to model both usefulness and unexpectedness;
- **SOG** [6] that introduces feature diversification to promote the recommendation serendipity; and
- **PURS** [8], an advanced framework that unifies relevance and unexpectedness into a hybrid utility function as we described before.

Moreover, we implemented four state-of-the-art accuracy-oriented approaches, in order to see whether the accuracy of our method could be comparable to theirs, and furthermore whether it would be feasible to enhance their unexpectedness by integrating TUR into their framework.

- **DIN** [17] that introduces a local activation unit to adaptively learn representation vector for each user to capture her interests;
- **PNN** [12] that extracts high-order feature interactions by introducing a product layer between the embedding layer and the fully connected layer;
- **Wide&Deep** [3] that jointly trains a linear model for feature processing and a feed-forward neural network for feature learning; and
- **DeepFM** [4] that combines the factorization machine and a feed-forward neural network to learn feature interactions.

Each method was trained and tested three times and the means of their accuracy and unexpectedness performance are reported in Tables 2 and 3. More details on codes and parameters can be found in https://github.com/greenblue96/TUR.

Table 2. Comparison of TUR with unexpectedness-oriented and accuracy-oriented baselines

Method	Accuracy	Unexpectedness	
	(AUC)	(MAE)	(RMSE)
Full-Auralist	0.5660	0.5048	0.6087
HOM-LIN	0.5352	0.4963	0.5978
UNEXP-AUG	0.4991	0.4157	0.5236
SOG	0.5142	0.3831	0.4716
PURS	0.6273	0.5260	0.6373
DIN	0.6186	–	–
PNN	0.5950	–	–
Wide&Deep	0.5930	–	–
DeepFM	**0.6305**	–	–
TUR	0.5920	**0.3267**	**0.4083**
Improvement	–	14.72%	15.50%

Note: The improvement is against the second-best performed method ($p < 0.001$ by Student's t-test).

Table 3. Comparison between TUR and PURS being integrated with other accuracy-oriented methods

Method		Accuracy	Unexpectedness	
		(AUC)	(MAE)	(RMSE)
DIN	+PURS	0.6857	0.5234	0.6337
	+TUR	0.6854	↑**0.3338**	↑**0.4183**
PNN	+PURS	0.6859	0.5234	0.6337
	+TUR	0.6852	↑**0.3350**	↑**0.4199**
Wide&Deep	+PURS	0.6854	0.5181	0.6269
	+TUR	0.6815	↑**0.3411**	↑**0.4278**
DeepFM	+PURS	0.6807	0.5209	0.6306
	+TUR	0.6799	↑**0.3251**	↑**0.4053**
Average improvement		–	35.99%	33.80%

Note: ↑ indicates that xxx+TUR (e.g., DIN+TUR) achieves significantly better prediction than xxx+PURS regarding the corresponding unexpectedness metric ($p < 0.001$ by Student's t-test).

5 Results

Metrics. To evaluate *accuracy*, we adopted the weighted **AUC**, the same as that used in PURS [8]. To evaluate *unexpectedness*, we calculated **MAE** and **RMSE** between the predicted unexpectedness and the user's real unexpectedness feedback after performing min-max normalization. For TUR and PURS, the predicted unexpectedness is the unexpectedness score. For HOM-LIN, it is the linear distance from i to u's expectation set. For UNEXP-AUG, it is the linear combination of i's rareness and its dissimilarity to u's profile. For Auralist [15] and SOG [6], we only use components except the relevance estimation for unexpectedness prediction.

Comparison with Baselines. There are several interesting observations (see Table 2): *First*, regarding recommendation accuracy, TUR outperforms four unexpectedness-oriented baselines, with 4.59% significant improvement ($p < 0.001$ by Student's t-test) than the second-best baseline Full-Auralist in terms of AUC. We also find that the accuracy of TUR can be comparable to those of the state-of-the-art accuracy-oriented algorithms (e.g., PNN and Wide&Deep), but be slightly lower than DIN and DeepFM. *Second*, as for unexpectedness, TUR performs significantly better than all the unexpectedness-oriented baselines, with 14.72% and 15.50% improvements on the second-best baseline SOG in terms of MAE and RMSE respectively, and 37.89% and 35.93% against PURS that TUR extends.

Integration with Accuracy-Oriented Method. Given that both TUR and PURS can act as a utility-based framework to be integrated with other accuracy-oriented methods, we are interested in comparing the two frameworks in this regard. Note that the integration can be simply done by replacing the relevance estimation (e.g., $Rel(u, i)$ in Equation (3) for TUR) with the output of an accuracy-oriented algorithm (i.e., DIN, PNN, Wide&Deep, or DeepFM).

From Table 3 we can see that all the combinations are basically equivalent regarding accuracy (with AUC ranging from 0.6799 to 0.6859), and the results are obviously better than those in Table 2 (in which the best accuracy is 0.6305 by DeepFM). More notably, the comparison between TUR and PURS shows that, when being integrated with the same accuracy-oriented algorithm, the former always significantly outperforms the latter regarding the unexpectedness metrics, with the best MAE and RMSE (0.3251 and 0.4053 respectively) obtained by DeepFM+TUR among all. On average, TUR shows an improvement of 35.99% w.r.t. MAE and 33.80% w.r.t. RMSE compared to to PURS. The results indicate that TUR can be more effective than PURS in terms of boosting the recommendation unexpectedness, while not compromising accuracy (and even increasing) when it is integrated with the accuracy-oriented method.

Table 4. Results of the ablation study

Method	Accuracy	Unexpectedness		Method	Accuracy	Unexpectedness	
	(AUC)	(MAE)	(RMSE)		(AUC)	(MAE)	(RMSE)
TUR^0	0.6854	**0.3338**$^{1-6}$	**0.4183**$^{1-6}$				
TUR$_I{}^1$	0.6855	0.5191	0.6281	TUR$_{IE}{}^4$	0.6856^0	0.4193$^{1-3}$	0.5282$^{1-3}$
TUR$_E{}^2$	0.6856^{0,6}	0.4386^{1,3}	0.5545^{1,3}	TUR$_{IT}{}^5$	0.6873^{0,2,4,6}	0.4988^1	0.6010^1
TUR$_T{}^3$	**0.6874**	0.4960^{1,5}	0.5973^{1,5}	TUR$_{ET}{}^6$	0.6854	0.3443$^{1-5}$	0.4320$^{1-5}$

Note: The superscript indicates that the corresponding method is significantly better than the numbered one.

Ablation Study. As shown above, DIN+TUR's overall performance is more satisfactory regarding both accuracy and unexpectedness, we therefore conducted ablation study on this version that is abbreviated as TUR henceforth. Six variants were implemented: TUR$_I$, TUR$_E$, TUR$_T$,TUR$_{IE}$, TUR$_{IT}$, and TUR$_{ET}$, where the subscript letter indicates the component considered by the variant (i.e., I, E, and T for implicit distance, explicit distance, and timeliness, respectively). Results show that all the six variants are inferior to the complete version of TUR (i.e., DIN+TUR) in capturing users' unexpectedness perception, with at least 3.15% worse w.r.t. MAE and 3.28% worse w.r.t. RMSE than TUR (see Table 4). It hence suggests that these three components, i.e., *implicit distance*, *explicit distance*, and *recommendation timeliness*, all contribute to TUR's unexpectedness prediction to a certain extent. We also notice that the timeliness module can help largely increase unexpectedness. For instance, the comparison of TUR with TUR$_{IE}$ reveals that the former obtains 25.61% and 26.27% improvements on unexpectedness in terms of MAE and RMSE respectively. Another interesting finding is that, the differences of TUR$_{IE}$ from TUR$_I$ and TUR$_E$ are both significant ($p < 0.001$), which verifies our assumption of taking into account their complementarity for enhancing unexpectedness prediction.

6 Conclusions

In this work, we propose the Temporal Unexpected Recommendation (TUR) approach. Specifically, grounded on the utility theory, we model user preference as a utility function accommodating both recommendation relevance and unexpectedness. For the latter, we particularly unify three components, i.e., *implicit distance* defined in the latent space, *explicit distance* computed over hierarchical category taxonomy, and *timeliness* learned through the time-aware GRU. Experiments on an e-commerce dataset containing users' real feedback show the superiority of our method to several baselines by significantly improving unexpectedness, while not compromising accuracy. In the future, we will be engaged in generalizing the findings to other domains, and considering leveraging users' personal characteristics (e.g., curiosity [13]) to further enhance unexpectedness.

Acknowledgements. This work was supported by Hong Kong Research Grants Council (RGC) (project RGC/HKBU12201620).

References

1. Adamopoulos, P., Tuzhilin, A.: On unexpectedness in recommender systems: or how to better expect the unexpected. TIST **5**(4), 1–32 (2014)
2. Chen, L., Yang, Y., Wang, N., Yang, K., Yuan, Q.: How serendipity improves user satisfaction with recommendations? a large-scale user evaluation. In: WWW, pp. 240–250 (2019)
3. Cheng, H.T., et al.: Wide & deep learning for recommender systems. In: DLRS, pp. 7–10 (2016)
4. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: DeepFM: a factorization-machine based neural network for CTR prediction. arXiv preprint arXiv:1703.04247 (2017)
5. Karpus, A., Vagliano, I., Goczyła, K.: Serendipitous recommendations through ontology-based contextual pre-filtering. In: BDAS, pp. 246–259 (2017)
6. Kotkov, D., Veijalainen, J., Wang, S.: How does serendipity affect diversity in recommender systems? a serendipity-oriented greedy algorithm. Computing **102**(2), 393–411 (2020)
7. Kotkov, D., Wang, S., Veijalainen, J.: A survey of serendipity in recommender systems. Knowl.-Based Syst. **111**, 180–192 (2016)
8. Li, P., Que, M., Jiang, Z., Hu, Y., Tuzhilin, A.: PURS: personalized unexpected recommender system for improving user satisfaction. In: RecSys, pp. 279–288 (2020)
9. Li, X., Jiang, W., Chen, W., Wu, J., Wang, G.: Haes: a new hybrid approach for movie recommendation with elastic serendipity. In: CIKM, pp. 1503–1512 (2019)
10. Li, X., Jiang, W., Chen, W., Wu, J., Wang, G., Li, K.: Directional and explainable serendipity recommendation. In: WWW, pp. 122–132 (2020)
11. Onuma, K., Tong, H., Faloutsos, C.: Tangent: a novel, surprise me, recommendation algorithm. In: KDD, pp. 657–666 (2009)
12. Qu, Y., et al.: Product-based neural networks for user response prediction. In: ICDM, pp. 1149–1154 (2016)
13. Wang, N., Chen, L., Yang, Y.: The impacts of item features and user characteristics on users' perceived serendipity of recommendations. In: UMAP, pp. 266–274 (2020)

14. Yu, Z., Lian, J., Mahmoody, A., Liu, G., Xie, X.: Adaptive user modeling with long and short-term preferences for personalized recommendation. In: IJCAI, pp. 4213–4219 (2019)
15. Zhang, Y.C., Séaghdha, D.Ó., Quercia, D., Jambor, T.: Auralist: introducing serendipity into music recommendation. In: WSDM, pp. 13–22 (2012)
16. Zheng, Q., Chan, C.K., Ip, H.H.: An unexpectedness-augmented utility model for making serendipitous recommendation. In: ICDM, pp. 216–230 (2015)
17. Zhou, G., et al.: Deep interest network for click-through rate prediction. In: KDD, pp. 1059–1068 (2018)