EXPLOITING POST-CLICK BEHAVIORS FOR RECOMMENDER SYSTEM

Hai-bin Zhai

CNCERT/CC

ABSTRACT

In recent years, recommender systems become an effective technique to address the information overload problem on the Web, targeting the delivery of most relevant items and information to individual users based on their historical behaviours. To build an effective recommender system, implicit feedback like user click has been harvested. Since click data is usually very noisy, recent works also leverage dwell time as a proxy to optimize user engagement. However, dwell time is just one-dimensional post-click behaviour. The multi-dimensional post-click behaviours, ***e.g.,*** *reading comments, browsing images, and other sub-modules, have not been fully exploited. To address this issue, in this paper, we study leveraging post-click behaviours to improve user engagement in recommendation systems. We first take an E-commerce recommender system as an example to define post-click behaviours and demonstrate their effects with respect to user conversions. Based on the analysis, we then propose a tree-based labelling model, which provides a new perspective to understand user engagement beyond CTR or dwell time. The labelling model can be further incorporated into state-of-the-art recommendation methods. Extensive experiments on the dataset from JD.com, a real-world E-commerce website, demonstrate the competitive performance of the proposed method in both offline and online scenarios.*

KEYWORDS

Recommender Systems, Post-Click Behaviours, Learning to Rank, Tree-based Model.

1. INTRODUCTION

Recommender system as an effective technique to address the information overload problem [4,8,5] has been widely deployed in real-world systems, such as personalized music recommendation on Spotify (https://www.spotify.com), e-commerce recommendation in JD.com (https://www.jd.com/), image recommendation on Pinterests (https://www.pinterest.com/). Currently, recommendation algorithms generally take **relevance** as the main objective, which reflects users' preference on the recommended item. In practice, relevance can be represented with explicit user feedback such as user ratings (e.g., 'like' or 'dislike'). However, they are usually extremely sparse or even not available in real-world applications. Instead, an alternative is to exploit implicit feedback to model users' preference. Click-behaviour is a widely used implicit feedback [12,3,7], which indicates users' interests and thus is often labelled as positive samples. However, user click data is generally highly noisy. For example, a user may click an item casually a then quickly leave the lad page without any further engaged behaviours, which might not indicate high relevance or deep user engagement. To address this issue, dwell time (the amount of time spent within an item page) as a complementing signal has been utilized as a proxy to optimize user engagement [11,10,1,9].

Although optimizing dwell time achieves promising results, the overall dwell time just describes one dimension as user post-click engagement. For example, in news feeds recommendation, dwell time provides a hint on user satisfaction/relevance, but reading user comments on the targeted news may show a stronger signal for user engagement. In E-commerce recommendation scenarios, after clicking a product and landing the item page, a series of post-click actions *e.g.*, reading comments, clicking images, browsing different sub-modules within the item page provides complementary signals to the overall dwell time. For example, after clicking a recommended cell phone, the user usually reads comments and reviews specifications if she is a serious customer. However, previous works rarely consider multiple post-click behaviours in recommender systems.

In this work, **post-click behaviour** is defined as any effective action within the item page after the user clicks an item. The specific problem we are addressing is to leverage post-click behaviours to optimize user engagement in recommendation systems. However, this study brings several challenges: **1)** post-click behaviours are essentially heterogeneous that are difficult for modelling and assessment; **2)** post-click behaviours can be significantly different depending on the item content and the context. For example, a user generally spends more time when browsing mobile phones than fast consuming products, such as food and drinks; **3)** how to incorporate post-click behaviours into existing recommendation models remains a problem. To address these issues, we propose a tree-based labelling model, which provides a new perspective to understand user engagement beyond CTR or dwell time. The labelling model can be further incorporated into state-of-the-art recommendation methods. Extensive experiments on the dataset from JD.com, a real-world E-commerce website, demonstrate the competitive performance of the proposed method in both offline and online.

2. Effects of Post-Click Behaviours

In this section, we take E-commerce recommender system as an example to demonstrate the effects of post-click behaviours. In E-commerce scenarios, ordering is naturally a ground truth label for clicks: if one finally purchased a recommended item, the click on the item should be significant. By leveraging such ordering signals, we will demonstrate effective patterns of post-click behaviours. Here conversion rate is the main metric for analysis, which calculates the percent of user clicks ending with ordering. Formally, it can be formulated as follows:

$$Conversion\,rate = \frac{\#\,clicks\,ended\,with\,ordering}{\#\,all\,clicks}$$

2.1. Post-click Behaviours

The terminology of post-click behaviours in this work is summarized as follows.

- **Modules**: Within an item (a product) page, there are serval basic modules: main page (basic information, price, etc.), item pictures, specifications (more parameters and details) and comments. The browsing modules usually reflect user’s interests.
- **Behaviours**: We consider two types of post-click behaviours: browsing modules (clicks) and dwell time. In particular, we count the number of clicks in a given browsing module. Furthermore, besides the total dwell time within the item page, we also zoom into the dwell time in each browsed module.
- **Orders**: Making an order is a highly strong signal to measure user interest and satisfaction in E-commerce recommendations.

The definitions of post-click behaviours considered in this study are summarized in Table 1. In particular, we use product id p_i and category id c_i to denote the clicked product and category, and use y to indicate whether the user makes an order. Considering both modules and behaviour types, we have 9 variables $x_1 \sim x_5$ and 4 dwell time variables $x_6 \sim x_9$.

Table 1. Post-click Behaviours Definition in our Dataset

Variable	**Type**	**Name**	**Description**
p_i	String	Product ID	ID of products
c_i	String	Category ID	Category ID of Products
x_1	Int	Main_Page_Click	Number of clicks within the main page
x_2	Int	Detail_Page_Click	Number of clicks within the specification page
x_3	Int	Comment_Page_Click	Number of clicks within the comment page
x_4	Int	Main_Pic_Click	Number of clicks within all thumbnail pictures
x_5	Int	All_Click	Number of all clicks within the product item page
x_6	Float	Main_Page_Dwell_Time	Dwell time within the main page
x_7	Float	Detail_Page_Dwell_Time	Dwell time within the specification page
x_8	Float	Comment_Page_Dwell_Time	Dwell time within the comment page
x_9	Float	All_Dwell_Time	Dwell time within the product item page
y	Binary	Order	Dwell time within the product item page

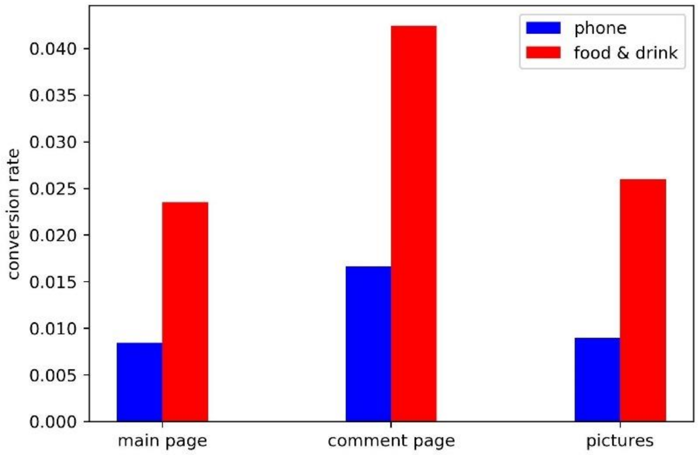
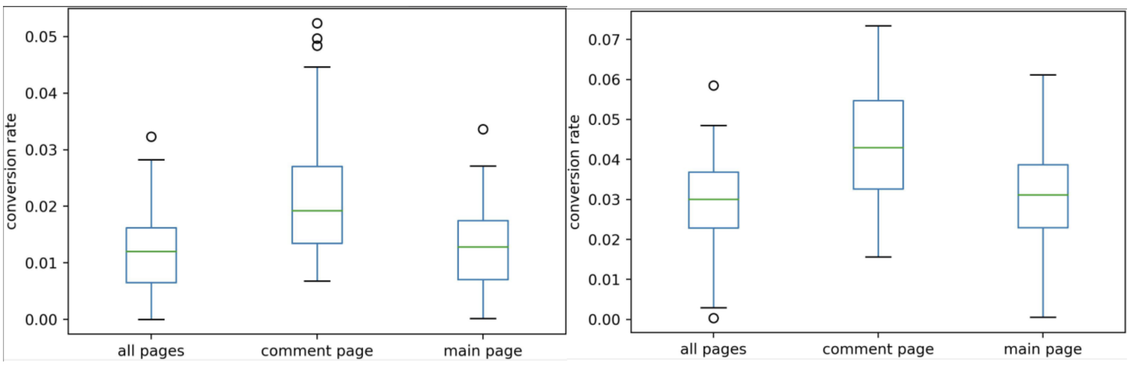


Fig. 1. Click Module Source vs. Conversion Rate



(a) Phones (b) Foods & Drinks

Fig. 2. Distribution of dwell time on both categories.

2.2. Analysis

In this section, we illustrate the effects of defined post-click behaviours. In particular, we examine why post-click behaviours are effective to measure user engagement and how post-click behaviours vary on different product categories. In doing this, we choose two representative product categories “Food & Drinks” and “Mobile Phones” as examples, and collect users’ post-click behaviours from JD.com, an E-commerce platform in China for analysis.

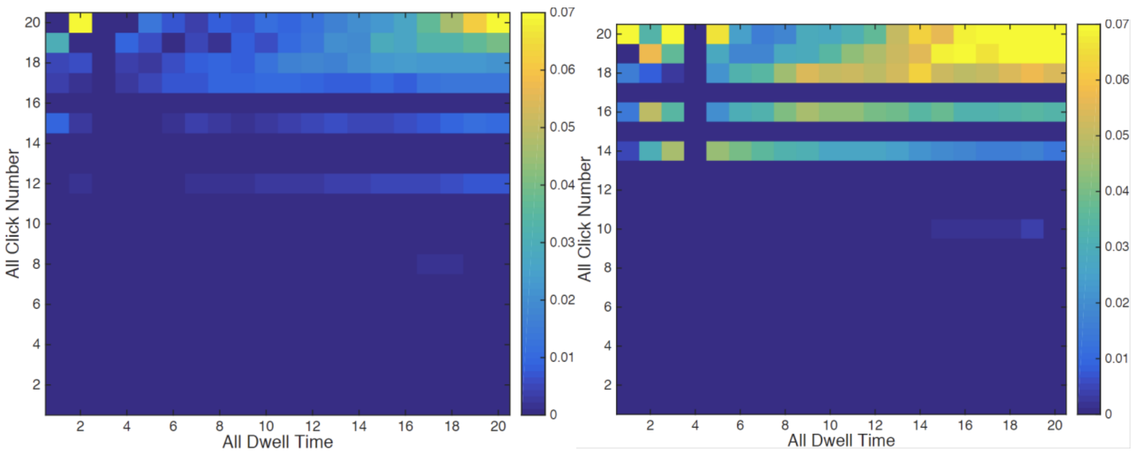


Fig. 3. All_Dwell_Time and All_Click with respect to conversion rate on mobile phones (left) and food& drinks (right) where pixel colour presents the conversion rate value.

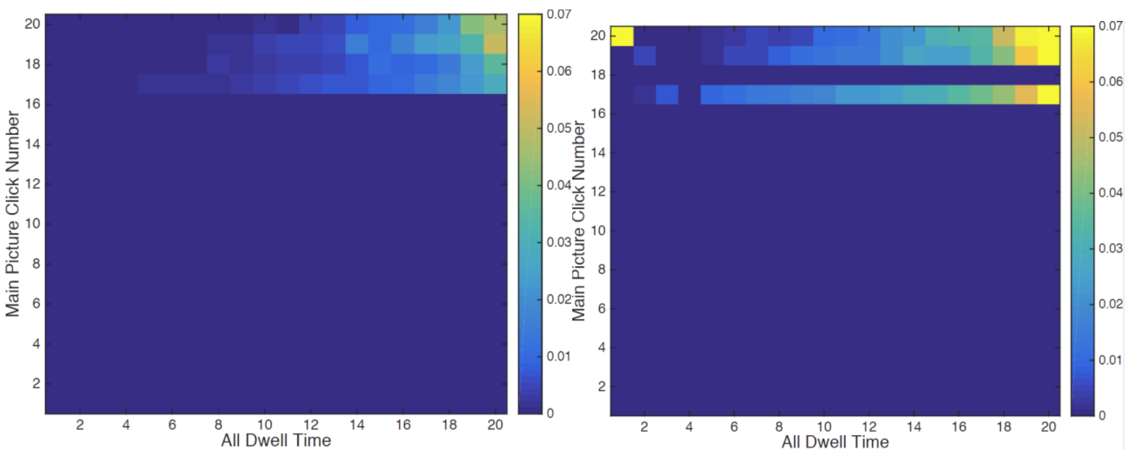


Fig.4. All_Dwell_Time and Main_pic_Click with respect to conversion rate on mobile phones (left) and food & drinks (right) where pixel colour presents the conversion rate value.

First, Figure 1 illustrates that the patterns of post-click behaviours vary considerably on different product categories. For example, the conversion rate of food and drinks is much higher than that of mobile phones. It results from the fact that most items in this category are fast moving consumer goods with lower price and shorter purchasing cycle. In contrast, mobile phones are much expensive, and therefore a user should be more cautious and spends more browsing time to make an order decision. Figure 2 report the distribution of dwell time with respect to conversion rate on each module respectively. The obvious difference in conversion rate further confirms that user post-click behaviours vary significantly on these two categories.

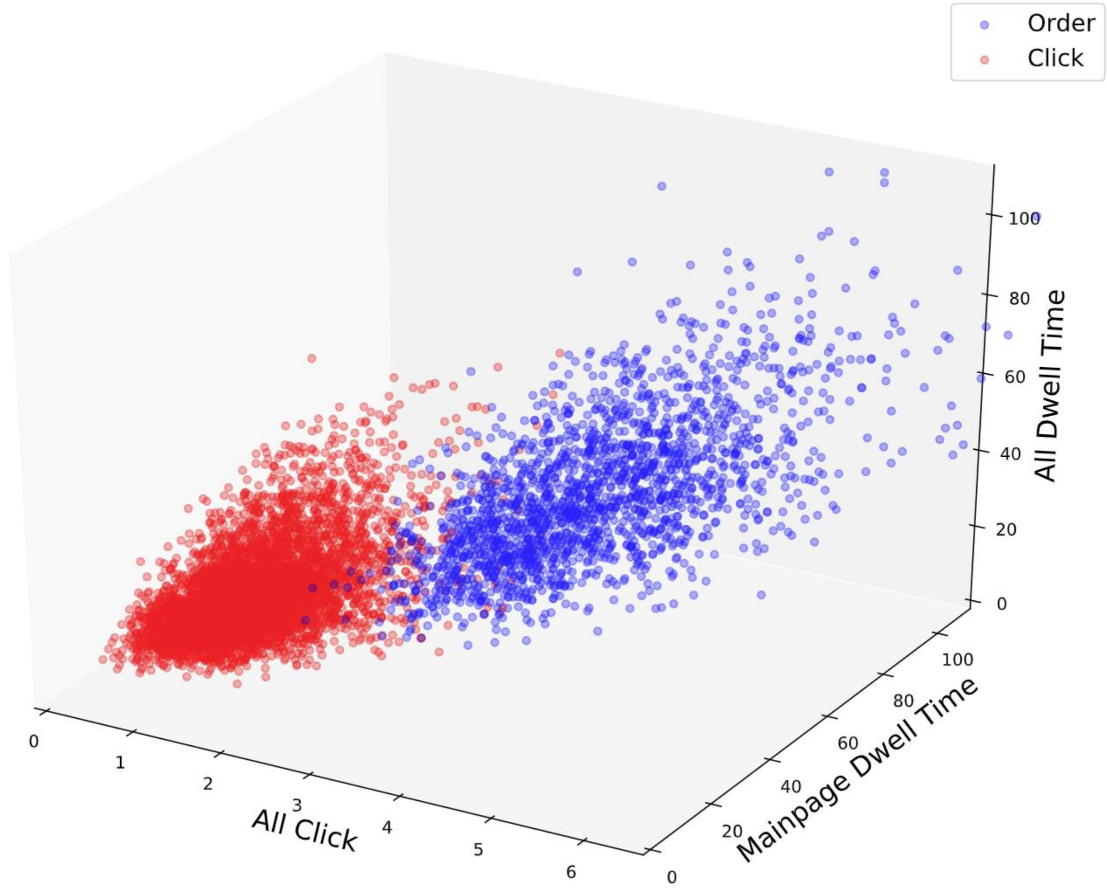


Fig. 5. User Ordering vs. User Engagement from three perspective (All Click, Mainpage Dwell Time and All Dwell Time) on Food&Drinks. Each point indicates the average performance from 10 random click or order samples.

Second, Figure 1 and Figure 2 also show how browsing module source relate to conversion rate. The results show that "Comment" and "Picture" modules reveal a higher conversion rate that "Main Page" on both categories. It means that if a user reads the comment page or views additional product pictures, she is more likely to make an order. But if the user only clicks the product main page (the default browsing module) and then quickly leaves, she is much less likely to buy the product eventually.

Last, post-click behaviours are related to user ordering and user engagement. Figure 3 demonstrates post-click behaviours (all dwell time and all number of clicks) with respect to conversion rate in individual bins. It is not surprising that the top-right corner is much brighter than the bottom-left in both figures. It means that the conversion rate increases as the number of click and overall dwell time increase. Another interesting discovery is that, some pixels in the top-left corner are still very bright, which indicates that users may click many times within a very short time before making orders. It furthermore supports our previous claim that one-dimensional post-click behaviours (overall dwell time) in pioneer studiers could not fully describe user engagement. Figure 4 reports similar findings using all dwell time and the number of main picture clicks. Figure 5 shows how ordering relates to user engagement. It is clear that a user is more likely to present stronger user engagement (making more clicks and spending more time) within the item page is she makes an order, indicating that ordering is a strong signal for user post-click engagement.

To sum up, we make the following revealing observations about post-click behaviours: **1)** post-click behaviours vary considerably on different product categories and different browsing modules; **2)** post-click behaviours are related to user ordering and user engagement; **3)** one-dimensional post-click behaviour (overall dwell time) could not fully describe post-click user engagement. These interesting observations demonstrate the challenges of modelling post-click behaviours, but they illustrate the potential of performance gains by leveraging post-click behaviours in recommender systems.

3. PROBLEM FORMULATION

In this work, our target is to exploit post-click behaviours in recommender systems. To better understand the specific problem studied here, we define two concepts, post-click instance and post-click labelling as follows.

Definition 1 (Post-click Instance). *Let P be a set of items, U be a set of users, and $\mathcal{A}$ be as set of post-click behaviours. A post-click instance $I = \langle u, p, A \rangle$ can be defined as a tuple with a set of post-click interactions $A \subseteq \mathcal{A}$ between a user $u \in U$and a product $p \in P$.*

For example, Denis clicked an iPhone X on an E-commerce website. After that, he stayed in the item page for 5 minutes, clicked 3 pictures and clicked 10 comments. Then this post-clicked instance can be presented as $\langle Denis, iPhone\ X, \{All\_Dwell\_Time = 300, Main\_Pic\_Click = 3, Comment\_Page\_Click = 10\}\rangle$.

Table 2. Statistics of the ranking dataset

Datasets	Training	Test_Click	Test_Order
Negative Samples	3,612,914	983,554	93,930
Clicks	247,915	60,779	N.A.
Orders	12,068	N.A.	2,416

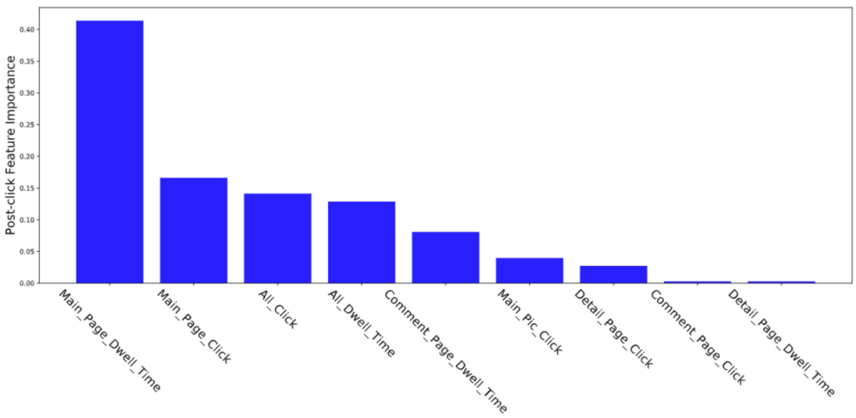


Fig.6. Post-click behaviour feature importance.

To keep consistence, we represent non-click instances as $I_{\emptyset} = \langle u, p, \emptyset \rangle$, indicating the user did not click the item.

Definition 2 (Post-click Labelling). *Let $\mathcal{I}_i$ be a post-click instance. The post-click labelling problem aims to find a mapping strategy $f: \mathcal{I}_i \rightarrow r_i \in \mathbb{R}_{\geq 0}$ to measure user post-click engagement and interest.*

With these two definitions, the specific problem we study can be stated as: *Given a set of post-click instances $\langle u_i, p_j, A_k \rangle$, we aim to leverage post-click instances to optimize user engagement in the existing recommendation system.*

4. EXPERIMENT

In this section, we evaluate our method on a real-world dataset from JD.COM to demonstrated the effectiveness of our proposed method in both offline and online tests.

4.1. Experimental Settings

We collect two data sets in our experiments, namely *post-click dataset* and *ranking dataset* respectively.

The post-click dataset consists of users' post-click instances, which is used to build the labelling model. This dataset was collected from a recommendation bucket of JD.com, spanning one month period. We removed product categories with less than $10K$ instances, which are insufficient for label modelling. By doing this, we have 104,498,003 clicks and 978,673 orders in total. Note that we do not make prediction on orders, so the test set is not needed here.

The ranking dataset consists of historical recommendation impressions, which is exploited to train the ranking model. It was sampled from the same recommendation bucket of JD.com within 6 days. Then we took the first 5 as the training data and the last one for click prediction, referred to as Test_Click. Furthermore, we removed all groups of user visits with no ordering from Test_Click to construct Test_Order dataset for order prediction. The ranking dataset is shown in Table 2.

We experimentally evaluated the following methods for a comparative study:

- **Order Model** uses binary labels as learning targets, where 1 is order and 0 is not.
- **Click model** uses binary labels as learning targets, where 1 is click and 0 is not.
- **All Click as Weight** uses the numbers of All_Click as learning weights for all click instances.
- **All Click as Target** users the numbers of All_Click as learning targets for all click instances.
- **All Dwell Time as Weight** uses All_Dwell_Time values as learning weights for all click instances. Following [10], the weight $w = \log(t)$ for a click instance's dwell time t
- **All Dwell Time as Target** uses All_Dwell_Time values as learning targets for all click instances.
- **Post-click as Target** uses the post-click labels as learning targets for all click instances.
- **Post-click as Weight (LR), Post-click as Target (LR)** uses the post-click label learned from logistic regression (LR).
- **Post-click as Weight (MLP), Post-click as Target (MLP)** uses the post-click label learned from multi-layer perceptron (MLP).

4.2. Experimental Results

We implemented random forest for post-click labelling with the scikit-learn python package [6], and GBDT model for ranking with XGBoost [2] in our experiments. All experiments were run on a Tesla K20 GPU machine.

Table 3. Click and Order Prediction

Data Type	Click Prediction	*precision@2*	*precision@4*	*precision@6*	*precision@8*
Click	Click Model	0.1577	0.1440	0.1368	0.1322
Order	Order Model	0.1395	0.1298	0.1249	0.1221
5*Post-click	All Click as Weight	**0.1594**	0.1453	0.1373	**0.1328**
	All Click as Target	0.1578	0.1431	0.1363	0.1319
	All Dwell Time as Weight	0.1592	0.1453	**0.1375**	0.1326
	All Dwell Time as Target	0.1583	0.1441	0.1370	0.1319
5*Post-click	Post-click as Weight (LR)	0.1590	0.1453	0.1368	0.1325
	Post-click as Target (LR)	0.1579	0.1442	0.1362	0.1316
	Post-click as Weight (MLP)	0.1533	0.1416	0.1316	0.1297
	Post-click as Target (MLP)	0.1540	0.1428	0.1323	0.1312
	Post-click as Weight	**0.1594**	**0.1454**	0.1371	**0.1328**
	Post-Click as Target	0.1581	0.1445	0.1366	0.1317
Data Type	**Order Prediction**	*precision@2*	*precision@4*	*precision@6*	*precision@8*
Click	Click Model	0.1433	0.1293	0.1236	0.1209
Order	Order Model	**0.1642**	**0.1404**	**0.1313**	**0.1270**
5*Post-click	All Click as Weight	0.1460	0.1286	0.1232	0.1203
	All Click as Target	0.1446	0.1287	0.1234	0.1200
	All Dwell Time as Weight	0.1430	0.1299	0.1248	0.1211
	All Dwell Time as Target	0.1461	0.1297	0.1246	0.1209
5*Post-click	Post-click as Weight (LR)	0.1482	0.1315	0.1249	0.1214
	Post-click as Target (LR)	0.1488	**0.1321**	0.1250	0.1216
	Post-click as Weight (MLP)	0.1431	0.1295	0.1246	0.1213
	Post-click as Target (MLP)	0.1473	0.1304	0.1243	0.1214
	Post-click as Weight	0.1490	0.1318	0.1250	**0.1219**
	Post-click as Target	**0.1501**	**0.1321**	**0.1251**	**0.1219**

Figure 6 demonstrates the feature importance of post-click behaviours in the tree-based labelling model on Food&Drinks post-click dataset. The result shows that, beyond one-dimensional post-click behaviour (overall dwell time), several other post-click behaviours are also significant, which points out the necessity of modelling multiple post-click behaviours to optimize user engagement, as discussed in section 2.

We calculate *precision@k*, which is the proportion of the targeted items among the top k predictions on both *Test_Click* and *Test_Order* respectively. For each group of user impression, precision at top-k positions can be defined as follows.

$$precision@k = \frac{\text{\# positive instances in top k results}}{\text{k}}$$

We calculate the average performance over all groups for $k = 2,4,6,8$ in our experiments. The click and order ranking results are shown in Table 3.

These results reveal several insights: 1) The click model presents good performance in click predictions but poor results in order ranking. This is not surprising as it ignores ordering information during the training procedure. With the same reason, the order model achieves best performance in order prediction but obtains the worst result in click prediction. Actually, it tends to recommend cheap products with higher conversions, leading to degraded performance in CTR and revenue. We will illustrate this point in the following case study and online test. 2) Using individual post-click behaviour as user feedback, such dwell time, also outperforms two baselines in click prediction, as reported in [10]. However, its performance in the order prediction task becomes degenerate. Here, we argue that just optimizing one dimensional post-click behaviour, either dwell time or click number, does not necessarily deliver much benefit on user conversion. Whereas, by leveraging ordering information, out method achieves improvement in order prediction. 3) In general, either as the target or as the weight, the post-click model with random forest can outperform the click model and the order model in click ranking. It suggests that exploiting post-click behaviours brings additional insights about user engagement to enhance recommendation. Tree-based post-click models also outperform all other methods on order ranking except the order model. Since train and test datasets are sampled homogeneously and order data is extremely sparse; as a result, the over-fitting probably leads to a fact that the order model could perform the best in the order ranking task. However, when doing online test, we actually obtain reverse results, due to more generalization power of the post-click models. 4) When comparing different labelling methods for post-click behaviours, we find that random forest performs better than logistic regression, indicating that the ensemble model leads to better relevance for user engagement than a linear model. Additionally, the MLP model does not perform well. This may be caused by the lack of sufficient positive order data to train the labelling model. For some product categories, the click/order ratio is substantially high, and few user ordering cannot generate sufficient positive samples to fit an effective MLP.

Besides user clicks and conversions, we further test out methods on overall user engagement in comparison with two baselines. The overall user engagement can be generally defined as all effective interactions between user and the recommender system, which is approximately measured by All_Click and All_Dwell_Time in our problem. To measure engagement in a unified metric, we define *CDF@k*, the average empirical cumulative distribution function (CDF), of total user engagement r_k at top k predictions. The empirical CDF is a non-parametric estimator of CDF of a random variable where it assigns a probability of $1/n$ to each instance and calculates the sum of the assigned probabilities up to and including each variable value. Given a set of values $X = \{x_1, x_2, x_3, \ldots, x_n\}$ for the overall user engagement variable x, the average empirical CDF at top-*k* position *CDF@k* is defined as:

$$CDF@k = \frac{\sum_{i=1}^{n} I(x_i \leq r_k)}{n}$$

where $I(\cdot)$ is an indicator function that returns 1 if the given condition is satisfied and 0 otherwise. The results on *CDF@k* are reported in Table 4.

It is observed that the best performance is achieved by either post-click as target or post-click as weight in each user engagement term. Highly similar performance can be also obtained when we test on other post-click behaviours not only increase user clicks or conversions, but also brings more user engagement where users tend to spend more time and perform more clicks after an item-level click. Leveraging user post-click behaviours, out method brings fine-grained and deep understanding about user engagement beyond CTR or overall dwell time.

Table 4. User Engagement Evaluation

All_Click	*CDF@2*	*CDF@4*	*CDF@6*	*CDF@8*
Order Model	0.207	0.345	0.457	0.554
Click_Model	0.234	0.388	0.512	0.617

Post-click as Target	0.235	0.391	0.513	0.614
Post-click as Weight	**0.238**	**0.395**	**0.515**	**0.621**
All_Dwell_Time	*CDF@2*	*CDF@4*	*CDF@6*	*CDF@8*
Order Model	0.268	0.451	0.599	0.727
Click Model	0.303	0.507	0.668	0.805
Post-click as Target	0.304	0.509	0.670	0.802
Post-click as Weight	**0.306**	**0.513**	**0.671**	**0.808**

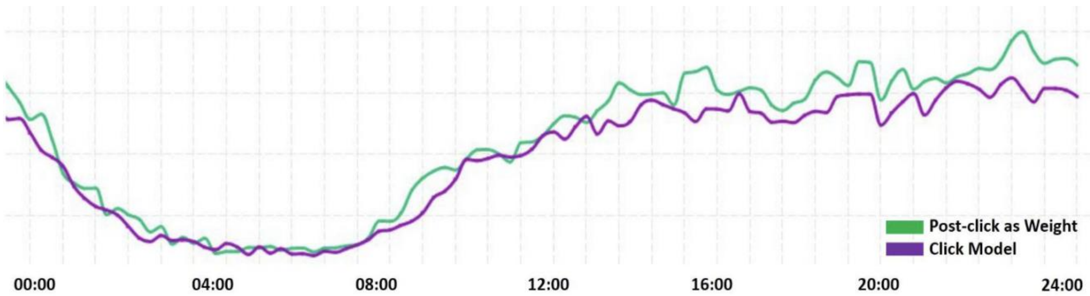


Fig.7. The relative performance comparisons on real-time click.

4.3. Online Test

We continue to validate empirical findings via online recommendation scenarios at JD.com. Without disclosing sensitive numbers, Figure 7 shows a one-day A/B test on the real-time user click as a representative example, and it is clear that the post-click model consistently outperforms the click baseline model.

Table.5. Relative performance on number of click, order and GMV per visiting user on online buckets.

Metric	Order Model	Click Model	Post-click as Weight
Click	-42.6%	1.0	+15.5%
Order	+20.9%	1.0	+22.6%
GMV	-11.3%	1.0	+27.7%

We also use three buckets to compare post-click as weight to two baselines with respect to three metrics, which are the number of click, order and Gross Merchandise Volume (GMV) per visiting user respectively. Table 5 reports relative performance where click model result is set to be 1.0.

It is revealed that the post-click model achieves best performance in all three aspects, indicating that our method brings better recommendations. It is noted that the post-click model even outperforms the order model on order-oriented metric evaluation. The reason is that the order model cannot attract more users to bring sufficient number of clicks, though its conversion rate could be very high.

5. Conclusion

In this paper, we have proposed a novel MDP based method – Div-FMCTS – to directly maximize the trade-off between accuracy and diversity for the top-N recommendation. The learning the of Div-FMCTS is decomposed into iteration between searching with MCTS and generalizing those plans with a policy-value neural network. To faster the tree search process, a novel structure pruning technique has been incorporated into the node expansion to narrow down searching space for MCTS. Additionally, we theoretically and empirically verify that the diversity recommendation can be equivalently solved by planning under the sub-problems. Extensive experiments on four benchmark datasets have demonstrated the effectiveness of our proposed method.

REFERENCES

[1] Veronika Bogina and Tsvi Kuflik. Incorporating dwell time in session-based recommendations with recurrent neural networks. In *CEUR Workshop Proceddings*, volume 1922, pages 57-59, 2017.

[2] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, pages 785-794, 2016.

[3] Yi Fang and Luo Si. A latent pairwise preference learning approach for recommendation from implicit feedback. In *CIKM*, pages 2567-2570, 2012.

[4] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76-80, 2003.

[5] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics reports*, 519(1):1-49, 2012

[6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825-2830, 2011.

[7] Steffen Rendle, Christoph Freudenthalear, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452-461. AUAI Press, 2009

[8] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *EC*, pages 158-167, 2000.

[9] Ke Sun, Tieyun Qian, Hongzhi Yin, Tong Chen, Yiqi Chen, and Ling Chen. What can history tell us? In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1593-1602, 2019.

[10] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. Beyond clicks: dwell time for personalization. In *RecSys*, pages 113-120, 2014.

[11] Peifeng Yin, Ping Luo, Wang-Chien Lee, and Min Wang. Silence is also evidence: interpreting dwell time for recommendation from psychological perspective. In *WSDM*, pages 989-997, 2013.

[12] Runlong Yu, Yunzhou Zhang, Yuyang Ye, Le Wu, Chao Wang, Qi Liu, and Enhong Chen. Multiple pairwise ranking with implicit feedback. In *Proceedings of the 27th ACM International conference on Information and Knowledge Management*, pages 1727-1730, 2018.

AUTHORS

Haibin Zhai is a senior engineer at CNCERT/CC. His main research areas include distributed network, big data.