Item Recommendation on Monotonic Behavior Chains

Mengting Wan
University of California, San Diego
m5wan@ucsd.edu

Julian McAuley
University of California, San Diego
jmcauley@ucsd.edu

ABSTRACT

'Explicit' and 'implicit' feedback in recommender systems have been studied for many years, as two relatively isolated areas. However many real-world systems involve a spectrum of both implicit and explicit signals, ranging from clicks and purchases, to ratings and reviews. A natural question is whether implicit signals (which are dense but noisy) might help to predict explicit signals (which are sparse but reliable), or *vice versa*. Thus in this paper, we propose an item recommendation framework which jointly models this full spectrum of interactions. Our main observation is that in many settings, feedback signals exhibit monotonic dependency structures, i.e., any signal necessarily implies the presence of a weaker (or more implicit) signal (a 'review' action implies a 'purchase' action, which implies a 'click' action, etc.). We refer to these structures as 'monotonic behavior chains,' for which we develop new algorithms that exploit these dependencies. Using several new and existing datasets that exhibit a variety of feedback types, we demonstrate the quantitative performance of our approaches. We also perform qualitative analysis to uncover the relationships between different stages of implicit vs. explicit signals.

ACM Reference Format:
Mengting Wan and Julian McAuley. 2018. Item Recommendation on Monotonic Behavior Chains. In *Twelfth ACM Conference on Recommender Systems (RecSys '18), October 2–7, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240323.3240369

1 INTRODUCTION

User feedback in recommender systems is usually classified into two categories: 'explicit' feedback—where users directly express their preferences (e.g. ratings), and 'implicit' feedback—where users indirectly reveal their interests through actions (e.g. clicks). These two paradigms have long been studied as two separate topics and different techniques have been developed to address each of their distinct properties.

Beyond the narrow definitions of explicit versus implicit feedback, we notice that multiple types of user feedback are abundant in many real-world information systems. For example, users' views, clicks, purchases and rating scores are commonly available on e-commerce platforms. All of these signals reflect (or imply) users' interests regarding items from different perspectives. Although

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240369

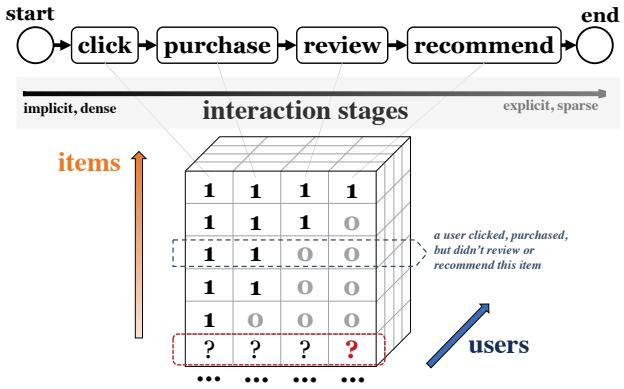


Figure 1: Illustration of monotonic behavior chains and the associated item recommendation problems.

there has been a line of work where the connections between implicit and explicit interactions are considered [8, 9, 12, 15, 18–20], most focus on improving numeric rating predictions by leveraging other signals as auxiliary information. These studies motivate us to bridge the gap between implicit and explicit signals, but our primary goal is to build a unified recommendation framework for a more general purpose, where several types of user feedback can be simultaneously considered regardless of their specific semantics. Specifically we are interested in (1) how to properly represent a spectrum of users' responses; and (2) how to efficiently harness the connections among these interactions and provide personalized item recommendations.

User-Item Interactions as Behavior Chains. We typically observe relatively a few explicit responses (such as numeric rating scores), but abundant implicit feedback such as 'click' and 'purchase' actions. Although these interactions are heterogeneous in terms of both representations and data distributions, they can be aligned on what we refer to as 'behavior chains.' As shown in Figure 1, different user-item interactions in e-commerce systems can be encoded as binary states on a chain, which semantically represents if a user clicks, purchases, reviews or recommends, (e.g. a rating score larger than some threshold) the item. Specifically, the highlighted vector $(1, 1, 0, 0)$ in Figure 1 encodes that a user clicked, purchased, but didn't review or recommend a product. By following the links on these chains, users gradually 'activate' interaction stages which increasingly imply more explicit preferences toward items. Such representations provide us with not only a template to unify different types of interactions but also a prototype to simulate users' decision-making processes.

Monotonicity on Behavior Chains. One notable property of such behavior chains is their *monotonicity*. That is, once a user decides to 'stop' at a stage, then the subsequent interactions by definition will not be observed. For instance, we cannot observe a user's

'review' or 'recommend' actions if the item was not purchased by this user. By properly leveraging these monotonicity constraints, we hope to distinguish critical versus nonessential information. For instance, 'not review' and 'not recommend' are nonessential (i.e., already implied) given that the item was not purchased by the user. Because of this structure, for each user we can define a binary matrix on these behavior chains, which starts with *the most implicit* (and *densest*) responses, and ends with *the sparsest* but *the most explicit* responses. In each row of this matrix, elements are monotonically non-increasing from left to right.

Recommendation on Monotonic Behavior Chains. Based on the above representations, we can describe our primary goal in this paper as follows:

> **Goal:** *Given historical observations of users' behavior chains, we seek to estimate their responses toward unobserved items by appropriately leveraging the monotonicity assumption implied by the data.*

Note for each interaction stage on a behavior chain, we are able to define an associated one-class recommendation problem. We regard the recommendation performance on the most explicit (i.e., the last) stage as our primary evaluation criterion but also investigate the performance with respect to the full interaction spectrum.

Contributions. We describe our contributions as follows.

- We observe a common scenario where multiple types of user-item interactions can be aligned on a monotonic behavior chain, and propose a unified item recommendation problem based on this representation.
- We propose a new algorithm—**chainRec** which effectively models multiple types of interactions and efficiently exploits the monotonicity among these actions. In particular, we design a scoring function on top of users' behavioral intentions in order to make use of all types of responses, explicitly model users' target intents, and preserve the monotonic constraints in the resulting user preference scores. We also develop a new optimization criterion which takes advantage of the monotonicity and automatically focuses on the most critical information in users' feedback data.
- We evaluate the model on five different real-world datasets where our experiments indicate the proposed algorithm substantially outperforms baselines.
- We contribute a new large-scale dataset (of book reviews) for this problem. It contains information from more than 200 million user-item interactions and covers four different interaction types (shelve – read – rate – recommend).

2 RELATED WORK

Traditional item recommendation systems often rely on a suite of collaborative filtering techniques to learn from explicit feedback such as rating scores. Typical model-based techniques include matrix factorization (**MF**) methods [13], which seek to learn item and user embeddings and use the inner product to approximate observed ratings. As providing explicit feedback often requires additional cognitive effort [2], these interactions may be *sparse* or unavailable in real-world scenarios. In such cases, the above **MF** methods can be extended to model the more abundant implicit signals that are disclosed via users' observable actions such as clicks and purchases [7, 17, 22]. In order to address the one-class property of this setting (i.e., only positive instances can be observed), several approaches including the pairwise ranking method **BPR** [22] and the pointwise optimization method **WRMF** [7, 17] have been proposed.

Although explicit and implicit data are commonly studied as two separate topics, there are several studies that seek to connect these signals [8, 15, 18–20]. These methods are summarized in a recent survey [9]. Specifically in the music recommendation domain, a positive correlation between implicit (e.g. play counts) and explicit feedback (e.g. ratings) has been found; regression models thus can be built to predict users' rating scores from implicit signals [19, 20]. In addition, a factorized neighborhood model was proposed to directly estimate users' ratings where implicit signals are used to locate and regularize the neighborhood items [12]. Furthermore, several methods have been proposed to jointly factorize users' rating scores and implicit responses with shared user and item latent factors [15, 18]. Unlike these methods which specifically handle numeric (i.e., 'star') ratings and regard other feedback as side-information, we seek to build a framework where several types of (binary) user feedback can be aligned without giving special treatment to rating scores.

In a recent study [24], tensor decomposition techniques are applied to model users' different types of activities and negative sampling strategies are introduced to address the one-class problem. Another line of relevant work includes session-based recommendation and modeling 'micro-behavior' in each user session [3, 6, 14, 23, 25]. These methods differ from each other in precise details, but they typically focus on embedding micro-behaviors within each user session (e.g. views, clicks, dwell time) into a predictive framework (e.g. a recurrent neural network) to estimate the users' next actions. Although all of the above methods seek to fuse users' activities, neither of them explores the 'strength' of each type of signal or the potential monotonic dependencies among these activities.

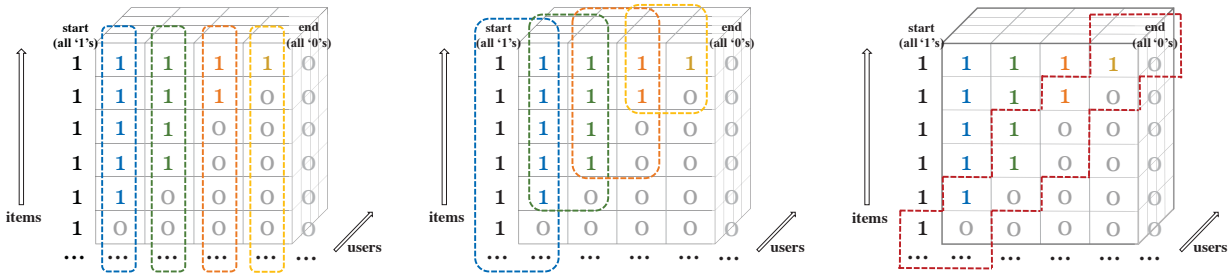
3 PROBLEM DEFINITION AND PRELIMINARY LEARNING STRATEGIES

In this section, we formally define monotonic behavior chains and investigate several preliminary learning strategies for the proposed item recommendation problem. Notation used throughout the paper is provided in Table 1.

Suppose for a user u and an item i, we have labels for a chain of user-item interactions $\boldsymbol{y}_{ui} = [y_{ui,1}, \ldots, y_{ui,L}]^T$, where L is the number of interaction stages and $\forall l = 1, \ldots, L$:

$$y_{ui,l} = \begin{cases} 1, & \text{if the interaction } (u,i) \text{ at stage } l \text{ is observed;} \\ 0, & \text{otherwise.} \end{cases}$$

That is if a user performs an action (e.g. click) on an item, it is regarded as a 'positive' instance; otherwise, as in traditional one-class settings, rather than simply treating the non-click as 'negative,' it could be that the user is simply not aware of the item, or the cost is too high (etc.). Typically we expect items interacted with by a user to be ranked higher than other items in the final recommendation. These behavior chains $\{\boldsymbol{y}_{ui}\}$ are said to be *monotonic* if $y_{ui,1} \geq$



(a) sliceOpt (b) condOpt (c) edgeOpt

Figure 2: Illustration of different optimization criteria.

Notation	Description
$\mathcal{U}, \mathcal{I}, \mathcal{I}_u^+$	user set, item set, items user u interacted with
$\mathcal{I}_{u,l}^+, \mathcal{I}_{u,l}^-$	items u interacted with at stage l; items u did not interact with at stage l
$y_{ui,l}$, $s_{ui,l}$	observation of user u's interaction on item i at stage l, u's preference score on item i at stage l
l_{ui}^*	$:= \arg\max_l \{y_{ui,l} = 1\}$, the last stage where the interaction (u, i) is observed
b_0, b_i, b_u	global bias, item bias, user bias
$\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_u, \boldsymbol{\gamma}_l$	item embedding vector, user embedding vector, stage embedding vector
$p_{ui,l}$	$:= P(y_{ui,l} = 1)$, marginal probability of user u interacting with item i at stage l
$p_{ui,l\|l-1}$	$:= P(y_{ui,l} = 1 \| y_{ui,l-1} = 1)$, conditional probability of u interacting with i at stage l given a positive observation at stage $l-1$.
$p_{ui,\cap}$	$:= \frac{P(y_{ui,l^*}=1, y_{ui,l^*+1}=0)}{P(y_{ui,l^*}=1)P(y_{ui,l^*+1}=0)}$, the (exponential of the) pointwise mutual information between two consecutive stages on the edge of the interaction chain for (u, i).

Table 1: Notation.

$y_{ui,2} \geq \ldots \geq y_{ui,L}, \forall u, i$. To simplify notation at boundaries, we assume two 'pseudo' stages $l = 0$ and $l = L + 1$, where we always have $y_{ui,0} = 1$ and $y_{ui,L+1} = 0, \forall u, i$.

Then recommendation problems on these monotonic behavior chains can be formulated as estimating the ranking scores of unobserved items (i.e., items that a user has never interacted with) at each stage, where the same underlying ranking mechanisms can be used to approximate the observed feedback $y_{ui,l}$.

Learning Preferences Independent of Stages. A naïve approach to solve our problem would be to ignore inter-stage dependencies and simply learn preference/ranking models for each stage independently. A representative objective function for stage l would be the *pointwise* binary cross-entropy (e.g. **LogisticMF** [10], **NeuMF** [5]):

$$-\sum_{u,i}\Big(y_{ui,l}\log\sigma(s_{ui,l}) + c_{ui,l}(1-y_{ui,l})\log(1-\sigma(s_{ui,l}))\Big) \quad (1)$$

where $s_{ui,l}$ is u's preference ranking score regarding item i, $\sigma(\cdot)$ is the sigmoid function and $c_{ui,l}$ is a customized weight to balance positive and negative observations. In practice, c_{ui} can be implemented through sampling techniques during training [5, 17]. That is for each positive (u, i) pair in $\mathcal{I}_{u,l}^+ = \{i | y_{ui,l} = 1\}$, we can sample N items with which the user did not interact at stage l, compose a 'balanced' negative itemset $\mathcal{I}_{u,l}^-$, and update the binary cross-entropy loss function:

$$-\sum_u\Big(\sum_{i\in\mathcal{I}_{u,l}^+}\log\sigma(s_{ui,l}) + \sum_{i'\in\mathcal{I}_{u,l}^-}\log(1-\sigma(s_{ui',l}))\Big). \quad (2)$$

Another popular objective function is the *pairwise* ranking loss (e.g. **BPR** [22]):

$$-\sum_{u,i\in\mathcal{I}_{u,l}^+,i'\in\mathcal{I}_{u,l}^-}\log\sigma(s_{ui,l} - s_{ui',l}), \quad (3)$$

which seeks to maximize a pairwise difference between observed positive and unobserved 'negative' instances.

Here independent parameters are applied to model the ranking scores $s_{ui,l}$ for different stages; one popular underlying approach is the latent factor model:

$$s_{ui,l} = b_{0,l} + b_{i,l} + b_{u,l} + \langle\boldsymbol{\gamma}_{i,l}, \boldsymbol{\gamma}_{u,l}\rangle, \quad (4)$$

where for each stage l, $b_{0,l}$ is the global offset, $b_{i,l}, b_{u,l}$ are item and user biases, and $\boldsymbol{\gamma}_{i,l}, \boldsymbol{\gamma}_{u,l}$ are K-dimensional embeddings to capture items' latent features and users' latent preferences toward these features. Here $\langle\cdot,\cdot\rangle$ denotes the inner product such that $\langle\boldsymbol{\gamma}_{i,l}, \boldsymbol{\gamma}_{u,l}\rangle$ captures the 'compatibility' between user u and item i on stage l.

Learning Preferences Jointly on Different Stages. Note that the above models ignore the underlying relationships among different interaction stages. Given that all of these interactions ought to reflect users' preferences from different perspectives, we can extend the assumption applied in existing studies [15, 18] that better item and user representations could be learned by jointly modeling different types of interactions. This results in a joint objective function that extends Eq. (2):

$$-\sum_{u,l}\Big(\sum_{i\in\mathcal{I}_{u,l}^+}\log\sigma(s_{ui,l}) + \sum_{i'\in\mathcal{I}_{u,l}^-}\log(1-\sigma(s_{ui',l}))\Big). \quad (5)$$

Critically, the preference score $s_{ui,l}$ can be modeled by *shared* item and user embeddings, combined with a set of stage-specific weights $\boldsymbol{\gamma}_l$ on different latent dimensions, i.e.,

$$s_{ui,l} = b_0 + b_i + b_u + \langle\boldsymbol{\gamma}_l, \boldsymbol{\gamma}_i \circ \boldsymbol{\gamma}_u\rangle, \quad (6)$$

which is equivalent to a variant of the **CP/PARAFAC** tensor decomposition framework [1]. Here $\circ$ denotes the Hadamard product.

Note that together with the above learning strategies, optimization criteria of these approaches focus on the estimations within each vertical 'slice' (i.e., each stage l) of the observation matrices, but do not involve any horizontal connections among slices such as the monotonicity we discussed above. We thus refer them as 'slicewise' optimizations (**sliceOpt**, see Figure 2a).

Learning Preferences Conditioned on Previous Stages. We next aim to explore the 'monotonicity' property of these behavior chains. An obvious assumption would be that any observations of an interaction stage should be conditioned on the presence of the previous stage (e.g. a 'purchase' action should be conditioned on the presence of a 'click' action). Therefore instead of approximating the marginal probability of each stage directly, we seek to model the conditional probability of the behavior 'escalation' from a weaker to a stronger interaction. Specifically we consider the following conditional optimization criterion (**condOpt**, see Figure 2b):

$$-\sum_{u,l}\sum_{i\in\mathcal{I}^+_{u,l-1}}\Big(y_{ui,l}\log p_{ui,l|l-1}+c_{ui,l}(1-y_{ui,l})\log(1-p_{ui,l|l-1})\Big) \tag{7}$$

where $p_{ui,l|l-1}$ is shorthand for $P(y_{ui,l}=1|y_{ui,l-1}=1)$; similar sampling techniques as in Eq. (2) can be used. Suppose we have $p_{ui,l|l-1}=\sigma(\delta_{ui,l})$. Then $\delta_{ui,l}$ can be factorized using stage-independent item/user embeddings (Eq. (4)) or shared item/user embeddings (Eq. (6)). We use the joint probability

$$s_{ui,l}\coloneqq P(y_{ui,1}=\ldots=y_{ui,l}=1)=\prod_{l'=1}^{l}p_{ui,l'|l'-1}$$

as the preference ranking score for item recommendations at stage l. This preference score naturally inherits the monotonicity in the interaction labels, i.e., $s_{ui,1}\geq\ldots\geq s_{ui,L}$.

As shown in Figure 2b, this learning strategy gradually narrows its training scope by conditioning on previous observed interactions. As positive instances for explicit feedback are relatively scarce, following this conditional optimization criterion may lead to difficulty capturing behavior escalations that happen at later stages of behavior chains. On the other hand, this strategy can be regarded as being analogous to executing a sequence of 'AND' operators on the interaction chains. That is, in order to reach the most explicit stage, all of the behavior escalations have to be 'activated.' One potential drawback of this philosophy is that it tends to propagate failures to the subsequent stages. For example, it is difficult to interpret a user's relative preferences toward (say) an item that has been reviewed but not recommended versus an item that has been purchased but not reviewed. Combined with its limitation in handling data scarcity, the above learning strategy may have difficulty bypassing the noisy, implicit (and 'weakly' negative) data and thus fail to learn users' preferences accurately. Such an observation motivates us to develop new techniques to carefully model the internal logic behind the monotonicity of observed interactions.

4 THE PROPOSED ALGORITHM

Based on the above investigation, we develop a new algorithm—**chainRec**—that exhibits the following properties: (1) it takes advantage of all stages of interactions to learn item/user representations for preference ranking; (2) the ultimate preference scores generated from the model explicitly preserve the monotonicity of the interaction matrix; (3) replacing the above 'AND' philosophy, we try to understand which interactions directly come from users' *intrinsic* behavior intentions, versus which are *derived* from (stronger) subsequent intentions. We use two techniques to achieve these properties: monotonic preference scoring functions and an edgewise optimization criterion.

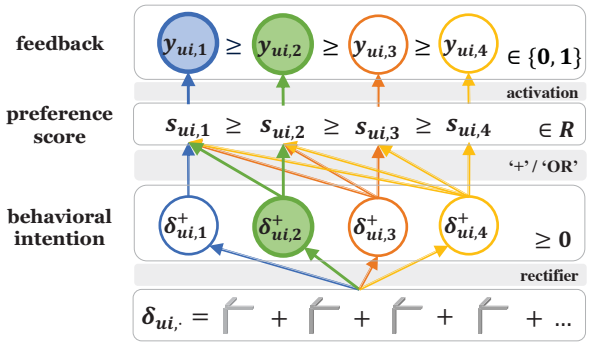


Figure 3: Illustration of our monotonic preference scoring function. In this example, only the behavioral intention $\delta^+_{ui,2}$ is activated. The observation $y_{ui,2}=1$ directly comes from its activated associated intention $\delta^+_{ui,2}$, while $y_{ui,1}=1$ is derived by its subsequent behavioral intention $\delta^+_{ui,2}$.

4.1 Monotonic Scoring Function

We still model the marginal probability $P(y_{ui,l})$ but from a different prospective:

$$p_{ui,l}\coloneqq P(y_{ui,l}=1)=\sigma(s_{ui,l})=\frac{1}{1+\exp(-s_{ui,l})}, \tag{8}$$

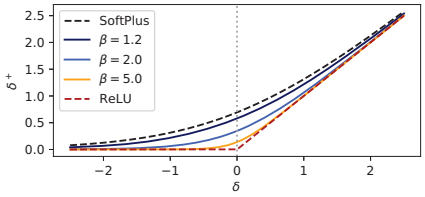
where $s_{ui,l}$ is the preference score. Instead of directly decomposing the response $y_{ui,l}$, we introduce an additional layer for users' behavioral intentions. Here we use a similar **CP/PARAFAC** tensor decomposition format as in Eq. (6) to factorize an intention score for each stage l:

$$\delta_{ui,l}=\left\langle\boldsymbol{\gamma}_l,\ \boldsymbol{\gamma}_i\circ\boldsymbol{\gamma}_u\right\rangle. \tag{9}$$

We then pass this intention score to a parametric rectifier such that a user's specific behavioral intention is activated if and only if this score exceeds zero:

$$\delta^+_{ui,l}=\frac{1}{\beta}\log(1+\exp(\beta\delta_{ui,l})). \tag{10}$$

This rectifier becomes a softplus function when $\beta=1$ and approximates a rectified linear unit (**ReLU**) as $\beta\to\infty$:



We assume $\beta\geq 1$ is a parameter which will be automatically learned during training.

On top of this layer, we assume that a user-item interaction stage can be observed if its associated *or* subsequent behavioral intentions are activated (as shown in Figure 3). We encode this soft logic in the final observations as follows:

$$s_{ui,l} = b_0 + b_i + b_u + \sum_{l'=l}^{L} \delta^+_{ui,l'}. \tag{11}$$

Note that the most explicit interaction can only be observed when its associated behavioral intention is activated. By modeling users' intentions in this format, both preference scores and the resulting probabilities preserve the monotonicity in observations $y_{ui,l}$, i.e., $s_{ui,l} - s_{ui,l+1} = \delta^+_{ui,l} \geq 0$.

4.2 Edgewise Optimization Criterion

In addition to this scoring function, we can encode the monotonicity of $y_{ui,l}$ into a probabilistic framework by enforcing these equivalent constraints

$$\begin{aligned}
&P(y_{ui,l'} = 1 | y_{ui,l} = 1) = 1, && \forall l' < l;\\
&P(y_{ui,l'} = 0 | y_{ui,l} = 0) = 1, && \forall l' > l;\\
&P(y_{ui,1} = 1, \ldots, y_{ui,l} = 1) = P(y_{ui,l} = 1), && \forall l;\\
&P(y_{ui,l+1} = 0, \ldots, y_{ui,L} = 0) = P(y_{ui,l+1} = 0), && \forall l.
\end{aligned} \tag{12}$$

These constraints help us prune the redundant information within the joint probability such that we can obtain the following reduced objective function on the 'edges' (i.e., two consecutive stages where users exhibit different responses) of users' behavior chains (**edgeOpt**):

$$\sum_{u,i} \log P(y_{ui,1}, \ldots, y_{ui,L}) = \sum_{u,i} \log P(y_{ui,l^*_{ui}} = 1, y_{ui,l^*_{ui}+1} = 0) \tag{13}$$

where $l^*_{ui} = \arg\max_l \{y_{ui,l} = 1\}$ is the last stage at which the interaction (u, i) can be observed.[1]

As shown in Figure 2, **edgeOpt** differs from **sliceOpt** and **condOpt** in that it focuses on the most critical signals—observations at 'edges' of the behavior chains. The reason **edgeOpt** allows us to do this is the previous and the subsequent interactions are already implied by the monotonicity and guaranteed by applying monotonic scoring functions.

Notice that we are still facing the one-class problem, which means we generally trust the positive interactions $y_{ui,l^*} = 1$ but are not confident in unobserved 'negative' instances $y_{ui,l^*+1} = 0$. Therefore, similar to the weighting techniques used in previous one-class collaborative filtering studies [7, 10, 17], we seek to separate information contained in these two consecutive stages from their joint probability and rebalance positive and negative instances. Specifically we have

$$P(y_{ui,l^*} = 1, y_{ui,l^*+1} = 0) = p_{ui,l^*}\,(1 - p_{ui,l^*+1})\,p_{ui,\cap}, \tag{14}$$

and $p_{ui,\cap}$ denotes the (exponential of the) pointwise mutual information (**PMI**) between two consecutive stages on the edge

$$p_{ui,\cap} \coloneqq \frac{P(y_{ui,l^*} = 1, y_{ui,l^*+1} = 0)}{P(y_{ui,l^*} = 1)\,P(y_{ui,l^*+1} = 0)}.$$

[1]For brevity we may omit the subscript ui and use l^* in subsequent paragraphs.

Algorithm 1 chainRec

for each user u, and each item $i \in \mathcal{I}^+_u$ **do**
  Locate the last positively interacted stage l^*_{ui}
  Update the associated parameters Θ_{ui} based on the gradients

$$\frac{\partial}{\partial \Theta_{ui}} \log p_{ui,l^*_{ui}}$$

  Sample N contrastive items based on the given sampling scheme
  for each contrastive item i' **do**
    Locate the last positively interacted stage $l^*_{ui'}$
    Update the associated parameters $\Theta_{ui'}$ based on the gradients

$$\frac{\partial}{\partial \Theta_{ui'}} \Big(\log \left(1 - p_{ui',l^*_{ui'}+1}\right) + \log p_{ui',\cap} \Big)$$

  end for
end for

By applying Eq. (12), Eq. (10) and Eq. (11), we have an explicit formula to calculate this information:

$$\begin{aligned}
p_{ui,\cap} =& \frac{p_{ui,l^*} - P(y_{ui,l^*} = 1, y_{ui,l^*+1} = 1)}{p_{ui,l^*}\,(1 - p_{ui,l^*+1})} = \frac{p_{ui,l^*} - p_{ui,l^*+1}}{p_{ui,l^*}\,(1 - p_{ui,l^*+1})}\\
=& 1 - \exp(-\delta^+_{ui,l^*}).
\end{aligned} \tag{15}$$

Then we obtain the following rebalanced objective:

$$\begin{aligned}
&\sum_u \Big(\sum_{i \in \mathcal{I}^+_u} \log p_{ui,l^*} + \sum_{i' \in \mathcal{I}} c_{ui',l^*} \big(\log \left(1 - p_{ui',l^*+1}\right) + \log p_{ui',\cap} \big) \Big)\\
\approx &\sum_u \Big(\sum_{i \in \mathcal{I}^+_u} \log p_{ui,l^*} + \sum_{i' \in \bar{\mathcal{I}}_u} \big(\log \left(1 - p_{ui',l^*+1}\right) + \log p_{ui',\cap} \big) \Big).
\end{aligned} \tag{16}$$

We propose the following two sampling schemes to compose the above contrastive item set $\bar{\mathcal{I}}_u$:

- **Uniform Sampling.** $c_{ui,l^*} \propto 1$. For each positive user-item pair (u, i) (i.e., $l_{ui}* > 0$), we uniformly sample N items regardless of their labels;
- **Stagewise Sampling.** $c_{ui,l^*} \propto \frac{|\mathcal{I}^+_{u,l^*+1}|}{|\mathcal{I}^+_{u,l^*}|}$. For each positive user-item pair (u, i), we sample N items based on their associated edge stage l^*_{ui}.

We briefly describe the proposed method **chainRec** in algorithm 1. Note that, in spite of the relatively complex derivation, the final algorithm is straightforward. We apply a standard ℓ_2 regularizer[2] on item and user embeddings $\boldsymbol{\gamma}_i, \boldsymbol{\gamma}_u$ and **ADAM** [11] for optimization. As our primary goal is to rank unobserved items based users' most explicit preferences, we track the cross-entropy loss for the last stage L on a held-out validation set and stop training once it no longer decreases. All results are reported on the test set and all hyperparameters are selected based on the performance on the validation set.

[2]The hyperparameter λ is selected from {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0}

Dataset	#item	#user	#interaction	distribution of interactions	#inter. /#item	#inter. /#user
Steam	8,696	24,110	2,447,847	purchase (100.0%), play (64.0%), review (2.2%), recommend (2.0%)	281.49	101.53
YooChoose	19,034	509,126	2,292,077	click (100.0%), purchase (45.7%)	120.42	4.50
Yelp	119,340	1,005,382	4,731,170	review (100.0%), recommend (71.1%)	39.64	4.71
GoogleLocal	539,767	3,063,444	5,968,216	review (100.0%), recommend (85.0%)	11.06	1.95
Goodreads	1,561,465	808,749	225,394,930	shelve (100.0%), read (49.1%), rate (45.9%), recommend (32.0%)	144.35	278.70

Table 2: Basic dataset statistics.

5 EXPERIMENTS

We evaluate **chainRec** and alternatives on five real-world datasets, where multiple types of user-item interactions are available. In particular we are interested in determining (1) whether recommendation performance for the sparsest (and thus 'most explicit') feedback can be improved by leveraging other types of interactions; (2) to what extent accounting for monotonicity can help with ranking performance; (3) whether recommendation performance on the dense (and implicit) stages can be improved by appropriately leveraging more explicit signals and monotonicity together.

5.1 Datasets

We consider four public datasets and contribute an additional large-scale dataset which contains various interaction types. These data cover different types of behavior chains and vary significantly in data sparsity.

- ***Steam*** [21]. This dataset covers a group of Australian users on the *Steam* video game distribution network[3] and was recently introduced for the task of bundle recommendation [21]. This dataset includes users' purchase information, play time of games they purchased, reviews, and thumbs-ups (i.e., 'recommended' or 'not recommended'). Based on these data we are able to build a 'purchase – play – review – recommend' chain for each user-item pair. Surprisingly, around 36% of purchased games are never played by users. Similarly the 'review' and 'recommend' actions are significantly sparser than the 'purchase' and 'play' actions (Table 2).
- ***YooChoose.***[4] This is a dataset provided by *YooChoose*[5] in the 2015 *RecSys Challenge*, which contains a series of click sessions and the purchase events that occurred in these sessions. Note that user IDs are not available in this dataset, thus we treat session IDs as user IDs and build 'click – purchase' chains for this data.
- ***Yelp.***[6] We use the Round 11 version of the *Yelp Challenge* data. We regard the reviews where rating scores are larger than 3 as 'recommend' actions and build 'review – recommend' behavior chains.
- ***GoogleLocal*** [4]. This dataset covers reviews about local businesses worldwide. It is a relatively sparse dataset, especially in terms of the number of interactions per user. We use the same criteria as the ***Yelp*** dataset to build 'review – recommend' chains.
- ***Goodreads.*** We introduce a new large-scale dataset from the book review website *Goodreads*.[7] This data contains 229,154,523 records collected from 876,145 users' public book shelves and covers 2,360,655 books (with detailed meta-data including authors, series, editions, publishers, numbers of pages, languages of book contents, similar books and top user-generated shelf names for these books). Each record contains information of a user's multiple interactions regarding an item, including date added to shelf, reading progress, rating score, and review text if available, thus making 'shelve – read – rate – recommend' chains available for our recommendation problem.

We apply the same preprocessing criteria for all five datasets: we discard users who have never reached the last stage of any behavior chain and items with fewer than 5 associated interactions in the system. Statistics and distributions of the above datasets after preprocessing are included in Table 2. For each dataset, we sample 100,000 interaction chains for validation and another 100,000 for testing. Within each of these two sets, each interaction chain corresponds to a different user. Data and code are available at https://github.com/MengtingWan/chainRec.

5.2 Comparison Methods and Evaluation Methodology

We consider three groups of methods for comparisons. We first consider methods where interaction stages are treated independently:

- **itemPop.** We count the observed interactions in the training set as preference scores for each stage. Thus items are ranked based on their popularity.
- **bprMF** [22]. Is a state-of-the-art pairwise ranking model for one-class recommendation. Independent latent factor models (Eq. (4)) are used for different interaction stages.
- **WRMF** [7, 17]. Is another line of models which optimizes the mean squared error between estimated preference scores and labels; additional weights are introduced to adjust unobserved interactions.
- **logMF** [10]. Similarly, independent latent factor models are applied here but the model optimizes a binary cross-entropy loss as in Eq. (2).

Next we consider alternative methods where the relationships among different interaction stages are involved.

- **condMF.** We adopt the conditional optimization criterion (Eq. (7)) and use independent latent factor models (Eq. (4)) to estimate the conditional probability $p_{ui,l|l-1}$.

[3] https://store.steampowered.com/
[4] http://2015.recsyschallenge.com/
[5] https://www.yoochoose.com/
[6] https://www.yelp.com/dataset/challenge
[7] https://www.goodreads.com/

- **condTF.** We apply the same optimization criterion but use tensor decomposition (Eq. (6)) to estimate the conditional probability.
- **sliceTF.** This is a combination of joint slicewise optimization (Eq. (5)) and tensor decomposition (Eq. (6)). Notice that this method can be regarded as extending the philosophy of existing work which jointly models different types of signals to our one-class problem on behavior chains [15, 18].
- **sliceTF (monotonic).** Uses the same learning strategy but replaces the original tensor decomposition by the monotonic scoring function (Eq. (11)).

Last we evaluate two implementations of the proposed algorithm—**chainRec (uniform)** and **chainRec (stagewise)**, where uniform sampling and stagewise sampling strategies are applied respectively.

By comparing methods from the first and the second groups, we evaluate whether incorporating multiple types of signals simultaneously can help with recommendation performance; by comparing the second and the last group of methods, we evaluate the effectiveness of the proposed techniques to leverage the special monotonic structure of these behavior chains.

We rank items based on the preference score $s_{ui,l}$ for each stage l, and consider the Area Under the ROC Curve (AUC) as an overall ranking metric, and Normalized Discounted Cumulative Gain (NDCG) as a top-biased evaluation metric. We fix the number of contrastive samples for each positive user-item pair to $N = 1$ in all methods where such procedures are involved (e.g. Eq. (16))

5.3 Quantitative Results

We report detailed results with embedding dimensionality set to $K = 16$. We later perform a parameter study to assess sensitivity with respect to this parameter.

We include results for the primary item recommendation task in Table 3, where performance is evaluated based on users' feedback at the last stage. From this table we notice that the proposed **chainRec** algorithm significantly outperforms other baselines on most datasets in all metrics, though it performs slightly worse than **sliceTF** on ***Goodreads***. One notable advantage of **sliceTF** is that its plain linear structure makes it straightforward to optimize. Another possible reason for its good performance on ***Goodreads*** could be that it is a relatively dense dataset such that we have sufficient observations to implicitly learn the monotonicity property without needing to explicitly enforce it via model design. We observe that the second group of methods generally outperforms the first group of methods, which indicates that incorporating a spectrum of signals do help to predict the most sparse but explicit feedback. Surprisingly **sliceTF** consistently outperforms **sliceTF (monotonic)**. One possible reason could be that monotonic scoring functions behave as a kind of regularization on parameters and excessively and redundantly enforcing them may harm performance. Compared with this, the success of **chainRec** particularly validates the effectiveness of the proposed edgewise training strategy. Note that the non-personalized method **itemPop** performs as a strong baseline on the ***Steam*** dataset, possibly because the collected users' reviews are biased towards popular games as they were more likely to be exposed on the platform.

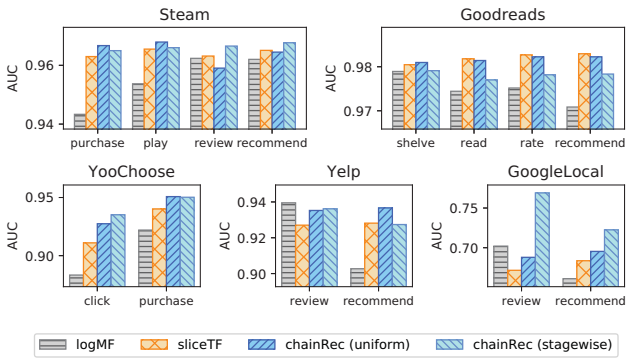


Figure 4: Results of item recommendation tasks on all stages in terms of AUC.

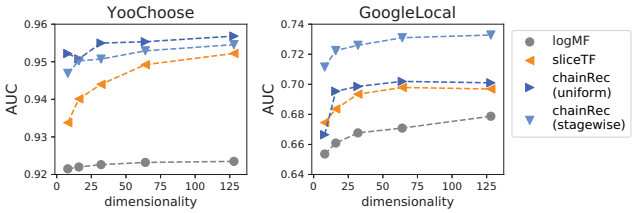


Figure 5: Sensitivity analysis w.r.t. dimensionality K on two datasets for the primary item recommendation task.

Next we evaluate item recommendation performance for each interaction stage separately. Results in terms of the AUC are included in Figure 4. For brevity we compare **chainRec** and two representative baselines, **logMF** and **sliceTF**, from the first and the second groups respectively.[8] Here we find that **chainRec** and **sliceTF** yield better recommendation results on nearly all stages compared to training stage-specific standalone models (i.e., **logMF**), which implies that sufficiently exploiting users' interaction chains and appropriately leveraging monotonicity can help us to predict users' preferences across all stages generally.

We also vary the embedding dimension $K \in \{8, 16, 32, 64, 128\}$ and report the item recommendation performance at the last stage on ***YooChoose*** and ***GoogleLocal*** datasets in Figure 5, where **chainRec** still dominates other baselines as we increase the dimensionality.

5.4 Qualitative Analysis

We apply the proposed **chainRec** algorithm on the ***Goodreads*** dataset and conduct case studies to explore the relationships among different stages.

We first seek to understand the stage-specific item embeddings learned from our model, by comparing these representations with the meta-data of items. We visualize the low-dimensional vectors of the normalized stage-specific item embeddings $\boldsymbol{\gamma}_i \circ \boldsymbol{\gamma}_l / ||\boldsymbol{\gamma}_l||$ in Figure 6 via t-SNE [16]. Here we categorize books into eight different genres: fantasy/paranormal, history/biography, romance, mystery/thriller, young adult, comics/graphic, children, and poetry. These genres are highlighted using different colors in Figure 6, from which we notice that these genre clusters are significantly

[8] Other baselines in general perform similarly to or weaker than these two methods.

Dataset	Metric	(a) itemPop	bprMF	WRMF	logMF	(b) condMF	condTF	sliceTF	sliceTF (m.)	(c) chainRec (uniform)	chainRec (stage.)	%impr. vs. (a)	%impr. vs. (b)
Steam	AUC	0.955	0.963	0.963	0.962	0.961	0.959	0.967	0.957	0.964	0.968	0.44%	0.06%
	NDCG	0.318	0.318	0.314	0.319	0.298	0.310	0.278	0.266	0.319	0.323	1.21%	4.23%
YooChoose	AUC	0.914	0.924	0.920	0.922	0.929	0.920	0.940	0.928	0.951	0.950	2.90%	1.13%
	NDCG	0.140	0.152	0.154	0.150	0.124	0.133	0.185	0.154	0.199	0.176	28.73%	7.09%
Yelp	AUC	0.838	0.921	0.912	0.903	0.900	0.838	0.928	0.918	0.937	0.927	1.71%	0.91%
	NDCG	0.093	0.105	0.096	0.100	0.090	0.088	0.107	0.096	0.108	0.102	3.05%	0.60%
GoogleLocal	AUC	0.597	0.661	0.625	0.661	0.679	0.616	0.684	0.667	0.695	0.722	9.31%	5.69%
	NDCG	0.064	0.067	0.064	0.066	0.064	0.063	0.070	0.065	0.072	0.072	8.36%	2.92%
Goodreads	AUC	0.938	0.971	0.963	0.971	0.904	0.933	0.984	0.934	0.982	0.978	1.17%	-0.17%
	NDCG	0.124	0.125	0.098	0.127	0.072	0.104	0.132	0.121	0.132	0.113	3.94%	0.00%

Table 3: Results of the primary item recommendation task, which is evaluated based on users' most explicit feedback. The best performance is underlined and the last two columns show the percentage improvement of chainRec over the strongest baseline within each group.

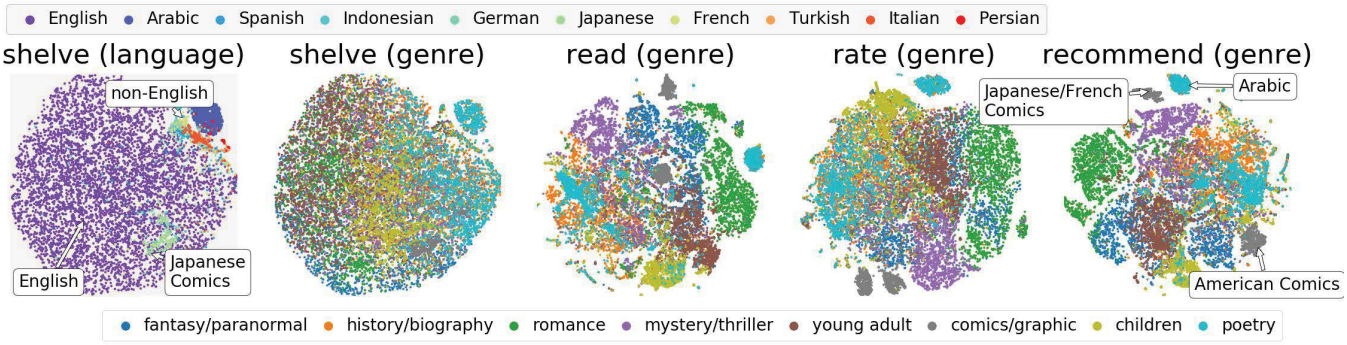


Figure 6: 2d t-SNE visualizations of item embeddings projected on different interaction stages (i.e., $\gamma_i \circ \bar{\gamma}_l$, where $\bar{\gamma}_l = \gamma_l / ||\gamma_l||$ is the normalized stage-specific scalar). Different languages and genres of books are highlighted using different colors.

visible for 'read' and 'recommend' actions but especially obscure for the 'shelve' action. This indicates that users tend to shelve books no matter what genres they belong to, but these features become important when users decide to read or recommend books. We then investigate the 'shelve' action by highlighting the languages of book contents in Figure 6. Here we see a clear separation between non-English books and English books, with the only exception being a group of Japanese comic books that are mixed with English books. This observation indicates that language plays an important role when users shelve books. We further investigate the review texts associated with these Japanese comic books and find that some English users might have mistakenly shelved Japanese editions despite the fact that what they read were English editions.

6 CONCLUSIONS AND FUTURE WORK

In this study, we proposed an item recommendation framework to model the full spectrum of users' feedback. We observe that users' interactions often exhibit monotonic structure, i.e., the presence of a stronger (or more explicit) interaction necessarily implies the presence of a weaker (or more implicit) signal. After investigating alternative models, we proposed a new recommendation algorithm—**chainRec** which exploits all types of interactions and efficiently harnesses their monotonic dependencies. We contribute a new public dataset and validate the effectiveness of **chainRec** by quantitative and qualitative results on new and existing datasets.

We note that the monotonicity structures studied in this work are widely observable and a number of topics can be further explored along this trajectory. Beyond recommendation tasks, such monotonic dependency structures and the associated predictive models can potentially be extended to other areas such as (e.g.) medical diagnosis where dependencies exist between progressive symptoms. This monotonic chain structure and the proposed algorithm can also be extended to more general tree structures, where different branches (e.g. 'click–bookmark' and 'click–purchase–recommend' in e-commerce systems) can be modeled simultaneously. Empirically, we only consider the binary representation of each interaction stage but counts of interactions (e.g. play counts of music tracks) could be incorporated as confidences for these binary observations. We also plan to investigate more advanced sampling schemes, and further analysis of the edgewise optimization strategy.

ACKNOWLEDGMENTS

We thank Wang-Cheng Kang, Jianmo Ni and Shuai Tang for thoughtful discussions.

REFERENCES

[1] R. Bro. Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 1997.
[2] S. C. Gadanho and N. Lhuillier. Addressing uncertainty in implicit preferences. In *RecSys*, 2007.
[3] T. Gurbanov and F. Ricci. Action prediction models for recommender systems based on collaborative filtering and sequence mining hybridization. In *Proceedings of the Symposium on Applied Computing*, 2017.
[4] R. He, W.-C. Kang, and J. McAuley. Translation-based recommendation. In *RecSys*, 2017.
[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *WWW*, 2017.
[6] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. In *ICLR*, 2015.
[7] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, 2008.
[8] G. Jawaheer, M. Szomszor, and P. Kostkova. Comparison of implicit and explicit feedback from an online music recommendation service. In *HetRec*, 2010.
[9] G. Jawaheer, P. Weller, and P. Kostkova. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems*, 2014.
[10] C. C. Johnson. Logistic matrix factorization for implicit feedback data. *NIPS*, 2014.
[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[12] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *TKDD*, 2010.
[13] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 2009.
[14] T. Lang and M. Rettenmeier. Understanding consumer behavior with recurrent neural networks. In *Workshop on Machine Learning Methods for Recommender Systems*, 2017.
[15] N. N. Liu, E. W. Xiang, M. Zhao, and Q. Yang. Unifying explicit and implicit feedback for collaborative filtering. In *CIKM*, 2010.
[16] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 2008.
[17] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *ICDM*, 2008.
[18] W. Pan, N. N. Liu, E. W. Xiang, and Q. Yang. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI*, 2011.
[19] D. Parra and X. Amatriain. Walk the talk: Analyzing the relation between implicit and explicit feedback for preference elicitation. In *UMAP*, 2011.
[20] D. Parra, A. Karatzoglou, X. Amatriain, and I. Yavuz. Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. In *CARS*, 2011.
[21] A. Pathak, K. Gupta, and J. McAuley. Generating and personalizing bundle recommendations on steam. In *SIGIR*, 2017.
[22] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.
[23] C. Wu and M. Yan. Session-aware information embedding for e-commerce product recommendation. In *CIKM*, 2017.
[24] H. Yin, H. Chen, X. Sun, H. Wang, Y. Wang, and Q. V. H. Nguyen. Sptf: A scalable probabilistic tensor factorization model for semantic-aware behavior prediction. In *ICDM*, 2017.
[25] M. Zhou, Z. Ding, J. Tang, and D. Yin. Micro behaviors: A new perspective in e-commerce recommender systems. In *WSDM*. ACM, 2018.