BRIDGING RECOMMENDATION AND MARKETING VIA RECURRENT INTENSITY MODELING

Yifei Ma, Ge Liu & Anoop Deoras
AWS AI Labs
{yifeim,gliua,adeoras}@amazon.com

ABSTRACT

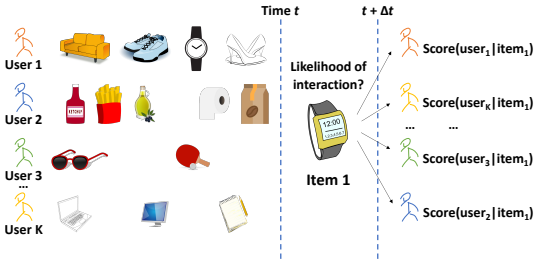
This paper studies some unexplored connections between personalized recommendation and marketing systems. Obviously, the two systems are different, in two main ways. Firstly, personalized item-recommendation (ItemRec) is user-centric, whereas marketing recommends the best user-state segments (UserRec) on behalf of its item providers. (We treat different temporal states of the same user as separate marketing opportunities.) To overcome this difference, we realize a novel connection to Marked-Temporal Point Processes (MTPPs), where we view both problems as different projections from a unified temporal intensity model for all user-item pairs. In this way, we derive Recurrent Intensity Models (RIMs) as unifying extensions from recurrent ItemRec models, though the connection can be more general. The second difference is in the temporal domains where they operate. While recommendation happens in real-time as each user appears, marketers often aim to reach a certain percentage of audience in the distribution of all user-states in a period of time. We formulate both considerations into a constrained optimization problem we call online match (OnlnMtch) and derive a Dual algorithm based on dual decomposition. Dual allows us to make ItemRec decisions in real time, while satisfying long-term marketing constraints in expectation. Finally, our connections between recommendation and marketing lead to novel applications. We run experiments where we use marketing as an alternative to cold-start item exploration, by setting a positive minimal-exposure constraint for every item over the user-state distribution in a future period of time. Our experiments are scalable to infinite streams of user-states and open-sourced.[1]

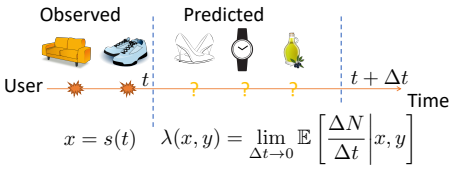
1 INTRODUCTION

Many ML applications today involve decision making based on streams of events. In recommender systems (RecSys), reasoning on the stream of events in a user's past history has allowed real-time user-based item recommendations (ItemRec), leading to significant impacts (Hidasi et al., 2016; Hidasi & Karatzoglou, 2018; Ma et al., 2020; Chen et al., 2019; Kang & McAuley, 2018). However, being user-centric, ItemRec has limitations in creating a globally inclusive environment for the item providers. For example, ItemRec tends to over-expose a few popular items to improve the immediate user satisfaction, yet this causes the selected items to get even more popular, discouraging the other providers from creating seminal items with higher potential values in the future. This is an important limitation because, as a RecSys involves the participation of both users and item providers, we must consider the success of the item providers for the long-term richness of the RecSys environment (Singh & Joachims, 2018; Mladenov et al., 2020; Su et al., 2021).

To cultivate the seed users for the new items, marketing via sponsored advertisements has been a practical strategy. Note that we do not dismiss the possibility of uncontrolled marketing which hurts the utility of the users, but we argue that a minimal amount of controlled marketing in a trusted environment may actually yield better trade-offs between recommendation relevance and item-diversity. For example, we may assign each item provider an exploration budget so that their items get exposed to a minimal percentage of users in a target period of time. (We treat two temporal states of the same user as two user(-state)s with separate marketing opportunities.) We further integrate marketing into

[1] https://github.com/awslabs/recurrent-intensity-model-experiments





(a) Definition of the intensity model $\lambda(x, y)$ between User-State x and Item y.

Figure 1: UserRec for an item. Users are considered positive if they indeed interact with the given item in a future time window. A matching hit requires both relevance and activeness of the user. An active user may have multiple hits with different items, whereas an inactive user may have zero. In addition, the user hits in a given item are often correlated with their past histories with similar items.

recommender systems to create an online matching (OnlnMtch) environment. Notice that OnlnMtch implies exposure planning for each item over all user opportunities in the target period of time. This yields an advantage over other exploration methods like contextual bandits (Li et al., 2010) and Thompson sampling (Agrawal & Goyal, 2013; Kawale et al., 2015), because it lowers the requirement for online updates. As a result, OnlnMtch allows direct benchmarks on the relevance-diversity trade-offs using holdout back-test periods in real-world datasets, shown in Section 5.

To make it happen, we contribute two algorithms: a *RIM* solution to the *UserRec* marketing problem on behalf of the item providers and a *Dual* algorithm for the OnlnMtch problem with its planning challenge. Both are realistic problems with their own applications, which we describe now.

First, in the marketing setup, our goal is to *predict* the relevance of each user with respect to the given item. We validate the quality of our prediction by comparing our top-user recommendations (UserRec) with the set of users who truly interacted with the given item in the holdout back-test period. See Figure 1 for illustration. Besides direct ranking outputs, the raw prediction scores are also used in the OnlnMtch step.

Related to cold-start item promotion, UserRec can be seen as an extension to targeted advertising, where the advertisers bid on search queries or related items to increase the exposure of their chosen items (Chang et al., 2020). In our work, a search query or a related item may be seen as a single event – often the last event – in a user's history. By extending to the full histories of the actual users, we obviously gain more information, but also create a novel problem in the space of information retrieval, which is to compare the users as random time series instead of random variables.

At a high level, we encode the user time series into hidden states by reusing most of sequential ItemRec models (Cho et al., 2014; Hidasi & Karatzoglou, 2018; Ma et al., 2020; Wu et al., 2017; Bai et al., 2019). However, we cannot directly compare the user states conditioned on the provided item, which leads to a "Bayesian paradox" we show in Figure 2. Instead, we realize that our tests as well as real marketing campaigns often have implicit temporal sensitivity. E.g., we must compare between a relevant but inactive user and an active but less relevant user, based on the likelihood of the desired user-item interactions in a near-future time period. We introduce the *intensity* concept in temporal point processes (TPPs) (Reinhart, 2018; Rizoiu et al., 2017) to account for the novel time dimension. Correspondingly, we develop Recurrent Intensity Models (RIMs) to extend from recurrent ItemRec models. See Figure 1a for detailed definitions in intensity models. Note that while TPPs have been widely discussed in the space of user-churn prediction, news trends or item dynamics, and physical events (Jia & Benson, 2019; Du et al., 2016; Leskovec et al., 2009; Chen et al., 2020; Hosseini et al., 2017; Wang et al., 2019a;b), we believe TPPs are relatively novel in the UserRec domain, which opens new venues for applications. Related, TPP-based UserRec can be used in social recommendation (Li et al., 2021).

In Appendix A, we will discuss another example of UserRec where we aim to send email reminders to users who leave their online-shopping carts abandoned for a period of time (Lowe & Craig, 2021). Different from likelihood-based UserRec, the focus there is on the causality between two event types: the reminder event and the checkout event. Our RIM framework can extend to this case, but we omit the discussion for now, because similar intervention signals may not be easily available in the most general cases. Our current discussion is analogous to implicit-feedback recommendation.

Table 1: Three marketing scenarios where recurrent intensity models may apply

	UserRec	Offline Match	Online Match
Pick top users among current users for one marketed item	✓	✓	✓
Market for multiple items at the same time; set minimal-exposure constraints for every item, combined with max-capacity constraints for every user		✓	✓
Responsive marketing when a user appears just like RecSys; distribute marketing materials in a future period of time			✓

Now that they are trained, we consider other scenarios for RIMs to best utilize the real-time properties of these sequence models. Table 1 summarizes these new scenarios. Among them, OnlnMtch yields the best integration between recommendation and marketing. On the other hand, it implies that the marketing materials will be distributed in a future period of time, because the users may appear online at random times, which we must plan ahead.

We develop a *Dual* planner based on the empirical user-state distribution from a past period of time. First, we set the desired exposure rate for every item as a constraint in the optimization of the global utility for all user-item matches. While the item-exposure constraints couple all users in the primal form, thanks to optimization duality, they become independent slack variables in the dual form. As a result, the dual form allows real-time decisions just like the user-centric ItemRec, while meeting the item-exposure constraints in expectation over a similar user-state distribution in a future time period. Our Dual algorithm is conceptually similar to (Huang et al., 2016; Mehta, 2013; Ding et al., 2019), but we use a different approach via entropic regularization similar to Sinkhorn's algorithm (Cuturi, 2013). We found our variant to be easier to converge in our experiments.

The rest of the paper will show details of our RIM model, Dual planner, and experiments. Additional discussions on related work can be found in the appendix.

2 Recurrent Intensity Modeling (RIM)

We are interested in modeling a temporal process represented by a series of event times and types of a given user. Let $\mathcal{S} = [(t_j, y_j) : \forall t_1 < t_2 < \cdots < t_j < \ldots, \forall y_j \in \mathcal{Y}]$ to be a random process that generates user behaviors with a continuous time variable t and a type variable (i.e., the consumed item in the event) in a large-cardinal set $y \in \mathcal{Y}$. We consider a *user-time* as a unique instance variable $X = \mathcal{S}(t) = \{(t_j, y_j) \in \mathcal{S} : t_j < t\}$, with observable user history until time t. We use lower-case $x = s(t)$ to indicate its realization and $x \in \mathcal{X}$ to be the set of all possible user-time instances in the recommendation problem. In the ItemRec construct, the sequence models are called upon the arrival of a user event to predict the item choice by a conditional categorical distribution:

$$P(Y = y|x) = \frac{\lambda(x,y)}{\sum_{y' \in \mathcal{Y}} \lambda(x,y')}, \text{ where } \lambda(x,y) = \exp\left(w_y^\top h(x) + b_y + m(x)\right), \tag{1}$$

where $h(\cdot)$ is a sequence encoder function, w_y is a row in the decoder weight, which is also interpreted as the hidden embedding of a specific item, b_y is a global bias in the item choices, and $m(x)$ represents any extra terms that cannot be learned because they cancel between the nominator and the denominator. The $m(x)$ term reminds us that the raw ItemRec scores $\lambda(x, y)$ cannot be used to compare across users. We aim to repurpose $\lambda(x, y)$ for UserRec by associating $m(x)$ with a user prior term that reflects their activeness in the RecSys, an important factor in UserRec.

Timeless Bayesian versus temporal intensity Given our goal to $\text{select}(x|y)$, our first thought is to frame it by Bayes rule $p(x|y) \propto p(y|x)p(x)$, i.e., relating $m(x)$ to $\log p(x)$ and other normalization terms. However, this leads to a paradox shown in Figure 2. The paradox is because we represent a user as a timeless variable x, instead of a temporal point process (TPP) through the lens of $s(t)$.

TPP predicts *when* the next event will happen for a user in addition to *what* type it is. TPP is different from other temporal models in that it predicts the event time as a moving target, whereas the other models aggregate time as a feature. The temporal concept is best explained from a counting perspective. Here, we model the absolute number of user-item interactions, $\Delta N(x, y)$, in a future

Truncated user history in last week (x)	$P(y=D\|x)$
A, B	0.4
C	0.8
$\emptyset$	
$\emptyset$	0.3
$\emptyset$	

$$\text{(Bayesian)} \times \begin{cases} p(x=[A,B]) = 0.2 \\ p(x=C) = 0.2 \\ p(x=\emptyset) = 0.6 \end{cases} \Rightarrow \begin{cases} p(x=[A,B]|y=D) \propto 0.2 \times 0.4 = 0.08 \\ p(x=C|y=D) \propto 0.2 \times 0.8 = 0.16 \\ p(x=\emptyset|y=D) \propto 0.6 \times 0.3 = 0.18 \end{cases}$$

$$\text{(MTPP)} \times \begin{cases} \lambda(x=[A,B]) = 2 \\ \lambda(x=C) = 1 \\ \lambda(x=\emptyset) = \mu \end{cases} \Rightarrow \begin{cases} \lambda(x=[A,B]|y=D) \propto 2 \times 0.4 = 0.8 \\ \lambda(x=C|y=D) \propto 1 \times 0.8 = 0.8 \\ \lambda(x=\emptyset|y=D) \propto \mu \times 0.3 = 0.3\mu \end{cases}$$

Figure 2: Bayesian paradox. We start with an ItemRec model $p(y|x)$ that assigns nonzero scores to all users, including cold or inactive users, who are abundant in RecSys due to power-law effects. The Bayes rule incorrectly associates the concentration in quantity with the engagement prior of each user, leading to non-intuitive predictions. Instead, our RIM proposal treats each user as a draw from a MTPP and we use the intensity parameter (defined in Figure 1a) to predict their future engagements. $\mu(>0)$ is a background intensity assigned to all processes, including inactive users.

time window, $[t, t+\Delta t)$, where $s(t)$ coincides with x. We adopt a Poisson assumption that the rate at which the desired events happen is smooth in time, given by an *intensity* parameter:

$$\lambda(t; s(t), y) = \lambda(x, y) = \lim_{\Delta t \to 0} \mathbb{E}\left[\tfrac{\Delta N}{\Delta t} \middle| x, y\right] = \mathbb{E}\left[\tfrac{\mathrm{d}N}{\mathrm{d}t} \middle| x, y\right]. \tag{2}$$

The definition of the intensity parameter is visualized in Figure 1. Besides matching the expectations, we may further prescribe $\Delta N(x,y) \sim \text{Poisson}\left(\lambda(x,y)\Delta t\right)$ as a parametric distribution, especially in short time windows. The Poisson log-likelihood is:

$$\log P\left(\Delta N(x,y) = n | \lambda(x,y)\Delta t\right) = n \log(\lambda(x,y)\Delta t) - \lambda(x,y)\Delta t - \log(n!). \tag{3}$$

The intensity parameter connects three variables of our interests: user sequence x, item y and our prediction target $\Delta N(x,y)$ in a future time window $[t, t+\Delta t)$. Slicing $\Delta N(x,y)$ in different directions yields different aspects of the problem. Most importantly, we slice by Δt to connect it to the distribution of future event times. We expect an active user to have higher intensities in unit time or, equivalently, shorter intervals between events.

Temporal Point Processes (TPPs) We focus on the connection between the intensity parameter $\lambda(x) = \lambda(t; s(t))$ and the time t_* for the next interaction. (For this paragraph, we consider the interactions with any next items in general and omit the item suffix.) Intuitively, we split the time interval until the next event into a large set of infinitesimal windows, i.e., $[t, t_*) = \cup_\tau [\tau, \tau + \mathrm{d}t)$, and insert Poisson likelihood with label $n(\tau) = 0$ to all previous windows and $n(\tau_*) = 1$ to the last window. Plugging in equation 3 we get

$$\begin{aligned} \sum\nolimits_{t \le \tau < t_*} \log P\big(\mathrm{d}N(\tau) = n(\tau) \big| \lambda(\tau; s(\tau))\mathrm{d}t\big) &= \log\big(\lambda(\tau_*; s(\tau_*))\mathrm{d}t\big) - \sum\nolimits_{t \le \tau < t_*} \lambda(\tau; s(\tau))\mathrm{d}\tau \\ \overset{\mathrm{d}\tau \to 0}{\Rightarrow} \quad \log p\big(T_* = t_* \big| t; \lambda\big) &= \log\big(\lambda(t_*; s(t_*))\big) - \int_t^{t_*} \lambda(\tau; s(\tau))\, \mathrm{d}\tau. \end{aligned} \tag{4}$$

Note that the $\mathrm{d}t$ inside the logarithmic term is moved to the left side when we convert the probability mass function to probability density function in the continuous variable T_*. Reinhart (2018); Chen (2016) discuss rigorous derivations as well as additional properties.

Equation 4 is known as the inter-arrival time distribution, which directly formulates a loss function to learn the intensity parameters, without the need to explicitly bucketize time in equation 2. The intensity functions can further be time-varying based on the user state $x = s(t)$. The numerical integration may be solved by importance sampling (Mei & Eisner, 2017), ODE solvers (Chen et al., 2018), or analytical forms in the case of Hawkes processes, which we use in this paper for simplicity.

Marked-TPP Decomposition While TPPs can be built for each event type (i.e., item choice) with cross-channel influences from other event types, this may lead to drastically different models than existing sequential ItemRec models. Instead, we utilize a Marked-TPP decomposition,

$$\lambda(x, y) = \lambda(x) p(y|x), \tag{5}$$

where $p(y|x)$ comes from a regular sequence model (1). MTPP is the framework of our RIM proposals. It resembles Bayes rule and considers the user sequence $x = s(t)$ as a point process.

3 ESTIMATION OF THE INTENSITY PARAMETERS

While RIM is motivated to estimate the intensity of each user-item pair separately, we find it more convenient to utilize existing sequential ItemRec models. Based on (5), we keep the RNN (or Transformer) model for its categorical prediction of the preference *direction*, $p(y|x)$, while introducing separate estimators for the *user-intensity prior*, $\lambda(x)$.

RNN-Pop is our simplest model, where we use the length of the user histories in the training sets. This approach works due to a homogeneous intensity assumption, where the number of past events directly correlates with the future event intensity. It also assumes that all user activities are recorded over a comparable period of time, which often holds in practice, because RecSys often truncates the training sets by a fixed time for scalability reasons. RNN-Pop is used in the example in Figure 2.

RNN-Hawkes To extend the naive RNN-Pop models, we may break the homogeneity assumption to consider user state changes that affect future event intensity. In this regard, Hawkes process assumes a positive stimulation through past user events and a gradual churn-out effect after a period of inactivity. It models these effects via an exponentially-decaying kernel in the intensity parameters

$$\lambda(t) = \mu + \sum_{j:t_j<t} \sum_{r=1}^{R} \alpha_r \phi_r(t - t_j), \text{ where } \phi_r(\Delta t) = \frac{1}{s_r} \exp\left(-\frac{\Delta t}{s_r}\right) 1_{\Delta t>0}, \text{ and } \mu, \alpha, s > 0.$$

Here, we extend it with a mixture of R latent kernels with learned coefficients and learn it with *tick* software package Bacry et al. (2017). We design the R kernels to have log-spaced half-lives between 10^{-3} and $10t_{\max}$ where $t_{\max}$ is the largest temporal span in user histories. A large-s_r kernel approximates the RNN-Pop model with a step function after each observation, whereas a small-s_r kernel suggests a fast-diminishing effect, often within a short browsing session.

RNN-Hawkes-Poisson (RNN-HP) For long prediction horizons, we may also calibrate Hawkes scores to eliminate the effects of fast-diminishing kernels. In fact, the accurate long-horizon model,

$$\bar{\Lambda} \sim \mathbb{E}[\Delta N(x, y)] = \mathbb{E}\left[\int_t^{t+\Delta t} \lambda(\tau; \mathcal{S}(\tau), y)\, \mathrm{d}\tau \middle| \mathcal{S}(t) = x\right], \tag{6}$$

takes the expectation with respect to stochastic integrals. To calibrate for the long-terms intensity scores, we notice that the Hawkes states $\phi(t)$ are already positive. We thus model the final scores as

$$\bar{\Lambda} = \phi(t) \odot \mathrm{softplus}(w) = \phi(t) \odot \log(1 + \exp(w)). \tag{7}$$

Our additive formulation is inspired by (Mei & Eisner, 2017) and modified by allocating the softplus to each coordinate to improve its numerical stability. Another approach is to replace (6) with over-dispersed distributions like negative binomial (Salinas et al., 2020), which we leave as future work.

Other Baseline Methods Practical personalized marketing models are often highly customized, e.g., with profitability or seasonality concerns, and they are not always designed to cover all items in the catalog, which requires training labels for all user-item pairs, including user-item pairs with zero activities in a time period. Despite so, we illustrate the general idea to use explicit recency, frequency, monetization (RFM) features for user segmentation (Fader et al., 2005). Here, frequency and monetary spending in related items are correlated with a customer's future life-time value (CLV) with respect to the given item and so is the time elapsed since last purchase with an inverse relation.

We build a model based on graph-convolutional feature aggregation similar to **GCMC** (Berg et al., 2017), followed by Bayesian-personalized ranking loss (BPR) (Rendle et al., 2009) against sampled negative users and items in the large space of all user-item pairs. Notice that this GCMC model uses labels from an explicit time period, which is inferior to RIM because (a) it has fewer usable labels and (b) it bucketizes the labels instead of preserving their continuous time and order. We attempt a mitigation we call **GCMC-Ex** that creates multiple periods to extract more labels and order information. However, we do notice that perfecting this remedy leads to a temporal integration similar to our TPP derivation (4). More details of this model can be found in Section B.

We also consider matrix factorization (MF) methods, such as **ALS** (Hu et al., 2008) and **LogisticMF** (Johnson, 2014) from Implicit package[2] and **BPR** (Rendle et al., 2009) from LightFM package (Kula, 2015). Notably, LightFM-BPR is limited to sample either users or items, but not both. We include our own BPR model for completeness. Unlike RIM and GCMC, MF methods do not consider temporal features.

[2] https://github.com/benfred/implicit

Algorithm 1 Dual planner for online (and offline) matching

Require: approximately normalized $\lambda(\cdot,\cdot) \in [0,1]$ with high probability; $\alpha; \beta$;
user-state distribution $P(\mathcal{X})$, often from a past period of time; step-size γ
Ensure: $\hat{v}_y$; return items for a user with a random state x by $\text{TopK}\{\lambda_{xy} - \hat{v}_y : y \in \mathcal{Y}\}$
init $\hat{v}_y \sim \text{Unif}(-1, 0), \forall y \in \mathcal{Y}$
for k in $[0, 1, \ldots, 100]$ **do**
set $\epsilon = 0.8^k$; sample a batch of user-states $X' \sim P(\mathcal{X})$; compute $\{\lambda_{xy} : \forall x \in X', y \in \mathcal{Y}\}$
find root $r_x(\hat{\pi}_\epsilon(\lambda_{x:}, u'_x, \hat{v})) = 0$ for every $x \in X'$ given $\hat{v}$; project to $u'_x \geq 0$
find root $r_y(\hat{\pi}_\epsilon(\lambda_{:y}, u', v'_y)) = 0$ for every $y \in \mathcal{Y}$ given u'; project to $v'_y \leq 0$
take step $\hat{v}_y = (1-\gamma)\hat{v}_y + \gamma v'_y, \forall y \in \mathcal{Y}$

4 Online Matching and Dual Planning

So far, we have learned RIM models as extensions from RNN/Transformers using MTPP intuitions. They often yield better accuracy than the time-bucketized RFM baselines and timeless MF models. Further, RIMs yield greater impacts in the offline and online matching scenarios in Table 1. Both new scenarios can be formulated by a constrained optimization problem, which we describe now.

OnlnMtch as Constrained Optimization We start by reformulating offline UserRec by constrained optimization. Say we want to send marketing emails to Top-C users at a given time to promote Item y. We represent the space of all user(-state)s at the given time as $\mathcal{X} \ni x$ and rewrite $\lambda_{xy} = \lambda(x, y)$ for convenience. We express the top-user recommendation as binary assignments $\pi_{xy} \in \{0, 1\}, \forall x \in \mathcal{X}$ and it is easy to see that π_{xy} can be equivalently solved by:

$$\max_{\pi \in \{0,1\}} \sum\nolimits_{x \in \mathcal{X}} (\lambda_{xy} \pi_{xy}), \text{ s.t. } \sum\nolimits_{x \in \mathcal{X}} \pi_{xy} \leq C. \tag{8}$$

For online marketing or matching, we extend the user-state set to a draw from a distribution $P(\mathcal{X})$, which unions all user-states in a period of time by sampling every user at regular time intervals or based on their actual times of interactions, whichever is appropriate for the application. Each sample point is counted as an independent user-state instance. We also replace the summation in (8) with Lebesgue integral in the probability space. We then solve for the OnlnMtch assignments $\pi_{xy} \in \{0, 1\}, \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$ with a linearly-relaxed optimization problem:

$$\max_{0 \leq \pi \leq 1} \mathbb{E}_{xy}(\lambda_{xy} \pi_{xy}), \text{ s.t.} \tag{9}$$

$$\begin{cases} \mathbb{E}_y(\pi_{xy}) \leq \alpha, \forall x \in \mathcal{X} & \rightarrow \text{user-capacity constraint with dual variable } u_x; \\ \mathbb{E}_x(\pi_{xy}) \geq \beta, \forall y \in \mathcal{Y} & \rightarrow \text{item minimal-exposure constraint with dual variable } v_y; \\ \mathbb{E}_x(\pi_{xy}) \leq \beta', \forall y \in \mathcal{Y} & \rightarrow \text{item maximal-exposure constraint that rewrites (8);} \end{cases} \tag{10}$$

where we express the thresholds in relative terms, i.e., $\alpha = {}^{K}/_{|\mathcal{Y}|}$ for Top-K ItemRec given a user-state x and $\beta' = {}^{C}/_{|\mathcal{X}|}$ for Top-C UserRec on behalf of an item y. Unique to OnlnMtch, we can also insert a minimal-exposure constraint for each item with threshold $\beta, \forall y \in \mathcal{Y}$. Without loss of generality, we keep β and omit β' to keep the discussions simple.

For the OnlnMtch problem, our challenge is that the item constraints are measured over a distribution of users in a period of time, while OnlnMtch requires real-time recommendation / marketing decisions. This challenge can be met with optimization duality, which we describe next.

Dual algorithm We take ideas from optimization duality to rewrite the constraints in a Lagrangian form. I.e., equation 9 (leaving out the omitted β' term) is equivalent to

$$\max_{\pi} \min_{u \geq 0, v \leq 0} L_\epsilon(\pi, u, v) = \mathbb{E}_{xy}(\lambda_{xy}\pi_{xy}) - \mathbb{E}_x(u_x r_x(\pi)) - \mathbb{E}_y(v_y r_y(\pi)) + \epsilon E_{xy}(H(\pi_{xy}))$$

$$\text{where} \begin{cases} r_x(\pi) = \mathbb{E}_y(\pi_{xy}) - \alpha & \rightarrow \text{user-capacity constraint;} \\ r_y(\pi) = \mathbb{E}_x(\pi_{xy}) - \beta & \rightarrow \text{item minimal-exposure constraint;} \\ H(\pi) = -\pi \log(\pi) - (1-\pi)\log(1-\pi) & \\ & \rightarrow \text{we use entropic trap for } 0 \leq \pi \leq 1 \text{ with annealed } \epsilon > 0; \\ & \quad \text{our algorithm will ensure that } \pi \text{ never escapes from this trap.} \end{cases}$$

Next, we apply the standard trick to define a dual objective by switching the order of operations:

$$\min_{u \geq 0, v \leq 0} \left\{ d_\epsilon(u, v) = \max_\pi L_\epsilon(\pi, u, v) \right\}. \tag{11}$$

After some derivations in Lemma S1 in Section C.1, we find an analytical solution to the inner-loop:

$$\hat{\pi}_{xy} = \text{sigmoid}\left(\frac{\lambda_{xy} - u_x - v_y}{\epsilon}\right); \text{ define as } \hat{\pi}_\epsilon(\lambda_{xy}, u_x, v_y). \tag{12}$$

The solution intuitively suggests that the assignment of a user-item pair not only depends on their predicted intensity score, but also on the global dual variables u_x and v_y. For example, if $\lambda_{xy_1} > \lambda_{xy_2}$ for the majority of users while we intend to keep the global exposure rates for y_1 and y_2 to be equal, then we may choose $v_{y_1} > v_{y_2}$, causing the final item-assignments to be reverted for some users to achieve the desired exposure outcomes. Notice that the item-dual variables become nonpositive if our aim is to boost the item exposures to at least the predefined β portion.

Finally, we discuss the solution to the outer-loop, which also yields an intuitive algorithm. To do so, we plug (12) to (11). The derivations in Lemma S2 in Section C.1) lead us to a clean result:

$$\frac{\partial d_\epsilon(u, v)}{\partial u_x} = -r_x(\hat{\pi}_\epsilon(\lambda_{x:}, u_x, v)); \quad \frac{\partial d_\epsilon(u, v)}{\partial v_y} = -r_y(\hat{\pi}_\epsilon(\lambda_{:y}, u, v_y)). \tag{13}$$

Setting both values to zero yields an iterative algorithm that alternates between solving u and v. We use simulated annealing from a larger initial $\epsilon > 0$ to produce more stable results. See Algorithm 1 for more details. In the extreme case where $\epsilon \to 0$, the two root-finding steps degenerate as

$$\begin{cases} \text{Find } u_x = \text{TopKThreshold}(\{\lambda_{xy} - \hat{v}_y : y \in \mathcal{Y}\}) \text{ for every } x \in \mathcal{X}; \text{ project to } u_x \geq 0 \\ \text{Find } v_y = \text{Quantile}(\{\lambda_{xy} - u_x : x \in \mathcal{X}\}, \beta) \text{ for every } y \in \mathcal{Y}; \text{ project to } v_y \leq 0 \end{cases} \tag{14}$$

For larger ϵ, we use bisection search to solve for (13) in each step, which is possible because the gradient functions are monotonic with respect to the variables u_x and v_y, respectively. The computational complexity is similar to finding quantiles by full sorts. See Appendix C.2 for further discussions on the computational and storage complexity.

Connections to Online Bidding and Pacing. Real-time bidding is a common practice in the online advertising industry. The bidding price from an advertiser depends on the expected reward due to the placement, the estimated competitor-pricing distribution, as well as its pacing plan to manage the campaign budget over a period of time for best results. In our Dual algorithm, we may see flavors of all the components: the dual variable v_y affects the overall ranking of the item in each user-matching opportunity, receives influences from all other items in each iteration of the alternating algorithm (14), and is calibrated against the empirical user-state distribution to achieve smooth pacing over time. Additionally, our Dual planner may extend to future time periods if we assume stationary user-state distribution, which we empirically observe, as well as dual-variable distribution for the other items, which we control.

Dual planner has direct applications in guaranteed display advertising, where advertisers buy contracts to show their ads to a guaranteed size of audience in a period of time (Bharadwaj et al., 2010). Further, Huang et al. (2016); Mehta (2013); Ding et al. (2019) analyzed the online regrets in such guarantees with slightly different algorithms. These algorithms are designed to minimize online-regrets in adversarial settings. Comparatively, our work is under stochastic and stationary assumptions, but we may consider adjustments after short periods of time for more practicality. Also, we found some difficulties in the convergence of their algorithms, potentially due to the skewed intensity-score distributions in our experiments. We leave the detailed comparisons for future work.

5 EXPERIMENTS

We conduct experiments in three RecSys datasets with unique properties.

- **Movielens (ML)** (Harper & Konstan, 2015) is a movie rating website that collects user preferences to study RecSys. We use a small-scale ML-1M in an 2003 release with one million rating events between six thousand users and three thousand items to easily validate our ideas.

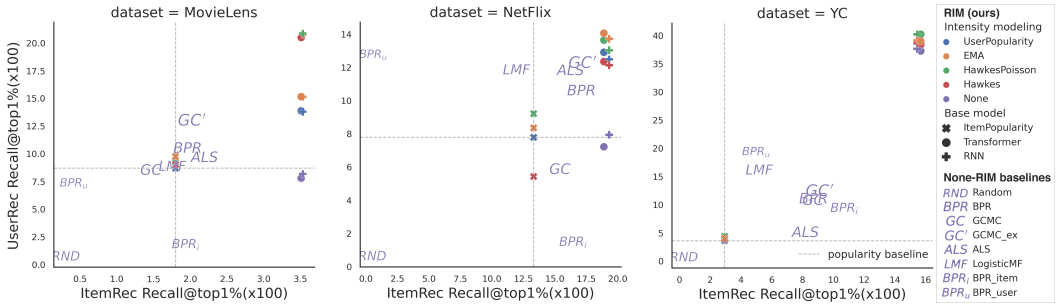


Figure 3: Suitability of different methods for ItemRec and UserRec tasks, plotted on different axes.

- **Netflix (NF)**[3] is a movie rental service where movie ratings from a user's past affects their personalized ItemRec in the future. NF is widely used in rating prediction and ItemRec, but less so in UserRec and OnlnMtch.
- **Yoochoose (YC)**[4] is a dataset used in RecSys 2015 Challenge with six months of activities for product recommendation such as general tools, toys cloths, electronics, and much more.

We highlight some data preparation steps. Most of our RIM methods are self-supervised sequence models. Thus, we only need to hold out test windows with equal horizons for the evaluation of UserRec, ItemRec, and Mtch. We use absolute time windows on NF and consider only the observed users/items in the training data as testing targets, because we may not know which new users/items will show up in the testing windows without temporal leakage from the test set. Besides temporal splits, it is common for sequence models to split train/test sets by users. We consider user-splits on ML and YC, because most users are active at different, non-overlapping clock times. We hold out time windows only on the test users (Group-B in Table S1), with equal starting time relative to their first event as well as equal size or horizon. All training users (Group A) and the observed histories of the testing users (Group B left part) are considered training data.

RNN-HP and GCMC(*) require further splitting of the training set. On NF, we create a set-back window between $[T', T)$ from all users and on ML/YC, we keep the same time $[T, T + \Delta T)$ but change the user base to Group A for validation. Similarly, to simulate OnlnMtch, we must calibrate Online-Dual on different users or times to test the transferability of the user-state distributions in online scenarios. We reuse the set-back window as RNN-HP and GCMC(*) for simplicity. Statistics of our data splits are shown in Table S1.

5.1 UserRec and ItemRec Results

We first compare various models for the separate tasks of ItemRec and UserRec to validate our proposals in Section 3. On the ItemRec dimension, we observe the usual story that user-conditional models outperform non-personalized models, but they are then topped by sequence models. However, the comparison becomes interesting in the UserRec direction, where the adoption of raw probability scores in RNN and Transformers could lead to inferior results than the simple heuristic that recommends top users unconditionally of the given item, i.e., UserPopularity or the more advanced TPP models for the marginal prediction, $\lambda(x)$. This is because the user-conditional probability scores in RNN/Transformer fail to identify the active users with higher intensity priors. On the other hand, with our RIM proposals, the sequence models become top in the UserRec direction as well.

For the baseline BPR methods, we also compare our implementation with open-source LightFM package. LightFM only allows User/Item sampling, but not both at the same time. In contrast, our version samples both sides concurrently, allowing a single model to perform well in both tasks. We also compare GCMC with GCMC*, a variant that includes multiple time periods to extract more labels and relations. We include similar plots for the precision metrics in the appendix.

[3] https://www.kaggle.com/netflix-inc/netflix-prize-data

[4] https://www.kaggle.com/phhasian0710/yoochoose

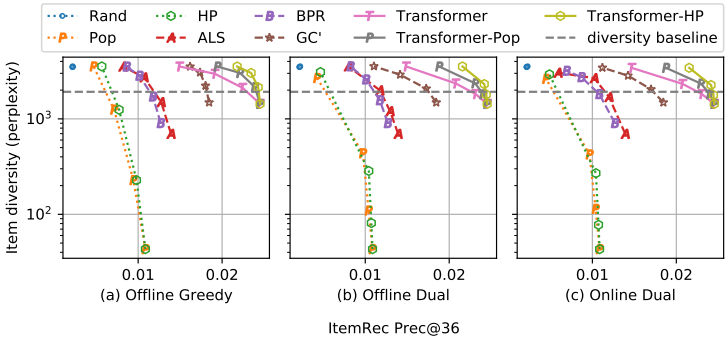


Figure 4: Matching experiments where we set minimal-exposure constraints for every item in additional to regular ItemRec settings. Showing a subset of representative methods on ML dataset.

5.2 Offline and Online Matching Experiments

To validate the discussions in Section 4, we integrate UserRec into ItemRec by setting minimal-exposure constraints. We fix $\alpha = 1\%$ in equation 9 and vary $0 \leq \beta \leq 1\%$ as item min-exposure constraints. (Section F covers a more common marketing scenario where the user-to-item capacity ratio is greater than one, which leads to even larger margins in our RIM methods. Here, we focus on the case with tighter constraints to highlight the subtleties and new possibilities in our Mtch setting.) Choosing $\beta = 0$ corresponds to the unconstrained ItemRec problem. As we increase β, we observe an increase in item-exposure diversity, measured by

$$\text{perplexity} = \exp\left(-\sum_y \left[\left(\frac{\pi(y)}{\sum_y \pi(y)}\right) \log\left(\frac{\pi(y)}{\sum_y \pi(y)}\right)\right]\right), \tag{15}$$

where $\pi(y)$ is proportional to the total number of times that y is recommended in front of a user.

Figure 4 compares the performance of various methods in three matching environments: Offline-greedy, Offline-Dual and Online-Dual. Within each environment, we see that all methods exhibit trade-offs between ItemRec relevance and item diversity. Comparatively, Transformer-Pop/HP yields best performance and compromises least relevance to achieve better diversity, which supports our claim that better modeling of user intensities leads to better welfare for both users and items. Notice that showing diversity effects in offline experiments is a challenging task due to the vast amount of missing data for alternative treatment effects. We attribute our success to the consideration of new problem dimensions. We show similar results in Section F on other datasets and settings.

We see similar results across different environments in Figure 4(b)&(c). Notably, the Online-Dual solver supports both real-time decisions and long-term exposure constraints over the distribution of all user-states in a future period of time. The consistent performance across different scenarios demonstrates great practicality in the integration of ItemRec and UserRec in RecSys.

6 Conclusion

RecSys that purely focuses on the immediate user satisfaction may hurt the overall diversity of the items being recommended. Instead, we study an exploratory marketing mechanism where each item will be guaranteed a minimal exposure rate to a suitable audience in a period of time. In this way, we address the pain points in marketing and allow the item creators to focus their expertise on creation. We believe this improves the long-term diversity and liveliness of the RecSys.

We make two contributions. Firstly, we turn a sequential ItemRec model around to recommend users on behalf of a given item. Our challenge is to treat the users as time series instead of random variables. We introduce novel connections to the intensity parameter of a temporal point process and develop a RIM solution that combines RNN (Transformer) and TPP. Our second contribution is to allow a RecSys to guarantee a minimal (maximal) exposure rate for every item that we intend to nourish (downsize) in a future time period. We call this scenario OnlnMtch and introduce Dual planner as a solution. All of our methods learn by user mini-batches to achieve unlimited scalability in the size of the user-state set. Also, compared with other exploration methods that require online feedback, our methods have unique advantages in the ease of benchmarks using offline datasets.

ACKNOWLEDGEMENTS

We thank Vaibhav Sethi and Balakrishnan (Murali) Narayanaswamy for some early discussions on the idea to repurpose ItemRec models for UserRec applications. We thank Danielle Robinson and Yuyang (Bernie) Wang for their discussions on the topic of Neural-ODEs, which inspired us to look further into the literature around temporal point processes. We appreciate the discussions with Lihong Li and Hsiang-Fu Yu on the general topic of targeted advertising. We finally appreciate the anonymous reviewers for this and our earlier versions for their constructive comments and additional references.

REFERENCES

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*. PMLR, 2013.

E. Bacry, M. Bompaire, S. Gaïffas, and S. Poulsen. tick: a Python library for statistical learning, with a particular emphasis on time-dependent modeling. *ArXiv e-prints*, July 2017.

Ting Bai, Lixin Zou, Wayne Xin Zhao, Pan Du, Weidong Liu, Jian-Yun Nie, and Ji-Rong Wen. Ctrec: A long-short demands evolution model for continuous-time recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, 2019.

Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

Andrey Bernstein, Shie Mannor, and Nahum Shimkin. Online classification with specificity constraints. *Advances in Neural Information Processing Systems*, 23:190–198, 2010.

Vijay Bharadwaj, Wenjing Ma, Michael Schwarz, Jayavel Shanmugasundaram, Erik Vee, Jack Xie, and Jian Yang. Pricing guaranteed contracts in online display advertising. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, 2010.

Alex Boyd, Robert Bamler, Stephan Mandt, and Padhraic Smyth. User-dependent neural sequence models for continuous-time event data. *Advances in Neural Information Processing Systems*, 33: 21488–21499, 2020.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H. Chi. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *International Conference on Learning Representations*, 2020.

Yuanda Chen. Thinning algorithms for simulating point processes. *Florida State University, Tallahassee, FL*, 2016.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26:2292–2300, 2013.

Weicong Ding, Dinesh Govindaraj, and S V N Vishwanathan. Whole page optimization with global constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, 2019.

Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J. Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, 2015.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1555–1564, 2016.

Peter S. Fader, Bruce G. S. Hardie, and Ka Lok Lee. Rfm and clv: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, 42:415 – 430, 2005.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 843–852, 2018.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks, 2016.

Seyed Abbas Hosseini, Keivan Alizadeh, Ali Khodadadi, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid R. Rabiee. Recurrent poisson factorization for temporal recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, 2017.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 263–272, 2008.

Jim C. Huang, Rodolphe Jenatton, and Cedric Archambeau. Online dual decomposition for performance and delivery-based distributed ad allocation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 117–126, 2016.

Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pp. 402–411. PMLR, 2016.

Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 652–661. PMLR, 2016.

Christopher C Johnson. Logistic matrix factorization for implicit feedback data. *NeurIPS 2014 Workshop on Distributed Machine Learning and Matrix Computations*, 2014.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 2018.

Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in Neural Information Processing Systems*, 2015.

Maciej Kula. Metadata embeddings for user and item cold-start recommendations. In *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015.*, CEUR Workshop Proceedings, 2015.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, 2009.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pp. 661–670, 2010.

Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. Learning temporal point processes via reinforcement learning. In *NeurIPS*, 2018.

Yunzhe Li, Yue Ding, Bo Chen, Xin Xin, Yule Wang, Yuxiang Shi, Ruiming Tang, and Dong Wang. Extracting attentive social temporal excitation for sequential recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 998–1007, 2021.

Ryan Lowe and Andy Craig. Solving abandoned cart scenarios using amazon pinpoint event-triggered journeys. AWS Messaging & Targeting Blog, 2021.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, 2017.

Yifei Ma, Balakrishnan (Murali) Narayanaswamy, Haibin Lin, and Hao Ding. Temporal-contextual recommendation in real-time. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '20, 2020.

Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.

Hongyuan Mei and Jason Eisner. The neural hawkes process: a neurally self-modulating multivariate point process. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017.

Martin Mladenov, Elliot Creager, Omer Ben-Porat, Kevin Swersky, Richard Zemel, and Craig Boutilier. Optimizing long-term social welfare in recommender systems: A constrained matching approach. In *International Conference on Machine Learning*, pp. 6987–6998. PMLR, 2020.

Alex Reinhart. A Review of Self-Exciting Spatio-Temporal Point Processes and Their Applications. *Statistical Science*, 33(3):299 – 318, 2018.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pp. 452–461, 2009.

MA Rizoiu, Y Lee, S Mishra, and L Xie. A tutorial on hawkes processes for events in social media. *Frontiers of Multimedia Research*, 2017.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, 2018.

Yi Su, Magd Bayoumi, and Thorsten Joachims. Optimizing rankings for recommendation in matching markets. *arXiv preprint arXiv:2106.01941*, 2021.

Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *The World Wide Web Conference*, WWW '19, 2019a.

Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *The World Wide Web Conference*, WWW '19, 2019b.

Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. Recurrent recommender networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, 2017.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International Conference on Machine Learning*, pp. 1717–1726. PMLR, 2016.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International Conference on Machine Learning*, pp. 11692–11702. PMLR, 2020.

A Related Work and Discussions

Non-sequential marketing strategies Marketing is a general area studied from multiple perspectives. From a system's perspective, our example of keyword and related-item bidding comes from close examination of large-scale extreme classifier models (Chang et al., 2020). The push notification example considers explicit user states, which are commonly expressed via Recency, Frequency, and Monetary values (RFM) (Fader et al., 2005). By far, the most common and straightforward approach is to use Gradient Boosted Decision Trees to regress on the binary user-item engagement labels using these RFM features. At a simpler level, without the manually constructed temporal features, the loss function is shared with traditional matrix factorization approaches, such as Alternating Least Squares (Hu et al., 2008), Bayesian Personalized Ranking (Rendle et al., 2009), and Logistic Matrix Factorization (Johnson, 2014), all of which serve as UserRec baselines.

Neural temporal point processes We further receive inspirations from temporal point processes (Reinhart, 2018; Chen, 2016; Du et al., 2016; 2015), Neural-ODEs (Chen et al., 2018), and their intersections (Jia & Benson, 2019; Chen et al., 2020; Mei & Eisner, 2017; Zuo et al., 2020). Most of the works are great introduction of various aspects of TPPs and their modern development in GPU environments. However, most of the works focus on the prediction for better temporal modeling without novel applications of the intensity parameters for different applications. Along the line of better temporal models, (Jia & Benson, 2019) shows how Neural-ODE can be applied to a MTPP construction and (Boyd et al., 2020) focuses on the particular challenge that personalized user histories "exhibit large predictive uncertainty at the beginning of the sequences". We do observe that many users churn out shortly after initial engagement. To model this temporal heterogeneity near the users' beginning, we use a simple trick, where we add another related time series with exactly one event for every user to mark their start time.

Causality In our work, we consider item exposure as the primary factor and assume positive downstream effects afterwards. We could also analyze the temporal correlation between an intervention event and the change in the user's temporal intensity in another event type. In the abandoned-cart example, let $\tilde{y}$ be the email-reminder event at time $\tilde{t}$ and y_* be the desired user-checkout event at time t_*. We then define the intervention state of the user as $\tilde{x} = x \cup \{(\tilde{t}, \tilde{y})\}$ and model the temporal intensity of the subsequent checkout event by

$$p(T_* = t_* | \tilde{t}; \lambda) = \log(\lambda(\tilde{x}, y_*)) - (t_* - \tilde{t})\lambda(\tilde{x}, y_*), \tag{16}$$

or its time-varying integral form similar to equation 4. Another difference to the likelihood-based intensity models in our main text is the inference routine. To choose which users to send the email reminders, we must also consider the users in the hypothetical intervention states. I.e., we augment the user time series as $\tilde{x} = x \cup \{(\tilde{t}, \tilde{y})\}$ before computing their intensity scores in the effect items.

Besides the (Granger) causality research, temporal intervention effects can also help prevent bad consequences in marketing. For example, we may combine the positive effects of a marketing email through likelihood-based intensity modeling as well as the risks of undesirable effects like user churn-out through causality models. Some interesting work on temporal causality include (Xu et al., 2016; Li et al., 2018; Jiang & Li, 2016; Lundberg & Lee, 2017).

Convex optimization and online regrets On the long-term constraints, we reference from Sinkhorn's algorithm from optimal transport (Cuturi, 2013; Genevay et al., 2018) and the literature in long-term marketing via online campaigns (Huang et al., 2016; Jenatton et al., 2016; Ding et al., 2019), with convergence analysis dating back to (Mehta, 2013). Online convergence analysis further leads to novel applications such as online classification with specificity constraints (Bernstein et al., 2010).

Diversity and Exploration In our discussion of the Dual solution for OnlnMtch, we mention a dual solution in the form:

$$\max_{\pi \in (0,1), r_x(\pi) \leq 0} \mathbb{E}_y[(\lambda(x, y) - v(y))\pi(x, y)] = \mathrm{TopK}\{\lambda(x, y) - v(y) : \lambda(x, y) - v(y)\}.$$

This solution is intrinsically connected to the idea of bandit exploration with upper-confidence bound (UCB) (Li et al., 2010; Chu et al., 2011), where our adoption of the the minimal-exposure constraint

leads to a positive boost to the raw prediction scores, i.e., $-v(y) \geq 0$. Further exploration in this direction may lead to additional benefits in the OnlnMtch paradigm.

B GCMC MODEL DETAILS

The basic idea in GCMC-CLV is to split user behavioral data in time to create a temporal-prediction scenario. We hold out a targeting window from the observed user behaviors as CLV labels and build RFM features for every user at the start of the window. Here, every user is represented by the aggregation of the items found in the user's own history, which reflects the frequency of interactions in these items. Due to the sparsity of the user-item pairs in the observed user histories, we use a graph convolution layer for the aggregation, which also extracts the semantic item embeddings similarly to graph-convolutional matrix factorization (GCMC). As for recency, we construct a single user bias term based on the recency of the last purchase regardless of the item choices. We omit the per-item recency terms due to their high (inverse-)correlation with the per-item frequency features. We neither have monetary information in our experiments.

The main challenge in the scalable CLV approach is the vast amount of implicit-negative labels. While we observe positive CLVs in the target window, we should also include every unobserved user-item pair as a zero CLV label. This leads to an imbalanced label distribution, i.e., a significant amount of zeros which we call negatives, and a small portion of non-zeros which we treat as positive (binary) labels. To address the imbalance, we adopt a strategy of negative sampling to augment every positive user-item pair with negative users ($\tilde{u}$) and items ($\tilde{\imath}$), followed by sigmoid-triplet loss:

$$\begin{aligned}\text{loss} = 0.5\Big(&\mathbb{E}_{\tilde{\imath}} \log(\text{sigmoid}(\text{score}(u, \imath) - \text{score}(u, \tilde{\imath})) \\ &+ \mathbb{E}_{\tilde{u}} \log(\text{sigmoid}(\text{score}(u, \imath) - \text{score}(\tilde{u}, \imath))\Big)\end{aligned}$$

Details of our negative sampling strategy is similar to Bayesian Pairwise-Ranking (BPR). However, open-source BPR implementations are often limited to sampling only negative items, leaving negative users unconsidered and causing biases in the CLV predictions. We thus include our implementation, with the additional benefit of better integration with GCMC framework. We sample the negative users (and items similarly) according to

$$p(u) \propto (C(u) + 1)^{0.5},$$

where $C(u)$ is the total number of appearances of the user (or item) in the training data.

Finally, the BPR scores are translation-invariant, because BPR only focuses on the relative comparisons instead of the absolute values. This would be a problem when we port the scores for Dual, where we expect positive intensity scores. We use softplus activation to preserve the larger values in the raw scores. Our earlier attempt with sigmoid activation leads to saturation in the higher score values, which causes issues due to over-exploration.

C DUAL ALGORITHM: ADDITIONAL DETAILS

C.1 DETAILED PROOFS OF PRIMAL AND DUAL SOLUTIONS

Lemma S1. *Show that the solution to* $\max_\pi L_\epsilon(\pi, u, v)$ *is* $\hat{\pi}_{xy} = \text{sigmoid}\left(\frac{\lambda_{xy} - u_x - v_y}{\epsilon}\right), \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$.

Proof. The proof is done by taking derivatives with respect to $\pi_{xy}, \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$. We first expand out the Lagrangian form

$$\begin{aligned}L_\epsilon(\pi, u, v) = &\mathbb{E}_{xy}(\lambda_{xy}\pi_{xy}) - \mathbb{E}_x(u_x(\mathbb{E}_y(\pi_{xy}) - \alpha)) - \mathbb{E}_y(v_y(\mathbb{E}_x(\pi_{xy}) - \beta)) \\ &+ \epsilon\mathbb{E}_{xy}(-\pi_{xy}\log(\pi_{xy}) - (1 - \pi_{xy})\log(1 - \pi_{xy})).\end{aligned} \tag{17}$$

Then, we take a derivative with respect to π_{xy} as

$$\begin{aligned}\frac{\partial L_\epsilon(\pi, u, v)}{\partial \pi_{xy}} &= \lambda_{xy} - u_x - v_y - \epsilon \log(\pi_{xy}) - \epsilon + \epsilon \log(1 - \pi_{xy}) + \epsilon \\ &= \lambda_{xy} - u_x - v_y - \epsilon \log\left(\frac{\pi_{xy}}{1 - \pi_{xy}}\right).\end{aligned}$$

Setting the derivative to zero yields the desired solution

$$\log\left(\frac{\pi_{xy}}{1 - \pi_{xy}}\right) = \frac{\lambda_{xy} - u_x - v_y}{\epsilon} \quad \Rightarrow \quad \pi_{xy} = \text{sigmoid}\left(\frac{\lambda_{xy} - u_x - v_y}{\epsilon}\right).$$

□

Lemma S2. *Show that the solution to* $\min_{u \geq 0} d_\epsilon(u, v)$ *is in the roots of*

$$0 = -r(\hat{\pi}_\epsilon(\lambda_{x:}, u_x, v)) = \alpha - \mathbb{E}_y(\hat{\pi}_\epsilon(\lambda_{xy} - u_x - v_y)),$$

projected to $u \geq 0, \forall x \in \mathcal{X}$*. The other part of* $\min_{v \leq 0} d_\epsilon(u, v)$ *can be similarly derived.*

Proof. The proof is done by plugging in the primal solution from Lemma S1 to the Lagrangian form and then taking derivatives with respect to $u_x, \forall x \in \mathcal{X}$. We first rearrange the Lagrangian form:

$$\begin{aligned}L_\epsilon(\pi, u, v) = \mathbb{E}_{xy}(\underline{(\lambda_{xy} - u_x - v_y)\pi_{xy}}) + \alpha \mathbb{E}_x(u_x) + \beta \mathbb{E}_y(v_y) \\ + \mathbb{E}_{xy}\left(\underline{-\epsilon \pi_{xy} \log\left(\frac{\pi_{xy}}{1 - \pi_{xy}}\right)} - \epsilon \log(1 - \pi_{xy})\right)\end{aligned}$$

This simplifies the substitution of the solutions from Lemma S1. Specifically, the underlined terms may cancel. Define $z_{xy} = \frac{\lambda_{xy} - u_x - v_y}{\epsilon}$, such that $\hat{\pi}_{xy} = \frac{e^{z_{xy}}}{1 + e^{z_{xy}}}$. The rest of the terms are

$$\begin{aligned}d_\epsilon(u, v) &= \alpha \mathbb{E}_x(u_x) + \beta \mathbb{E}_y(v_y) + \mathbb{E}_{xy}(-\epsilon \log(1 - \hat{\pi}_{xy})) \\ &= \alpha \mathbb{E}_x(u_x) + \beta \mathbb{E}_y(v_y) + \mathbb{E}_{xy}\left(\epsilon \log\left(1 + e^{z_{xy}}\right)\right).\end{aligned}$$

Taking the derivative with respect to u_x leads to

$$\frac{\partial d_\epsilon(u, v)}{\partial u_x} = \alpha + \mathbb{E}_y\left(\epsilon \frac{e^{z_{xy}}}{1 + e^{z_{xy}}} \frac{\partial z_{xy}}{\partial u_x}\right) = \alpha - \mathbb{E}_y\left(\frac{e^{z_{xy}}}{1 + e^{z_{xy}}}\right).$$

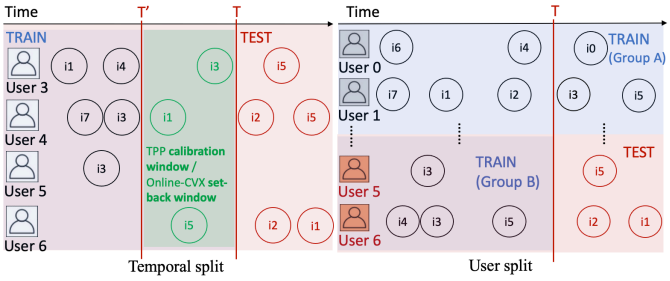
Setting the gradients to zero leads to a solution to the unconstrained problem $\min_u d_\epsilon(u, v)$. Additionally, since the gradients are monotone and independent with respect to $u_x, \forall x \in \mathcal{X}$, we may also obtain a solution to the constrained problem in the space of $u \geq 0$ by direct projections. □

C.2 Complexity and Storage Complexity

With a batch-size of B and an catalog size of $|\mathcal{Y}| = n$, this leads to a temporary storage complexity of $O(Bn)$ and a computational complexity of $O(Bn(\log(n) + \log(B)))$ for each mini-batch. Notice that to enable fast computations on streams of users, we must also avoid the storage of all user-item scores, but use a combination of sparse and low-rank matrices. For example, we may delay the final dot-product layer between the user and item representations in RNN/Transformer models, but instead just store the D-dimensional representations using as little as $O((B + n)D)$ storage. The computational complexity for the dot-product layer is $O(BnD)$, which is comparable with sorting / bisection costs in the Dual algorithm. In our experiments, we use $100m/B$ mini-batches based on m total user states in empirical data.

D Data Statistics and Offline Precision Metrics

Figure/Table S1 shows our data splitting strategies and resulting statistics in the three datasets we consider in Section 5. With temporal split, we carve out a prediction time window between $(T, T + \Delta T]$ and a calibration/validation window between $(T', T]$ for time-bucketed models like Hawkes-Poisson and GCMC and the observation of user-state distribution for Online-Dual simulation. With user split, we pick Group B user to construct a test window between $(T, T + \Delta T]$.



	NF	ML	YC
# warm users	32238	3020	71784
# warm items	16217	3555	11431
# training events	2437151	376736	1087267
# test events	187096	120191	49500
Test start	2005/6/15	1520s	458s
Test window	14 days	1520s	458s
K=1% items	163	36	115
C=1% users	323	31	718

Table S1: Data split and statistics

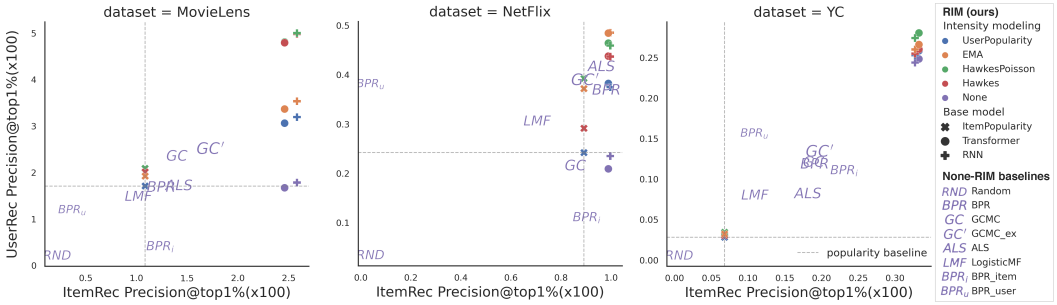


Figure S1: Suitability of different methods for ItemRec and UserRec tasks, plotted on different axes.

For calibration/validation, we use Group A users with the same temporal splits. Our definition of user-state distribution allows sampling by either users or time or both users and time at the same time.

Figure S1 shows the offline precision benchmark results which resemble our Figure 3 on recall metrics in Section 5.1. The trends are similar. We show recall in the main text because it caters more to user-welfare, whereas precision connects to the global objective in Mtch experiments. Conversely, the precision numbers agree with the unconstrained optimization results in Figure 4,S2,S3 and Figure S4,S5,S6 for different combinations of tasks and datasets, respectively.

E ITEMREC WITH MIN-EXPOSURE CONSTRAINTS (NF AND YC)

Similar to the Netflix results in the main text, we show two more Mtch experiments on NF (Figure S2) and YC (Figure S3). Comparing the Pareto fronts, we observe RNN-TPP variants > RNN > MF > Non-personalized models, which is consistent with our results on ML dataset.

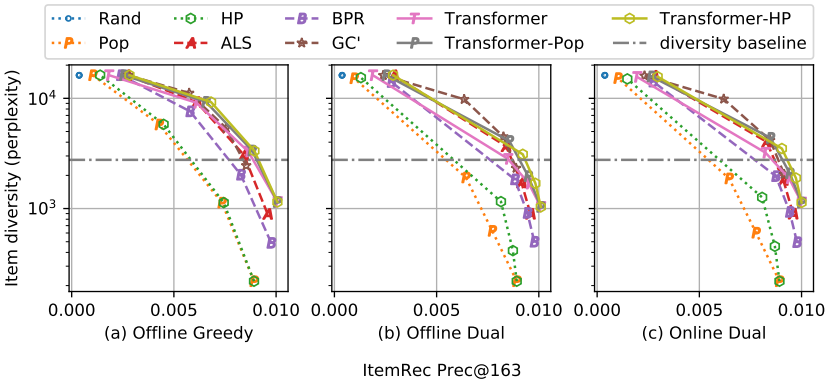


Figure S2: ItemRec with min-exposure constraints in Netflix dataset.

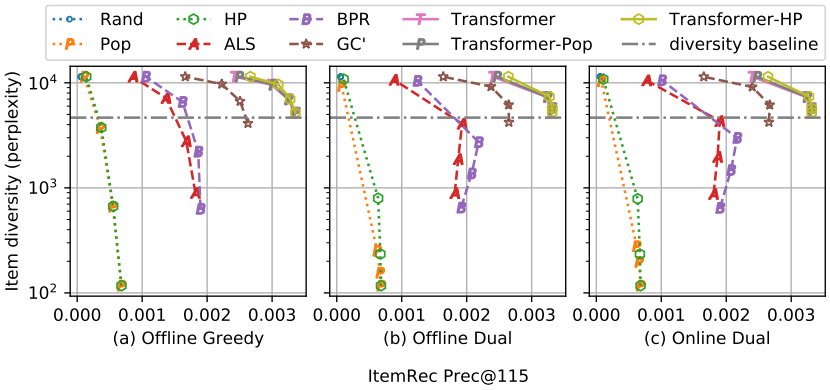


Figure S3: ItemRec with min-exposure constraints in Yoochoose dataset.

F USERREC WITH MAX-USER-CAPACITY CONSTRAINTS

We consider $1\% \leq \alpha \leq 100\%$ and $\beta = 1\%$, both as upper-bound constraints in equation 9. In the limit $\alpha = 100\%$, the problem becomes UserRec without the consideration of user capacities. This yields best UserRec relevance, with a risk to overwhelm the users. As α decreases, the users are expected to take a more balanced load of incoming items. We use a similar perplexity metric to measure the user-load balances.

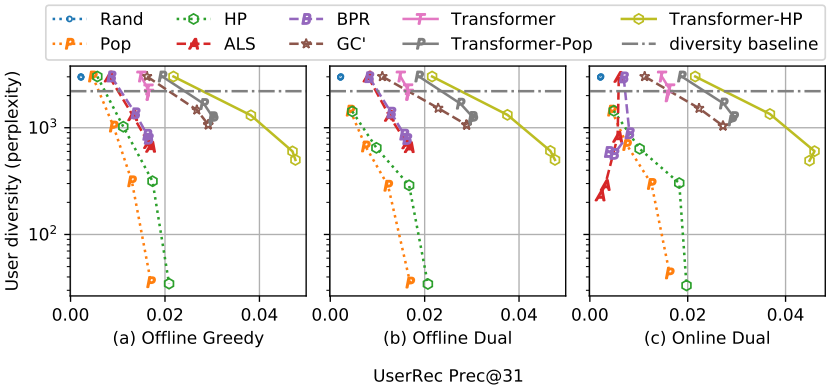


Figure S4: UserRec with user max-capacity constraints in ML-1M dataset.

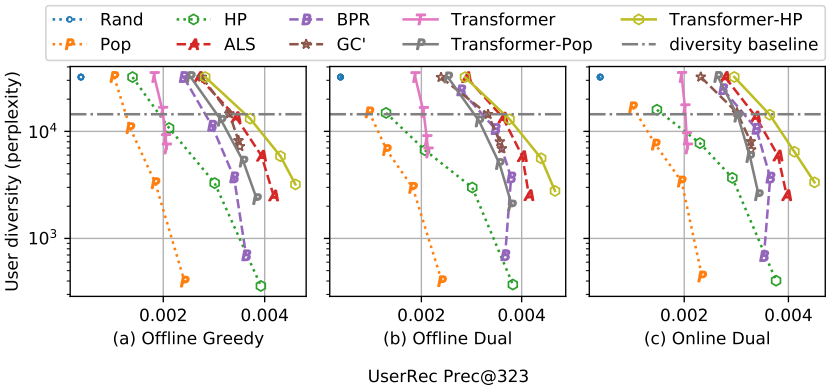


Figure S5: UserRec with user max-capacity constraints in Netflix dataset.

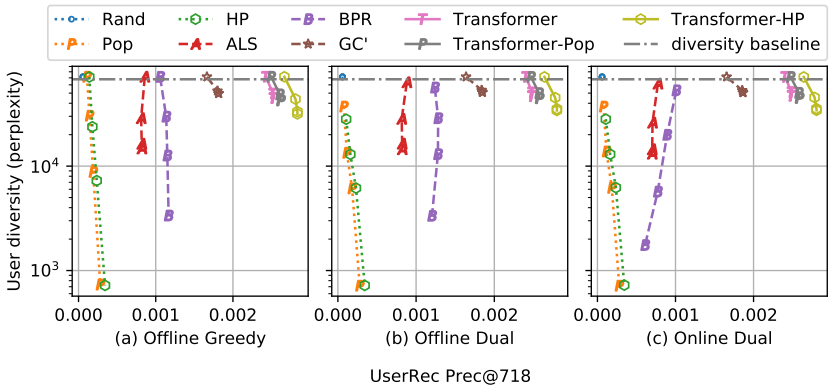


Figure S6: UserRec with user max-capacity constraints in Yoochoose dataset.

Figure S4, S5, S6 show the trade-off between relevance and user diversity, which is an indirect result of capacity limitations per user. From the curves, we find larger advantages in the RNN-TPP variants. This is because the problem is more focused on the advertising perspective, giving larger user-to-item capacity ratio, which highlights our RIM contributions. The rightmost points in the first two plots agree with the unconstrained offline UserRec precision metrics in Figure S1.

For online UserRec, we observe that most methods, including all of RIM and GCMC methods follow similar trends as offline UserRec (Figure S4.c, S5.c, and S6.c). However, we do notice some artifacts in MF methods on ML and YC datasets. The artifacts are related to the under-delivery of recommendations with respect to maximal user-capacity constraints, which leads to drops in the precision metrics. For more details, we show the average percentage of users delivered per marketed item in Figure S7. The ideal percentage is 1% to agree with the β-constraint. Notice that constraint dissatisfaction is a possibility in online ItemRec as well, but they do not affect our general observation of improvements in diversity-relevance trade-offs and we thus omit the discussions for simplicity. Our discussions here extend to online ItemRec as well.

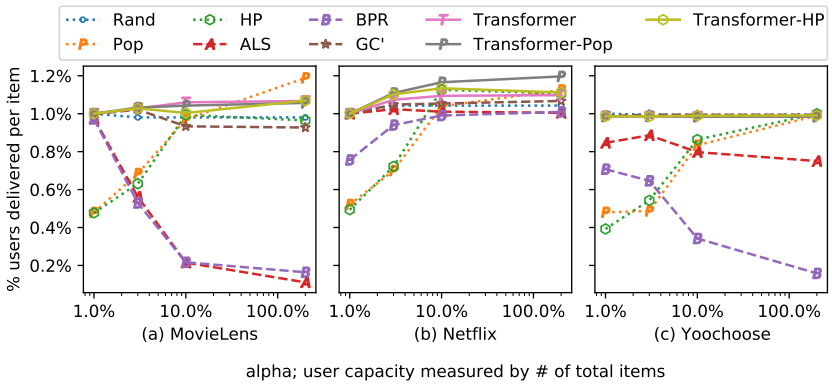


Figure S7: Average percentage of users actually delivered per item. We set $\beta = 1\%$ and $\alpha \geq 1\%$, so the ideal outcome should be 1%. However, when we simulate OnlnMtch in a future period of time, any biases in the empirical user-state distribution could cause the outcomes to be different. We observe the bias to be more obvious in ALS and BPR methods on MovieLens and Yoochoose datasets and less so with our RIM and GC methods in all datasets, showing another benefit of stability with sequential models.

What is special in online UserRec is that the total number of recommendations is highly related to the satisfaction of the β-constraints. For example, in the extreme case where $\alpha = 100\%$ and $\beta = 1\%$, the user capacities are entirely unconstrained, yet the total number of item exposures have to be controlled. The other settings where $1\% \leq \alpha \leq 100\%$ respect the capacity of each individual

user, but still over-provision the overall user capacities for the desired number of item exposures. In these cases, the β-constraints are observed by transferring the dual parameters, $v(y)$, from the calibration set to the test set through either temporal split (Figure S1.a) or user split (Figure S1.b). I.e., Online-Dual does not use any user statistics in the test set, which are unavailable ahead of time in true online settings. Despite so, the β-constraints are largely satisfied with RIM and GCMC methods in all datasets as well as MF methods on NF dataset with temporal splits.

The artifacts are only limited to MF methods on ML and YC datasets, both of which use user-splits. Upon further inspection, we realize that the artifacts are due to the fact that we reuse a part of training data for calibration purposes, i.e., the $[T, T + \Delta T)$ time window for Group-A users in Figure S1.b. This causes Group-A users to have more training data, leading to higher recommendation scores during the learning of $v(y)$. As a result, $v(y)$ tends to be higher, causing over-penalization when they are transferred to Group-B users, who have less training data and generally lower recommendation scores. This eventually translates to overall lower item-exposure rates compared with what is calibrated on the validation set, shown in Figure S7. Notice that these artifacts are much less obvious on NF datasets, where the total training-data time (6 months) overwhelms the difference (15 days) between these two splits. They are neither significant in RIM / GCMC methods, where the user training data is averaged against the user time span, improving the stability of the user-state distribution. In fact, the artifacts with MF methods reveal a side benefit of our RIM (and GCMC) methods in our novel design of online experiments.