# BRIDGING RECOMMENDATION AND MARKETING VIA RECURRENT INTENSITY MODELING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

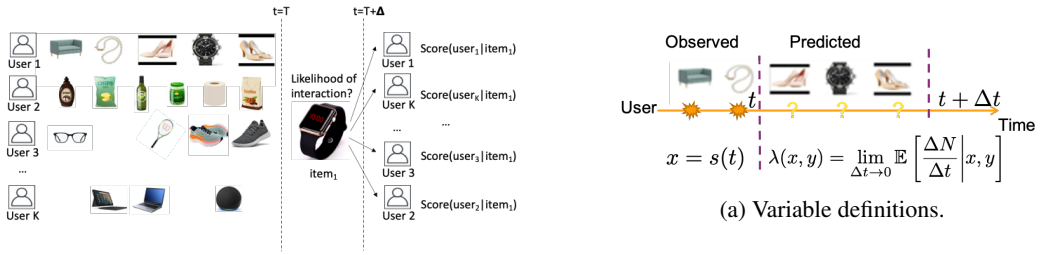
Personalized item-recommendation (ItemRec) and targeted item-marketing are two sides of the same coin. While ItemRec chooses the best items immediately upon user requests, marketing considers the strategic item exposition in user-space and time, under budget constraints. Inspired by recent advances in sequential ItemRec models, we study the reverse problem we call UserRec, which is to select users based on their real-time states on behalf of the item providers. UserRec turns out to be nontrivial and our key insights include a novel connection to Marked-Temporal Point Processes (MTPPs), with which we build a family of models we call Recurrent Intensity Models (RIMs). To further integrate with ItemRec regimes where real-time decisions are required, we discuss and improve an online optimal decision strategy we call CVX-Online, based on the duality theories in convex optimization. The union of ItemRec and UserRec reveals new dimensions for exploration. We conduct experiments on three real-world datasets where we show that RIM is highly effective on both UserRec and ItemRec. With CVX-Online, we also successfully diversify item exposures with minimal compromises in relevance, thanks to the new ability to explore in the user-state distributions. We open-source our evaluation framework with a rich set of baselines.

## 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 producers. 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 producers 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 producers, we must consider the success of the item producers for the long-term richness of the RecSys environment.

We consider a UserRec problem which aims to recommend users on behalf of the item providers, based on their likelihood to interact with a given item in the near future. UserRec is a mathematical generalization of several item-marketing scenarios. For example, item providers may choose to bid on search keywords or previously browsed items as a strategy to increase the exposure of their items. These searching or browsing events may be considered as part of the user-history features, with which the likelihood scores can be predicted. As another example, a user may be in a particular state where a nudging via push notifications could lead to positive outcomes. Instead of hand-crafting a filter on the states, we may learn latent representations based on a likelihood objective for UserRec. While UserRec is a great way to communicate the needs of the item providers, this channel cannot be abused. Bidding too quickly may deplete the advertisers' budgets and sending push notifications too often could turn the users away. Thus, two challenges stand out: how to *predict* the relevance for UserRec and how to *decide* with pre-allocated marketing budgets against future users in real time.

First, the *prediction* problem can be challenging because the user state is *evolving over time*. A past event may stimulate future engagements, whereas sustained inactivity leads to churns. We must




(a) Variable definitions.

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, user hits in a given item are often correlated with their past histories with similar items.

redefine user identity to address *Ship of Theseus Paradox: whether an object that has had all of its components replaced remains fundamentally the same object.* We note that some of these problems are also present in ItemRec in less obvious ways. For example, the state of an item may change after a price change or a phenomenal review (Wu et al., 2017; Ma et al., 2020). Nonetheless, state changes are often more critical with users than items.

To date, most RecSys have adopted a hidden-state user representation to solve online ItemRec with evolving user states. The user states are based on the user's past event histories, passed through sequence models such as recurrent neural networks (RNNs) or Transformers, for the prediction of the item distribution for their next choice (Cho et al., 2014; Hidasi & Karatzoglou, 2018; Ma et al., 2020; Kang & McAuley, 2018). However, the ItemRec formulation may not directly translate to UserRec, because the item choice is formulated as a user-conditional distribution without an identifiable *user prior*. Specifically, due to the softmax activation across all items for the given user, an inactive user may get similar final scores as an active user of the same type. Further, imputing the user prior term with an incorrect "Bayesian" interpretation may lead to a paradox (Figure 2).

The cornerstone in UserRec is to predict the *intensity* of a user's activity in a given item type in a comparable future time window, based on the user's past data. We observe a clear connection to Marked Temporal Point Processes (MTPPs) (Du et al., 2016; Reinhart, 2018), where a complete temporal model includes the prediction of both event-times and types (i.e., items). In the analogy, by placing the TPP-intensity parameters learned from the event time variables, we find theoretical consistency in the final scores to be considered for both UserRec and ItemRec. We call our strategy *Recurrent Intensity Modeling (RIM)*. While we discuss some related MTPP work in appendix, most of them stop at the prediction of event time or data likelihood. We claim novelty in the application of the *intensity* predictions to the UserRec problem, as a solution to the "Bayesian" paradox.

Our next challenge is *online decision* under pre-allocated budgets for a distribution of user(-state)s, often over a period of time. For example, we may aim to expose an item to the Top-1% of all users who may visit the website in the next week and be interested in this item. The challenge is that some UserRec decisions must be made in real-time as the user arrives at a suitable state, e.g., when the user visits the website, despite uncertainties in the user's relevance and/or purchase intents compared with others who may arrive at a later time. This is analogous to online secretary/marriage problem, where at most one candidate can be selected for the whole time and we aim to do so optimally with minimal regrets in hindsight. Further, we focus on the interaction between multiple users and items for their combined benefits. We develop an independent solution that resembles online dual decomposition (Huang et al., 2016; Mehta, 2013; Jenatton et al., 2016), yet we introduce entropic regularization similar to Sinkhorn's algorithm (Cuturi, 2013) to improve its convergence speed in our experiments. We call this algorithm Online-CVX and it complements our RIM predictions to complete a full story of online user-item matching (OnlnMtch) with broader impacts.

Additional discussions on related work can be found in the appendix. Our contributions are:

1. A novel connection between UserRec and ItemRec via Recurrent Intensity Modeling;
2. An improved Online-CVX program to handle long-term budget/exposure constraints;
3. Empirical results showing RIM's effectiveness in UserRec and ItemRec, and its superiority in managing relevance-diversity trade-offs;

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

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

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

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.

4. All RIM model components: RNN, TPP, CVX, can scale to infinite user pools via mini-batching.

## 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 and a type variable (i.e., the consumed item in the event) in a large-cardinal set $\mathcal{Y}$. 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.

### 2.1 Marked Temporal Point Processes (MTPP)

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 single variable $x$, instead of a TPP, $s(t)$.

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 events, $\Delta N(x,y)$, in a future 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:

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

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.[1] The Poisson log-likelihood is:

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

[1] Poisson may be over-dispersed in longer horizons, see discussions in Salinas et al. (2020).

The definition of the intensity parameter is visualized in Figure 1. 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 $\Delta t$ to connect it to the distribution of future event times. We expect an active user to have shorter intervals between events and higher intensities in unit time.

**Temporal Point Processes (TPPs).** We focus on the connection between the intensity parameter $\lambda(x) = \lambda(t; s(t))$ and the event-time variable $t$ for a specific item $y$ which we omit. Intuitively, we split the time interval between two adjacent events into a large set of infinitesimal windows, i.e., $(t_k, t_{k+1}] = \cup_\tau (\tau, \tau + \mathrm{d}t]$, 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

$$\sum_\tau \log P\left(\mathrm{d}N(\tau) = n(\tau) | \lambda(\tau; s(\tau))\mathrm{d}t\right) = \log\left(\lambda(\tau_*; s(\tau_*))\mathrm{d}t\right) - \sum_\tau \lambda(\tau; s(\tau))\mathrm{d}\tau$$

$$\overset{\mathrm{d}\tau \to 0}{\Rightarrow} \quad \log p(T_{\text{next}} = t_{k+1} | t_k; \lambda) = \log \lambda(t_{k+1}; s(t_{k+1})) - \int_{t_k}^{t_{k+1}} \lambda(\tau; s(\tau))\, \mathrm{d}\tau. \tag{4}$$

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 $T_{\text{next}}$. 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 explicit bucket time in equation 2. The intensity functions can further be time-varying based on user states. 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.

**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 in equation 1. MTPP resembles Bayes rule and considers the user sequence $x = s(t)$ as a point process. It is the framework of our RIM proposals.

## 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 equation 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 user histories from the collected data. This approach works due to a homogeneous intensity assumption, i.e., the number of past events directly correlates with the future event intensity. It also assumes that all user activities are collected over a comparable period of time, which often holds when, in practice, RecSys often truncates the training datasets by a fixed time for scalability concerns. This is the example shown 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 sustained inactivity. It models these effects via an exponentially-decaying kernel in the intensity parameters

$$\lambda(t) = \mu + \sum_{j: t_j < t} \sum_{r=1}^{R} \alpha_r \phi_r(t - t_j), \tag{6}$$

$$\text{where } \phi_r(t) = \tfrac{1}{s_r} \exp\left(-\tfrac{t}{s_r}\right) 1_{t>0}, \text{ and } \mu, \alpha, s > 0.$$

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,

$$\bar{\Lambda} \sim \mathbb{E}[\Delta N(x, y)] = \mathbb{E}\left[\int_t^{t+\Delta t} \lambda(\tau; \mathcal{S}(\tau), y)\, \mathrm{d}\tau \middle| \mathcal{S}(t) = x\right], \tag{7}$$

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

$$\bar{\Lambda} = \phi(t) \odot \mathrm{softplus}(w) = \phi(t) \odot \log(1 + \exp(w)). \tag{8}$$

Our additive formulation is inspired by (Mei & Eisner, 2017) and we slightly improve it by allocating the softplus to each coordinate to improve numerical stability.

**Other Baseline Methods** Practical item 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 in equation 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.

## 4 Recommendation as a Matching Problem

Now that we have described *how* we create RIM scores to allow UserRec by comparing the users' real-time states, with the obvious benefits of simplification and generalization thanks to the continuous-time auto-regressive nature in TPP/RNN/Transformer models, we then suggest *why* this may lead to broader impacts. Our main proposal is to integrate UserRec into ItemRec to form a matching problem (Mtch). Mtch caters to the benefits of both users and items, where the item provider's benefits are shown as exposure guarantees with best user choices on their behalf. We will frame the problem, address its eminent scalability challenges, and discuss its connection to online optimal decision processes, as time is an irreversible direction in the user-state distribution.

**Formulation of the Matching Problem** In UserRec, we make marketing decisions in the space of user-time that is appropriate for the application, e.g., to consider push notifications for all active users at a given time or users in a particular stage of an experience. We represent the user-time space as $\mathcal{X} \ni x$. UserRec then picks Top-$C$ users, which can equivalently be represented as the solution of a (multi-)assignment vector $\pi(x|y) : \mathcal{X} \to \{0, 1\}$:

$$\max_{\pi(x|y)\in\{0,1\}} \sum\nolimits_{x\in\mathcal{X}} \left[\pi(x|y)\lambda(x, y)\right], \text{ s.t. } \sum\nolimits_{x\in\mathcal{X}} \pi(x|y) \le C, \forall y \in \mathcal{Y}. \tag{9}$$

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

For online marketing or with large user pools, the user states can be seen as a draw from a distribution $P(\mathcal{X})$. We may replace the summation in equation 9 with Lebesgue integral in the probability space. In fact, we further extend the assignment problem to consider constraints in both UserRec and ItemRec for an Mtch solution $\pi : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ that maximizes the global objective

$$\max_{0 \le \pi(x,y) \le 1, \forall x, y} \mathbb{E}_{x,y} \left[ \pi(x, y)\lambda(x, y) \right], \quad \text{s.t.} \quad \begin{cases} \mathbb{E}_y[\pi(x, y)] \le \alpha(x), & \forall x \in \mathcal{X}; \\ \mathbb{E}_x[\pi(x, y)] \le \beta(y), & \forall y \in \mathcal{Y}, \end{cases} \tag{10}$$

where $\alpha(x)$ and $\beta(y)$ are user and item capacities in relative terms. Let $m = |\mathcal{X}|$ and $n = |\mathcal{Y}|$. Choosing $\alpha(x) = K/n$ and $\beta(y) = 1$ (unconstrained) reduces to the traditional Top-K ItemRec problem. Choosing $\alpha(x) = 1$ and $\beta(y) = C/m$ reduces to the offline UserRec problem equation 9. In its most general form, equation 10 simultaneously solves for both ItemRec and UserRec to control the global exposure rates of different items. We may also replace the $\beta$-constraints with lower bounds to guarantee the minimal exposure of the given items. This corresponds to a guaranteed delivery contract in display advertising and our diversity pursuits by promoting cold-start items.

### 4.1 Offline Greedy Solution

We simply sort the $\lambda(x, y)$ in both users and items and then match user-item pairs one at a time until the constraints are satisfied. For lower-bound $\beta$-constraints, we first solve for the common factors by setting $\alpha = \min(\alpha, \beta)$, followed by a "top-off" of the remaining user capacities with respect to the original $\alpha$. While Offline-Greedy appears fast with in-memory datasets, costing $O(mn \log(mn))$ time, it has fundamental limitations in the lack of mini-batch sampling from infinite user distributions or the adaptation to online decision scenarios, which we aim to resolve.

### 4.2 Offline CVX Solution

We craft a solution via online dual decomposition in convex optimization (CVX), which is analogous to quantile-thresholding $v(y)$ in the marketing of a single item $y$, i.e. $\pi(x, y) = 1 \Leftrightarrow \lambda(x, y) > v(y)$ when a new user-state $x$ is considered. By using online algorithms, we may learn $v(y)$ on streams of user mini-batches. Our CVX solution further resolves the competing conditions of multiple co-occuring item-marketing campaigns. We propose an improvement over online subgradient methods (Huang et al., 2016; Mehta, 2013; Jenatton et al., 2016) with faster convergence in our experiments.

We first describe the subgradient CVX approach as it yields stronger intuitions . The general idea is to convert the constraints in equation 10 as Lagrangian multiplers:

$$\max_{\substack{0 \le \pi \le 1 \\ r_x(\pi) \le 0}} \min_{v \ge 0} L(\pi, v) = \mathbb{E}_{x,y}[\lambda(x, y)\pi(x, y)] - \mathbb{E}_y[v(y)r_y(\pi)], \quad \text{where } r_x(\pi) = \mathbb{E}_y[\pi(x, y)] - \alpha(x) \text{ and } r_y(\pi) = \mathbb{E}_x[\pi(x, y)] - \beta(y). \tag{11}$$

The Lagrangian form is parametrized by $(\lambda, \alpha, \beta)$ and it introduces dual variables $v(y)$ to penalize constraint violations with the inner-loop, e.g., setting $\min_{v(y) \ge 0}[-v(y)r_y(\pi)] = -\infty$, if $r_y(\pi) > 0$ violates the Top-K ItemRec constraint. When we use minimal-exposure constraints to promote diversity, i.e., bounding $r_y(\pi) \ge 0$, the corresponding dual requirement changes to $v(y) \le 0$.

The Lagrangian approach converts the constrained optimization to a minimax saddle point problem. The next step is to change the order of operations to form its dual problem:

$$\min_{v \ge 0} \max_{\substack{0 \le \pi \le 1 \\ r_x(\pi) \le 0}} L(\pi, v) = \min_{v \ge 0} \left\{ \mathbb{E}_x \left[ \max_{\substack{0 \le \pi \le 1 \\ r_x(\pi) \le 0}} \mathbb{E}_y[(\lambda(x, y) - v(y))\pi(x, y)] \right] + \mathbb{E}_y[v(y)\beta(y)] \right\} \tag{12}$$

Note that the inner-max loop can be solved by picking the top items by the modified ranking scores,

$$\pi(x|y, v) = \text{TopK}\{\lambda(x, y) - v(y) : \lambda(x, y) - v(y) \ge 0\}, \tag{13}$$

while the outer loop is convex. The dual problem can be solved by stochastic (sub-)gradient descents (SSGD) through all users in mini-batches, with decreasing learning rates such that $\sum_k \eta_k = \infty, \sum_k \eta_k^2 < \infty$ to guarantee its convergence. In practice, however, we observe slower convergence with oscillatory patterns (Figure 3). This is because the dual objective is non-differentiable when $\lambda(x, y) - v(y) = 0$, causing the solver to bounce between the opposing sides in the convex hull. Besides, setting one learning rate for all dimensions may not handle heterogeneous score distributions.

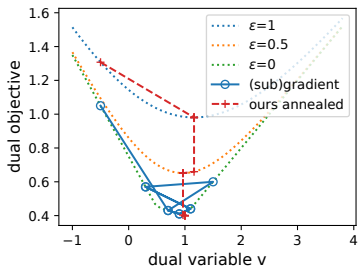



Figure 3: Toy problem with two users and one item with score matrix $(\lambda(x, y)) = (1, 0.5)^\top$, subject to UserRec equality constraint $\mathbb{E}_x[\pi(x, y)] = 0.4$. We plot the dual objective with different entropic-regularizer $\epsilon$ as well as the solution paths of different methods. As $\epsilon \to 0$, (sub)gradient steps become oscillatory without carefully-designed decreasing learning rates. To avoid the tuning of step sizes, we develop a novel solution via annealed alternating optimization, which converges better in our experiments.

**Entropic Regularization and Alternating Optimization** To improve practical convergence of CVX on the set of observed users, we independently develop an entropic-regularized version with direct inspirations from Sinkhorn's algorithm in optimal transport (Cuturi, 2013). The trick is to replace the $0 \leq \pi \leq 1$ constraint with an entropic regularizer, $H(\pi) = -\pi \log(\pi) - (1 - \pi) \log(1 - \pi)$. To smooth out the objective function, we further incorporate the primal constraint $r_x(\pi) \leq 0$ via another dual variable $u(x) \geq 0$. The new dual objective and Lagrangian form become

$$\min_{u \geq 0, v \geq 0} d_\epsilon(u, v) = \left[ \max_\pi \Big( L(\pi, v) - \mathbb{E}_x[u(x) r_x(\pi)] + \epsilon \mathbb{E}_{x,y}[H(\pi(x, y))] \Big) \right], \tag{14}$$

where the new Lagrangian, if expanded, is symmetric in $u(x)$ and $v(y)$. As a subroutine in the dual problem, we find an explicit solution to the primal variable $\pi(x, y)$ as:

$$\pi(x, y) = f(u(x), v(y), \lambda(x, y)) = \text{sigmoid}\left( \frac{\lambda(x, y) - u(x) - v(y)}{\epsilon} \right), \tag{15}$$

which is similar to equation 13 but with soft assignments. Plugging $\pi$ in the dual form leads to functional gradients with respect to the dual variables: $\frac{\partial d_\epsilon(u,v)}{\partial u(x)} = -r_x(f(u(x), v, \lambda))$ and vice-versa for $v(y)$. Particularly, the dual gradients are monotone in every coordinate of $u(x)$ and $v(y)$, which allows efficient bisection search for their roots in the range of: $\inf_{y \in \mathcal{Y}}[\lambda(x, y) - v(y)] \leq \epsilon \sigma^{-1}(\alpha(x)) + u(x) \leq \sup_{y \in \mathcal{Y}}[\lambda(x, y) - v(y)]$, and vice-versa for $v(y)$.

The introduction of $\epsilon$ allows exact solutions for $u(x)$ and $v(y)$ in $O(\log n)$ time in each step of alternating optimization. Alternating with exact solutions is more efficient than subgradient descents in our experiments, illustrated in Figure 3. See Algorithm 1 and further discussions in Appendix C.

### 4.3 Online CVX Solution

Similar to online secretary / marriage problem, Online Matching (OnlnMtch) considers the optimal choices of future users for item exposition under a pre-allocated budget constraint for the whole time. This is made possible by the CVX programs by carrying over the learned $v(y)$ variables to future users, assuming that the past and future users share similar state distributions over time. In essence, we subtract the RIM score $\lambda(x, y)$ by the CVX parameter $v(y)$ to make assignment decisions $\pi(x, y)$. Notice that the absence of either RIM or CVX disables OnlnMtch.

While we assume that user-state distributions stay constant, Mehta (2013); Huang et al. (2016); Jenatton et al. (2016) further analyze the adversarial robustness against online changes which may lead to unsatisfied constraints and/or sub-optimal objectives in hindsight. We leave this as future work, as we can show significant empirical improvements in item diversity despite the said challenges.

## 5 Experiments

We conduct experiments in three RecSys datasets with unique properties (code available at anonym).

- **Movielens (ML)** (Harper & Konstan, 2015) is a movie rating website that collects user preferences to study RecSys. Over time, it released several versions of datasets. 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.

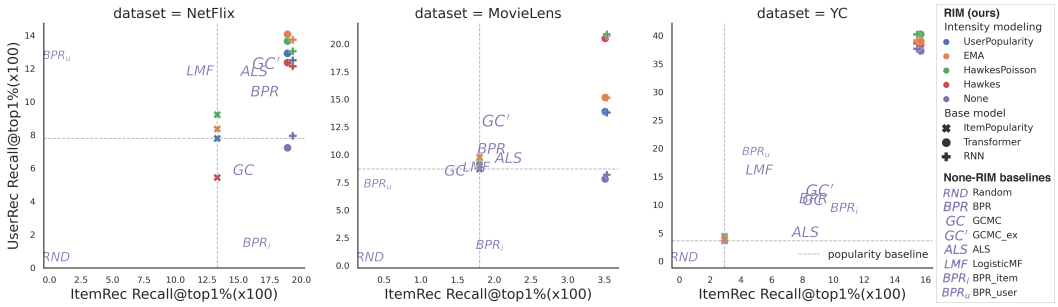



Figure 4: Suitability of different methods for ItemRec and UserRec tasks, plotted on different axes.

- **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-CVX 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 user-marginal 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.

### 5.2 Offline and Online Matching Experiments

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

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

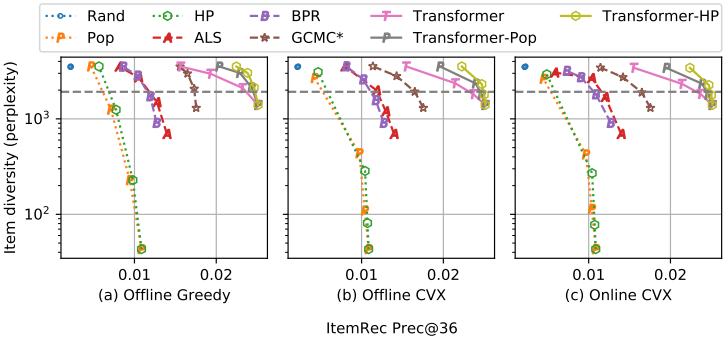



Figure 5: Mtch experiments where we set minimal-exposure constraints for every item in additional to regular ItemRec settings. Showing a subset of representative methods on ML dataset.

To validate the discussions in Section 4, we integrate UserRec into ItemRec by setting minimal-exposure constraints. We fix $\alpha = 1\%$ in equation 10 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 $\beta$, we observe an increase in item-exposure diversity, measured by

$$\text{perplexity} = \exp\left(-\sum_y \left[\left(\frac{\pi(y)}{\sum_y \pi(y)}\right) \log\left(\frac{\pi(y)}{\sum_y \pi(y)}\right)\right]\right), \tag{16}$$

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

Figure 5 compares the performance of various methods as well as three Mtch strategies: Offline-greedy, Offline-CVX and Online-CVX. Within each Mtch strategy, 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 with different Mtch solvers in Figure 5(b)&(c). Notably, the Online-CVX solver allows some users or user events to be unseen during training and the match-maker has to make real-time decisions for each user without buffering. The consistent performance across different scenarios demonstrates great practicality in the integration of ItemRec and UserRec in RecSys.

## 6 Conclusion

We study UserRec in an online RecSys paradigm, where each user is represented by their history of item consumption events, generalizing a variety of item-marketing scenarios. This problem is of great values in keeping the diversity and liveliness of RecSys; by addressing the pain points in item marketing, the item creators can focus on their expertise in creation. Our main challenge is to compare user sequences in variable lengths and duration. We novelly connect this challenge to the intensity parameter estimation of a temporal point process. With the insights, we develop a three-part RIM solution combining RNN (Transformer), TPP, and CVX, where all component adhere to the same user-mini-batch learning strategy, allowing RIM to achieve unlimited scalability in the size of the user set. Our models evaluate well in both UserRec and ItemRec on three RecSys datasets that were previously used mainly for ItemRec.

RecSys is an area of broad social impacts. The techniques we develop are aimed to improve the diversity and welfare of all participating parties, i.e., the users and items. We use global optimality as a surrogate in Mtch reformulation of RecSys. We show promising connections between RIM and better relevance-diversity trade-offs. On the flip side, bridging recommendation and marketing may give us better tools to understand some unintended consequences of modern RecSys.

## REFERENCES


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.

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.

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*, pp. 3163–3171, 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*, pp. 208–214. JMLR Workshop and Conference Proceedings, 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, pp. 3153–3161, 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, pp. 219–228, New York, NY, USA, 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.

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.

Maciej Kula. Metadata embeddings for user and item cold-start recommendations. In Toine Bogers and Marijn Koolen (eds.), *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.*, volume 1448 of *CEUR Workshop Proceedings*, pp. 14–21. CEUR-WS.org, 2015.

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.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf`.

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.

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.

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.

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.

**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 CVX 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.

**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 action and the change in hazard rate in a user's temporal state. While (Granger) causality is a useful consideration in general, e.g., to prevent sending bad emails with large negative effects, it appears more relevant when the marketed item is a discount coupon and the goal is to attract cold or inactive users. Some interesting work on temporal causality include (Xu et al., 2016; Li et al., 2018; Jiang & Li, 2016; Lundberg & Lee, 2017).

# 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:

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

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.

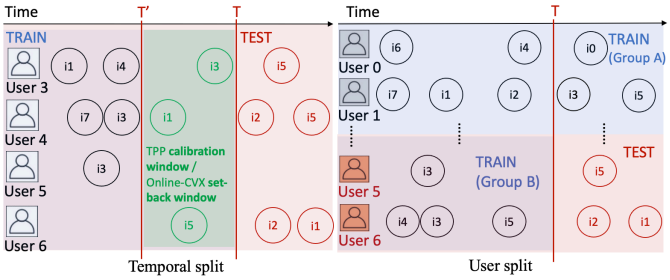
Finally, the BPR scores are translation-invariant, because BPR only focuses on the relative comparisons instead of the absolute values. This would be a problem when we port the scores for CVX, where we expect positive intensity scores. We use softplus activation to preserve the larger values in the raw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

## C CVX Algorithm and Discussions

**Algorithm 1** CVX for offline or online matching

**Require:** $\lambda(x, y), \alpha(x), \beta(y)$, batch-to-full ratio $\gamma$
**Ensure:** Dual variable $\{v(y) : \forall y \in \mathcal{Y}\}$
  Init $v^0(y), \forall y$ randomly within sign-constraints
  **for** $k$ in $[0, 1, \ldots, 100]$ **do**
    Receive user-minibatch $X$; compute $\{\lambda(x, y) : \forall x \in X, y \in \mathcal{Y}\}$; set $\epsilon = 0.9^k$
    Find root $\frac{\partial d_\epsilon(u,v)}{\partial u(x)} = 0$ for every $x \in X$ given $v^k(y)$; project to $u(x) \geq 0$
    Find root $\frac{\partial d_\epsilon(u,v)}{\partial v(y)} = 0$ for every $y \in \mathcal{Y}$ given $u(x)$; project if needed
    Take step $v^k(y) = (1 - \gamma)v^{k-1}(y) + \gamma v(y), \forall y$
  **end for**

Compared with the original subgradient solutions, our $\epsilon$-regularized approach yields similar or better computational efficiency. Both approaches are trained by iterating through the user pool via mini-batches just like a sequence model. They also cost similar amounts of computation for each user, $O(|\mathcal{Y}|)$, with additional constants in $\log(K)$ or $b$-bits of desired numerical precision, respectively. The true benefit of our proposed $\epsilon$-method is in its alternating-direction minimization of $v(y)$ based on mini-batches of users. This automatically tunes for the optimal learning rate which we find tricky to design in the practical application of subgradient methods. On the limitation side, the alternating solution can take overly small steps if we directly input a small $\epsilon \to 0$. We mitigate this issue by annealing $\epsilon$ from a larger starting point.




| | 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

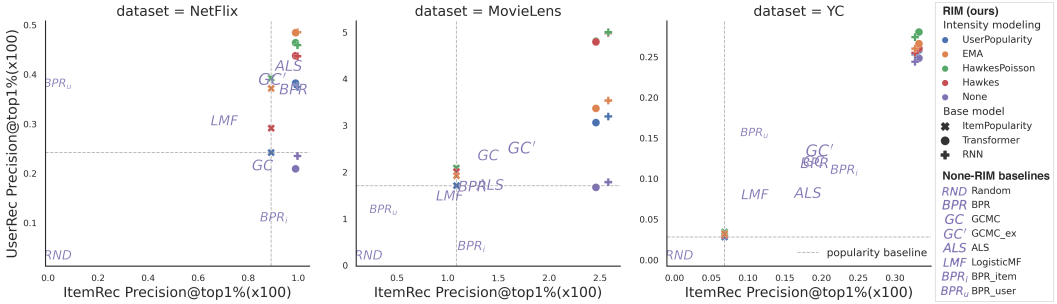



Figure S1: Suitability of different methods for ItemRec and UserRec tasks, plotted on different axes.

## 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-CVX simulation. With user split, we pick Group B user to construct a test window between $(T, T + \Delta T]$. 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 4 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 5,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.

## F UserRec with Max-User-Capacity Constraints

We consider $1\% \leq \alpha \leq 100\%$ and $\beta = 1\%$, both as upper-bound constraints in equation 10. 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 $\alpha$ 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.

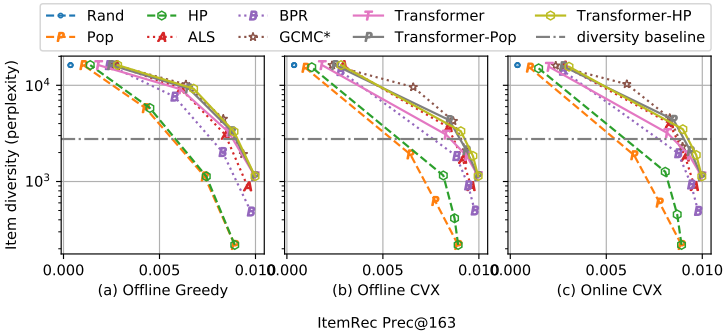



Figure S2: ItemRec with min-exposure constraints in Netflix dataset.

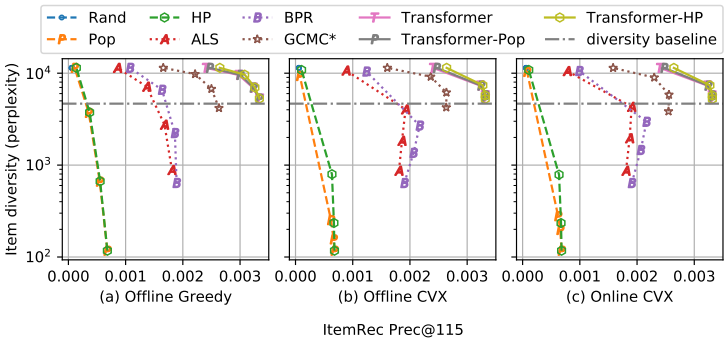



Figure S3: ItemRec with min-exposure constraints in Yoochoose dataset.

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 under-serving of recommendations with respect to maximal user capacity constraints, which leads to drops in the precision metrics. For more details, we show the total number of recommendations relative to the $\beta$-constraints in Figure S4.d, S5.d, and S6.d. 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.

What is special in online UserRec is that the total number of recommendations is highly related to the satisfaction of the $\beta$-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 $\beta$-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-CVX does not use any user statistics in the test set, which are unavailable ahead of time in true online settings. Despite so, the $\beta$-constraints are largely satisfied with RIM and GCMC methods in all datasets as well as MF methods on NF dataset with temporal splits (Figure S4.d, S5.d, and S6.d).

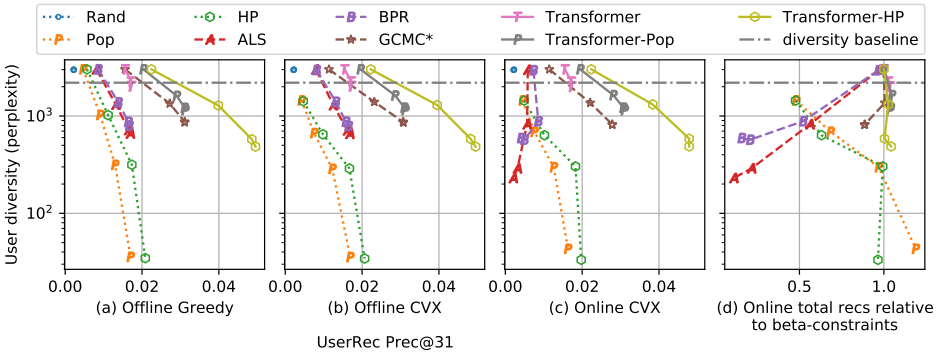



Figure S4: UserRec with user max-capacity constraints in ML-1M dataset.

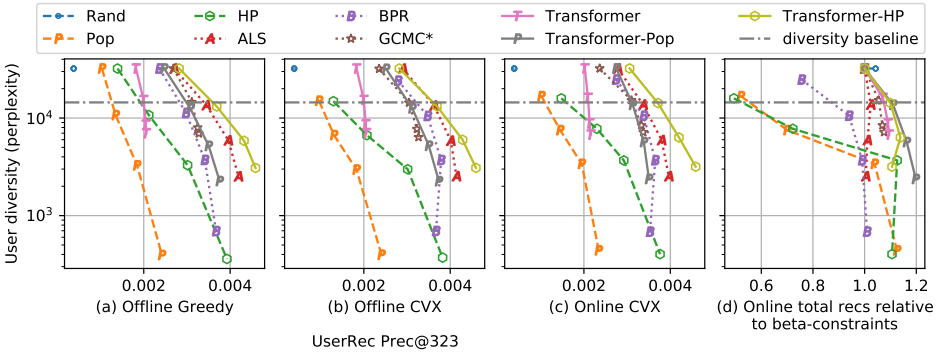



Figure S5: UserRec with user max-capacity constraints in Netflix dataset.

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 S4.d and Figure S6.d. 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.

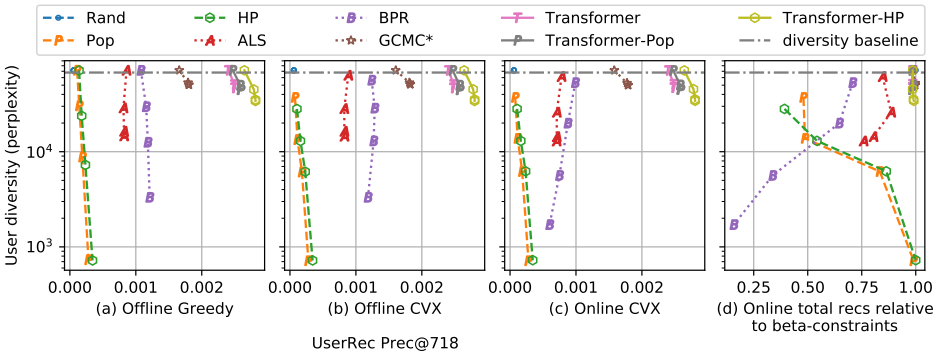



Figure S6: UserRec with user max-capacity constraints in Yoochoose dataset.