BRIDGING RECOMMENDATION AND MARKETING VIA RECURRENT INTENSITY MODELING

Anonymous authors
Paper under double-blind review

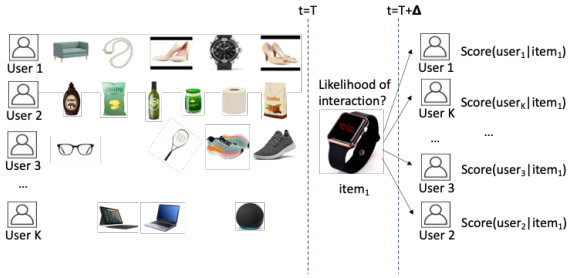
ABSTRACT

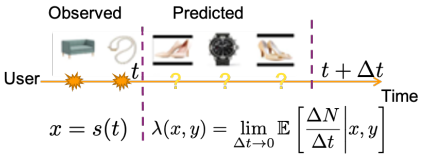
Personalized item-recommendation (ItemRec) and targeted item-marketing are two sides of the same coin. The key in the latter problem, which we call UserRec, is to recommend users in their temporal contexts on behalf of the item providers. As ItemRec progresses to using end-to-end sequence models which directly ingest the item-consumption history of each user to build real-time hidden states, we study whether we can turn these models around to recommend users in their real-time contexts. Our main challenge is to compare these item-consumption sequences of variable lengths and duration. To overcome it, we discover a novel connection to Temporal Point Process (TPP), where we use its intensity parameter to predict the user's likelihood of future engagements, based on but not biased by the past observations. We develop a Recurrent Intensity Modeling (RIM) trick to convert sequential ItemRec models to UserRec based on Marked-TPP decomposition. RIM allows us to rethink recommendation in a Matching (Mtch) scenario, where the benefits of the users (e.g., ItemRec relevance) and item providers (e.g., item-exposure guarantees) are considered at the same time. Notably, as RIM is based on sequence models, we extend to Online Matching (OnlnMtch) to serve ItemRec in real-time while satisfying the item guarantees by approximate ratios in the long-term, borrowing techniques from online dual decomposition in convex optimization. Finally, to complement our discussions on RIM and Mtch, we provide empirical simulations on real-world datasets with open-source codes.

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 consider to 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, while sending too many push notifications could get the user disengaged in return. Therefore, two prominent issues stand out: how to maximize the relevance of UserRec and how to control for the marketing budgets.





(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.

First, the relevance 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 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), for the prediction of the exact item choice upon their arrival (Cho et al., 2014; Hidasi & Karatzoglou, 2018; Ma et al., 2020). 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*. To make matters worse, we must compare users with different history lengths and activity patterns for their likelihood of future engagements, which is otherwise omitted in ItemRec because it serves one user at a time.

The cornerstone in UserRec is to model user-marginal priors unbiasedly to predict the intensity of user-item engagements in comparable time windows. 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-time 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 work stops at the prediction of event time or data likelihood. We claim novelty in the application of the *intensity* parameters to form unbiased user-marginal priors. In contrast, a common mistake, where we fell prey in the past, is to naively consider the user-marginal prior based on the *data likelihood* of the observations in the user's past events. The difference between data likelihood and intensity is shown in equation 4. An obvious sign that the data likelihood solution may be flawed is that it introduces biases against users with longer history lengths.

Our next challenge is on budget allocation, often over multi-day marketing campaigns. An example is to control the item-exposure rate with respect to the number of user visits to the RecSys. 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 RecSys, whereas the final set of recommended users should stay close to the offline optimum in retrospect, i.e., with top RIM-scores. An intuitive solution is to approximate the empirical score distribution and make decisions by checking against the percentage thresholds. To extend this idea to competing item-marketing campaigns and to further improve its empirical convergence, we develop a strategy that combines online dual decomposition (Huang et al., 2016; Mehta, 2013; Jenatton et al., 2016) and Sinkhorn's algorithm in optimal transport (Cuturi, 2013).

To summarize, our contributions are:

1. A novel connection between UserRec and ItemRec via Recurrent Intensity Modeling;

2. An improved convex optimization(CVX) algorithm 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;
4. All RIM model components: RNN, TPP, CVX, can scale to infinite user pools via mini-batching.

Additional discussions on related work can be found in the appendix.

2 Problem Formulation

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 out 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. The lack of calibration of $m(x)$ is related to the fact that the ItemRec formulation excludes the prediction of the event times. In the next section we show how this gap can be fixed and, as a hint, we overload the notation with temporal intensity parameters.

2.1 Recurrent Intensity Modeling (RIM)

To calibrate $\lambda(x, y)$ for both ItemRec and UserRec problems, we modify the prediction targets. We model the absolute number of user-item events, $\Delta N(x, y)$, in a future time window, $(t, t + \Delta t]$, where t coincides with the observation boundary in 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 distribution on counting numbers, especially in short time windows. The Poisson likelihood is:

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

A visualization of the intensity parameter is shown in Figure 1. The intensity parameter connects three variables of our interests: user sequence x, item y and the expected number of xy-engagements in time window $(t, t + \Delta t]$. Slicing $\Delta N(x, y)$ in different directions yields different aspects of the problem. Proposition 1 shows how it connects back to equation 1 by slicing in y with fixed x and t.

Proposition 1 (Poisson-Multinomial Connection)**.** *Suppose $\{\Delta N(y) \sim \text{Poisson}(\lambda(y)\Delta t) : y \in \mathcal{Y}\}$ is a set of independent Poisson random variables. Then, conditioned on the number of total events $\sum_{y \in \mathcal{Y}} \Delta N(y) = n$, the distribution of each variable follows a multinomial distribution $(\Delta N(y)|n) \sim \text{Mult}(n, P)$, where $P(y) = {}^{\lambda(y)}/_{\sum_{y'} \lambda(y')}$. The categorical distribution in equation 1 is a special case with $n = 1$, omitting the dependency on x.*

Matrix Factorization (MF). Next, we consider direct prediction of $\Delta N(x, y)$ for all xy-pairs on fixed time windows $(t, t + \Delta t]$. This yields implicit-feedback MF, such as BPR (Rendle et al.,

2009), ALS (Hu et al., 2008), and ImplicitMF (Johnson, 2014). The drawback is that each user is considered only once per time window. To consider rolling-predictions like sequence models, we must construct multiple time windows and, on top of that, suffer from the discretization errors by using fixed time windows instead of modeling the continuous time variables.

Temporal Point Processes (TPPs). To address these drawbacks, TPPs allow us to learn the intensity parameter λ by direct likelihood in the event-time variable t. Intuitively, they break the time interval between two adjacent events, $(t_k, t_{k+1}]$, into infinitesimal windows $\cup_\tau(\tau, \tau + \mathrm{d}t]$ and assign $n = 0$ to all previous windows and $n = 1$ only to the last window. By repeated application of equation 3 on these infinitesimal windows, we obtain likelihood

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

omitting dependencies on y. Note that the $\mathrm{d}t$ inside the logarithmic term in equation 3 is removed when we convert the probability mass function to probability density function in continuous variable T_{next}. Reinhart (2018); Chen (2016) discuss rigorous derivations as well as additional properties.

TPPs are much improved over MF. They model exact time variables and utilize rolling-prediction just like sequence models. 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 sequence ItemRec models. Instead, we get back to Proposition 1 for a MTPP decomposition, $\lambda(x, y) = \lambda(x)p(y|x)$, where $p(y|x)$ may come from a regular sequence model in equation 1. MTPP is the framework of our RIM proposal. We will discuss more modeling details in Section 3.

2.2 Applications

Our main contribution in RIM is to consider its broader impacts in applications such as UserRec and user-item Mtch. UserRec finds users who are likely to engage with a given item y. Thanks to RIM, the decision can be 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 be equivalently represented as the solution of an assignment vector $\pi(x; y) : \mathcal{X} \to \{0, 1\}$:

$$\max_{\pi(x;y)\in\{0,1\}} \sum\nolimits_{x\in\mathcal{X}} \left[\pi(x; y)\lambda(x, y)\right], \text{ s.t. } \sum\nolimits_{x\in\mathcal{X}} \pi(x; y) \le C, \forall y \in \mathcal{Y}. \tag{5}$$

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 5 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

$$\max_{0\le\pi(x,y)\le 1, \forall x,y} \mathbb{E}_{x,y}\left[\pi(x, y)\lambda(x, y)\right], \text{ s.t. } \begin{cases} \mathbb{E}_y[\pi(x, y)] \le \alpha(x), & \forall x \in \mathcal{X}; \\ \mathbb{E}_x[\pi(x, y)] \le \beta(y), & \forall y \in \mathcal{Y}, \end{cases} \tag{6}$$

where $\alpha(x)$ and $\beta(y)$ are user and item capacities in relative terms. Choosing $\alpha(x) = 1$ (unconstrained) and $\beta(y) = {}^{C}/_{|\mathcal{X}|}$ reduces to the offline UserRec problem equation 5. Choosing $\alpha(x) = {}^{K}/_{|\mathcal{Y}|}$ and $\beta(y) = 1$ reduces to the traditional Top-K ItemRec problem. In its most general form, equation 6 simultaneously solves for both ItemRec and UserRec to control the global exposure rates of different items, e.g., for item exploration purposes. In these settings, we may further replace the β-upper-bound constraints with equality or lower-bound constraints. We will discuss solutions and implications in Section 4 and show experiments in Section 5.2.

3 Model Details

While RIM is motivated to estimate the intensity of each user-item pair separately, we find it more convenient to utilize existing RNN-ItemRec models. Based on Proposition 1, we keep the RNN

model for its categorical prediction of the preference *direction*, $p(y|x)$, while introducing separate estimators for the user global engagement *norm*, $\lambda(x)$.

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 coincidentally holds because practical RecSys often truncates the training datasets by a fixed time for scalability concerns.

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

$$\lambda(t) = \mu + \sum_{j:t_j<t} \sum_{r=1}^{R} \alpha_r \phi_r(t - t_j), \tag{7}$$

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

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,

$$\bar{\Lambda} \sim \mathbb{E}[\Delta N(x, y)] = \mathbb{E}\left[\int_t^{t+\Delta t} \lambda(\tau; \mathcal{S}(\tau), y)\, \mathrm{d}\tau \middle| \mathcal{S}(t) = x\right], \tag{8}$$

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

$$\bar{\Lambda} = \phi(t) \odot \mathrm{softplus}(w) = \phi(t) \odot \log(1 + \exp(w)). \tag{9}$$

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.

4 Offline and Online Matching via Convex Optimization

Our RIM proposal allows to rethink recommendation in a Matching (Mtch) scenario, where the benefits of the users (e.g., ItemRec relevance) and item providers (e.g., item-exposure guarantees) are considered at the same time. This is shown earlier in equation 6 and we discuss its solutions now. We pay special attentions to the OnlnMtch scenario where we recommend users without seeing all of the users in advance. Instead, we may obtain statistical insights by looping through the observed users in a *training* set, hoping to generalize to new users or new events of the observed users between the training time and the actual inference time. 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. The CVX solution further resolves the race conditions of multiple item-marketing campaigns and we independently develop an entropic-regularized (ϵ) version inspired by Sinkhorn's algorithm in optimal transport (Cuturi, 2013) to improve its convergence properties in practice.

4.1 The Original Sub-Gradient Solution

The general solution to equation 6 is to convert its UserRec constraints as a Lagrangian multiplier

$$\max_{\substack{0 \le \pi(x,y) \le 1, \forall x, y \\ \mathbb{E}_y[\pi(x,y)] \le \alpha(x), \forall x}} \min_{v(y), \forall y} L(\pi, v) = \mathbb{E}_{x,y}[\lambda(x, y)\pi(x, y)] - \mathbb{E}_y[v(y)(\mathbb{E}_x[\pi(x, y)] - \beta(y))], \tag{10}$$

which is parametrized by (λ, α, β), where the dual variable $v(y)$ penalizes constraint violation by turning the inner-loop to infinity. For inequality-based UserRec, such as min-exposure guarantees $\mathbb{E}_x[\pi(x, y)] \geq \beta(y)$, we impose sign-constraints to the dual variables, i.e., $v(y) \leq 0$. We abbreviate the outer constraint as $\{0 \leq \pi \leq 1, \pi \sim \alpha\}$.

Then, since the objectives and the constraints are all linear, we may switch the order of min/max to form a dual problem with strong duality:

$$\min_{v(y)} d(v) = \left[\max_{\substack{0\leq\pi\leq 1\\ \pi\sim\alpha}} L(\pi, v)\right] = \mathbb{E}_x\left[\max_{\substack{0\leq\pi\leq 1\\ \pi\sim\alpha}} \mathbb{E}_y[(\lambda(x, y) - v(y))\pi(x, y)]\right] + \mathbb{E}_y[v(y)\beta(y)]. \quad (11)$$

Note that the inner-max loop can be solved by picking the top items by the modified ranking scores, $\lambda(x, y) - v(y)$, while the scores remain non-negative. For single-item marketing, this exactly reduces to thresholding by $\lambda(x, y) \geq v(y)$. The dual problem solves $v(y)$ by stochastic (sub-)gradient descents (SSGD) through all users in mini-batches. The dual objective is convex and, therefore, with additional regularization in $v(y)$ and decreasing learning rates, SSGD can achieve sublinear regret bounds as it scans through unseen users online with adversarially distributed $\lambda(x, y)$ scores. See Mahdavi et al. (2012); Huang et al. (2016); Jenatton et al. (2016) for further theoretical discussions.

While boasting online regret guarantees, the SSGD solver may suffer oscillatory patterns without carefully designed decreasing learning rates. 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, yet slow to converge to the final optima. See Figure 2 for a visualization.

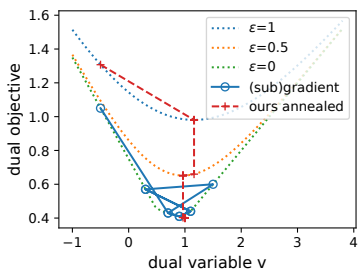


Figure 2: Toy problem with two users and one item with score matrix $(\lambda(x, y)) = (1, 0.5)^\top$, subject to UserRec constraint $\mathbb{E}_x[\pi(x, y)] = 0.4$. We plot the dual objective with different entropic-regularizer ϵ as well as the solution paths of different methods. As $\epsilon \to 0$, (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.

4.2 Entropic-Regularization and Alternating Optimization

To improve the convergence properties of CVX on the training set, 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 $\pi \sim \alpha$ via another dual variable $u(x) \geq 0$. The new dual objective and Lagrangian multiplier become

$$\min_v \min_{u\geq 0} d_\epsilon(u, v) = \left[\max_\pi L_\epsilon(\pi, u, v)\right], \text{ where}$$
$$L_\epsilon(\pi, u, v) = L(\pi, v) - \mathbb{E}_x[u(x)(\mathbb{E}_y[\pi(x, y)] - \alpha(x))] + \epsilon\mathbb{E}_{x,y}[H(\pi(x, y))], \quad (12)$$

where $L_\epsilon(\pi, u, v)$ is symmetric in $u(x)$ and $v(y)$ if we expand out the $L(\pi, v)$ term. As a subroutine in the dual problem, the primal variable $\pi(x, y)$ assumes an explicit solution given $u(x)$ and $v(y)$:

$$\pi(x, y) = \sigma\left(\frac{\lambda(x, y) - u(x) - v(y)}{\epsilon}\right), \text{ where } \sigma(z) = \frac{e^z}{1 + e^z}. \quad (13)$$

Plugging this in the dual form leads to functional gradients with respect to the dual variables:

$$\frac{\partial d_\epsilon(u, v)}{\partial u(x)} = \alpha(x) - \mathbb{E}_y\left[\sigma\left(-\frac{u(x)}{\epsilon} + \frac{\lambda(x, y) - v(y)}{\epsilon}\right)\right], \forall x; \text{ and vice-versa for } v(y). \quad (14)$$

Finally, unsurprisingly thanks to the convexity of the dual objectives, the gradients are monotone in every coordinate of the dual variables, which allows efficient bisection search for their roots in

Algorithm 1 CVX for offline or online matching

Require: $\lambda(x, y), \alpha(x), \beta(y)$, batch-to-full ratio γ
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

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 $v(y)$. As a result, we may efficiently solve for $u(x)$ and $v(y)$ alternatively, with additional projections to the sign-coordinates in the case of inequality constraints.

4.3 Discussions

Compared with the original subgradient solutions, our ϵ-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 ϵ-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 ϵ from a larger starting point. See Algorithm 1 for a summary of our proposed solution.

Finally, we also consider an **Offline-Greedy** solution in our experiments. We simply sort the $\lambda(x, y)$ in both users and items to match user-item pairs until the constraints are satisfied. For lower-bound β-constraints, we first solve for the common factors by setting $\alpha = \beta$, followed by a "top-off" of the remaining user capacities with respect to the original α. While Offline-Greedy appears fast with in-memory datasets, it has fundamental limits in the use of mini-batches on infinite data streams.

5 Experiments

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

- **Netflix (NF)**[1] 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.
- **Movielens (ML)** (Harper & Konstan, 2015)[2] 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.
- **Yoochoose (YC)**[3][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

[1] https://www.kaggle.com/netflix-inc/netflix-prize-data
[2] https://grouplens.org/datasets/movielens/1m/
[3] https://www.kaggle.com/phhasian0710/yoochoose
[4] https://recsys.acm.org/recsys15/challenge/

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.

Two methods require further splitting of the training set: (RNN-)Hawkes-Poisson to calibrate for long-horizon intensities and CVX-Online, where we assume no observation of the test user-states at training time. On NF, we create a set-back window between $(T', T]$ and let CVX-Online to only observe user events prior to T', allowing a surprise factor in the new events of these users during test time. On ML/YC, we train CVX-Online on a validation window for Group-A users and then transfer to Group-B users for testing. Notice that the data in the validation (target) windows are reused between the scoring models and the calibration steps. This is not ideal, but it avoids training the scoring models twice after the calibration steps to bridge the data gaps. Statistics of our data splits are shown in Table S1.

5.1 UserRec and ItemRec Results

Table 1: RecSys methods evaluated by Precision@1% and Recall@1% in both ItemRec and UserRec. RNN-TPP models are clear winners in both scenarios.

ItemRec (Top-K Items per User)							UserRec (Top-C Users per Item)						
Method	Precision ($\times$100)			Recall($\times$100)			$\times$ User Intensity	Precision ($\times$100)			Recall ($\times$100)		
	NF	ML	YC	NF	ML	YC		NF	ML	YC	NF	ML	YC
Rand	0.04	0.22	0.01	0.48	0.29	0.29	Rand	0.04	0.21	0.01	0.62	0.84	0.81
Pop	0.78	0.99	0.07	12.41	1.65	2.91	Pop	0.19	1.57	0.03	7.44	8.38	3.59
							Hawkes	0.26	1.80	0.03	5.30	8.47	3.83
							Hawkes-Poisson	0.35	1.94	0.03	9.06	8.93	4.38
BPR-Item	0.71	0.76	0.22	14.75	1.37	10.47	BPR-User	0.26	0.82	0.14	11.86	6.39	18.18
ALS	0.72	0.69	0.18	13.97	1.27	8.09	-	0.27	1.04	0.07	10.98	7.71	4.95
LogisticMF	0.58	0.67	0.11	10.31	1.12	5.34	-	0.24	1.14	0.08	11.31	7.29	15.76
RNN	**0.87**	**2.00**	**0.29**	**17.74**	**2.78**	**13.69**	(Uniform)	0.20	1.49	0.21	7.09	7.22	32.84
							Pop	0.29	2.73	0.22	11.44	12.64	34.33
							Hawkes	**0.39**	4.29	0.22	11.56	18.59	34.57
							Hawkes-Poisson	**0.39**	**4.34**	**0.25**	**12.13**	**18.71**	**36.90**

Table 1 shows results from the baselines and our proposed methods, where RNN-TPP models are clear winners in both scenarios. Further, in ItemRec, we observe a trend in Rand $<$ Pop $<$ MF models $<$ RNN, meeting our expectations. On UserRec, we also see contributions from both the prediction of *item distribution* and user *intensity parameters*. For distribution directions, we expect Rand $<$ Pop $<$ MF models $<$ RNN-Pop. For intensity priors, we expect a significant contribution in RNN-Pop models over the RNN (non-prior) baseline. Additionally, we consistently observe Pop $<$ Hawkes $<$ Hawkes-Poisson as we build more complex TPP models.

Comparing Precision and Recall, the Precision metrics directly reflect the optimality for equation 5, whereas the Recall metrics further discount against dominant users. Personalized RecSys often provides even bigger relative gains on Recall metrics, due to their ability to retrieve highly relevant items (or users) specific to the requests.

5.2 Offline and Online Matching Experiments

Previously, people study ItemRec and UserRec separately, without constraints on the global distribution of the targets. For ItemRec, this may lead to poor global diversity, when, e.g., the same set of popular items are recommended to every user. In UserRec, this may lead to disproportional concentration to frequent users, exceeding their physical capacities. We thus study global constraints in the targets to trade-off relevance and global diversity or feasibility.

Here, we fix $\alpha = 1\%$ in equation 6 and vary $\beta \leq 1\%$ as item min-exposure constraints. In Section C, we fix $\beta = 1\%$ and vary $\alpha \geq 1\%$ as user max-capacity constraints. Both settings yield the same

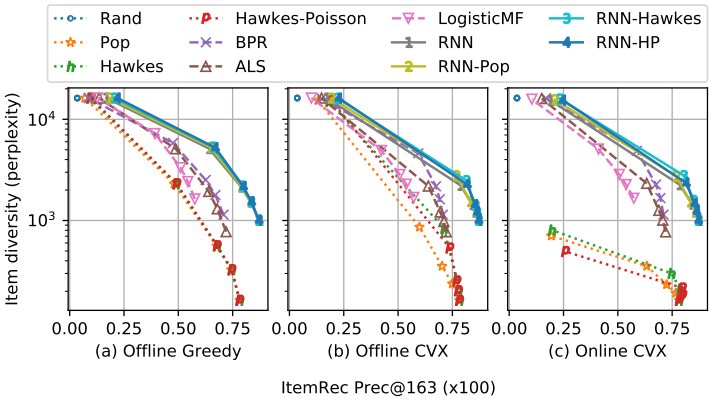


Figure 3: Matching experiments to trade-off ItemRec relevance and item diversity on NF dataset.

total number of user-item pairs, making precision metrics comparable across the board. They are also solvable by real-time decisions as each user is considered, using CVX dual decomposition.

For $\beta \leq 1\%$, choosing $\beta = 0$ corresponds to the unconstrained ItemRec problem. As we increase β, we observe an increase in item-exposure diversity, measured by

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

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

Figure 3 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. Comparing the Pareto fronts, we see clear winners from RNN-TPP variants. The difference becomes larger as we come to the diversity-side of the curves, where we need to weigh user-intensities more in tighter user-item matches. (The gap further increases in Section C experiments.) Next in performance are the MF-based models. While they may not compete favorably in relevance alone, presumably due to strong sequential and popularity biases in Netflix, their benefits are better shown by considering their diversity contributions.

Comparing across different Mtch strategies, we see CVX solvers often yield slightly better Pareto fronts. More importantly, it generalizes to OnlnMtch scenarios, where the users or some user events are 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 potentials in mixing ItemRec and UserRec under similar considerations.

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, 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.

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

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14: 3207–3260, 2013.

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.

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.

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.

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.

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.

Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: Online convex optimization with long term constraints. *Journal of Machine Learning Research*, 13(81):2503–2528, 2012.

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.

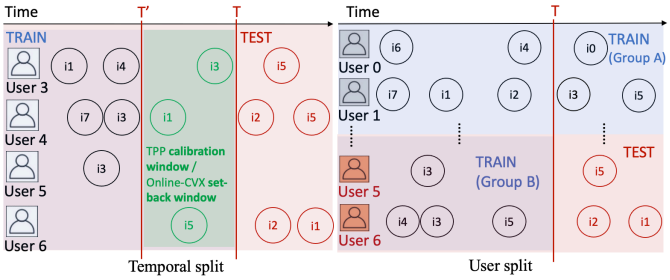
Alex Reinhart. A Review of Self-Exciting Spatio-Temporal Point Processes and Their Applications. *Statistical Science*, 33(3):299 – 318, 2018.

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.

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.



	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

A Related Work

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.

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.

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).

Other notable trends in marketing include causal effect modeling as well as exploration. Some related works include (Bottou et al., 2013; Xu et al., 2016; Li et al., 2018; 2010).

B ItemRec with Min-Exposure Constraints (ML and YC)

Similar to the Netflix results in the main text, we show two more Mtch experiments on MF (Figure S1) and YC (Figure S2). Comparing the Pareto fronts, we observe RNN-TPP variants > RNN > MF > Non-personalized models, which is consistent with our results on NF dataset.

C UserRec with Max-User-Capacity Constraints

We consider $\alpha \geq 1\%$ and $\beta = 1\%$, both as upper-bound constraints in equation 6. In the limit $\alpha = 100\%$, the problem becomes UserRec without the consideration of user capacities. This yields

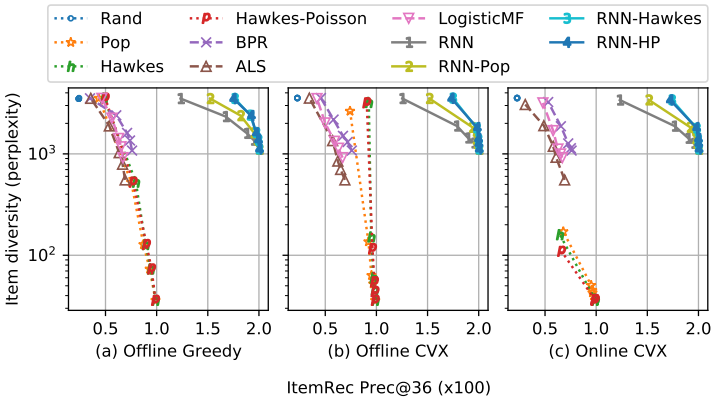


Figure S1: ItemRec with min-exposure constraints in ML-1M dataset.

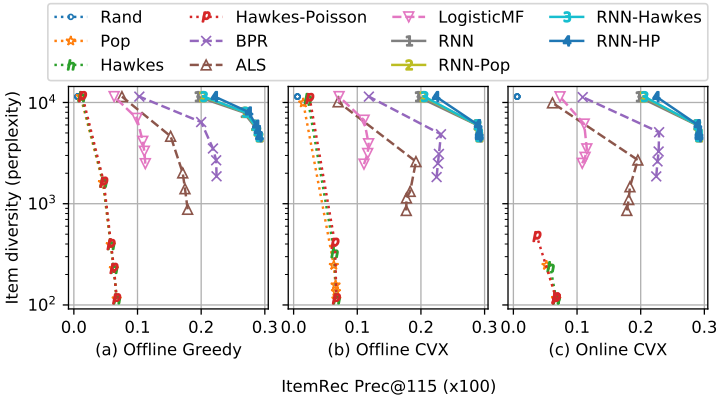


Figure S2: ItemRec with min-exposure constraints in Yoochoose dataset.

best UserRec relevance, with a risk to overwhelm the users. As α decreases, the users are forced to take a more balanced load. We use a similar perplexity metric to measure the user-load balances.

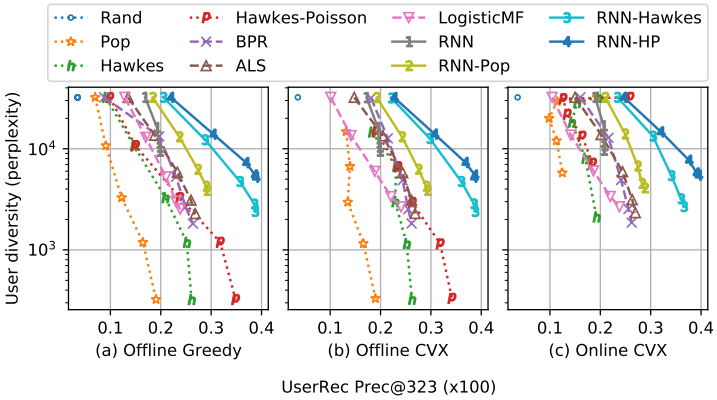


Figure S3: UserRec with user max-capacity constraints in Netflix dataset.

Figure S3&S4&S5 show the trade-off between relevance and user diversity. With a further focus on the user-intensity priors, we find larger advantages in the RNN-TPP variants. Further, the matching problem can be solved by Online-CVX, demonstrating the capability to market multiple competing items in an unseen user pool.

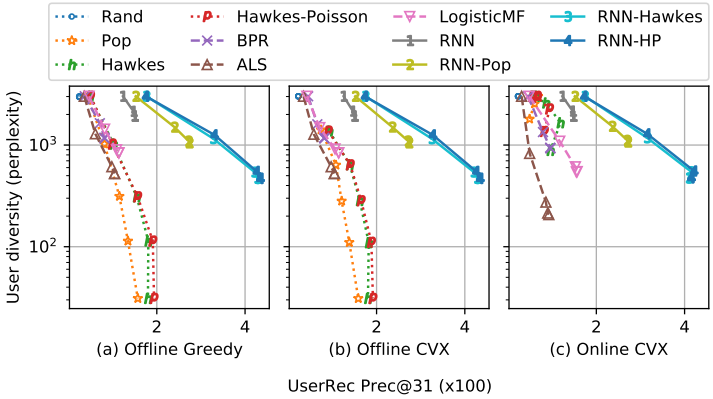


Figure S4: UserRec with user max-capacity constraints in ML-1M dataset.

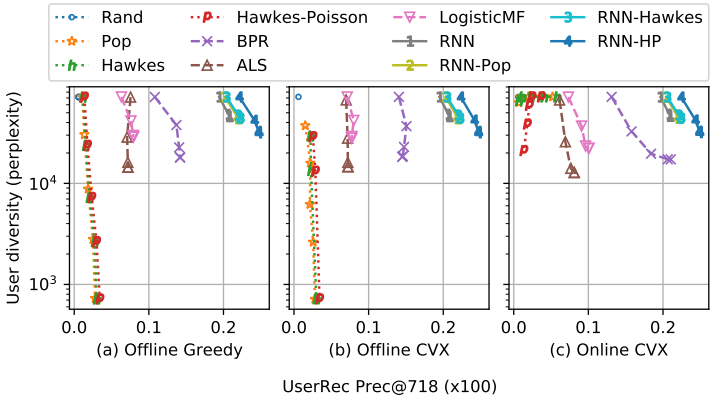


Figure S5: UserRec with user max-capacity constraints in Yoochoose dataset.