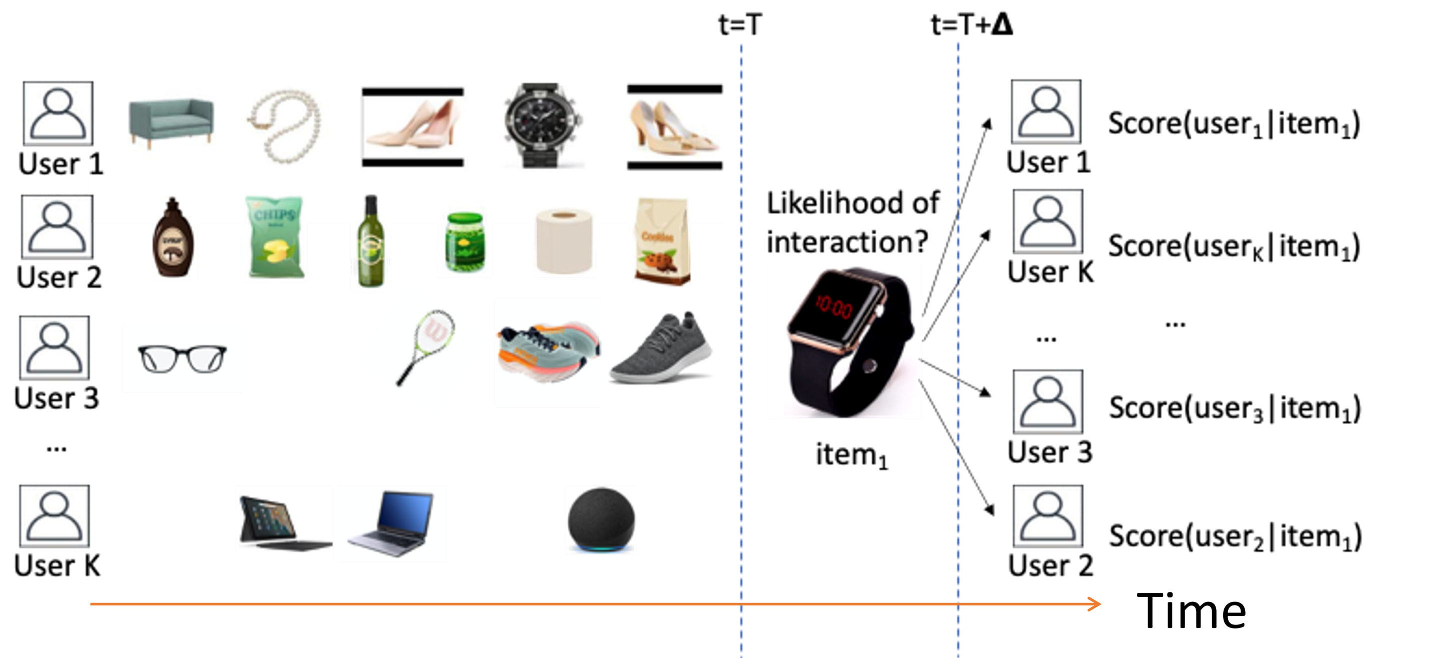
Recurrent Intensity Model for User Recommendation and Online Matching

AWS Labs

Yifei Ma, Ge Liu, Anoop Deoras

ICML 2021 – Time Series Workshop

1. Problem Definition



UserRec: We aim to recommend users on behalf of item producers, where the users are represented by their recurrent hidden states from past behaviors. Our models eliminate the need of extra features or labels to achieve the greatest generality.
OnlnMtch: One benefit of UserRec is to allow fair exposure of all items in the catalog. We quantify the benefits via a constrained online-matching problem.

2. Main Challenges

User events happen at irregular times. Unlike ItemRec, we cannot directly compare users at any particular time point.

(no events happen at any particular time point)

If we compare by time intervals, a user may have multiple events. How do we compare event sequences or sets of different sizes?

3. Proposed Methods

Goal is to estimate the expected future intensity in Item-y

$$\lambda_y(h(t)) \sim \mathbb{E}\left[\frac{\mathrm{d}N_y}{\mathrm{d}t}\middle|h(t)\right], \text{ where } \begin{cases} \mathrm{d}N_y & \#\text{events associated with Item-}y; \\ \mathrm{d}t & \text{next infinitesimal time interval;} \\ h(t) & \text{user state at time } t. \end{cases}$$

For simplicity, decompose as {ItemRec RNN} x {User-Intensity TPP}

$$\lambda_y(h(t)) = p(y|h(t)) \times \lambda(t; h(t))$$

- RNN predicts the item type just like traditional ItemRec

$$p(y|h(t)) = \frac{\exp(w_y^\top h(t) + b_y)}{\sum_{\tilde{y}} \exp(w_{\tilde{y}}^\top h(t) + b_{\tilde{y}})}$$

- TPP (Hawkes) models the user-intensity with exponential kernels

$$\lambda(t) = \mu + \sum_{j:t_j<t} \sum_{r=1}^{R} \frac{\alpha_r}{s_r} \exp\left(-\frac{t-t_j}{s_r}\right)$$

Learn TPP by MLE $\max\left[\sum_i \log\lambda(t_i) - \int_0^T \lambda(t)\,\mathrm{d}t\right]$ or MSE $\min\left[\int_0^T \lambda(t)^2\,\mathrm{d}t - 2\int_0^T \lambda(t)\,\mathrm{d}N(t)\right]$

Assign Recommendations via Constrained Optimization

$$\max_{\pi(h,y)\in\{0,1\}} \iint [\pi(h,y)\lambda_y(h)]\,\mathrm{d}P_H(h)\,\mathrm{d}P_A(y)$$
$$\text{s.t.} \begin{cases} \int \pi(h,y)\,\mathrm{d}P_A(y) \le k(h), & \forall h \in H; \\ \int \pi(h,y)\,\mathrm{d}P_H(h) \le c(y), & \forall y \in A; \end{cases}$$

where P_H = observed user-state distribution;
P_A = observed item distribution.

Special Case	k(h)	c(y)
ItemRec	1%	100%
UserRec	100%	1%
OnlnMtch (1-to-1)	1%	1%

Solve via greedy allocation when all users and items are observed

Other methods

- Pop score ∝ {item visitation counts} x {user history length}
- Bayesian Personalized Ranking (BPR) learns to rank users/items in a validation window
- Hawkes-Poisson fine-tunes $\Lambda = \mathbb{E}\left[\int_T^{T+\Delta T} \lambda(t)\,\mathrm{d}t\right]$ from the Hawkes-kernel states

4. Experiments

Datasets:

- Netflix: observe 6 months of events; predict in next 2 weeks
- Movielens: 4068 seconds of observed events => 4068s interval
- Yoochoose: 457 seconds of observed events => 457s interval

	NF	ML	YC
# warm users	32238	3020	71784
# warm items	16217	3669	11431
# training events	2437151	762016	1087267
# clean test events	187096	37597	49500
UserRec C(y)	323	31	718
ItemRec K(h)	163	37	115

Traditional ItemRec and Proposed UserRec

- Relevance measured by precision and diversity measured by global perplexity
- RNN-TPPs show improvements in both metrics over non-personalized baselines

ItemRec (Top-K items per user)							UserRec (Top-C users per item)						
Method	Precision (x100)			Item Perplexity			x User Intensity	Precision (x100)			User Perplexity		
	NF	ML	YC	NF	ML	YC		NF	ML	YC	NF	ML	YC
Rand	0.04	0.37	0.006	16192	3610	11423	Rand	0.04	0.37	0.006	32140	2980	71472
Pop	0.78	1.74	0.067	163	37	115	Pop	0.19	3.03	0.028	323	31	718
							Hawkes	0.26	4.44	0.030			
							Hawkes-Poisson	0.35	4.48	0.034			
BPR-Item	0.71	0.73	0.222	1139	684	1774	BPR-User	0.26	0.84	0.147	1843	777	18629
RNN	0.86	2.09	0.306	1015	1049	5024	(Uniform)	0.19	1.43	0.223	8910	1952	45992
							Pop	0.28	2.81	0.233	3520	1041	44620
							Hawkes	0.38	5.42	0.235	2268	375	44880
							Hawkes-Poisson	0.38	5.15	0.258	4720	442	34533

Proposed OnlnMtch

Set k(h)=1%; control global item diversity by c(y)=[1%, 3%, 10%, 30%, 100%];
Set c(y)=1%; control global user diversity by k(h)=[1%, 3%, 10%, 30%, 100%];
RNN-TPP models dominate the Pareto front of relevance-diversity trade-offs