

Recurrent Poisson Factorization for Temporal Recommendation

Seyed Abbas Hosseini¹, Ali Khodadadi¹, Keivan Alizadeh, Ali Arabzadeh¹, Mehrdad Farajtabar¹, Hongyuan Zha, and Hamid R. Rabiee¹, *Senior Member, IEEE*

Abstract—Poisson Factorization (PF) is the gold standard framework for recommendation systems with *implicit feedback* whose variants show state-of-the-art performance on real-world recommendation tasks. However, they do not explicitly take into account the temporal behavior of users which is essential to recommend the right item to the right user at the right time. In this paper, we introduce *Recurrent Poisson Factorization* (RPF) framework that generalizes the classical PF methods by utilizing a Poisson process for modeling the implicit feedback. RPF treats time as a natural constituent of the model, and takes important factors for recommendation into consideration to provide a rich family of time-sensitive factorization models. They include *Hierarchical RPF* that captures the consumption heterogeneity among users and items, *Dynamic RPF* that handles dynamic user preferences and item specifications, *Social RPF* that models the social-aspect of product adoption, *Item-Item RPF* that considers the inter-item correlations, and *eXtended Item-Item RPF* that utilizes items' metadata to better infer the correlation among engagement patterns of users with items. We also develop an efficient variational algorithm for approximate inference that scales up to massive datasets. We demonstrate RPF's superior performance over many state-of-the-art methods on synthetic dataset, and wide variety of large scale real-world datasets.

Index Terms—Poisson factorization, poisson process, temporal recommender system

1 INTRODUCTION

RECOMMENDATIONS are the main drive for consumer purchase in online retailers and the most effective factor in user engagement in online services. However, delivering the right item at the right time to the right person is a challenging task due to many reasons. First, the user preference is not directly observable and must be learnt from *implicit feedback*. Moreover, users may have different behavioral patterns; Some tend to explore and test new products while others use a limited number of products, frequently. In addition to this heterogeneity, their interest and preference change over time and may exhibit seasonality. Moreover, there are many common patterns in engagement of users with different items over time that can be utilized to suggest the right item to the user at the right moment which can't be learnt unless user-item interactions are modeled efficiently. On the other hand, the ubiquity and increasing presence of social networks in everyday lives transform the purchasing process or service adoption from a simple matching of user preference and item specification to a more complex one affected by the peer influence.

Poisson Factorization (PF) is the gold standard framework for recommendation systems with implicit feedback [1], [2], [3], [4], [5]. The PF based methods utilize the user interactions with items in terms of the count of item usages, where this implicit feedback signal helps to infer the latent user preference and item specification. There is a wealth of research to adapt the PF framework to tackle the aforementioned problems. For example, diversity and heterogeneity of users and products is addressed by an innovative hierarchical probabilistic model called Hierarchical Poisson Factorization (HPF) [1], and Dynamic Poisson Factorization (DPF) captures the time evolving latent factors with a Kalman filter [3]. Moreover, to incorporate social network information into a traditional factorization method, Social Poisson Factorization (SPF) enriched the preference-based recommendations [2]. However, the existing variants of PF based recommendation systems do not explicitly take into account the *temporal* behavior and the *recurrent* activities of users.

The time-sensitive product adoption data carries a great deal of information on usage dynamics which is otherwise neglected when the user-item interaction is only reported by the aggregated number of times they happen [6], [7], [8]. Furthermore, it allows successful predictions of the returning time which not only allow a web company to keep track of the evolving user preferences, but also result in more clever marketing strategies and time-sensitive recommendation. After all, online retailers need not blindly advertise all the times and make the users indifferent. Traditionally, in PF based recommendation the state of each user is often assumed to be binary—either adopting a product or not, or an aggregated count/rating variable. However, such assumption does not capture the recurrent nature of product usage, where the frequency and time of the usage matters. It's noteworthy that the

• S.A. Hosseini, A. Khodadadi, K. Alizadeh, A. Arabzadeh, and H.R. Rabiee are with the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. E-mail: {a_hosseini, khodadadi, kalizadeh, arabzadeh}@ce.sharif.edu, rabiee@sharif.edu.

• M. Farajtabar and H. Zha are with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332. E-mail: mehrdad@gatech.edu, zha@cc.gatech.edu.

Manuscript received 5 Nov. 2017; revised 4 Oct. 2018; accepted 1 Nov. 2018. Date of publication 6 Nov. 2018; date of current version 5 Dec. 2019.

(Corresponding author: Hamid R. Rabiee.)

Recommended for acceptance by E. Terzi.

Digital Object Identifier no. 10.1109/TKDE.2018.2879796

classical time-window or instance-decay approaches and manually discretization do not work, as they lose too much information when discarding data instances or coarse-grained discretizing of the time [9], [10]. Alternatively, one can improve the accuracy by more granular approximation at the cost of increased computational complexity.

In this paper, we introduce *Recurrent Poisson Factorization* (RPF) for recommendation systems. RPF is a mathematical framework based on *Poisson processes* [11], which generalizes the classical PF methods by modeling the product adoption as a non-homogeneous Poisson process over time rather than a Poisson distribution on the aggregated count values. RPF jointly models the time and the type of items with which the user interacts and hence is able not only to predict the item but also the time that she will engage with an item. A distinctive feature of RPF is its interpretability and intuitive parameterization; In addition to following the classic PF methods on learning a latent representation for each user and item, it provides a sound and rigorous framework to model the effect of previous actions of the user and her friends on future service usages and preferences. RPF framework not only extends the basic PF method [1], but also is applicable to the many of its variants. Basically, we establish the foundation for replacing Poisson distribution with the Poisson process whenever there is a call for temporal and recurrent service adoption. Notably, we extend the well-known HPF, DPF, and SPF methods to their recurrent counterparts, HRPF, DRPF, and SRPF, respectively. Moreover, we introduce Item-Item RPF (IIRPF) which is able to take into account the effect of engagement of a user with an item on the user's future tendency to adopt other items. We also propose an eXtension of IIRPF (XIIRPF) which is able to utilize the item's metadata such as category, genre and location to better infer the correlation among engagement pattern of users with items. These downstream variants of RPF accompanied by its well-founded mathematical essence makes it a compelling replacement for classic PF when the temporal granularity and frequency of actions matter. In summary, the overall highlights of this paper are:¹

- We establish a previously unexplored connection between Poisson factorization framework and Poisson processes, which allows to recommend the right item to the right user at the right time by inferring the dynamic user interests over time and predicting the time that users need each item.
- We show that the proposed RPF framework is applicable to many variants of basic PF. It can handle dynamic user preference and item specification (DRPF). It can capture the heterogeneity and diversity among users and items (HRPF), and it is able to incorporate the peer influence and the network of users (SRPF). Moreover, it is able to infer the effect of a user-item interaction on future engagement of the user with other items (IIRPF), and utilize items metadata (X-IIRPF) to better infer item-item correlation and recommend items to users more efficiently.
- We propose a scalable variational inference algorithm based on mean-field approximation for RPF that scales up to millions of user-item interactions.

- We conduct several experiments on synthetic and real datasets to demonstrate the performance of our model. We also compared our results with state-of-the-art recommendation methods on a wide range of applications. To this end, we utilized three datasets from different areas, i.e., *Tianchi* M-Commerce data, *Foursquare* users' check-in data, users' watching history on a *IPTV* service. We also collected *Last.fm* dataset consists of the listening history of more than 1000 users over 6 months including more than 450K user-song pairs.²

Related Work. The works most closely related to ours are roughly divided into two groups: Poisson factorization methods and Poisson point processes. These two lines of research for recommendation systems are progressing independently, and, to the best of our knowledge, we are the very first to systematically combine these two and propose a unified framework for Poisson process factorization.

Matrix factorization algorithms are widely used for recommendation in order to infer the user preferences based on similar consumption patterns among users [13]. Traditional Factorization methods used user ratings to products in order to estimate their preferences. Since users rarely provide such explicit feedbacks, more recent methods such as Poisson factorization infer such preferences from user implicit feedback. Different variants of PF are able to consider the heterogeneity among users, dynamic user interests over time and peer influence among users [1], [2], [3]. Furthermore, the nonparametric version of PF is able to effectively estimate the dimension of latent feature space [4]. Collaborative Topic Poisson Factorization (CTPF) is another variant of PF which is customized for article recommendation and is able to consider the content of articles to infer the user preferences more effectively [5]. Moreover, Cofactor [14] is a recent matrix factorization method which models the co-occurrence of items in users' history to better learn the users interests. The method proposed in [15] is another recent matrix factorization method which considers the exposure of users with items to better recommend items to users.

Although PF and other matrix factorization methods provide a general framework for recommendation with many desirable features, they lose a great deal of information due to ignoring the temporal patterns of user actions.

More recently, point process methods have attracted a lot of interest in modeling marked temporal events [16], [17], [18], user behavior modeling [19], [20], [21] and recommendation systems [7], [8], and optimization over networks [22], [23]. The method proposed in [24] models the adoption of competing products by a multidimensional point process. However, it doesn't provide a method for recommending appropriate items to users. The authors in [7] propose a compelling point process model for recommendation systems, however, the model lacks the peer influence within the network and suffers from non-dynamic user preferences and item specifications. Moreover, to model the heterogeneity between users they impose a low-rank regularizer which blindly set most latent dimensions of preferences to zero while keeping a few non-zero elements. In contrast, our

1. Preliminary version of this work appeared in [12].

2. The codes and data are available at <https://github.com/AHosseini/RPF>

approach captures the diversity between users by hierarchical Bayesian modeling. The method in [8] introduces a coevolutionary model of user and item latent features that captures the evolution and coevolution of users' and items' features over time without incorporating network influence or considering the heterogeneity among users. The last but not the least, all these works lack a systematic treatment of Poisson factorization methods and Poisson processes and their close connections.

2 PRELIMINARIES

2.1 Poisson Factorization

Poisson factorization is a general framework for recommendations based on users implicit feedbacks, such as clicks or purchases. PF methods use the number of times a user u consumed the product p to infer the interest of user to all products including non-consumed ones. Hierarchical Poisson Factorization (HPF) is the most basic model in this framework, which assumes the number of times that user u purchased or clicked an item p is a Poisson random variable r_{up} with rate λ_{up} . HPF models the interest of user u and properties of product p as K dimensional latent vectors θ_u and β_p , respectively, and use their inner product as the latent rate of r_{up} , that is:

$$r_{up} \sim \text{Poisson}(\theta_u^\top \beta_p). \quad (1)$$

In order to model the heterogeneous interests of users and the variety of items, HPF uses a hierarchical prior over latent feature vectors [1]. Although this model has rigorous statistical properties such as sparse representation and ability to capture the long-tailed user activities, unfortunately, it is not able to utilize the social relation among users that is inherent to many recommendation scenarios.

Social Poisson Factorization (SPF) is another development in PF framework that solves the above issue [2]. SPF models the number of times user u is engaged with an item p as a Poisson random variable. However, SPF assumes that the rate of r_{up} is the sum of $\theta_u^\top \beta_p$ and a weighted sum of the number of times user u 's friends have consumed this item. That is:

$$r_{up}|r_{-up} \sim \text{Poisson}\left(\theta_u^\top \beta_p + \sum_{v \in N(u)} \tau_{vu} r_{vp}\right), \quad (2)$$

where r_{-up} denotes all rates except r_{up} and $N(u)$ shows the set of people that user u follows in the social network and τ_{vu} is the latent impact of user v on user u . In this way, SPF incorporates the social relations among users with the user-item history in an effective way. Therefore, it's able to infer the influence of users on each other. However, since this model is not able to consider time of users' actions, the rate of adopting item p by user u will be increased by τ_{vu} any time after her friend v adopts this item. This is not true in real-world scenarios as the effect of users' actions on their friends usually decays after some time.

The PF framework has advanced continuously to handle more real-world and applied situations. The evolving user preferences and item specification (over time) was one of the serious challenges of recommender systems. In appeal to this call, Dynamic Poisson Factorization (DPF) was

introduced as a recommendation method based on Poisson factorization [3]. It basically solves this issue by considering time dependent feature vectors for users and items. DPF is a discrete-time approach which models the evolution of users and items latent features over time by a Kalman filter. In order to do so, DPF models latent features by a Gaussian state space model and exponentiates the state space model to make it nonnegative. Therefore, the rate of user u for adopting item p becomes:

$$r_{up,t} \sim \text{Poisson}\left(\sum_{k=1}^K e^{(\theta_{ukt} + \bar{\theta}_{uk})} e^{(\beta_{pkt} + \bar{\beta}_{pk})}\right), \quad (3)$$

where $\bar{\theta}_{uk}$ and $\bar{\beta}_{pk}$ are the static parts of k th element of the users u and product p 's feature vectors respectively and θ_{ukt} and β_{pkt} are the dynamic parts which change over time based on a Markov model.

One of the most limiting properties of the PF methods is that they summarize the history of user-item interactions in terms of a single count and hence discard a great deal of information. Furthermore, they are not usually well-suited to model the situations where the consumption is a recurrent process. Failing to answer time-sensitive questions is another drawback of the classic PF models. In the next section, we make a brief introduction to Poisson processes which are able to capture the temporal dynamics of user-item interactions more effectively.

2.2 Poisson Process

Poisson process is a stochastic process which is suitable for modeling timestamped events. This process assumes that the number of events in a time window $[t_0, t_1]$, $N_{[t_0, t_1]}$, is a random variable distributed as:

$$N_{[t_0, t_1]} \sim \text{Poisson}(\lambda \times (t_1 - t_0)), \quad (4)$$

where λ is the rate of occurring events. The main advantage of Poisson process over Poisson distribution in modeling timestamped events is the fact that it considers the interval in which the events occur and hence can estimate the rate of occurring events more accurately. To elaborate, consider two scenarios: in the first one, three events occur during one month e.g., at days 1, 10, 25. Now assume in the second scenario all the three events occur in a single day, say day 20. The PF based models consider both scenarios the same because Poisson distribution can only work with the aggregated counts. In contrast poisson processes can differentiate between the two scenarios: The first process has higher rate of occurrence in the beginning of the months, while the second process has high activity rates towards the end of the month.

In real-world applications events and times series data are not usually distributed uniformly over the time. For example, the distribution of inter-event durations in many applications such as posting emails, listening to music, twitting and checking into social networks have been shown to be bursty, heavy-tailed and periodic. Hence they can't be effectively modeled using poisson distribution and even homogeneous Poisson processes. *Non-homogeneous Poisson process* is a process with dynamic rate of event occurrence that can capture the complex longitudinal dependencies among events. This process can be uniquely defined using

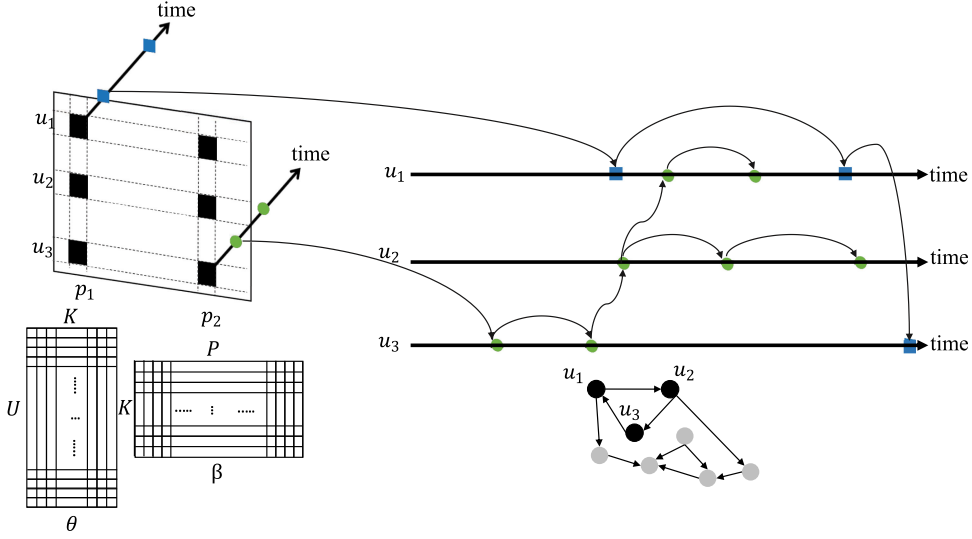


Fig. 1. The illustration of the RPF model. The left matrix represents the user-item engagement that are triggered by user intrinsic interest in different items. Each of these events may result in future engagement of the same user or one of her friends with the same item. These events and the triggering relations among them are represented in the right plot. The arrows show the triggering relationships among events. As it is depicted, all RPF variants infer the users' intrinsic interest in different items using a factorization model in which the users and items are modeled in a K -dimensional latent space. The social relations are also depicted in the right bottom. The arrows show the following relationship among users.

the so-called *conditional intensity function*, $\lambda^*(t)$. It encodes the expected rate of occurring event at time t given the history of past events, $\mathcal{H}(t)$, i.e.,

$$\lambda^*(t)dt = \mathbb{P}\{\text{event in } [t, t+dt) | \mathcal{H}_t\}. \quad (5)$$

Using the definition of $\lambda^*(t)$ in Eq. (5), the likelihood a list of events (t_1, \dots, t_n) observed during a time window $[0, T)$, can be written as

$$\mathcal{L}(t_1, \dots, t_n) = \prod_{i=1}^n \lambda^*(t_i) \underbrace{\exp\left(-\int_0^T \lambda^*(s)ds\right)}_{S(T)}. \quad (6)$$

Roughly speaking, the first term denotes the likelihood of occurring event at times (t_1, \dots, t_n) , and the second term $S(T)$ denotes the likelihood of not occurring events elsewhere. Therefore, if the temporal pattern of occurring events is not uniform, the likelihood is maximized by an intensity function that describes the temporal dynamics the best.

One of the most important properties of Poisson processes which makes them appropriate candidates for modeling temporal events is the fact that the superposition of independent Poisson processes is itself a Poisson process with intensity equal to the summation of other intensities [11]. We use this property to propose a computationally appealing framework for modeling user preferences over time, predicting their future actions, and timely recommending them the appropriate services and products.

3 RECURRENT POISSON FACTORIZATION

Intuitively, engagement of a user with an item is driven by three main factors. Intrinsic user preferences, previous pleasant experiences of her friends and her own ones with that item and the context of the items with which she was engaged. For example, the left matrix in Fig. 1 shows that user u_1 and u_3 have listened to music p_1 and p_2 respectively, because of their own initiatives. The social network among

the users is plotted in the bottom right. The upper right plot shows the events that are triggered by previous events and the arrows show the triggering relation among them. Since user u_2 follows user u_3 and user u_3 listens to music p_2 , after some time, user u_2 also listens to that music. Furthermore, interest in this song may become viral and u_1 also plays it under influence from her followee, user u_2 . Moreover, in many applications, users engage with specific set of items concurrently or adopting an item encourages the user to buy some other products. For example, as it is depicted in Fig. 2, Mary that buys a pair of running shoes is influenced contextually to buy a smart band and track her activity and hence recommending a smart band to her right after buying the running shoes, has a high chance to convert and hence learning such inter-item relations can immensely improve the time-sensitive recommendation systems and increase business revenue.

Recurrent Poisson Factorization (RPF) is a mathematical framework which models such behavioral patterns and is able to recommend the right item to the right user at the right time by utilizing the recurring temporal patterns in user-item engagement. In the followings, we first specify some notations and then discuss the generative model of RPF.

3.1 Notations and Conventions

Let $\mathcal{H}(T) = \{e_i\}_{i=1}^{M(T)}$ denote the set of user-item engagements until time T where $M(T)$ is the number of engagements up to time T . The engagement e_i is a triple (t_i, u_i, p_i) which indicates that at time t_i , user u_i engaged with product p_i . For clarity, we use the following notation for the remainder of this paper. $\mathcal{H}_{up}(t)$ denotes the set of interactions of user u with item p until time t . We use dot notation to represent union over the dotted variable, e.g., $\mathcal{H}_u(\cdot)$ represents the events of user u , before time t , with any product, and $\mathcal{H}_{up}(t)$ denotes the interactions of all users except u , before time t , with item p . In addition to user-item engagement data, we may have the users social relations. In that case, a user u may follow a set of users which is denoted by N_u .

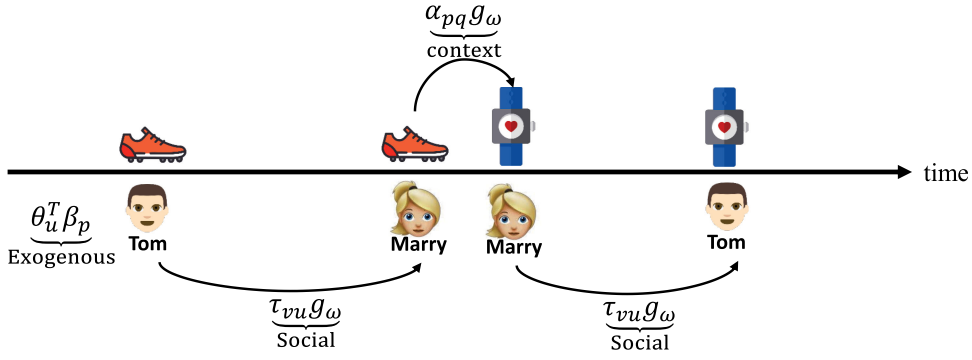


Fig. 2. IIRPF considers the effect of user-item engagement in future engagement of the user with other products. For example, Tom buys running shoes based on his intrinsic tendency which encourages his friend, Mary, to buy running shoes. Mary also buys a smart band to track her activity. Tom also buys a smart watch under the influence of Mary. IIRPF is able to model this process and infers the driving factor of each event.

3.2 Proposed Generative Model

The main idea of RPF is modeling the time of engagement of users with different products as a set of dependent Poisson processes where the intensity of each process consists of two main parts. The intrinsic intensity which denotes the matching between user preferences and items attributes, and the extrinsic intensity which represents the tendency of user to engage with an item triggered by previous user-item interactions. RPF assumes a hierarchical matrix factorization model for intrinsic users interests on items and models the triggering effect of interaction of a user with an item on future user-item engagements using a set of cascade Poisson processes[25]. In other words, it is assumed that the times of interacting user u with item p is generated by a Poisson process with the following intensity:

$$\lambda_{up}(t) = \theta_u(t)^\top \beta_p(t) + \sum_{e \in H(t)} \kappa_{up}(e, t), \quad (7)$$

where $\theta_u(t)$ and $\beta_p(t)$ are the K dimensional latent vectors which denote the interests of user u and attributes of item p at time t , respectively. The dot product between user interests and the item attributes essentially shows the similarity between them and serves as the user intrinsic intensity to engage with this item. In order to model the effect of previous user-item interactions on the tendency of user u to engage with item p at time t , it is assumed that each of the previous user-item engagements e , triggers a Poisson process which increases the tendency of user u to engage with item p at time t by $\kappa_{up}(e, t)$. Since the effect of this interaction usually decreases over time, the exponential function is widely used. That is, in the simplest form, the kernel $\kappa_{up}(e, t)$ can be defined as:

$$\kappa_{up}(e, t) = e^{-w(t-t_e)} \mathcal{I}(p_e = p, u_e = u), \quad (8)$$

where, \mathcal{I} is the indicator function which shows that the tendency of user u to adopt item p only depends on her own experiences with that item. $\kappa_{up}(e, t)$ can be any integrable function of time and can take a variety of functional forms to capture the complex temporal properties of preference propagation among users and items.

As it was mentioned before, RPF is a general framework for recommendation which can be customized to effectively include different types of information and priors and

beliefs. We will introduce three variants of this model in the following.

3.2.1 Hierarchical Recurrent Poisson Factorization (HRPF)

HRPF is the most basic variant of RPF which assumes that the users and items latent vectors do not change over time and engagements of user u with product p depends only on her own experience with that product. Therefore, the intensity of engagement of user u with product p in HRPF model is:

$$\lambda_{up}(t) = \theta_u^\top \beta_p + \sum_{e \in \mathcal{H}_{up}(t)} g_\omega(t_e, t), \quad (9)$$

where $g_\omega(t, t_e)$ is the temporal kernel capturing the complex longitudinal dependencies among user-item engagements. This kernel can be any integrable function that represents the impact of an event at time t_i on another event at time t . The most important feature of HRPF is its ability to promote and model the diversity of users and items. To this end, HRPF assumes a hierarchical prior over the set of $\{\theta_u\}_{u=1}^U$ and $\{\beta_p\}_{p=1}^P$. Since the gamma distribution is conjugate with the likelihood in Eq. (6) and also constrains the latent vectors to be nonnegative and sparse, we use the same idea as in [1] and consider a gamma prior over these variables with a latent rate parameter, i.e.,

$$\begin{aligned} \theta_u &\sim \text{Gamma}(a^{\theta, \text{shp}}, \eta_u) \\ \beta_p &\sim \text{Gamma}(a^{\beta, \text{shp}}, \xi_p), \end{aligned} \quad (10)$$

where, η_u and ξ_p are gamma distributed random variables equip RPF with a heterogeneous users and items latent features.

3.2.2 Social Recurrent Poisson Factorization (SRPF)

SRPF is a variant of RPF that is able to model the impact of social network on users' actions. In this model, we assume the actions of user u are either triggered by her intrinsic interests or by previous actions of her friends or her own. To this end, we propose the following intensity function for user u 's interaction with item p :

$$\lambda_{up}(t) = \theta_u^\top \beta_p + \sum_{v \in \mathcal{N}_u} \sum_{e \in H_{vp}(t)} \tau_{vu} g_\omega(t_e, t), \quad (11)$$

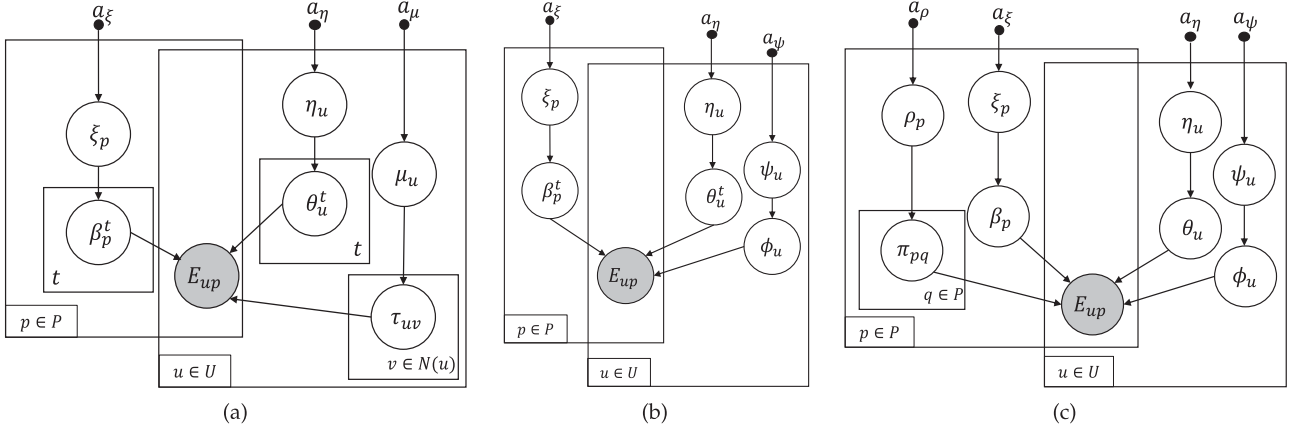


Fig. 3. The graphical model of the RPF. It shows that different effecting factors are considered in RPF, i.e., similarity between users preferences and items features, social relations among users and the causality among adoption patterns in which using one product trigger the interaction between the user and other products. Moreover, the hierarchical structure of the model allows to handle the diversity of users and products. Moreover, the time plates over users and items features shows the dynamic nature of the model.

where τ_{vu} is a latent influence of user v on user u . In order to make the influence matrix τ nonnegative and sparse, we assume a gamma prior over these variables which is also conjugate with likelihood in Eq. (6). That is:

$$\tau_{vu} \sim \text{Gamma}(a^{\tau, \text{shp}}, \mu_v), \quad (12)$$

In order to model the diversity of items and users, SRPF uses the same hierarchical prior model for user and items latent vectors as HRPF.

3.2.3 Dynamic Recurrent Poisson Factorization (DRPF)

DRPF is another variant of RPF which models the dynamic interests of users and popularity of items over time. DRPF proposes the following intensity for engaging user u with product p :

$$\lambda_{up}(t) = \theta_u^\top(t) \beta_p(t) + \sum_{e \in H_{up}(t)} \tau_{ue} g_\omega(t_i, t), \quad (13)$$

where, $\theta_u(t)$ and $\beta_p(t)$ are two processes which denote the preferences of user u and attributes of item p at time t , respectively. τ_{uu} is a latent parameter which shows the amount of influence of user u by her own previous interactions with items. In order to parametrize $\theta_u(t)$ and $\beta_p(t)$, we assume that these are nonnegative linear combinations of simpler processes:

$$\theta_u(t) = \sum_{i=1}^I \theta_u^i h_i(t) \quad (14)$$

$$\beta_p(t) = \sum_{j=1}^J \beta_p^j l_j(t), \quad (15)$$

where $h_i(t)$ and $l_j(t)$ are known functions over time and θ_u^i are nonnegative latent variables that should be learnt from data. For example, since the users have similar activity patterns in similar days of week or similar hours of the day, we can use the 24-dimension function $h(t)$, which $h_i(t)$ is 1 only in the i th hour of the day. In addition, an extra 7 dimensions for weekdays is also added:

$$h_i(t) = \begin{cases} \mathcal{I}(\text{hour}(t) = i) & 1 \leq i \leq 24 \\ \mathcal{I}(\text{day}(t) = i - 24) & 25 \leq i \leq 31, \end{cases} \quad (16)$$

and hence we can learn the preference of users in different hours of the day and different days of the week.

3.2.4 Dynamic Social Recurrent Poisson Factorization (DSRPF)

DSRPF is a variant of RPF which combines the features of SRPF and DRPF to jointly models the dynamic interests of users and popularity of items over time and peer influence among users in social network. DSRPF uses the following intensity function for the user-item engagement:

$$\lambda_{up}(t) = \theta_u^\top(t) \beta_p(t) + \sum_{v \in \mathcal{N}_u} \sum_{e \in H_{vp}(t)} \tau_{vu} g_\omega(t_i, t). \quad (17)$$

Fig. 3a depicts the graphical representation of DSRPF model. Since DSRPF includes all the latent variables included in HRPF, SRPF and DRPF, the graphical representation of these models can be simply extracted from Fig. 3a.

3.2.5 Item-Item Recurrent Poisson Factorization (IIRPF)

As it was previously mentioned, engaging a user with an item may encourage the user to engage with some other products and learning such inter-item relations can improve the temporal recommendation. IIRPF is a variant of RPF which infers the effect of interaction of a user with an item on its future engagements with other products. This is done by learning co-occurrence pattern of items to recommend the items in which that users may be interested at specific times. The general form for IIRPF intensity function to model the users-item preference over time is as follows:

$$\lambda_{up}(t) = \theta_u^\top(t) \beta_p(t) + \phi_u \sum_{e \in H_u(t)} \alpha_{pep} g_\omega(t_e, t), \quad (18)$$

where ϕ_u is a latent factor showing how much user u cares about her previous interactions with other items in choosing an item and α_{pq} is a latent factor which shows the influence of interaction of users on item p on the tendency of users to

engage with item q . In most of the applications, the number of items is too large and hence learning matrix α is a challenging task. We use the factorization technique one more time here to reduce the latent space dimension. Since α_{pq} is larger for product pairs that are adopted by a set of users sequentially, we can assume that these products have similar *beta* and hence in order to reduce the latent space of the model, we can use the following factorization:

$$\alpha_{pq} = \beta_p^\top \beta_q \quad p \neq q. \quad (19)$$

Moreover, for modeling the effect of previous interaction of the user or her friends with the same item, we can use the same idea as in SRPF and hence the more general form of the intensity function of IIRPF is:

$$\begin{aligned} \lambda_{up}(t) = & \theta_u^\top \beta_p + \sum_{v \in N(u)} \sum_{e \in H_{vp}(t)} \tau_{vu} g_{w_1}(t_e, t) \\ & + \phi_u \sum_{e \in H_{up}(t)} \beta_p^\top \beta_{p_e} g_{w_2}(t_e, t), \end{aligned} \quad (20)$$

where g_{w_1} and g_{w_2} are two temporal kernel functions which show the effect of previous interactions of the user and her friends with the same item on her current interest on the item and the effect of her previous interactions with other items on her current tendency to engage with item p respectively.

3.2.6 eXtended Item-Item Recurrent Poisson Factorization

IIRPF is a general method for modeling the inter-item relations and the effect of interaction of a user with an item on engagement with other items. The factorized model proposed in Eq. (19) for α_{pq} is a good one for reducing the latent space dimension when no metadata is available about items. However, in many applications, we have metadata about items which can help us to learn the inter-item relations more efficiently. For example, in many applications, items are grouped into different classes, categories or genres so that we can assume those in a category have very similar characteristics. For example, the genre of the music or movies, the type of the venues and the category of the items in M-Commerce websites. Moreover, we usually have more information about the items so that we can differentiate among those in one category and hence recommend more appropriately. For instance, the actors in a movie, the tags assigned to a music and the exact location of a venue. We can utilize the category of items and other metadata to better infer the co-occurrence pattern of items. X-IIRPF is an extension of IIRPF which assumes that we have such metadata about them, and uses it to infer the inter-item relations more efficiently. The intensity function of X-IIRPF is the same as IIRPF but α_{pq} is defined as:

$$\alpha_{pq} = \pi_{b_p b_q} d_v(l_p, l_q), \quad (21)$$

where $\pi_{b_p b_q}$ is the interest of users for adopting item of category b_p after an item from category b_q . d_v is any kernel function that shows the items with similar features are more likely to be adopted one after another. For example, in location recommendation, since users tend to visit nearby locations in a sequence, $d_v(l_p, l_q)$ should be a decreasing function of the

distance between venues and in movie recommendation, users are usually fan of specific actors or directors and hence $d_v(l_p, l_q)$ should be an increasing function of the number of common actors in movies p and q . The graphical representation of this model is depicted in Fig. 3c. Although this is a good candidate for applications in which we have this kind of metadata, IIRPF is a general method that can be extended in a variety of ways to utilize different types of items' metadata.

3.3 Prediction and Recommendation Using RPF

RPF models the tendency of users to engage with different items over time using the proposed intensity function and hence is able to recommend the most appropriate item to the users at any time. In order to do so, we should infer the posterior of the latent variables of the model and compute the expected intensity of user to engage with different items and recommend the items to the user sorted based on the expected intensity. As it is discussed in Section 4, we estimate the posterior of latent variables with a factorized distribution over all latent variables which is most similar to the posterior using Bayesian mean field approximation and hence the expected intensity could be calculated as follows:

$$\begin{aligned} \mathbb{E}[\lambda_{up}(t)] = & \mathbb{E}[\theta_u]^\top \mathbb{E}[\beta_p] + \sum_{v \in N_u} \sum_{e \in H_{vp}(t)} \mathbb{E}[\tau_{vu}] g_{w_1}(t_e, t) \\ & + \mathbb{E}[\phi_u] \sum_{e \in H_{up}(t)} \mathbb{E}[\alpha_{p_e p}] g_{w_2}(t_e, t), \end{aligned} \quad (22)$$

where $\mathbb{E}[\alpha_{pq}]$ in IIRPF and XIIRPF models is $\mathbb{E}[\beta_p]^\top \mathbb{E}[\beta_q]$ and $\mathbb{E}[\pi_{b_p b_q}] d_v(l_p, l_q)$ respectively.

Moreover, RPF models are able to predict the returning time of the users to products. To achieve this end, we adopt the approach introduced in [7], i.e., we sample the time of next event from the Poisson process with intensity $\mathbb{E}[\lambda_{up}(t)]$ using Ogata's thinning algorithm [26] and report the sample mean as the expected returning time of user to the item.

4 INFERENCE

As it was mentioned in Section 3.3, the performance of the model to recommend the right item to the users and predicting the returning time of the users accurately, strongly depends on inferring the posterior of the latent variables efficiently. Since XIIRPF model with dynamic users' and items' features is the most general form of RPF and has all features of other variants of RPF, in this section we propose a scalable inference algorithm for Dynamic XIIRPF based on mean-field variational inference [27].

In order to make the model conditionally conjugate and obtain simple updates, we introduce an auxiliary variable s_n for each user-item interaction e_n , which denotes the factor that triggered e_n . Since the triggering factor of engaging a user with an item is either one of its previous user-item interactions or matching between one of the user u 's feature vector components and product p 's features, therefore, we can define the conditional intensity of an event and its triggering factor as:

$$\lambda_{up}^*(t, s) = \begin{cases} \theta_{u s}^i \beta_{p s}^j h_i(t) l_j(t) & -K \times I \times J < s \leq 0 \\ \tau_{u s} g_{w_1}(t - t_s) & 0 < s < N, p_s = p \\ \phi_u \alpha_{p s} g_{w_2}(t - t_s) & 0 < s < N, u_s = u, p_s \neq p. \end{cases} \quad (23)$$

Algorithm 1. Dynamic XIIRPF Variational Inference

```

1: Initialize  $\gamma$  randomly
2: while  $\Delta \text{Elbo} > \epsilon$  do ▷ check for model convergence
3:   for  $e_n \in D$  do
4:     for  $m = -I \times J \times K : -1$  do
5:        $\gamma_n^s(m) \propto \exp(\mathbb{E}[\ln(\theta_{u_n k_m}^m)] + \mathbb{E}[\ln(\beta_{p_n k_m}^m)])$ 
6:       for  $m = 1 : n - 1$  do
7:         if  $p_m = p_n$  then
8:            $\gamma_n^s(m) \propto \exp(\mathbb{E}[\ln(\tau_{u_m u_n})]) g_{\omega_1}(t_m, t_n)$ 
9:         else if  $u_m = u_n$  then
10:           $\gamma_n^s(m) \propto \exp(\mathbb{E}[\ln(\phi_{u_n} \alpha_{p_m p_n})]) g_{\omega_2}(t_m, t_n)$ 
11:       for all users  $u \in U$  do
12:          $\gamma_u^{\mu, \text{shp}} \leftarrow \lambda_u^{\mu, \text{shp}} + |\{v : u \in N(v)\}| \lambda_{uv}^{\tau, \text{shp}}$ 
13:          $\gamma_u^{\mu, \text{rte}} \leftarrow \lambda_u^{\mu, \text{rte}} + \sum_{v: u \in N(v)} \mathbb{E}[\tau_{uv}]$ 
14:          $\gamma_u^{\eta, \text{shp}} \leftarrow \lambda_u^{\eta, \text{shp}} + K I \lambda_u^{\theta, \text{shp}}$  ▷ Update  $q(\eta_u)$ 
15:          $\gamma_u^{\eta, \text{rte}} \leftarrow \lambda_u^{\eta, \text{rte}} + \sum_{k=1}^K \sum_{i=1}^I \mathbb{E}[\theta_{uk}^i]$ 
16:          $\gamma_u^{\psi, \text{shp}} \leftarrow \lambda_u^{\psi, \text{shp}} + \lambda_u^{\phi, \text{shp}}$  ▷ Update  $q(\psi_u)$ 
17:          $\gamma_u^{\psi, \text{rte}} \leftarrow \lambda_u^{\psi, \text{rte}} + \mathbb{E}[\phi_u]$ 
18:          $\gamma_u^{\phi, \text{shp}} \leftarrow \lambda_u^{\phi, \text{shp}} + \mathbb{E}[\bar{c}_u]$  ▷ Update  $q(\phi_u)$ 
19:          $\gamma_u^{\phi, \text{rte}} \leftarrow \sum_{e \in E_u} \alpha_{pe} G_{\omega}(T - t_e) + \mathbb{E}[\psi_u]$ 
20:         for  $v \in N(u)$  do ▷ Update  $q(\tau_{uv})$ 
21:            $\gamma_{vu}^{\tau, \text{shp}} \leftarrow \lambda_{vu}^{\tau, \text{shp}} + \mathbb{E}[c_{vu}]$ 
22:            $\gamma_{vu}^{\tau, \text{rte}} \leftarrow \sum_{e \in H_v} G_{\omega}(T - t_e) + \mathbb{E}[\mu_v]$ 
23:         for  $k = 1 : K$  do ▷ Update  $q(\theta_{uk}^i)$ 
24:           for  $i = 1 : I$  do
25:              $\gamma_{uki}^{\theta, \text{shp}} \leftarrow \lambda_{uki}^{\theta, \text{shp}} + \sum_{p \in P} \sum_{j=1}^J \mathbb{E}[c_{upk}^{ij}]$ 
26:              $\gamma_{uki}^{\theta, \text{rte}} \leftarrow \sum_{j \in J} F_{ij}(T) \sum_{p \in P} \mathbb{E}[\beta_{pk}^j] + \mathbb{E}[\eta_u]$ 
27:         for all products  $p \in P$  do
28:            $\gamma_p^{\xi, \text{shp}} \leftarrow \lambda_p^{\xi, \text{shp}} + K J \lambda_p^{\beta, \text{shp}}$ 
29:            $\gamma_p^{\xi, \text{rte}} \leftarrow \lambda_p^{\xi, \text{rte}} + \sum_{k=1}^K \sum_{j=1}^J \mathbb{E}[\beta_{pk}^j]$ 
30:           for  $k = 1 : K$  do ▷ Update  $q(\beta_{pk}^j)$ 
31:             for  $j = 1 : J$  do
32:                $\gamma_{pkj}^{\beta, \text{shp}} \leftarrow \lambda_{pkj}^{\beta, \text{shp}} + \sum_{u \in U} \sum_{i=1}^I \mathbb{E}[c_{upk}^{ij}]$ 
33:                $\gamma_{pkj}^{\beta, \text{rte}} \leftarrow \sum_{i \in I} F_{ij}(T) \sum_{u \in U} \mathbb{E}[\theta_{uk}^i] + \mathbb{E}[\xi_p]$ 
34:         for all product categories  $b \in B$  do
35:            $\gamma_b^{\rho, \text{shp}} \leftarrow \lambda_b^{\rho, \text{shp}} + |B| \lambda_{bb'}^{\pi, \text{shp}}$  ▷ Update  $q(\rho_b)$ 
36:            $\gamma_b^{\rho, \text{rte}} \leftarrow \lambda_b^{\rho, \text{rte}} + \sum_{b' \in B} \mathbb{E}[\pi_{bb'}]$ 
37:         for all product categories  $b' \in B$  do
38:            $\gamma_{bb'}^{\pi, \text{shp}} \leftarrow \lambda_{bb'}^{\pi, \text{shp}} + \mathbb{E}[c_{bb'}]$  ▷ Update  $q(\pi_{bb'})$ 
39:            $\gamma_{bb'}^{\pi, \text{rte}} \leftarrow \mathbb{E}[\rho_b] + \sum_{e \in \mathcal{H}_b} \phi_{ue} \sum_{q \in \mathcal{H}} d_v(t_{pe}, t_q) G_{\omega_2}(T - t_e)$ 

```

Since it's not possible to find the exact posterior of the latent variables, we consider a factorized distribution over all latent variables and find the distribution that is most similar to the posterior using Bayesian mean field approximation. Namely,

$$\begin{aligned}
& q(S, \beta, \theta, \tau, \pi, \phi, \xi, \eta, \mu, \psi, \rho) \\
&= \prod_{e \in E} q(s_e | \gamma_e^s) \prod_{p, k, j} q(\beta_{pk}^j | \gamma_{pk}^{\beta, j}) \prod_{u, k, i} q(\theta_{uk}^i | \gamma_{uk}^{\theta, i}) \\
& \prod_u q(\xi_u | \gamma_u^{\xi}) \prod_p q(\eta_p | \gamma_p^{\eta}) \prod_{u, v} q(\tau_{uv} | \gamma_{uv}^{\tau}) \prod_u q(\mu_u | \gamma_u^{\mu}) \\
& \prod_{b, b'} q(\pi_{bb'} | \gamma_{bb'}^{\pi}) \prod_b q(\rho_b | \gamma_b^{\rho}) \prod_u q(\phi_u | \gamma_u^{\phi}) \prod_u q(\psi_u | \gamma_u^{\psi}).
\end{aligned} \tag{24}$$

where, $q(s_e | \gamma_e^s)$ is a multinomial distribution and all other factors are gamma distributed. Using mean field theorem, we know that the optimal approximate functions are as follows:

$$\ln q_j^*(\cdot) = \mathbb{E}_{i \neq j} [\ln p(E, S, \beta, \theta, \tau, \pi, \phi, \xi, \eta, \mu, \psi, \rho)] + \text{const.} \quad (25)$$

Therefore, we iterate over all factors and update their parameters based on Eq. (25). The inference algorithm is described in Algorithm 1. The complexity of each iteration of the algorithm is $O(N^2 + UK + PK)$ where N is the number of train user-item interaction due to the loops in lines 6–10 and 11 and 27. However, it can be reduced to $O(N + UK + PK)$ using the dynamic programming technique. The scalability of the algorithm is investigated in experiments section. In this algorithm, c_{uv} denotes the number of events of user v triggered by a previous event of user u . Moreover, the functions $F_{ij}(T)$ and $G_{\omega}(T)$ denote the integrals of the functions $h_i(t)l_j(t)$ and $g_{\omega}(t)$, respectively.

5 EXPERIMENTS

In this section, we do extensive experiments to show the efficiency of the proposed inference algorithm and also to show the superior performance of different variants of RPF in comparison to state-of-the-art PF and time sensitive recommendation methods on large-scale real world datasets from different application domains. We show that RPF not only efficiently models the recurrent user behaviors over time, but also effectively captures the changes in trends and user interests where other temporal models are incapable of.

5.1 Synthetic Data

In order to evaluate the proposed inference algorithm, we first generate a synthetic dataset using SRPF method with known parameter values and then train a model with the generated data and evaluate performance of the inference algorithm in terms of accuracy in estimating the parameters, convergence rate, scalability and the number of events needed to learn the parameters accurately.

Experimental Setup. Our synthetic recommendation task consists of 1000 users and 1000 items. We generated a random network between users with an average degree 50. We sampled the parameters of RPF from the associated distributions introduced in the Section 3. Finally, we generated about 1M events using the Ogata's thinning algorithm [26]. We repeated each experiment 10 times and the results are reported by taking the average over these runs.

Results. Fig. 4 presents the performance of proposed RPF framework over synthetic data. We reported the Mean Absolute Error (MAE) between the true parameters of the model and the estimated ones. Fig. 4a shows that the proposed method only requires around 300 iterations to converge which shows the fast convergence of the inference algorithm. Fig. 4b presents the MAE over different number of train events. As it is expected, with the increase in the number of events, the performance of proposed method improves as well. Furthermore, it verifies that the proposed inference algorithm only requires a modest number of events to achieve a good performance. Additionally, we investigate how well the temporal patterns in data is captured by the

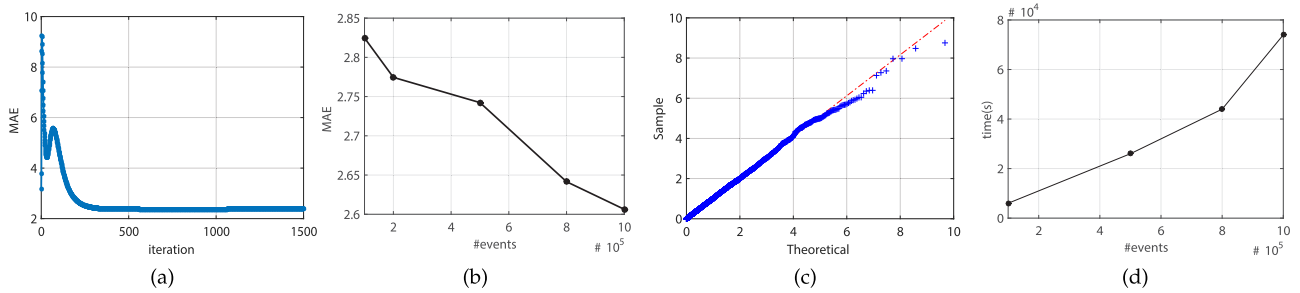


Fig. 4. The performance of proposed inference algorithm over synthetic data. (a) Mean Absolute Error (MAE) versus the number of iterations. (b) MAE versus the number of events. (c) Quantile plot of intensity integrals. (d) Run-time versus the number of events.

proposed algorithm. According to the time-change theorem [28], given all successive event times of a particular point process with intensity function $\lambda(t)$, the set of intensity integrals $\int_{t_i}^{t_{i+1}} \lambda(t) dt$ should conform to the unit-rate exponential distribution, if the samples are truly generated from the process $\lambda(t)$. Fig. 4c demonstrates the quantiles of intensity integrals computed using the learned intensities. The conformity of the empirical quantiles and the true ones suggests that the model captures the temporal dynamics very well. The last but not the least, we investigate the scalability of algorithm. Fig. 4d shows that with the increase in the number of events the inference algorithm scales almost linearly.

5.2 Real Data

To show the performance of different variants of RPF in capturing the dynamic preferences and the recurrent consumption behavior we compare it with the recent state-of-the-art recommendation methods on datasets from different application domains. Notably, we chose *DPF* [3], *SPF* [2], and *HPF* [1], from PF framework as the competitor baseline which are popular recommendation methods but do not take the temporal behavior of users into account. We also selected the *TSRS* [7], which is a low rank method based on Hawkes process and *COEVOLVE* [8], which is a coevolutionary latent feature process that captures the coevolving nature of users' and items' features over time.

We evaluate the performance of different variants of RPF and other competitor methods on four real datasets from different domains; *Last.fm*, *Alibaba Mobile Commerce*, *Foursquare*, and *IPTV*. All datasets contain timestamped actions which make them appropriate benchmark for comparing the proposed method to the other state-of-the-art methods. In order to compare the methods, we sort all events based on time and choose the first 80 percent as the train data and leave the rest as the test data. In order to evaluate the performance of each method, we train that model using the training events and use the learned parameters to predict the events in the test data. The details of each dataset is as follows:

Last.fm. This dataset contains the music listening logs of 1200 users and 1000 artists. There are around 418K events in total which spans a period of six months.

Tianchi Mobile Data. It contains the user interactions with items in Alibaba's mobile M-Commerce platform [29]. The dataset includes four behavior types: click, collect, add-to-cart, and payment. We only considered the click events and used the item categories as the recommendation targets. Our data contains roughly 1000 users, 2100 items, and the total 1.2M events.

Foursquare Data. This dataset contains Foursquare users' check-ins in London [30]. We selected the active users with more than 30 check-ins and the venues with more than 50 check-ins which resulted in 67K check-ins of 890 users in 1158 venues in London, which spans from Mar. 2011 to Sep. 2011. The venues of this dataset fall in 151 categories which are considered as the category metadata in X-IIRPF method. Moreover, the exact geolocation of these venues are available in this dataset which are treated as the meta-data to differentiate among the venues in one category in the X-IIRPF method.

IPTV Data. This dataset contains the users' history of watching TV programs on online TV streaming services [8]. The dataset contains 7100 users and 436 TV programs. Each TV program has 1420 features including 1073 actors, 312 directors, 22 genres, 8 countries and 5 years which are coded as a binary vector and are treated as meta-data for items in the X-IIRPF method. The dataset contains 2.4M events and spans for a period of 11 months.

5.2.1 Qualitative Analysis

We first explore the fitted model to confirm that the model can capture different patterns in data. Fig. 6 represents the summation of the learned intensity of users listening to the Britney Spears through time in the Last.fm dataset. We see a sudden increase in the learned intensity at 26th of August 2016, which is the time her new album, *Glory*, was released. It is interesting that there are two other jumps in the learned intensity before this sudden increase at August 11th and August 21st, which are when the track *Clumsy* and *Deluxe* version of the album were released. This is how RPF framework can capture different temporal dynamics in item consumptions.

Next, we investigate how much the proposed method can capture the similarity between users' preferences. To this end, we created the similarity matrix using the learned user preferences θ_u . Each a_{uv} entry of the defined similarity matrix is equal to $\theta_u^\top \theta_v$ which indicates that how much the latent preferences of users u and v are close. This is roughly proportional to their shared items of interest. We also created an empirical similarity matrix, using the Jaccard similarity between all pair of users based on the size of shared consumed items. Each b_{uv} entry of the defined similarity matrix is equal to $b_{uv} = \frac{|P_u \cap P_v|}{|P_u \cup P_v|}$ where P_u is the set of items user u was engaged with and hence it measures how much the interests of users u and v are close. Figs. 7a and 7b present the learned similarity matrix and the empirical one for the Last.fm dataset. Interestingly, these two matrices conform by a reasonable amount. We also highlighted the two

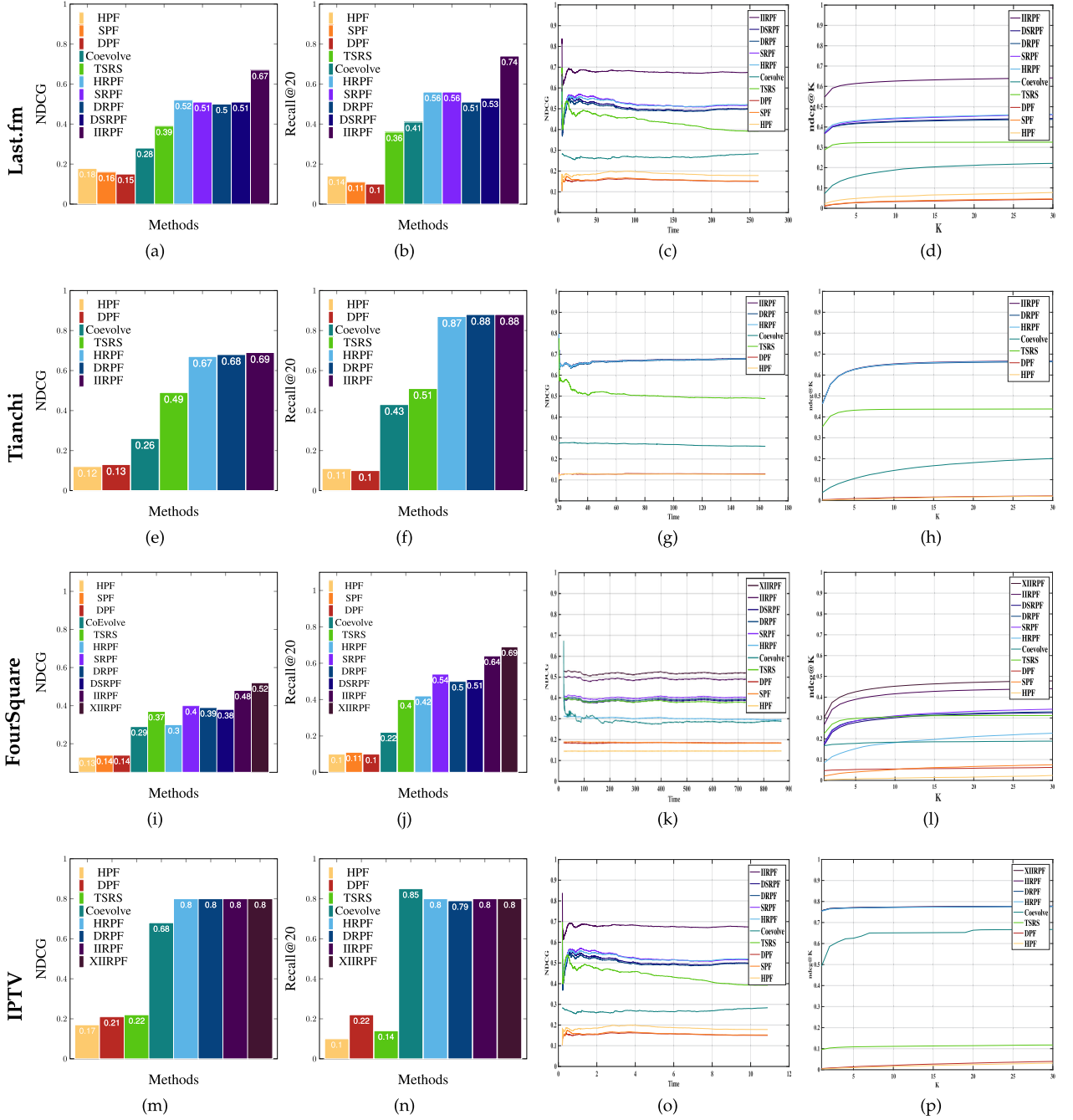


Fig. 5. Performance of different methods in item recommendations. The first two columns demonstrate the $NDCG@P$ and $Recall@20$ for different methods over different datasets, respectively, where P is the number of items in the dataset. The third column shows the $NDCG@20$ for different methods over time, i.e., the x -axis is the time after the last train event and the value at time t represent the $NDCG@20$ for all test events until t time units after the last train event. The fourth column shows the $NDCG@K$ for different methods over different values of k . The first row presents measures for *Last.fm* dataset; the second row presents the results for *Tianchi* dataset the third row presents the results for *Foursquare* dataset, and the fourth row presents the results for the *IPTV* dataset.

blocks of highly similar users in the matrices with squares. As it can be seen, there is a strong correlation between the two matrices, and the blocks match each other. It basically shows the proposed model can efficiently learn user preferences and can capture their similarities.

5.2.2 Quantitative Analysis

In order to evaluate the performance of RPF methods qualitatively, we measure their ability in predicting the next item that each user will be engaged with, i.e., *Item Prediction*, and

the time that next time she will return to the system, i.e., *Returning Time Prediction*. We will discuss these two tasks in the following.

Item Prediction. Our first quantitative task is predicting the items that each user will interact with. In order to compare RPF methods with others, we trained each model using the first 80 percent of data and the rest 20 percent is treated as the test data. We generate top k recommendations for each user, using items with the highest probability under each method. The methods are evaluated based on $NDCG@k$ and $Recall@k$ metrics:

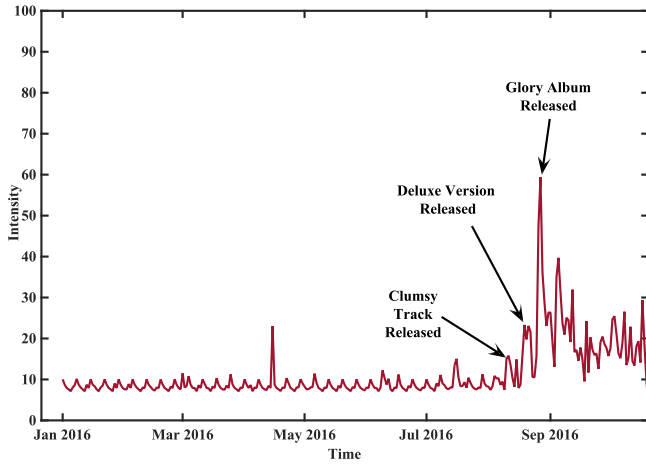


Fig. 6. The total learned intensity of users for listening to Britney Spears songs over time.

$$\text{Recall}@k = \frac{1}{N} \sum_{n=1}^N \mathcal{I}(\text{rank}(p_n) < k) \quad (26)$$

$$\text{NDCG}@k = \frac{1}{N} \sum_{n=1}^N \frac{\mathcal{I}(\text{rank}(p_n) < k)}{\log_2 1 + \text{rank}(p_n)}, \quad (27)$$

where N is the number of test data and \mathcal{I} is the indicator function. $\text{Recall}@K$ shows the percentage of items that are ranked in the first k . $\text{NDCG}@k$ is a weighted version of $\text{Recall}@K$ which the interested reader is referred to [3] for their detailed properties.

Figs. 5a, 5b, 5e, 5f, 5i, 5j, 5m, and 5n demonstrate the $\text{NDCG}@P$ and $\text{Recall}@20$ for different methods over different datasets, respectively where P is the number of items in the dataset. It is noteworthy that since there is no social network in Tianchi and IPTV datasets, the result of social based methods (SPF, SRPF, and DSRPF) are not reported on them. Besides, since there is no meta-data for Last.fm and Tianchi datasets, the results of X-IIRPF method are reported only for Foursquare and IPTV datasets. As it is evident in figures, RPF-based methods perform significantly better than competitors and TSRS and COEVOLVE methods perform marginally better than PF-based methods. The main reason may be due to the fact that PF-based methods ignore the time and summarize the history of interactions of a user with an item in a count variable while TSRS and COEVOLVE and all RPF variants treat time as a natural constituent of the model and consider the time of previous user-item engagements in their recommendation. Moreover, using factorization technique to infer the users' and items' latent features and considering different affecting factors (such as dynamics and diversity of users' interests and items' popularity, social influence among users and inter-item relations) are the main reasons that allow the RPF methods better predict the users' interests over time and recommend more appropriate items in comparison to TSRS and COEVOLVE methods. The COEVOLVE method considers both the impact of time and the user-item interactions, but it suffers from the high complexity and needs so much data per user-item to be trained. As a result, over the Foursquare dataset that has many items and few events per user-item, the COEVOLVE performs poorly, but for the IPTV dataset that has much

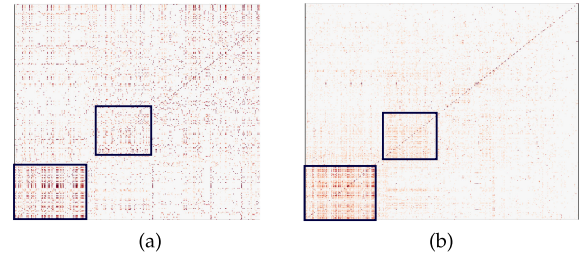


Fig. 7. (a) Learned similarity between user interests. (b) The empirical Jaccard similarity between user listening behaviors over Last.fm dataset.

more events per user-item, the COEVOLVE can be learned effectively and performs significantly better in all measures. The TSRS method itself has a more emphasis on previous user interactions than the intrinsic user preferences. It performs poorly over the IPTV dataset compared to other methods. To explain this observation, note that the IPTV dataset contains the user history of watching different TV programs and capturing the preferences of users is more central than paying attention to the timings. Therefore, the TSRS performs poorly over IPTV dataset in comparison to other temporal competitors.

Furthermore, IIRPF and X-IIRPF methods (which consider the impact of interaction of a user with an item on their engagement in the future) improve the performance almost over all datasets. The improvements are more significant on Last.fm and Foursquare datasets. This may be due to the fact that the previous locations that a user visited in Foursquare and the previous songs he listened on Last.fm has a notable impact on the future locations he will visit and the future songs he will play. Moreover, utilizing the meta-data in X-IIRPF method, improves the performance over Foursquare dataset while it has no significant effect on IPTV dataset. It may be due to two reasons. First, there are a few events per venue in the Foursquare dataset and hence learning the inter-items relations is challenging and hence utilizing the venues' metadata can immensely help. However, the number of events per item in IPTV dataset is the order of magnitude more than the Foursquare dataset and hence learning the tendency of users to items over time is a simpler task. Therefore, the performance of almost all RPF variants is satisfactory on this dataset and utilizing the metadata can't make any significant improvement. The second reason may be due to the nature of metadata of the two datasets. Users usually check-in nearby venues sequentially in Foursquare and hence the geolocation of venues provides a valuable information for predicting next place that the user will check-in. Moreover, there is a common pattern in user movements among venues with specific types, e.g., people usually go to a restaurant after cinema and hence considering the type of venues as the category of items in X-IIRPF helps to learn such behavioral patterns. However, in IPTV dataset, we have only genre and the artists of the movies as the metadata which may not be very informative on users' interest.

We also evaluated the impact of size of recommendation list (k) and the time interval between the test data and the last train data on the performance of different methods. Figs. 5c, 5g, 5k, and 5o show the $\text{NDCG}@20$ for different methods over time, i.e., the x -axis is the time after the last

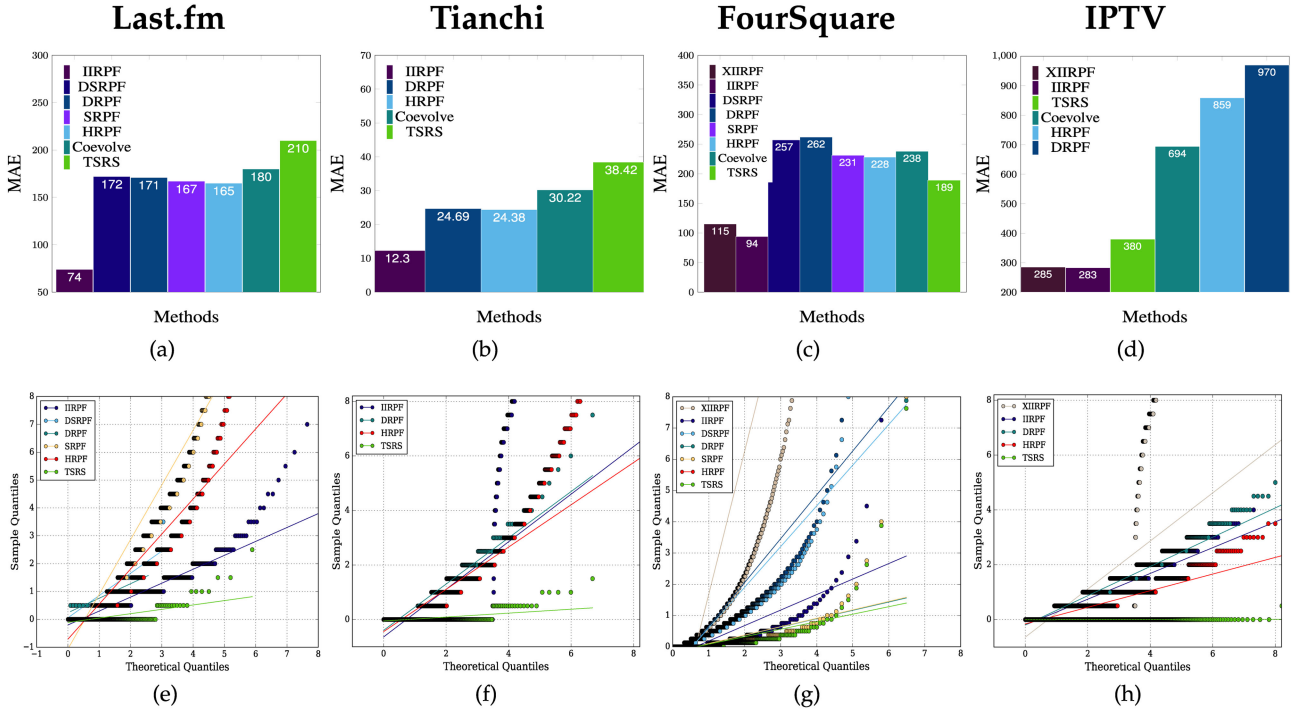


Fig. 8. Performance of different methods on returning-time prediction. (a), (b), (c), and (d) show MAE of different methods in predicting the returning time of users for Last.fm, Tianchi, Foursquare, and IPTV datasets. (e), (f), (g), and (h) show quantile plot of different methods for different datasets.

train event and the value at time t represent the NDCG@20 for all test events until t time units after the last train event. As it can be seen, the performance of different methods degrades over time due to the fact that the needs and interests of users change over time and hence predicting their far future needs become more and more challenging. However, RPF-based methods perform marginally better than competitors in all datasets and the relative order of different methods are preserved over time.

Figs. 5d, 5h, 5l, and 5p show the NDCG@k for different methods over different values of k . All methods exhibit an early increase in NDCG@k with increase of k . Moreover, RPF-based methods consistently being the best ones.

In a nutshell, COEVOLVE is focused on modeling the evolution of users' and items' features over time and doesn't consider the social influence of users on each other and inter-item relations explicitly and its parameter set is very large and hence needs a large set of data to be learnt. TSRS also pays more attention to the impact of previous events of the user, and less attention is paid to the exact modeling of intrinsic user preferences. It uses constraint optimizations to implicitly consider the impact of different items on each other, while in the RPF-based methods the user preferences are modeled using explicit random variables and are inferred through matrix factorization. Hence the RPF-based methods can better capture both the intrinsic user preferences and the impact of previous events. Among the RPF-based methods, IIRPF that models the inter-item relations has a better accuracy in predicting the users' interests and in the case that informative metadata about the items is available, X-IIRPF has a better performance.

Returning Time Prediction. Finally, we evaluated the performance of different methods in predicting the time when

a user will return to the system. To this end, we trained the model using the train data and predicted the time that each user will return to the system for the first time in the test data and computed the MAE between the estimated time and the ground-truth time for all users. Figs. 8a, 8b, 8c, and 8d, show the time prediction results. All RPF-based methods perform better than TSRS and COEVOLVE on the Last.fm and Tianchi datasets, while on Foursquare dataset the HRPF, SRPF, DRPF, and DSRPF methods perform closely to the TSRS and COEVOLVE. The TSRS and the COEVOLVE methods perform better than HRPF and DRPF methods on IPTV dataset. Interestingly, the new extensions to the RPF method (IIRPF, X-IIRPF) significantly improve the performance in time prediction over all datasets. The main reason may be the fact that these methods consider the impact of current user-item interaction on future user engagement with other items and hence they can better capture the temporal dynamics and predict the future return times. Besides, while utilizing meta-data improves the performance in item prediction for Foursquare dataset, it has no significant effect in time prediction which may be an indicator that the current location category influences the future locations to be visited but has no notable effect on the time of visits. In summary, the proposed RPF-based methods model the dynamics of user interests, consider the peer influence of the neighboring users consuming a product and consider the impact of user-item interaction on future engagement of user with other products and hence are more expressive than the competitors.

We also plotted the Quantile plot for different methods. To this end, we compare the theoretical quantiles from the exponential distribution with the ones that each model has learnt from real-world data. The closer the slope is to one,

the better a model matches the event patterns. Figs. 8e, 8f, 8g, and 8h show that the RPF-based models fit the observed event sequence better than the TSRS and COEVOLVE. Over the Last.fm dataset, HRPF that has less parameters, better fits the real data. On the Tianchi dataset, the DRPF and IIRPF has almost the same quantile. This is due to the large number of events per user-item on this dataset, which helps these methods to better capture the true dynamics of real data.

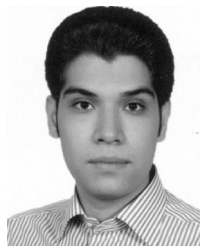
6 CONCLUSIONS

In this work, we presented a novel framework, *Recurrent Poisson Factorization* (RPF), for building recommendation systems from implicit feedback data. RPF extends the Poisson factorization (PF) methods by modeling the aggregated count data with a Poisson process, therefore, enabling the model to account for recurrent activities and answering time-sensitive queries. RPF can handle the change of users and items specification through time (DRPF), can consider socially-influenced user preferences (SRPF), can capture the heterogeneity amongst users and items (HRPF), can learn the effect of interaction of a user with an item on its future engagement with other items (IIRPF), and also is able to utilize the items metadata to better infer such relations among items (X-IIRPF). Experiments on synthetic data and two real world datasets demonstrate superiority of the proposed framework over several state-of-the-art methods on item prediction and returning time prediction—mainly due to its unique temporal capabilities. Furthermore, in terms of interpretability, our model is able to discover interesting patterns such as peer influence between users, trending products, and their temporal properties.

For future work, we would like to incorporate nonparametric models into RPF in order to add more flexibility on capturing dynamic preferences and peer influences. Moreover, although the proposed variational inference is able to scale to large datasets using implicit information, but it is not readily extensible to web-scale sparse datasets where the users activities has a heavy tail distribution. Learning temporal models within a sparse dataset is a challenging problem and is an interesting and important future work.

REFERENCES

- [1] P. Gopalan, J. M. Hofman, and D. M. Blei, "Scalable recommendation with hierarchical poisson factorization," in *Proc. 31st Conf. Uncertainty Artif. Intell.*, 2015, pp. 326–335.
- [2] A. J. Chaney, D. M. Blei, and T. Eliassi-Rad, "A probabilistic model for using social networks in personalized item recommendation," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 43–50.
- [3] L. Charlin, R. Ranganath, J. McInerney, and D. M. Blei, "Dynamic poisson factorization," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 155–162.
- [4] P. Gopalan, F. J. Ruiz, R. Ranganath, and D. M. Blei, "Bayesian nonparametric poisson factorization for recommendation systems," in *Proc. Int. Conf. Artif. Intell. Statistics*, 2014, pp. 275–283.
- [5] P. K. Gopalan, L. Charlin, and D. Blei, "Content-based recommendations with poisson factorization," in *Proc. Advances Neural Inf. Process. Syst.*, 2014, pp. 3176–3184.
- [6] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, pp. 89–97, 2010.
- [7] N. Du, Y. Wang, N. He, J. Sun, and L. Song, "Time-sensitive recommendation from recurrent user activities," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 3492–3500.
- [8] Y. Wang, N. Du, R. Trivedi, and L. Song, "Coevolutionary latent feature processes for continuous-time user-item interactions," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 4547–4555.
- [9] A. Ahmed, Y. Low, M. Aly, V. Josifovski, and A. J. Smola, "Scalable distributed inference of dynamic user interests for behavioral targeting," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 114–122.
- [10] S. A. Hosseini, H. R. Rabiee, H. Hafez, and A. Soltani-Farani, "Classifying a stream of infinite concepts: A bayesian non-parametric approach," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2014, pp. 1–16.
- [11] J. F. C. Kingman, *Poisson Processes*. Hoboken, NJ, USA: Wiley, 1993.
- [12] S. A. Hosseini, K. Alizadeh, A. Khodadadi, A. Arabzadeh, M. Farajtabar, H. Zha, and H. R. Rabiee, "Recurrent poisson factorization for temporal recommendation," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 847–855.
- [13] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.
- [14] D. Liang, J. Allosa, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 59–66.
- [15] D. Liang, L. Charlin, J. McInerney, and D. M. Blei, "Modeling user exposure in recommendation," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 951–961.
- [16] S. A. Hosseini, A. Khodadadi, A. Arabzadeh, and H. R. Rabiee, "Hnp3: A hierarchical nonparametric point process for modeling content diffusion over social media," in *Proc. IEEE 16th Int. Conf. Data Mining*, 2016, pp. 943–948.
- [17] M. Farajtabar, Y. Wang, M. G. Rodriguez, S. Li, H. Zha, and L. Song, "Coevolve: A joint point process model for information diffusion and network co-evolution," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 1954–1962.
- [18] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1555–1564.
- [19] A. Khodadadi, S. A. Hosseini, E. Tavakoli, and H. R. Rabiee, "Continuous-time user modeling in presence of badges: A probabilistic approach," *ACM Trans. Knowl. Discovery Data*, vol. 12, no. 3, 2018, Art. no. 37.
- [20] H. Jing and A. J. Smola, "Neural survival recommender," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 515–524.
- [21] L. Tran, M. Farajtabar, L. Song, and H. Zha, "Netcodec: Community detection from individual activities," in *Proc. SIAM Int. Conf. Data Mining*, 2015, pp. 91–99.
- [22] M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha, "Multistage campaigning in social networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2016, pp. 4718–4726.
- [23] M. Farajtabar, J. Yang, X. Ye, H. Xu, R. Trivedi, E. Khalil, S. Li, L. Song, and H. Zha, "Fake news mitigation via point process based intervention," *Int. Conf. Mach. Learn.*, pp. 1097–1106, 2017.
- [24] A. Zarezade, A. Khodadadi, M. Farajtabar, H. R. Rabiee, and H. Zha, "Correlated cascades: Compete or cooperate," *AAAI*, pp. 238–244, 2017.
- [25] A. Simma and M. I. Jordan, "Modeling events with cascades of poisson processes," in *Proc. Twenty-Sixth Conf. Uncertainty Artificial Intell.*, 2010, pp. 546–555.
- [26] Y. Ogata, "On lewis' simulation method for point processes," *IEEE Trans. Inf. Theory*, vol. 27, no. 1, pp. 23–31, Jan. 1981.
- [27] M. I. Jordan, *Learning in Graphical Models*. Berlin, Germany: Springer, 1998, vol. 89.
- [28] D. J. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. Berlin, Germany: Springer, 2007.
- [29] Z. Yi, D. Wang, K. Hu, and Q. Li, "Purchase behavior prediction in M-commerce with an optimized sampling methods," in *Proc. IEEE Int. Conf. Data Mining Workshop*, Nov. 2015, pp. 1085–1092.
- [30] D. Hristova, M. J. Williams, M. Musolesi, P. Panzarasa, and C. Mascolo, "Measuring urban social diversity using interconnected GEO-social networks," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 21–30.



Seyed Abbas Hosseini received the BSc degree in software engineering and MSc degree in artificial intelligence from the Sharif University of Technology, Tehran, Iran, in 2012 and 2014, respectively. He is currently working toward the PhD degree in the Department of Computer Engineering, Sharif University of Technology. Since 2014, he has worked as a technical consultant at Intelligent Data Processing Group (PegahTech) and is an adjunct lecturer at the Sharif University of Technology. His current research interests include modeling marked events, user profiling and their applications in recommendation and online advertising.



Ali Khodadadi received the BSc and MSc degrees in information technology engineering from the Sharif University of Technology, Tehran, Iran, in 2010 and 2012, respectively. He is currently working toward the PhD degree in the Department of Computer Engineering, Sharif University of Technology. His current research interests include machine learning with application in social and complex networks including inferring networks of diffusion, Bayesian community detection, multi-layer network analysis, and user activity modeling.



Keivan Alizadeh received the BSc degree in software engineering from the Sharif University of Technology, Tehran, Iran, in 2017. His current research interests include the application of temporal point processes for recommendation.



Ali Arabzadeh received the BSc degree in software engineering from the Sharif University of Technology, Tehran, Iran, in 2016. He is currently working toward the MSc degree in the Sharif University of Technology, Tehran, Iran. His current research interests include the application of temporal point processes to modeling dependent data.



Mehrdad Farajtabar received the BS and MS degrees from the Sharif University of Technology, Iran, in 2009 and 2012, respectively. Since 2013, he has been working toward the PhD degree in the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology. His research interests are machine learning and scalable data mining methods for the modeling, analysis, optimization, and control of processes that take place over networks.



Hongyuan Zha received the PhD degree in scientific computing from Stanford University, in 1993. He is a professor with the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, and East China Normal University. Since then, he has been working on information retrieval, machine learning applications, and numerical methods. He is the recipient of the Leslie Fox Prize (1991, second prize) of the Institute of Mathematics and its Applications, the Outstanding Paper Awards of the 26th International Conference on Advances in Neural Information Processing Systems (NIPS 2013), and the Best Student Paper Award (advisor) of the 34th ACM SIGIR International Conference on Information Retrieval (SIGIR 2011). He was an associate editor of the *IEEE Transactions on Knowledge and Data Engineering*.



Hamid R. Rabiee (SM'07) received the BS and MS degrees (with great distinction) in electrical engineering from CSULB, Long Beach, CA, in 1987 and 1989, the EEE degree in electrical and computer engineering from USC, Los Angeles, CA, in 1993, and the PhD degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 1996. From 1993 to 1996, he was a member of technical staff at AT&T Bell Laboratories. From 1996 to 1999, he worked as a senior software engineer at Intel Corporation. He was also with PSU, OGI, and OSU universities as an adjunct professor of electrical and computer engineering from 1996-2000. In September 2000, he joined the Sharif University of Technology, Tehran, Iran. He was also a visiting professor with the Imperial College of London for the 2017-2018 academic year. He is the founder of the Sharif University Advanced Information and Communication Technology Research Institute (AICT), ICT Innovation Center, Advanced Technologies Incubator (SATI), Digital Media Laboratory (DML), Mobile Value Added Services Laboratory (VASL), Bioinformatics and Computational Biology Laboratory (BCB), and Cognitive Neuroengineering Research Center. He has also been the founder of many successful High-Tech start-up companies in the field of ICT as an entrepreneur. He is currently a professor of computer engineering at the Sharif University of Technology, and director of AICT, DML, and VASL. He has been the initiator and director of many national and international level projects in the context of the Iran National ICT Development Plan and UNDP International Open Source Network (IOSN). He has received numerous awards and honors for his Industrial, scientific and academic contributions. He has acted as chairman in a number of national and international conferences, and holds three patents. He is also a member of the IFIP Working Group 10.3 on Concurrent Systems, and a senior member of the IEEE. His research interests include statistical machine learning, Bayesian statistics, data analytics and complex networks with applications in complex networks, multimedia systems, cloud and IoT privacy, bioinformatics, and brain networks.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.