Modeling Interdependent and Periodic Real-World Action Sequences

Takeshi Kurashima
NTT Corp. & Stanford University
kurashima.takeshi@lab.ntt.co.jp

Tim Althoff
Stanford University
althoff@cs.stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

ABSTRACT

Mobile health applications, including those that track activities such as exercise, sleep, and diet, are becoming widely used. Accurately predicting human actions in the real world is essential for targeted recommendations that could improve our health and for personalization of these applications. However, making such predictions is extremely difficult due to the complexities of human behavior, which consists of a large number of potential actions that vary over time, depend on each other, and are periodic. Previous work has not jointly modeled these dynamics and has largely focused on item consumption patterns instead of broader types of behaviors such as eating, commuting or exercising.

In this work, we develop a novel statistical model, called *TIPAS*, for Time-varying, Interdependent, and Periodic Action Sequences. Our approach is based on personalized, multivariate temporal point processes that model time-varying action propensities through a mixture of Gaussian intensities. Our model captures short-term and long-term periodic interdependencies between actions through Hawkes process-based self-excitations. We evaluate our approach on two activity logging datasets comprising 12 million real-world actions (*e.g.*, eating, sleep, and exercise) taken by 20 thousand users over 17 months. We demonstrate that our approach allows us to make successful predictions of future user actions and their timing. Specifically, TIPAS improves predictions of actions, and their timing, over existing methods across multiple datasets by up to 156%, and up to 37%, respectively. Performance improvements are particularly large for relatively rare and periodic actions such as walking and biking, improving over baselines by up to 256%. This demonstrates that explicit modeling of dependencies and periodicities in real-world behavior enables successful predictions of future actions, with implications for modeling human behavior, app personalization, and targeting of health interventions.

1 INTRODUCTION

Activity tracking applications for mobile health have become an important part of people's daily lives. A US-nationwide study in 2013 found that 69% of adults keep track of a health indicator, and 21% among them used an app or device to do so [24]. In activity logging applications such as Fitbit, Under Armour Record, and Argus, users might take one of many possible actions from a large and diverse space of potential actions at any point in time. Users continuously track many actions of their lives including exercise, diet, sleep, and commuting behavior with the goal of improving self-knowledge and personal well-being [6, 7, 45, 46]. User modeling is critical to making activity logging applications more useful by providing users with personalized experiences matching their specific objectives [12, 19, 23, 27, 54]. This has the potential to significantly improve people's health, for instance by preventing negative health outcomes and promoting the adoption and maintenance of healthy behaviors [4, 25, 42, 49]. However, successful personalization of systems rests on the ability to predict the user's next actions and when they will occur [17, 21, 54].

Predicting actions is important because these predictions facilitate personalization of the user interface and user experience in order to provide users with what they need, without them asking for it explicitly [41]. For example, in activity logging applications we can predict when the user will eat dinner and their future location in order to provide relevant recommendations [52]. Accurate and contextualized predictions could further help users to realize their personal goals by reminding them to measure their weight or notifying them about the exercise the next morning [46]. Besides predicting the action itself, it is also critical to predict its timing, so that recommendations and reminders can be made at the right time. For instance, diet reminders ideally are delivered just *before* meal choices are made [26, 42, 49]. More generally, predicting user actions also enables digital personal assistants that support users with relevant information including local recommendations, traffic, weather, events, and news [19].

However, human behavior is extremely complex, which makes accurate predictions very challenging. In particular, human behavior is (1) *time-varying*, (2) *interdependent*, and (3) *periodic*. First, real-world actions *vary over time*, for example based on time of day (*e.g.*, spending time with friends in the evenings) and day of week (*e.g.*, going hiking on weekends) [14, 35]. Second, actions are also *interdependent* in the short-term and the long-term (*e.g.*, brushing teeth before going to bed, or drinking water after workouts). Third, humans are creatures of habit [17] and exhibit *periodic* behaviors [5, 16, 18], such as brushing teeth every morning and evening.

Current user modeling techniques (*e.g.*, [8, 11, 17, 27, 32, 35, 37, 50, 55]) do not jointly model all these key aspects (time variation, interdependence, periodicity) of real-world action sequences. However, failing to account for any of them results in decreased predictive performance. For example, consider the task of predicting the time of a user's next meal. When not accounting for periodicity, one would miss the fact that the user's early lunch might lead to an earlier dinner as well. However, this could be a critical mistake if the user relies on timely diet reminders.

While great advances have been made in modeling specific aspects of behavior in narrow application domains, in particular in the space of recommender systems [35] or information retrieval [2,

WWW 2018, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5639-8/18/04.
https://doi.org/10.1145/3178876.3186161

3, 48], these lines of work have largely focused on consumption of items such as specific videos, songs, or websites [8, 11, 32, 35, 50]. In all these cases, users repeat the *same* high-level actions such as watching one video after another. In contrast, we consider predicting *which* higher-level action, out of many, the user will take next; for example, whether they will watch a movie or go for a run (not which specific movie or run). Furthermore, previous work has often focused on predicting short-term actions such as the next unix command [17], web page request [55], or TV episode watched [50]. Instead, we are interested in predicting longer-term actions such as a commute in the evening or a run the next morning.

This work. We present a new model for the task of predicting future user actions and their timing. First, we empirically demonstrate that action sequences exhibit time-varying, interdependent, and periodic patterns and that modeling them is critical to accurate predictions of user actions. Our model extends prior work on multivariate temporal point processes and is the first model to account for all three key properties. The model addresses (1) time-varying propensities of actions through mixture of Gaussians, (2) short-term dependencies between actions through a Hawkes process, and (3) long-term periodicity with time-dependent Weibull distributions. We call this model *TIPAS* referring to Time-varying Interdependent Periodic Action Sequences. TIPAS is personalized to each user through learning user-specific action preferences. We further develop an EM-based algorithm to fit this model using maximum likelihood estimation.

We demonstrate that TIPAS can scale to real-world datasets from Argus and Under Armour activity logging applications that capture 12 million actions taken by 20 thousand users over 17 months. We evaluate our model on these two activity logging datasets capturing ten different real-world actions, and demonstrate that we can predict the user's next logged activity (*e.g.*, run, eat, or sleep) and the timing of that activity (continuous, non-discretized timestamp) based on the user's previous actions and their timing.

Further, we show that TIPAS accurately captures all three fundamental behavioral patterns in real-world data. Using several domains of real-world actions, we demonstrate that our model outperforms eleven existing approaches on tasks of predicting actions by up to 156% as well as predicting when they will occur by up to 37%. Further, we show that performance improvements over baselines are particularly large for rare actions, increasing prediction accuracy over baselines by up to 256%. We find that these performance improvements are crucially enabled by modeling time-varying propensities of actions and their dependencies, and by modeling long-term periodicities of actions. Empirically, modeling time-varying propensities of actions yields 53% and 40% accuracy on the two activity logging datasets. Modeling short-term dependencies between actions improves this to 59% and 49%, respectively. Also capturing long-term periodicities of actions further improves this to 61% and 51%, respectively. Thus, capturing these three properties is essential to predicting periodic and interdependent human action sequences.

2 RELATED WORK

Predicting the next action. Much work has focused on predictions of next actions, including unix commands [17], user interface actions to enable interface adaption [27], web page requests allowing for prefetching and latency reduction [55], clicks on web search [3], user behavior anomalies [37], product item preferences [35, 44], online purchases [34], mobile apps used [10], and future location-based checkins [9, 13, 38]. Many of these works (*e.g.*, [9, 13, 32, 37]) have formulated the problem as a discrete-time sequence prediction task and used Markov models. However, Markov models assume unit time steps and are further unable to capture long-range dependencies since the overall state-space will grow exponentially in the number of time steps considered [19]. Other works have used LSTM models [30], which also assume discrete time steps and are limited in their interpretability.

In contrast, we also model and predict *when* the next action will occur, which is critical to surface recommendations and reminders at the right time. In addition, instead of specific web queries or item consumption, we consider a broader set of higher-level actions such as watching a movie, going for a run, or going to sleep.

Patterns of repeat consumption. Another line of work has studied repeated actions, in particular in the space of item consumption, including video binge watching [50], music listening [32], web page revisitation patterns [2], and repeated web search queries [48]. More recent work has focused on modeling these behaviors and proposed models based on patterns of boredom [11, 32] and recency [8].

Importantly, patterns of human actions in the real world, which are modeled in this work, are fundamentally different from patterns of item consumption due to their higher-level notion (*e.g.*, watching a movie, not which specific one). For example, patterns of boredom [11, 32] suggest that the probability of repeating an action within a short amount of time is unlikely. In contrast, we empirically observe the opposite in some cases, such as users commuting one way being extremely likely to commute back in the near future. More generally, real-world actions are characterized by more complex dynamics including time-varying behavior, interdependence, and periodicity of actions.

Temporal point processes. Recent work has considered temporal point processes [15] including Poisson and Hawkes [28] process-based models to predict the timing of future actions. Temporal point processes have been used to predict continuously time-varying item preferences [21], and to model user influence in a social network [31, 47, 53], the co-evolution of information and network structure [22], competition between products [51], mobility patterns in space and time [19], user return times [33], and temporal document clustering [20, 40]. Perhaps the closest works to ours are by Du et al. [19, 21], which also attempt to predict both future user actions *and* their timing.

We extend this line of work by explicitly modeling time-varying action propensities as well as developing a novel combination of Exponential and Weibull kernels to model short-term and long-term periodic dependencies between actions. Further, we demonstrate that these aspects are critical when predicting real-world user actions and their timing across two real-world activity logging datasets.

3 TASK DESCRIPTION

The task considered in this work is, given a user and her history, a timestamped sequence of her actions in the past, to predict the user's future actions and the timing of these actions.

Formally, let U be a set of users. Each user $u \in U$ has an action se-

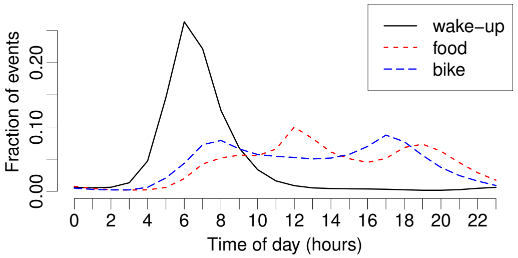


Figure 1: Fraction of events within each time-of-day window. Notice that action propensity is clearly non-uniform and sometimes multi-modal.

quence, which we represent as a user history $H_u = \{(a_{un}, t_{un})\}_{n=1}^{N_u}$ with a total of N_u events. Each element in H_u is an event consisting of an action and timestamp representing that user u takes action $a_{un} \in A$ at time $t_{un} \in \mathbb{R}^+$ $(0 \le t_{un} \le T)$. T denotes the end of our observation period. For example, a_{un} could correspond to watching a movie or going for a run (but not which specific movie or run). We assume that events are sorted by their timestamps, $t_{un} \le t_{un'}$ for $n < n'$. We denote the set of events before time t in user history H_u as $H_{ut} = \{(a', t') | (a', t') \in H_u \text{ and } t' < t\}$.

The task of predicting future user actions and when they will occur can now be formalized as follows. Given user history H_{ut} up until time t, predict the next K actions the user will take and their timing $\{(a_k, t'_k)\}_{k=1}^{K}$, where $t'_k > t$ (*i.e.*, these are the actions with the smallest $t'_k > t$ among all possible future user actions).

Here, we propose a novel multivariate temporal point process model for this prediction task and focus on the case of $K = 1$.

4 EMPIRICAL OBSERVATIONS

Next we make a series of empirical observations about important properties of real-world action sequences that will provide the basis for our statistical model TIPAS (Section 5). Accounting for these observations will lead to superior predictive models (Section 6).

4.1 Dataset Description

To illustrate critical properties of real-world actions we use a dataset of logged activities from a mobile activity logging application, Argus by Azumio, used in previous work on activity logging [6, 7, 45]. This smartphone app allows users to track their various daily activities including drink, sleep, heart rate, running, weight, food, walking, biking, workout, and stretching actions. For example, the drink action is logged to keep track of the user's daily fluid intake and the workout action is used to log various indoor exercises such as weightlifting or indoor-cycling. This dataset includes over four thousand active users taking 1.2 million actions over the course of seven months (all users logged at least two unique actions per day on average). Due to the popularity of the app, this set of users is very diverse in terms of age, gender, health status, country of origin, and other features [7]. We note that the following properties of real-world actions also hold in other datasets including Under Armour activity logging app data (Section 6.1).

4.2 Properties of Real-World Action Sequences

Next, we describe three important properties of real-world action sequences and present empirical justification for each. TIPAS will explicitly address all three properties (Section 5).

Time-varying propensities of actions. Human real-world actions vary over time, for example based on time of day (*e.g.*, having meals in the morning, at mid-day, and in the evening) and day of week (*e.g.*, working out on the weekends). This dynamic is evident in real-world data of human activities as illustrated in Figure 1. The figure shows the distribution of the timing of three types of actions throughout the day: wake-up (from sleep), food, and bike. First, we observe that all three distributions are clearly non-uniform over time. For example, wake-up actions are clustered at around 07:00 hours (7 am). Second, we observe significant differences in the propensities to take different actions. While for sleep we observe a uni-modal distribution concentrated in the early morning, we observe a bi-modal distribution for biking. The two modes in the morning and evening likely correspond to commute activity where users log their rides to and from work. We also observe two clear modes for food during lunch and dinner times. However, breakfast times seem to vary more widely across users and are more dispersed. Summarizing, we observe non-uniform, temporal distributions with varying number of modes that vary across actions.

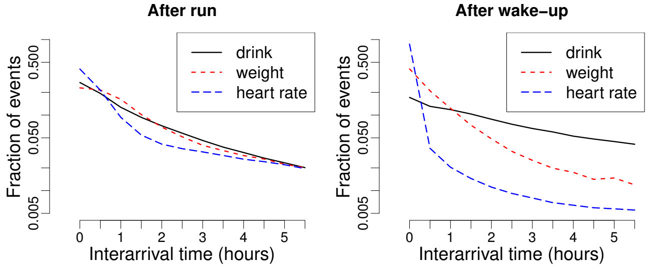


Figure 2: Fraction of interarrival times at each time window (log scale). Figure shows drink, weight, and heart rate measurement actions taken after run (left) and wake-up (right) actions. Notice that the likelihood of drink, weight, and heart-rate actions declines quickly after both run and wake-up actions. However, note that fraction of heart-rate actions decreases much quicker after wake-up than after runs.

Short-term dependencies between actions. Certain actions make it more likely that some other actions will follow shortly. For example, people might drink water right after exercising or stretch right before running. In order to examine the short-term correlations between actions, we extract interarrival times between pairs of actions (*i.e.*, the elapsed time between the two actions) from a set of action histories. Figure 2 shows the distribution of interarrival times for several pairs of user actions after run actions (left) and sleep actions (right). We make two important observations. First, the monotonically decreasing curves show that the likelihood of other actions is largest right after an action has happened. After this, the likelihood declines very quickly in a monotonic manner (note the log scale of the Y-axis). This points to a self-excitation dynamic of logged human actions. For example, users are very likely to follow up on runs or waking up from sleep with drinking water or measuring their heart rate or weight. Specifically, about 50% of the weight measurements which happen within 6 hours of waking up occur right within the first 30 minutes. Second, we find that the action dependency patterns vary across actions. For example, drinking is more common after runs than after waking up and heart rate measurements fall off more sharply right after waking up than after runs. In summary, human actions in the real world often trigger other actions within a short period but these patterns are different across actions. We can leverage these correlations among

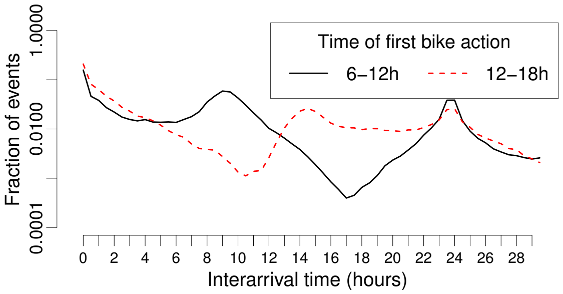


Figure 3: Density describing when the next biking action will occur (interarrival time) given that the prior bike action occurred between 6-12h (solid black line) or between 12-18h (red dashed line) after midnight (timing, not duration). Notice the multiple and different modes of the two distributions indicating that biking actions recur periodically but that the period timing depends on the time of day.

actions when predicting future events.

Long-term periodic effects. Humans exhibit periodic behaviors such as waking up at about the same time every morning or commuting back home after about 8 hours of work. Therefore, logged real-world actions likely follow periodic recurrence patterns in which the same action tends to recur at certain, regular intervals. While some of these periodic behaviors are rooted in intrinsic biological rhythms such as sleep [5], others are dictated by extrinsic factors (*e.g.*, when does one have to be in the office in the morning), or based on personal habits [17] (*e.g.*, measuring one's weight before breakfast). We illustrate these dynamics using interarrival times between bike events in real-world data. Figure 3 shows the distribution of interarrival times up to a maximum of 30h, where the two curves represent observed dynamics when the first of the two bike actions occurred during specific times of day (6-12h in solid black and 12-18h in dashed red line; note that these correspond to the timing and not the duration of the bike action).

We make two important observations. Previously, we had observed that short-term dependencies between actions exhibit monotonic decay. Here, we observe that this strong monotonic decay only holds within the first few hours and that we observe multiple additional peaks for both distributions after this initial phase. Second, we observe that these peaks occur at different times based on when the first action occurred. In the case of the distribution for bike actions following a 6-12h bike ride, we observe peaks at around 9 and 24 hours (interarrival times), and peaks at around 14 and 24 hours for bike actions following a 12-18h bike ride. This behavior is not unexpected. When biking in the morning (6-12h), the next bike ride will likely be a commute back around 9h later. However, if the bike ride happens in the evening (12-18h), the next bike ride is likely not during the middle of the night, but after 14 hours or at around 8:00h in the morning. In addition, both curves exhibit a daily, 24h, periodicity. Modeling these periodicities allows us to capture user-specific timing of, for example, a late evening commute signaling a later start the next morning. In conclusion, two important dynamics could help predicting future real-world actions: actions display periodic recurrence and the time of recurrence can depend on the time of day.

5 PROPOSED MODEL

In this section, we operationalize the insights gained from empirical observations (Section 4) in a probabilistic model based on temporal point processes, called TIPAS.

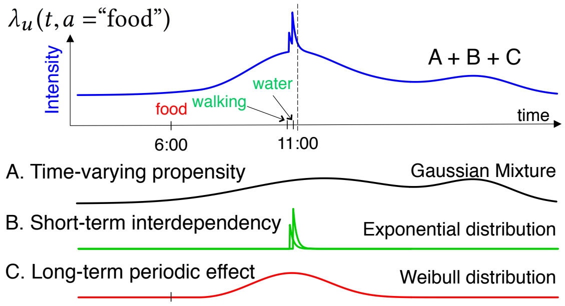

$\lambda_u(t, a =$"food"$)$
Intensity
A + B + C
water
walking
food
time
6:00
11:00
A. Time-varying propensity
Gaussian Mixture
B. Short-term interdependency
Exponential distribution
C. Long-term periodic effect
Weibull distribution

Figure 4: Conceptual model overview. Intensity function of "food" for user u is modeled by the sum of three types of influences; time-varying background intensity (A; black), short-term dependencies (B; green), and long-term periodic effects (C; red). (A) Time-varying background intensity models typical times for food (*e.g.*, having lunch around 12:00h). (B) Events of "walking" and "water" might trigger "food" action within a short period of time. (C) Due to the early breakfast (6:00h), we might expect an earlier lunch.

5.1 Background on Temporal Point Processes

A temporal point process is a random process whose realization consists of a list of discrete events localized in time, $\{t_n\}_{n\in\mathbb{N}}$ with $t_n \in \mathbb{R}^+$. We introduce univariate temporal point processes for ease of exposition, though we will be using multivariate point processes to model the joint occurrence dynamics of multiple different actions (description inspired by [22]; more background in [1]). Let H_t be the history of events before time t. Temporal point processes can be characterized via the conditional intensity function representing a stochastic model for the time of the next event given all the times of previous events. Formally, the conditional intensity function $\lambda(t)$ is the conditional probability of observing an event in a small window $[t, t + dt)$ given the history H_t; that is, $\lambda(t)dt = \mathbb{P}\{\text{event in}[t, t + dt)|H_t\}$. The conditional probability that no event happens during $[t, t')$ is $S(t') = \exp(-\int_t^{t'} \lambda(\tau)d\tau)$ and the conditional density that an event occurs at time t' is $f(t') = \lambda(t')S(t')$ [1]. Thus, the log-likelihood of a list of events $t_1, t_2, \ldots, t_n$ in an observation window $[0, T)$, where $T > t_n$, can be expressed as

$$\mathcal{L}(t_1, t_2, \ldots, t_n) = \sum_{i=1}^{n} \log \lambda(t_i) - \int_0^T \lambda(\tau)d\tau\,. \qquad (1)$$

The intensity λ can take various functional forms leading to a homogeneous Poisson process if $\lambda(t)$ is constant, to an inhomogeneous Poisson process if $\lambda(t)$ is time-varying but independent of the event history H_t, or to a Hawkes process if the intensity models mutual self-excitations between events [1]. Our TIPAS model is based on multivariate Hawkes processes [28].

5.2 Model Definition

We model user actions as a multivariate temporal point process with a time-varying intensity based on three factors based on our empirical observations (Section 4). The following intensity function models the rate that action a occurs at time t in user history u,

$$\lambda_u(t, a) = \alpha_{ua} + Time_u(t, a) + ShortTerm_u(t, a) + LongTerm_u(t, a). \quad (2)$$

Here, we use an additive decomposition of the intensity instead of modeling more complex interaction effects, because this approach is simple yet powerful and has been proven empirically successful as well [22, 31, 47]. This model is conceptually visualized in Figure 4. The figure shows how the overall intensity function $\lambda_u(t, a)$ (blue; here, a = food) is the sum of the time-varying propensity $Time_u(t, a)$ (black), short-term dependencies between actions $ShortTerm_u(t, a)$ (green), and long-term periodic effects $LongTerm_u(t, a)$ (red) between actions (for simplicity, we assume no personalization, *i.e.* $\alpha_{ua} = 0$). Note that our model does account for randomness, in the sense that not all actions may strictly conform to short-term and long-term patterns, through the personalized and time-varying baserates. In fact, learning model parameters from real data tries to account for all actions and will adapt distributional parameters to best explain all occurring actions. Next, we formally define each of the four factors in turn.

Personalized action preferences: α_{ua}**.** We include personalized user preferences for specific actions through a constant additive factor $\alpha_{ua} \geq 0$ for each action and user. Note that one could also model user preferences to be time-varying instead. However, this would lead to a very large number of parameters and we show in Section 6 that this simple model works well in practice.

Time-varying propensities of actions: $Time_u(t, a)$**.** Events can occur without influence from preceding events according to the background intensity function $Time_u(t, a)$. Having observed that the propensity of actions varies across time of day (Section 4.2), we model the background intensity of action a as a function of time-of-day through a Gaussian mixture model. We define:

$$Time_u(t, a) = \sum_{z \in Z} \frac{\beta_{az}}{\sqrt{2\pi\sigma_{az}^2}} \exp\Big(-\frac{\big(l_t - \mu_{az}\big)^2}{2\sigma_{az}^2}\Big) \ , \tag{3}$$

where $z \in Z$ represents the latent class of Gaussian mixture model (the number of mixtures can be determined through cross-validation). For each action a and latent mixture class z, $\mu_{az} > 0$ and $\sigma_{az} > 0$ denote the mean and standard deviation of the Gaussian distribution. The importance of that mixture on the overall intensity function $Time_u(t, a)$ is captured by $\beta_{az} \geq 0$. l_t corresponds to the time of day of timestamp t (*i.e.*, elapsed time since midnight). We show in Section 6.3 that Gaussian mixtures fit temporal variation in real-world data well.

Short-term dependencies between actions: $ShortTerm_u(t, a)$**.** To model short-term dependencies between actions, we consider how the rate at which action a occurs at time t (Equation 2) is influenced by actions a' which occurred at previous time $t' < t$. We model these influences as a Hawkes process exhibiting self-excitations using Exponential decay functions starting at the time of previous actions. As demonstrated in Section 4.2, the short-term influence of previous actions diminishes quickly and monotonically, making the Exponential distribution a natural choice for the decay function. We define:

$$ShortTerm_u(t, a) = \sum_{(t', a') \in H_{ut}} \theta_{a'a}\omega_{a'a} \exp(-\omega_{a'a}\Delta_{t't}) \ , \tag{4}$$

where $H_{ut} = \{(t', a')|(t', a') \in H_u \text{ and } t' < t\}$ is the set of events before time t in history u, and $\Delta_{t't} = t - t'$ is the time difference between time t' and time $t > t'$. Further, $\omega_{a'a} \geq 0$ determines how quickly action a' triggers action a (shape of Exponential distribution), and $\theta_{a'a} \geq 0$ determines how likely action a' triggers action a (scaling of distribution). We estimate these parameters for each pair of actions (a', a). Therefore, this component of the model captures the interdependencies between different actions (*e.g.*, drinking after running), as well as the self-exciting effects of actions (*e.g.*, running after running). We show in Section 6.3 that a Hawkes process with Exponential decay function fits short-term action dependencies in real-world data well.

Long-term periodic effects: $LongTerm_u(t, a)$**.** We model the long-term periodic effects between identical actions (*e.g.*, run to run) using Weibull distributions. The Weibull distribution is a continuous distribution with positive support (*i.e.*, for $\Delta_{t't} > 0$) that is well suited to model long-term effect patterns at different points in time and with different variance around its mean. We model the rate at which action a occurs at time t influenced by a previous event of action a at time t' as follows:

$$LongTerm_u(t, a) = \sum_{(t', a') \in H_{ut}^a} \phi_{c_{t'}a}\gamma_{c_{t'}a}\kappa_{c_{t'}a}\Delta_{t't}^{\kappa_{c_{t'}a}-1} \exp(-\gamma_{c_{t'}a}\Delta_{t't}^{\kappa_{c_{t'}a}}) \tag{5}$$

where $H_{ut}^a = \{(t', a')|(t', a') \in H_u \text{ and } t' < t \text{ and } a' = a\}$ is the set of events of action a before time t in history u, and $\Delta_{t't} = t - t'$ is again the time difference between time t' and time $t > t'$. As shown in Section 4.2, long-term effects vary based on the time of day of action a'. This is captured through the parameter $c_{t'} \in C$ that represents discretized time-of-day windows (*e.g.*, using four classes as 0-6h, 6-12h, 12-18h, and 18-24h). This allows us to learn time-of-day-dependent distributions modeling different periodicities. Parametrized by this time-of-day category $c_{t'}$ and by action a, $\gamma_{c_{t'}a} \geq 0$, $\phi_{c_{t'}a} \geq 0$ determine how quickly and how likely (influence) action a' (which occurred in time-of-day window $c_{t'}$) triggered its following event of action a. $\kappa_{c_{t'}a} \geq 0$ determines the shape of the Weibull distribution. In Section 6.3, we demonstrate that the Weibull distribution closely match periodic dynamics in real-world data.

5.3 Model Inference

We use maximum likelihood estimation to infer the parameters of our proposed model (Equation 2). The unknown parameters are $\boldsymbol{\alpha} = \{\{\alpha_{ua}\}_{u \in U}\}_{a \in A}$, $\boldsymbol{\beta} = \{\{\beta_{az}\}_{a \in A}\}_{z \in Z}$, $\boldsymbol{\mu} = \{\{\mu_{az}\}_{a \in A}\}_{z \in Z}$, $\boldsymbol{\sigma} = \{\{\sigma_{az}\}_{a \in A}\}_{z \in Z}$, $\boldsymbol{\Theta} = \{\{\theta_{a'a}\}_{a \in A}\}_{a' \in A}$, $\boldsymbol{\Omega} = \{\{\omega_{a'a}\}_{a \in A}\}_{a' \in A}$, $\boldsymbol{\Phi} = \{\{\phi_{ca}\}_{c \in C}\}_{a \in A}$, $\boldsymbol{\Gamma} = \{\{\gamma_{ca}\}_{c \in C}\}_{a \in A}$, and $\boldsymbol{K} = \{\{\kappa_{ca}\}_{c \in C}\}_{a \in A}$. The set of all parameters is denoted by $\boldsymbol{\Psi} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{\Phi}, \boldsymbol{\Gamma}, \boldsymbol{K}\}$.

The log-likelihood function (Equation 1), given a set of user histories $\mathcal{H} = \{H_u\}_{u \in U}$, can be expressed as:

$$\mathcal{L}(\Psi|\mathcal{H}) = \sum_{u \in U} \sum_{n=1}^{N_u} \log \lambda_u(t_{un}, a_{un}) - \sum_{u \in U} \int_0^T \sum_{a \in A} \lambda_u(t, a) dt \ , \tag{6}$$

where the last term, the expectation function, represents the expected number of events in the time period from 0 to T. Combining Equations (2)-(6), the log-likelihood can be written as follows:

$$\begin{aligned}
&\mathcal{L}(\Psi|\mathcal{H}) = \\
&\sum_{u \in U} \sum_{n=1}^{N_u} \log\Bigg\{\alpha_{ua_{un}} + \sum_{z \in Z} \frac{\beta_{a_{un}z}}{\sqrt{2\pi\sigma_{a_{un}z}^2}} \exp\Big(-\frac{\big(l_{t_{un}} - \mu_{a_{un}z}\big)^2}{2\sigma_{a_{un}z}^2}\Big) \\
&+ \sum_{m=1}^{n-1} \theta_{a_{um}a_{un}}\omega_{a_{um}a_{un}} \exp(-\omega_{a_{um}a_{un}}\Delta_{t_{um}t_{un}}) \\
&+ \sum_{l=1}^{n-1}\Big(I(a_{ul} = a_{un})\phi_{c_{ul}a_{un}}\gamma_{c_{ul}a_{un}}\kappa_{c_{ul}a_{un}}\Delta_{t_{ul}t_{un}}^{\kappa_{c_{ul}a_{un}}-1} \\
&\times \exp(-\gamma_{c_{ul}a_{un}}\Delta_{t_{ul}t_{un}}^{\kappa_{c_{ul}a_{un}}})\Big)\Bigg\} - \sum_{u \in U} \int_0^T \sum_{a \in A} \lambda_u(t, a) dt \ ,
\end{aligned} \tag{7}$$

where $c_{ul} \in C$ represents time-of-day category of l-th event of u, and $I(\cdot)$ is the indicator function. The integral in Equation 7 can be analytically calculated.

Inspired by previous work [22, 53], we develop an efficient inference algorithm to maximize the log-likelihood based on the EM algorithm. By iterating the E-step and the M-step until convergence, we obtain a local optimum solution for $\mathbf{\Psi}$.

E-step. Conceptually, we introduce latent variables $\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}$ to capture why each event was triggered either through user preference, time-varying background intensity, short-term action interdependencies, or long-term periodic effects. Let $p_{0,un}$ be the probability that the n-th event of user u was triggered by user preference, $p_{z,un}$ be the probability that the n-th event of user u was triggered by the time-varying background intensity function of latent class z, $q_{um,un}$ be the probability that the n-th event of user u was triggered by the short-term effect of the m-th event of user u, and $r_{ul,un}$ be the probability that the n-th event of user u was triggered by the long-term effect of the l-th event of user u.

In E-step, k-th estimate of $p^k_{0,un}$, $p^k_{z,un}$, $q^k_{um,un}$, and $r^k_{ul,un}$ are calculated by:

$$p^k_{0,un} = \frac{\alpha^k_{ua_{un}}}{R_{un}} \quad , \tag{8}$$

$$p^k_{z,un} = \frac{1}{R_{un}} \frac{\beta^k_{a_{un}z}}{\sqrt{2\pi(\sigma^k_{a_{un}z})^2}} \exp\Big(-\frac{\big(l_{t_{un}} - \mu^k_{a_{un}z}\big)^2}{2(\sigma^k_{a_{un}z})^2}\Big) \quad , \tag{9}$$

$$q^k_{um,un} = \frac{1}{R_{un}} \theta^k_{a_{um}a_{un}} \omega^k_{a_{um}a_{un}} \exp(-\omega^k_{a_{um}a_{un}} \Delta_{t_{um}t_{un}}) \quad , \tag{10}$$

$$\begin{aligned} r^k_{ul,un} = \Bigg\{ & \frac{1}{R_{un}} \phi^k_{c_{ul}a_{un}} \gamma^k_{c_{ul}a_{un}} \kappa^k_{c_{ul}a_{un}} \\ & \times \Delta^{\kappa^k_{c_{ul}a_{un}}-1}_{t_{ul}t_{un}} \exp(-\gamma^k_{c_{ul}a_{un}} \Delta^{\kappa^k_{c_{ul}a_{un}}}_{t_{ul}t_{un}}) \Bigg\} \quad , \end{aligned} \tag{11}$$

where $\mathbf{\Psi}^k = \{\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k, \boldsymbol{\mu}^k, \boldsymbol{\sigma}^k, \mathbf{\Theta}^k, \mathbf{\Omega}^k, \mathbf{\Phi}^k, \mathbf{\Gamma}^k, \boldsymbol{K}^k\}$ is the k-th estimate of parameters in the EM procedure, and R_{un} is the normalization factor in order to satisfy $p^k_{0,un} + \sum_{z \in Z} p^k_{z,un} + \sum_{m=1}^{n-1} q^k_{um,un} + \sum_{l=1}^{n-1} r^k_{ul,un} = 1$.

M-step. We use Jensen's inequality to provide a lower bound for the log-likelihood (Equation 7); this lower bound is often called the Q function. We obtain the next estimate of the parameters by taking the derivative of the Q function with respect to each parameter and setting them to zero:

$$\alpha^{k+1}_{ua} = \frac{\sum_{n=1}^{N_u} I(a_{un} = a) p^k_{0,un}}{T} \quad , \tag{12}$$

$$\beta^{k+1}_{az} = \frac{2\mathcal{T}}{|U|T} \times \frac{\sum_{u \in U} \sum_{n=1}^{N_u} I(a_{un} = a) p^k_{z,un}}{erf(\frac{\mu^k_{az}}{\sqrt{2}\sigma^k_{az}}) + erf(\frac{\mathcal{T} - \mu^k_{az}}{\sqrt{2}\sigma^k_{az}})} \quad , \tag{13}$$

$$\theta^{k+1}_{a'a} = \frac{\sum_{u \in U} \sum_{n=1}^{N_u} \sum_{m=1}^{n-1} I(a_{um} = a', a_{un} = a) q^k_{um,un}}{\sum_{u \in U} \sum_{n=1}^{N_u} I(a_{un} = a') \Big(1 - \exp\big(-\omega^k_{a'a}(T - t_{un})\big)\Big)} \quad , \tag{14}$$

$$\phi^{k+1}_{ca} = \frac{\sum_{u \in U} \sum_{n=1}^{N_u} \sum_{l=1}^{n-1} I(a_{ul} = a, a_{un} = a, c_{ul} = c) r^k_{ul,un}}{\sum_{u \in U} \sum_{n=1}^{N_u} I(a_{un} = a, c_{un} = c) \Big(1 - \exp\big(-\gamma^k_{ca}(T - t_{un})^{\kappa^k_{ca}}\big)\Big)} \quad , \tag{15}$$

Dataset Statistics	Argus	Under Armour
Observation period	7 months	10 months
	Jan-July '15	Jan-Oct '16
# unique actions	10	8
# total users	4,708	15,221
# total actions	2,140,757	9,733,645
Avg. # actions per user	454.7	639.5
Avg. # unique actions per user	6.3	6.8
Avg. # unique actions per user day	2.7	4.4

Table 1: Basic dataset statistics.

where $\mathcal{T}$ is the time period of a day (*i.e.*, 24 hours), $\frac{T}{\mathcal{T}}$ is the number of days representation of the observed period T, and where *erf* denotes the Gauss error function $erf(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^{x} e^{-t^2} \mathrm{d}t$. Because of the exponentials (exp and *erf*) within the expectation function (Equation 7), $\omega^{k+1}_{a'a}$, γ^{k+1}_{ca}, κ^{k+1}_{ca}, μ^{k+1}_{az}, and σ^{k+1}_{az} cannot be solved in closed form. However, by further considering a lower bound for these exponentials $\omega^{k+1}_{a'a}$ and γ^{k+1}_{ca} can be solved in closed form. Their update rules are as follows:

$$\begin{aligned} \omega^{k+1}_{a'a} = & \Bigg\{ \sum_{u \in U} \sum_{n=1}^{N_u} \sum_{m=1}^{n-1} I(a_{um} = a', a_{un} = a) q^k_{um,un} \Bigg\} \\ & / \Bigg\{ \sum_{u \in U} \sum_{n=1}^{N_u} \sum_{m=1}^{n-1} I(a_{um} = a', a_{un} = a) q^k_{um,un} \Delta_{t_{um}t_{un}} \\ & + \sum_{u \in U} \sum_{n=1}^{N_u} I(a_{un} = a') \theta^k_{a'a}(T - t_{un}) \exp\Big(-\omega^k_{a'a}(T - t_{un})\Big) \Bigg\} \quad , \end{aligned} \tag{16}$$

$$\begin{aligned} \gamma^{k+1}_{ca} = & \Bigg\{ \sum_{u \in U} \sum_{n=1}^{N_u} \sum_{l=1}^{n-1} I(a_{ul} = a, a_{un} = a, c_{ul} = c) r^k_{ul,un} \Bigg\} \\ & / \Bigg\{ \sum_{u \in U} \sum_{n=1}^{N_u} \sum_{l=1}^{n-1} I(a_{ul} = a, a_{un} = a, c_{ul} = c) r^k_{ul,un} \Delta^{\kappa^k_{ca}}_{t_{ul}t_{un}} \\ & + \sum_{u \in U} \sum_{n=1}^{N_u} I(a_{un} = a, c_{un} = c) \phi^k_{ca}(T - t_{un})^{\kappa^k_{ca}} \exp\Big(-\gamma^k_{ca}(T - t_{un})^{\kappa^k_{ca}}\Big) \Bigg\} \ . \end{aligned} \tag{17}$$

The other three parameters, κ^{k+1}_{ca}, μ^{k+1}_{az} and σ^{k+1}_{az}, are estimated by maximizing the Q function through the use of a gradient-based numerical optimization method; we used the Newton method. For more details on model inference see the Online Appendix [36].

6 EXPERIMENTS

This section evaluates the predictive performance of our proposed model on two real-world datasets on predicting the next user action and when it will occur. We compare against eleven different baselines on each dataset. However, since many baseline models are unable to make joint predictions of action and timing, we evaluate these two tasks separately. Importantly, this process allows us to identify the individual sources of error that would impact joint predictions. Our implementation is available at snap.stanford.edu/tipas.

6.1 Datasets

Our experiments use two real-world activity logging datasets. In total, these datasets comprise 12 millions real-world actions taken by 20 thousand users over 17 months.

Argus dataset. We use the activity logging data from the Argus mobile app described in Section 4.1. Users in this dataset can log 10 different actions (drink, sleep, heart rate, running, weight, food, walking, biking, workout, and stretching) and our goal is to predict which of these 10 actions a user will take next (and when). Our analyses include users who logged at least two unique actions per day on average (other users might only use the app to for example track their sleep making predictions of actions and their timing almost trivial; we find that our results are robust to different choices of this threshold). We consider 7 months of data from the app in a rolling window evaluation, where we use one month for training and the next for testing (*i.e.*, making out-of-sample predictions; without retraining). As shown in Table 1, the dataset includes 2.1 million actions by over 4 thousand users within the 7 month observation period.

Under Armour dataset (UA). We also use activity logging data from Under Armour mobile apps (*i.e.*, MapMyFitness and MyFitnessPal; focusing on users that are active in both apps). Users in this dataset can log 8 different types of actions (running, walking, biking, workout, breakfast, lunch, dinner, and snacks). Our analyses include users who logged at least four unique actions per day on average, leading to a similar number of unique actions per user on average compared to the Argus dataset (again, our results are robust to different choices of this threshold). We consider 10 months of data from the app and again perform a rolling window evaluation where we train on one month and test on the next. In total, this dataset comprises 15 thousand users taking 9.8 million actions (Table 1).

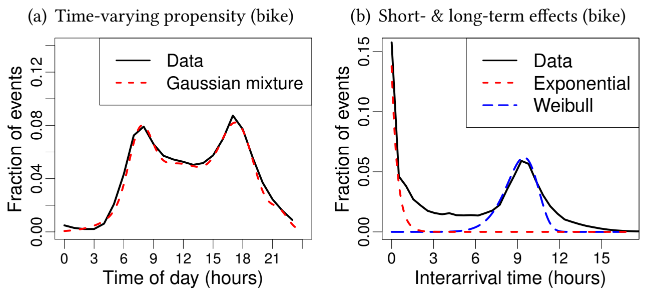
6.2 Model Learning

Note that our model has few core model parameters. In the context of the datasets described above, we have about 500 core model parameters ($\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\Theta}, \boldsymbol{\Omega}, \boldsymbol{\Phi}, \boldsymbol{\Gamma}, \boldsymbol{K}$) and about 25 thousand personalization parameters ($\boldsymbol{\alpha}$). This small, non-redundant set of parameters allows us to train the model efficiently and robustly, and explain model predictions through inspection and visualization of model parameters (Section 6.6), while performing competitively (Section 6.4). However, during training time (but not test time) we also have latent variables ($\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{r}$) that allow us to learn the core model parameters. These latent variables represent which actions trigger which other actions, leading to $O(|U|(max_{u \in U} N_u)^2)$ variables in the worst case. On both datasets, inference of both core model and latent parameters involves solving an optimization problem with over 200 million total variables (Section 5.3; we randomly initialize all parameters). Using our EM-based inference procedure we can robustly infer these parameters in less than ten hours using a single-threaded C++ implementation on a single machine. We find that one month of training data is enough to reliably train our model.

6.3 Validating Parametric Assumptions

In Section 5.2 we developed a model consisting of three parts: time variation modeled using a mixture of Gaussians ($\text{Time}_u(t, a)$), short-term dependencies between actions modeled by a Hawkes process with Exponential decay function ($\text{ShortTerm}_u(t, a)$), and long-term periodicity modeled through Weibull distributions ($\text{LongTerm}_u(t, a)$). Here, we test empirically whether these parametric assumptions hold true in real data. Using the Argus dataset, we inferred appropriate parameters for these distributions.

We demonstrate qualitatively in Figure 5 that the chosen distributions fit real-world dynamics well. Figure 5 (a) shows the time-varying propensity with superimposed mixture of Gaussian fit and (b) shows that, collectively, Exponential and Weibull distribution closely approximate the influence of previous actions (example data is for bike action as seen in Figure 1).


(a) Time-varying propensity (bike)

(b) Short- & long-term effects (bike)

Figure 5: Validation of parametric modeling assumptions (Section 5.2). (a) Mixture of Gaussian closely fits observed time-varying action propensity (here, for bike action). (b) Exponential and Weibull distributions collectively well-approximate short-term dependencies and long-term periodic effects of previous bike actions.

We have further quantitatively evaluated our parametric assumptions and compared our choices to alternative distributions (*e.g.*, Rayleigh and Power-law) through goodness-of-fit tests which have shown that the suggested distributions best fit real-world data.

6.4 Predicting the Next Action

First, we evaluate our proposed model in terms of its accuracy in predicting actions at a given time. The task is to predict the $n+1$-st action a_{un+1} of user u, given time t_{un+1} and past user history $\boldsymbol{H}_u = \{(a_{u1}, t_{u1}), \cdots, (a_{un}, t_{un})\}$. For each two month period in both datasets, we use the first month for training and the second month for testing and perform a rolling window evaluation, where we predict each test set event given all events that happened before it (without retraining). We use accuracy, the percentage of correct predictions, over all test events as our evaluation measure (the most common measure to evaluate recommender systems [29]). We also report macro-averaged recall [39] corresponding to averaging prediction accuracy equally weighted across all action types. This measure highlights differences in predictive performance on rare actions that do not affect the standard accuracy measure very much. We find very similar results using other classification metrics (*e.g.*, ROC AUC, F1). The number of mixtures for the time-varying action propensity (Equation 3) is set via cross-validation. We compare our proposed model against the following seven baseline models, which have proven competitive across a wide variety of prediction tasks and recommender systems:

- **Copy Model:** Simply repeats the user's last action. Several repeat consumption models are variants of this copy model (*e.g.*, [8, 11]).
- **Markov Model:** Predicts the next action based on the most recent actions of the user. We report first to fifth-order Markov models (sixth-order models did not significantly improve performance). Markov models have been used widely to predict next actions (*e.g.*, [9, 32]).
- **Hidden Markov Model (HMM):** This is a Markov model with hidden (unobserved) states. It predicts the next action based on the current, inferred state of the action sequence [37].
- **Factorizing Personalized Markov Chains (FPMC):** This is based on underlying Markov chains where the transitions matrices are user-specific. Matrix factorization models are used to address sparsity issues of these user-specific Markov chains [44].
- **Recurrent Neural Network (RNN):** Feedforward neural net-

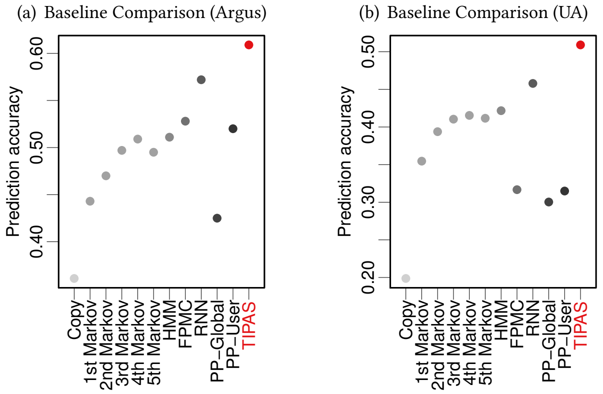


Figure 6: Accuracy when predicting actions. Higher is better. Comparing proposed TIPAS model (red) to baselines (gray). Error bars in all plots correspond to standard errors.

work structure using outputs from the hidden units at the prior time step as the inputs as the current time step. Assumes discrete time steps and no ready-to-use generalizations to continuous time domain exist.

- **PP-Global:** A global Poisson process model. The intensity function is constant over time and defined by $\lambda_u(t,a) = \alpha_a$.
- **PP-User:** A user-specific Poisson process model. The intensity function is constant over time and defined by $\lambda_u(t,a) = \alpha_{ua}$.

Note that Hawkes process models (*e.g.*, [20, 22, 28]) are closely related to the ShortTerm$_u(t,a)$ component of our model (Equation 4). Our proposed model **TIPAS** uses the intensity function of Eq. 2. We predict the most likely user action as $\hat{a} = \text{argmax}_a\, \lambda_u(t_{un+1}, a)$. We also compare the individual model components in an ablation study below.

Results: Comparison to baseline models. Figure 6 compares accuracy of next action prediction. We observe that the eleven baselines achieve accuracies of 36-57% on the Argus dataset and 20-46% on the Under Armour dataset with the RNN baseline performing best in both datasets. The limited predictive performance of these competitive baselines shows that this prediction task is non-trivial. TIPAS outperforms all baselines on both Argus (60.9%; 6-69% rel. improvement) and Under Armour datasets (50.9%; 11-156% rel. improvement). The small standard error across multiple dataset splits in the rolling window evaluation (Figure 6) demonstrates that our training procedure is robust and consistently shows good performance. We note that TIPAS performs particularly well on rare actions leading to 9-256% relative improvement in macro-averaged recall over baseline models (Online Appendix [36]).

Results: Comparison of individual model components. Note that TIPAS has three components (Equation 2): time-varying action propensities (Time), short-term interdependencies between actions (Short), and long-term periodic effects (Long). Here, we evaluate the performance of each of these components in an ablation study by comparing Time, Time+Short, and the full TIPAS model combining Time+Short+Long (Figure 7; all models include user personalized preferences α_{ua}). We find that modeling time-varying action propensities achieves an accuracy of 53% and 40% on the two datasets, respectively. Further, modeling short-term dependencies between actions improves this to 59% and 49%, and capturing long-term periodicities of actions further improves this to 61% and 51%, respectively. This demonstrates that capturing all three properties is essential to predicting actions in both datasets of human real-world action sequences. Further, we observe a bigger difference between the full Time+Short+Long model and the Time+Short model in terms of macro-averaged recall (7% and 5% relative MAR improvements compared to 3% and 4% in terms of accuracy on the Argus and Under Armour datasets, respectively). This indicates that modeling long-term periodicities is especially important for more rare actions such as walking and biking. In addition, we find that modeling long-term periodic effects discretized by time of day (0-6h, 6-12h, 12-18h, 18-24h) performs significantly better than not discretizing by time of day on both datasets. For example, actions such as biking and walking are periodic but vary based on time of day (Figure 3). Our full model captures these time-of-day dependent long-term effects and relatively improves macro-averaged recall of predicting biking and walking actions by 491-556% over Time model and 2-4% over Time+Short model.

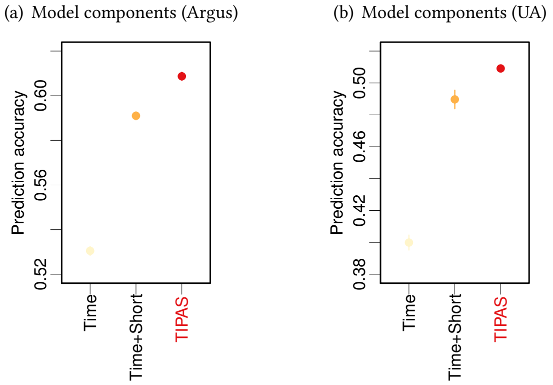


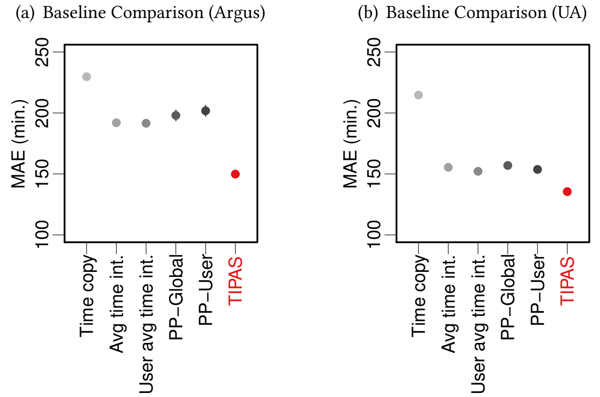
Figure 7: Ablation study comparing different model components on accuracy when predicting actions. Higher is better.

6.5 Predicting the Time of the Next Action

We now focus on the second aspect of modeling real-world actions: Predicting the time of the next action. Specifically, the task is the predict the $n+1$-th timestamp t_{un+1} in history u, given past events $H_u = \{(a_{u1}, t_{u1}), ..., (a_{un}, t_{un})\}$ (we do not assume that the next action a_{un+1} is given). Mean absolute error (MAE) is used as the evaluation metric. We use the same train/test paradigm as before (rolling window evaluation training one month and testing on the next). We restrict predictions to only events that will occur within the next 12 hours (*i.e.*, the time interval $t_{un+1} - t_{un} \leq 12$ hours) because these are the most important and actionable inferences (*e.g.*, predicting a sleep time many days from now may have large error, but it is also less relevant). In order to make time predictions based on TIPAS, we simulate the multivariate temporal point process using Ogata's modified thinning algorithm [43]. We simulate 100 samples and return the average time.

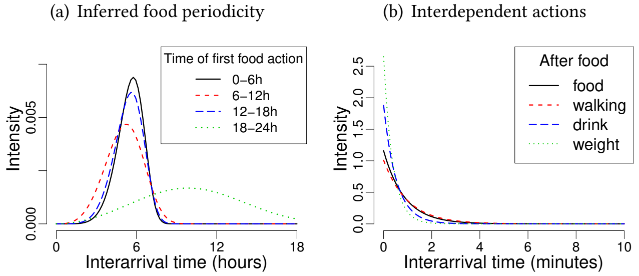
We compare our model to the following five baseline methods:

- **Time Copy Model:** Predicts the next time, t_{un+1}, based on the most recent time-interval of user u ($t_{un+1} = t_{un} + (t_{un} - t_{un-1})$).
- **Average Time Interval:** Predicts the next time t_{un+1} using the global average of time-intervals.
- **User Average Time Interval:** Predicts the next time t_{un+1} using the average of time-intervals for user u.
- **PP-Global:** A global Poisson process model. The intensity func-


(a) Baseline Comparison (Argus)

(b) Baseline Comparison (UA)

Figure 8: Mean absolute error (MAE) when predicting time of next actions. Lower is better. Comparison to baselines.


(a) Inferred food periodicity

(b) Interdependent actions

Figure 9: Visualization of inferred TIPAS model parameters for (a) periodicity of food actions and (b) interdependent actions following food actions. The learned dependencies allow to explain why specific actions are being predicted.

tion is constant over time and defined by $\lambda_u(t, a) = \alpha_a$.

- **PP-User:** A user-specific Poisson process model. The intensity function is constant over time: $\lambda_u(t, a) = \alpha_{ua}$.

We note that the other baselines (Markov models, HMM, FPMC, and RNN) used in Section 6.4 are unable to make any time predictions.

Results. Experimental results are shown in Figure 8. We observe that all baselines perform similarly except the Time Copy model which performs significantly worse on both datasets. TIPAS significantly outperforms all baselines across both datasets by 22-35% in the Argus dataset and 11-37% in the Under Armour dataset (relative improvement). Restricting predictions to events within the next 6 hours (instead of 12h as before), TIPAS outperforms the baselines even more significantly, improving upon them by 44-58% and 37-41% on the two datasets. TIPAS is able to make better timing predictions because it is able to leverage three key components. First, it is aware that certain actions only happen during certain parts of the day. For example, it will predict longer delays in the middle of the night when actions are unlikely to occur. Second, the model can exploit dependencies between actions. For instance, it might predict a very short time after a run because many users will drink water or check their heart rate soon after. Third, TIPAS is able to exploit periodicities in the data. For example, it might predict an evening time commute because it observed a commute in the morning. In summary, modeling these three key aspects of human behavior allows us to make better predictions of actions and their timing.

6.6 Model Explainability

TIPAS also allows for visualization of model parameters, which enables explanations of why certain predictions are made. This is especially important in the mobile health context, where model predictions may impact users' real-world health behaviors and therefore need to be explained and monitored.

The inferred model parameters for Equation 5 are shown in Figure 9a (specifically, $f(\Delta_{t't}) = \gamma_{c_{t'}a}\kappa_{c_{t'}a}\Delta_{t't}^{\kappa_{c_{t'}a}-1}\exp(-\gamma_{c_{t'}a}\Delta_{t't}^{\kappa_{c_{t'}a}})$ for a = food). These distributions correspond to when food events likely trigger other food events. The distributions show that meals are extremely periodic and that meals sharply determine the timing of the next meal, except for dinners after 18:00h, which do not precisely determine the timing of the next meal (18-24h, green). The periodicities vary between 5h after breakfast (6-12h) and 6h after lunch (12-18h). This is consistent with a typical schedule of meals at 7:00h, 12:00h, and 18:00h. Importantly, this enables us to correctly predict that earlier lunches may lead to earlier dinners. Such predictions are critical for correctly timed interventions, for instance making sure that diet reminders do not come to late.

Furthermore, TIPAS allows us to explain why an activity was predicted, based on the relative contributions of model components to the overall intensity function (Section 5.2). For example, after food actions users are likely to log other foods and drinks (Figure 9b; showing $f(\Delta_{t't}) = \omega_{a'a}\exp(-\omega_{a'a}\Delta_{t't})$ for a' = food). This makes sense as typical meals include both food and drinks, and users may choose to log parts of each meal separately. Interestingly, users also often log their weight right after food, indicating that they might be conscious of how their meal might have impacted their weight. Lastly, we observe walking actions right after meals. Users may walk back from a restaurant, or they might attempt to walk off some of their meal's calories.

While these results and examples are specific to mobile activity logging applications, the utility of our model may generalize other domains where behaviors are time-varying, interdependent, or periodic. Distributional choices for the individual may vary across domains but can easily be adapted in our model.

7 CONCLUSION

Accurately predicting the user's future actions is essential for personalization, user modeling, and timely interventions in mobile health applications. In this paper we demonstrated that real-world user behavior exhibits several complexities including a large number of potential actions, time-varying action propensities, dependencies between actions, and periodic behaviors. We proposed a novel statistical model based on multivariate temporal point processes that jointly models all these complexities of human behaviors. Empirically, we demonstrate that our model successfully captures these dynamics in two real-world datasets and that it significantly outperforms nine baselines on tasks of predicting the next user action and when this action will occur. Our model can serve as a foundation to predict more fine-gained attributes of real-world actions such as their duration, intensity, or exact location. Our results further have implications for modeling human behavior, app personalization, and targeting of health interventions.

Acknowledgments. This research has been supported in part by NIH BD2K, DARPA NGS2, ARO MURI, IARPA HFC, Stanford Data Science Initiative, and Chan Zuckerberg Biohub.

REFERENCES

[1] O. Aalen, O. Borgan, and H. Gjessing. *Survival and event history analysis: A process point of view.* 2008.

[2] E. Adar, J. Teevan, and S. T. Dumais. Large scale analysis of web revisitation patterns. In *SIGCHI*, 2008.

[3] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, 2006.

[4] T. Althoff. Population-scale pervasive health. *IEEE Pervasive Computing*, 16(4):75–79, 2017.

[5] T. Althoff, E. Horvitz, R. W. White, and J. Zeitzer. Harnessing the web for population-scale physiological sensing: A case study of sleep and performance. In *WWW*, 2017.

[6] T. Althoff, P. Jindal, and J. Leskovec. Online actions with offline impact: How online social networks influence online and offline user behavior. In *WSDM*, 2017.

[7] T. Althoff, R. Sosic, J. L. Hicks, A. C. King, S. L. Delp, and J. Leskovec. Large-scale physical activity data reveal worldwide activity inequality. *Nature*, 2017.

[8] A. Anderson, R. Kumar, A. Tomkins, and S. Vassilvitskii. The dynamics of repeat consumption. In *WWW*, 2014.

[9] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous computing*, 2003.

[10] R. Baeza-Yates, D. Jiang, F. Silvestri, and B. Harrison. Predicting the next app that you are going to use. In *WSDM*, 2015.

[11] A. R. Benson, R. Kumar, and A. Tomkins. Modeling user consumption sequences. In *WWW*, 2016.

[12] S. Berkovsky, T. Kuflik, and F. Ricci. Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction*, 2008.

[13] F. Bohnert, I. Zukerman, S. Berkovsky, T. Baldwin, and L. Sonenberg. Using interest and transition models to predict visitor locations in museums. *AI Communications*, 2008.

[14] J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec. Anyone can become a troll: Causes of trolling behavior in online discussions. In *CSCW*, 2017.

[15] D. R. Cox and V. Isham. *Point processes*. 1980.

[16] A. Das Sarma, S. Gollapudi, R. Panigrahy, and L. Zhajtabar, Yang. Understanding cyclic trends in social choices. In *WSDM*, 2012.

[17] B. D. Davison and H. Hirsh. Predicting sequences of user actions. In *AAAI/ICML WS on Predicting the Future*, 1998.

[18] A. Drutsa, G. Gusev, and P. Serdyukov. Periodicity in user engagement with a search engine and its application to online controlled experiments. *ACM TWEB*, 2017.

[19] 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 *KDD*, 2016.

[20] N. Du, M. Farajtabar, A. Ahmed, A. J. Smola, and L. Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *KDD*, 2015.

[21] N. Du, Y. Wang, N. He, J. Sun, and L. Song. Time-sensitive recommendation from recurrent user activities. In *NIPS*, 2015.

[22] 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 *NIPS*, 2015.

[23] G. Fischer. User modeling in human–computer interaction. *UMUAI*, 2001.

[24] S. Fox and M. Duggan. *Tracking for health.* Pew Research Center's Internet & American Life Project, 2013.

[25] J. Freyne and S. Berkovsky. Intelligent food planning: personalized recipe recommendation. In *IUI*, 2010.

[26] J. Freyne, J. Yin, E. Brindal, G. A. Hendrie, S. Berkovsky, and M. Noakes. Push notifications in diet apps: Influencing engagement times and tasks. *Int J of Human–Computer Interaction*, 2017.

[27] P. Gorniak and D. Poole. Predicting future user actions by observing unmodified applications. In *AAAI*, 2000.

[28] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 1971.

[29] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *TOIS*, 2004.

[30] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[31] T. Iwata, A. Shah, and Z. Ghahramani. Discovering latent influence in online social activities via shared cascade poisson processes. In *KDD*, 2013.

[32] K. Kapoor, K. Subbian, J. Srivastava, and P. Schrater. Just in time recommendations: Modeling the dynamics of boredom in activity streams. In *WSDM*, 2015.

[33] K. Kapoor, M. Sun, J. Srivastava, and T. Ye. A hazard based approach to user return time prediction. In *KDD*, 2014.

[34] F. Kooti, K. Lerman, L. M. Aiello, M. Grbovic, N. Djuric, and V. Radosavljevic. Portrait of an online shopper: Understanding and predicting consumer behavior. In *WSDM*, 2016.

[35] Y. Koren. Collaborative filtering with temporal dynamics. In *KDD*, 2009.

[36] T. Kurashima, T. Althoff, and J. Leskovec. Modeling Interdependent and Periodic Real-World Action Sequences. 2018. Online Appendix. http://bit.ly/2stjpB4.

[37] T. Lane. Hidden markov models for human/computer interface modeling. In *Proc. IJCAI WS on Learning about Users*, 1999.

[38] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, 2016.

[39] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval.* 2008.

[40] C. Mavroforakis, I. Valera, and M. G. Rodriguez. Modeling the dynamics of online learning activity. In *WWW*, 2017.

[41] M. D. Mulvenna, S. S. Anand, and A. G. Büchner. Personalization on the net using web mining: Introduction. *CACM*, 43(8):122–125, 2000.

[42] I. Nahum-Shani, S. N. Smith, B. J. Spring, L. M. Collins, K. Witkiewitz, A. Tewari, and S. A. Murphy. Just-in-time adaptive interventions (JITAIs) in mobile health: key components and design principles for ongoing health behavior support. *Ann Behav Med*, 2016.

[43] Y. Ogata. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 1981.

[44] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 2010.

[45] A. Shameli, T. Althoff, A. Saberi, and J. Leskovec. How gamification affects physical activity: Large-scale analysis of walking challenges in a mobile application. In *WWW*, 2017.

[46] M. Swan. The quantified self: Fundamental disruption in big data science and biological discovery. *Big Data*, 1(2):85–99, 2013.

[47] Y. Tanaka, T. Kurashima, Y. Fujiwara, T. Iwata, and H. Sawada. Inferring latent triggers of purchases with consideration of social effects and media advertisements. In *WSDM*, 2016.

[48] J. Teevan, E. Adar, R. Jones, and M. Potts. History repeats itself: Repeat queries in Yahoo's logs. In *SIGIR*, 2006.

[49] J. G. Thomas and D. S. Bond. Behavioral response to a just-in-time adaptive intervention (JITAI) to reduce sedentary behavior in obese adults: Implications for JITAI optimization. *Health Psychology*, 2015.

[50] W. Trouleau, A. Ashkan, W. Ding, and B. Eriksson. Just one more: Modeling binge watching behavior. In *KDD*, 2016.

[51] I. Valera and M. Gomez-Rodriguez. Modeling adoption and usage of competing products. In *ICDM*, 2015.

[52] R. Yu, A. Gelfand, S. Rajan, C. Shahabi, and Y. Liu. Geographic segmentation via latent poisson factor model. In *WSDM*, 2016.

[53] K. Zhou, H. Zha, and L. Song. Learning triggering kernels for multi-dimensional hawkes processes. In *ICML*, pages 1301–1309, 2013.

[54] I. Zukerman and D. W. Albrecht. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 2001.

[55] I. Zukerman, D. W. Albrecht, and A. E. Nicholson. Predicting users' requests on the WWW. In *UM*. 1999.