A State Transition Model for Mobile Notifications via Survival Analysis

Yiping Yuan
LinkedIn Corporation
ypyuan@linkedin.com

Jing Zhang
University of Maryland
jzhang86@umd.edu

Shaunak Chatterjee
LinkedIn Corporation
shchatte@linkedin.com

Shipeng Yu
LinkedIn Corporation
siyu@linkedin.com

Romer Rosales
LinkedIn Corporation
rrosales@linkedin.com

ABSTRACT

Mobile notifications have become a major communication channel for social networking services to keep users informed and engaged. As more mobile applications push notifications to users, they constantly face decisions on what to send, when and how. A lack of research and methodology commonly leads to heuristic decision making. Many notifications arrive at an inappropriate moment or introduce too many interruptions, failing to provide value to users and spurring users' complaints. In this paper we explore unique features of interactions between mobile notifications and user engagement. We propose a state transition framework to quantitatively evaluate the effectiveness of notifications. Within this framework, we develop a survival model for badging notifications assuming a log-linear structure and a Weibull distribution. Our results show that this model achieves more flexibility for applications and superior prediction accuracy than a logistic regression model. In particular, we provide an online use case on notification delivery time optimization to show how we make better decisions, drive more user engagement, and provide more value to users.

CCS CONCEPTS

• **Information systems** → Data mining; • **Mathematics of computing** → Survival analysis.

KEYWORDS

Mobile notifications; survival analysis; Weibull distribution; accelerated failure-time model

1 INTRODUCTION

Social networking services (e.g., Facebook, LinkedIn, Instagram, Twitter, WeChat) actively push information to their users through mobile notifications. As the content ecosystem and users' connection networks grow, more and more information is generated on the social networking site that is worth informing the users. On the other hand, users have limited attention span, regardless of how much value notifications could inform them of. The discrepancy between increasing content and limited user attention is the challenge many mobile applications are facing, especially those social networking applications.

A mobile notification is a message displayed to the user either through the mobile application *user interface* (UI) itself, or through the operating system's push notification services, such as *Apple Push Notification Service* (APNs). Instances of such messages include a user-to-user communication, a friend or connection request, an update from a friend or connection (e.g., birthday or job change), an article posted by a connection, etc. These notifications help keep the users informed of what is happening in their network. In addition, notifications also serve the purpose of promotions and product marketing for many mobile applications.

Compared with email communication, mobile notifications are more time sensitive and more promptly responded to [8, 26]. Without an established way to determine delivery time, mobile notifications often arrive at inconvenient moments, failing to provide value to a user. Moreover, due to the pervasive nature of smartphones, such inconvenience may lead to complaints or even disablement on future notification deliveries, causing a permanent loss to both service providers and users. In short, sending notifications at the right time with the right content in many cases is critical.

In this paper, we focus on a quantitative way to measure the effectiveness of a mobile notification and to learn the pattern of how the effectiveness differs from user to user and from time to time. The overall objective is to improve personalization and ensure better delivery time and volume optimization.

The interaction of a user with mobile notifications can be very complex and depends on numerous aspects [22, 23, 33]. It is common to link a notification event to one or more rewards to evaluate the effectiveness of a notification. For engagement, a typical reward is a visit from the user to the app. One challenge for such a study is how to attribute a reward, because users may receive multiple notifications before they open and visit the app. Simple strategies could be to attribute the reward to the most recent one

or to several notifications delivered within a look-back time period. Such strategies are hard to justify theoretically and could introduce significant bias in learning. Our strategy is to leverage the survival analysis to attribute a reward without ambiguity and bias [5, 14].

Survival analysis is commonly used within medical and epidemiological research to analyze data where the outcome variable is the time until the occurrence of an event of interest. For example, if the event of interest is heart attack, then the time to event or survival time can be the time in years until a person develops a heart attack. In survival analysis, subjects are usually followed over a specified time period and the focus is on the time at which the event of interest occurs. Survival time has two components that must be clearly defined: a beginning point and an endpoint that is reached either when the event occurs or when the follow-up time has ended. If the event does not occur by the follow-up time, the observation is called censored. The censored observations are known to have a certain amount of time where the event of interest did not occur and it is not clear whether the event would have occurred if the follow-up time were longer. Such censoring is very common in observational notification data.

We introduce survival analysis to notification modeling as a new domain. The beginning point in this case is the delivery time of a notification and the endpoint is the reward time (e.g., the time of a visit) or a next notification delivery time, whichever happens first. When the next notification occurs first, the observation is censored. In this paper, we apply an accelerated failure-time model [16, 32] with a Weibull distribution to our large-scale user data for the reward prediction. This turns out to be not just novel, but also superior in prediction performance compared to baseline models in our offline analysis.

We provide two example formulations for notification volume optimization (VO) and delivery time optimization (DTO) separately. We then present an online use case on notification DTO, where our model is used to make send decisions. The A/B test results show significant improvement on user engagement and content consumption over a non-DTO control and a baseline DTO model.

The major contributions of this paper can be summarized as follows.

- We develop a state transition model to measure the effectiveness of a notification through a delta effect $\Delta F(W_0, T)$ in Section 3.
- We propose to estimate the delta effect in the presence of censored data, using a log linear survival structure and a Weibull distribution.
- We conduct offline evaluations with real-world notification data to demonstrate the accuracy and flexibility of our engagement prediction.
- We carry out an online use case of determining the delivery time and show superiority of the proposed method with A/B tests in Section 7.

2 RELATED WORK

Email communication as a channel has a long history for social networking services. A volume optimization framework [11, 12] can simultaneously minimize the number of emails sent, control the negative complaints, and maximize user engagement. While we share similar goals for mobile notifications, there are unique mobile aspects to be considered. Moreover, the volume optimization framework focuses on solving a Multi-Objective Optimization (MOO) problem [1, 2], in which multiple objectives are optimized under given constraints. We focus on probabilistic nature of interactions between a user and a notification. Our work can be leveraged as a utility prediction model, which would be one of the utilities of interest in a MOO formulation for mobile notifications.

As more mobile applications push information to users, several studies have been carried out to understand how to make effective use of notifications. Sahami et al. [29] collect close to 200 million notifications from more than 40,000 users including users' subjective perceptions and present the first large-scale analysis of mobile notifications. A number of findings about the nature of notifications, such as shorter responding time, have shed light on how to effectively use them. Pielot et al. [26] carry out an one-week, in-situ study involving 15 mobile phones users and suggested that an increase in the number of notifications is associated with an increase in negative emotions. Both works do not attempt to model the interactions probabilistically.

Xu et al. [33] developed an app usage prediction model that leverages the user's day-to-day activities, app preferences and the surrounding environment. Mehrotra et al. [22] developed a classification model to predict notification acceptance by considering both content and context information. Pielot et al. [27] proposed a machine learning model to predict whether the user will view a message within the next few minutes or not after a notification is delivered. Their study also suggests that indicators of availability, such as the last time the user has been online, not only create social pressure, but are also weak predictors of attentiveness of the message. Pielot et al. [25] carried out a field study with hundreds of mobile phone users and built a machine-learning model to predict whether a user will click on the notification and subsequently engage with the content. The model can be used to determine the opportune moments to send notifications. These studies focus on cross-application study with complete device information, yet the scale of notifications and users are not comparable to our case.

On general user engagement, extensive studies [3, 4, 7, 17, 31] have been promoting relevant and high quality content to users to maximize long-term user engagement with the platform. Other works [9, 34] show that low-quality advertising has detrimental effect on long-term user engagement. Zhou et al. [36] developed an ad quality model based on logistic regression to identify offensive ads that affect user engagement. The focus has been on the quality instead of the timing.

Most applications of the survival analysis in the literature have been in medicine, biology or public health, but there is an increasing interest in its applications to social behavior. Survival techniques based on Weibull distributions have been applied to understanding and predicting dwell time on web services [21, 30]. Yu et al. [35] proposed a temporally heterogeneous survival framework to model social behavior dynamics, whose model parameters can be solved by maximum likelihood estimation. The model can be applied to user-to-user communication. Gomez-Rodriguez et al. [10] studied the formation of an information cascade in a network based on survival theory. Last but not least, Li et al. [20] applied survival analysis in modeling the career paths. They formulated

the problem as a multi-task learning and achieved favorable performance against several other state-of-the-art machine learning methods.

3 STATE TRANSITION MODEL VIA SURVIVAL ANALYSIS

A mobile notification may be delivered through one or many channels such as a sound, a badge count update on the app icon, and an alert shown on the lock screen or as banners. A UI push notification shown in Figure 1 refers to one with an alert showing the content of the message. Such notifications are more effective at drawing a user's attention but they can also be intrusive or even annoying. As suggested in studies [26, 29], the UI push channel is better for time-sensitive and potentially important notifications, e.g., a connection invitation or a user-to-user message. Other less time-sensitive ones, e.g., a connection's birthday or work anniversary, can be served as badging notifications, meaning we only push a badge count update as shown in Figure 2 and a user has to open the app to see the content as an in-app notification within the mobile application's UI in Figure 3. Unlike UI push notifications, such badging notifications are much less intrusive. On the other hand, the effect of them are more subtle. Users are not able to view and interact with the notification content without opening the app. It usually takes longer time for a user to respond to the badging than the UI push and it is harder to separate the effect of notifications from the organic visits. Attribution challenges also arise when multiple badging notifications have been delivered with more than one badge count. This challenge is further elaborated in Section 3.2 as data censoring. The content of badging notifications are usually less time-sensitive and hence we have more flexibility in their delivery time.
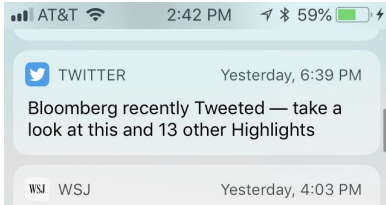

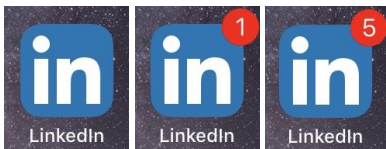**Figure 1: An example of UI push notifications**

Figure 2: Visual appearance of badge counts

In this section, we develop a state transition model to describe and predict the interactions between users and notifications. We focus on badging notifications, which has a more subtle effect to model and is less studied in the literature. The methodology can be extended to UI push notifications with possibly different distribution assumptions as they are responded to more quickly.
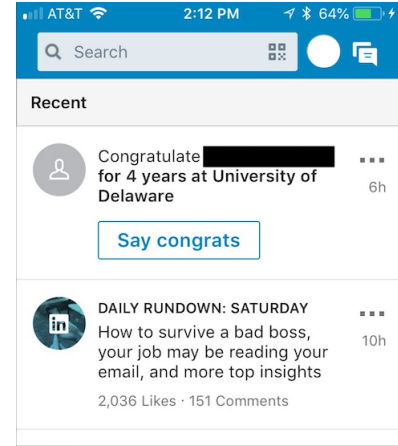

Figure 3: An example of in-app notification

3.1 State Transition Model

We aim to learn how notifications as interventions promote user engagement and bring more value to users. Notifications may change users' mobile context state in various ways. For badging notifications, the most obvious one would be the change of the outer badge count. They may also change the notification inventory within the app.

Let M be a notification event, s be a mobile context state, and t_s be the time to the next visit since the start time of the state s. Figure 4 shows how a state transition model works. After a notification M_0, a mobile context state stays at s_0. Then at any evaluation time, we consider whether or not to send a notification M_1 to a user, who has stayed in state s_0 for W_0 time. The mobile context state will change to s_1 if M_1 is received. Note that a user's visit can also change the state, so s_0 may start from the most recent visit event or the most recent notification event, whichever comes later.
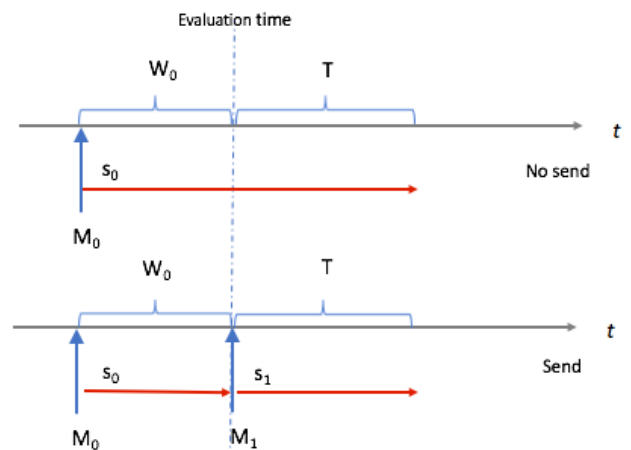

Figure 4: Illustration of state transition

In our state transition model, we assume that users' engagement behaviors depend on both their mobile context states and users'

characteristics. If M_1 is sent, then state s_1 kicks in and the probability of a user visiting our app within the next T time would be

$$P(\text{visit}|\text{send}) = \Pr(t_{s_1} < T \mid z, s_1) = F_{t_{s_1}|z, s_1}(T), \qquad (1)$$

where s_1 represents this user's new mobile context state after the notification is sent, z represent this user's features and $F_{t_{s_1}|z, s_1}$ is the cumulative distribution function of time-to-visit t_{s_1} given (z, s_1). T is the prediction window whose value is usually chosen based on the specific problem instance. For example, we can set it to be 24 hours if we want to focus on daily active users.

If we decide not to send a notification M_1, the user will stay in the current state s_0. Then the probability of the next visit within the next T time is

$$
\begin{aligned}
P(\text{visit}|\text{not send}) &= \Pr(t_{s_0} \le T + W_0 | z, s_0, t_{s_0} > W_0) \\
&= \frac{\Pr(W_0 < t_{s_0} \le T + W_0 | z, s_0)}{\Pr(t_{s_0} > W_0 | z, s_0)} \\
&= \frac{F_{t_{s_0}|z, s_0}(T + W_0) - F_{t_{s_0}|z, s_0}(W_0)}{1 - F_{t_{s_0}|z, s_0}(W_0)}, \qquad (2)
\end{aligned}
$$

which is the probability of time-to-visit from the last state t_{s_0} being less than or equal to $T + W_0$ given that t_{s_0} is already greater than W_0.

We name the difference between (1) and (2) the delta effect, which is a function of T and W_0 given z, s_0 and s_1,

$$
\begin{aligned}
\Delta F(W_0, T) =\;& P(\text{visit}|\text{send}) - P(\text{visit}|\text{not send}) \\
=\;& F_{t_{s_1}|z, s_1}(T) - \frac{F_{t_{s_0}|z, s_0}(T + W_0) - F_{t_{s_0}|z, s_0}(W_0)}{1 - F_{t_{s_0}|z, s_0}(W_0)}. \\
& \hspace{6cm} (3)
\end{aligned}
$$

The delta effect predicts the additional probability of visit in the next T time by sending a notification at the moment. The larger the delta effect is, the more motivation we have to deliver a notification.

In (3), we need to learn the distribution of users' time to visit in each state $F_{t|z, s}(T)$ to predict the delta effect. We explain how we estimate this distribution in Section 3.2.

3.2 Time-to-visit Forecasting

One of the challenges for learning $F_{t|z, s}(T)$ is that we can not always observe the time to visit after a notification send event, because we may send out another notification before the user's next visit. Figure 5 illustrates the mobile activities of a user. After T_1 with notification event M_1, we observe a visit V_1. And T_4 after notification event M_4 we observe a visit V_2. We do not observe a visit after M_2 and M_3 before their next notification events M_3 and M_4, respectively. In the latter cases, the two observations are censored. A censored observation only tells you that the visit event has not happened before the next notification arrives. Such censored observations are very common in notification training data, especially for less active users since the average time-to-visit after a notification delivery is longer.

Therefore, we observe either a visit time T_i, or a censored time U_i. An observation in survival analysis can be conveniently represented by a triplet $(X_i; T_i; \delta_i)$. Here X_i is a feature vector containing
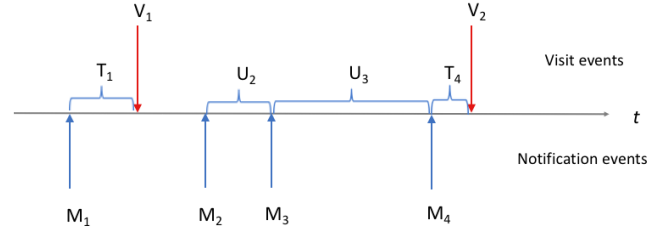

Figure 5: Right-censoring

both users' features z and state features s; δ_i is the censoring indicator, specifically, $\delta_i = 1$ for an uncensored instance and $\delta_i = 0$ for a censored instance; and T_i denotes the observed visit time if $\delta_i = 1$ and a censoring time if $\delta_i = 0$.

While it is possible to avoid or alleviate such censoring by collecting data from a controlled experiment, we argue that it is impractical in many cases. For example, in a controlled study we send a mobile notification to every user in the treatment group at the beginning of the experiment, and then monitor the next visit event without sending more notifications in-between. First, the treated users may get very negative user experience without being promptly notified. Secondly, the experiment may take a long time to observe a visit event for less active users. Lastly, it becomes too costly to repeat the experiment frequently when the model has to be re-trained over time with updated user bases and features.

Survival methods correctly incorporate information from both censored and uncensored observations for estimation, through maximizing the following likelihood function

$$
\begin{aligned}
L &= \prod_{i:\delta_i=1} f(t = T_i \mid X_i) \prod_{i:\delta_i=0} \Pr(t > T_i \mid X_i) \\
&= \prod_{i=1}^{n} \left(f_{t|X_i}(T_i) \right)^{\delta_i} \left(\left(1 - F_{t|X_i}(T_i)\right) \right)^{1-\delta_i}, \qquad (4)
\end{aligned}
$$

where $f_{t|X_i}$ is the probability density function.

A well-known survival model is the Cox proportional hazards model [6, 15]. It is a semi-parametric model built upon the assumption of proportional hazards. In other words, it assumes that the effects of the predictor variables on survival are constant over time and are additive in one scale. This assumption may not be realistic for our application. In addition, a nonparametric baseline hazard function from the Cox model is difficult to interpret and to conduct statistical inference with. A popular alternative survival model is the parametric log-linear model, which is also known as accelerated failure-time (AFT) model [16, 32]. In this model, the effect of changing a covariate is to accelerate or decelerate the time-to-event by some factor. The parametric form makes it much easier to evaluate $F(W_0, T)$ in (3). In addition, a property in Lemma 3.1 works well in practice to space notifications. Therefore, we use the AFT model for our time-to-event forecasting.

The AFT model proposes the following relationship between a random time-to-visit T_i and covariates X_i,

$$\log T_i = bX_i + \sigma\epsilon_i, \qquad (5)$$

where ϵ_i are *independent and identically distributed* (i.i.d.) random errors.

Popular distributions for ϵ_i are logistic, Gaussian and extreme value distributions, leading to log-logistic, log-Gaussian, and Weibull distributions for T_i, respectively. Based on our data analysis and prior knowledge, the time-to-visit for badging notifications given the users' features and state features does not quite depend on how much time has elapsed already, which is the memoryless property. The distribution may be close to an exponential distribution, a special case of the Weibull distribution. Therefore, we assume a Weibull distribution for $t \mid z, s$. The Weibull distribution is a flexible model for time-to-event data [18]. The probability density function and the cumulative distribution function of Weibull distribution are

$$f(T; \lambda, \alpha) = \begin{cases} \alpha \lambda T^{\alpha-1} e^{-\lambda T^{\alpha}} & T \geq 0, \\ 0 & T < 0, \end{cases} \qquad (6)$$

and

$$F(T; \lambda, \alpha) = \Pr(t \leq T) = 1 - e^{-\lambda T^{\alpha}} \quad T \geq 0. \qquad (7)$$

The exponential distribution is a special case when $\alpha = 1$.

Assume ϵ_i in (5) follows an extreme value distribution with

$$f_\epsilon(t) = e^{(t - e^t)}, F_\epsilon(t) = 1 - e^{-e^t}, \qquad (8)$$

then T_i follows Weibull distribution [18] with

$$\lambda_i = e^{-\mu_i/\sigma}; \quad \alpha = 1/\sigma, \qquad (9)$$

where $\mu_i = \boldsymbol{b} \boldsymbol{X}_i$. Note that the model assumes no heteroscedasticity for simplicity, which implies that σ as well as α are constants. It is possible to assume heteroscedasticity and model σ as a function of features $\boldsymbol{X}_i$, adding more personalization in estimating the distribution of time-to-visit T_i for different users at different states. On the other hand, the maximum likelihood estimation is going to be more computationally challenging.

LEMMA 3.1. *If $t_{s_0} \mid z, s_0$ follows a Weibull distribution $f(T; \lambda_0, \alpha_0)$ with $\alpha_0 \in (0, 1)$, then $P(\text{visit}|\text{not send})$ in (2) is decreasing and thus $\Delta F(W_0, T)$ in (3) is increasing in W_0, the time that has elapsed already in state s_0, for any given $T > 0$.*

PROOF. With the Weibull distribution in (7), $\Delta F(W_0, T)$ in (3) becomes

$$\Delta F(W_0, T)$$
$$= F_{t_{s_1} \mid z, s_1}(T) - \frac{F_{t_{s_0} \mid z, s_0}(T + W_0) - F_{t_{s_0} \mid z, s_0}(W_0)}{1 - F_{t_{s_0} \mid z, s_0}(W_0)}$$
$$= 1 - e^{-\lambda_1 (T)^{\alpha_1}} - \frac{\{1 - e^{-\lambda_0 (T+W_0)^{\alpha_0}}\} - \{1 - e^{-\lambda_0 (W_0)^{\alpha_0}}\}}{e^{-\lambda_0 (W_0)^{\alpha_0}}}$$
$$= e^{-\lambda_0 (T+W_0)^{\alpha_0} + \lambda_0 (W_0)^{\alpha_0}} - e^{-\lambda_1 (T)^{\alpha_1}}.$$

Taking derivative with respect to W_0,

$$\frac{\partial \Delta F(W_0, T)}{\partial W_0}$$
$$= e^{-\lambda_0 (T+W_0)^{\alpha_0} + \lambda_0 (W_0)^{\alpha_0}} \{-\lambda_0 \alpha_0 (T + W_0)^{\alpha_0 - 1} + \lambda_0 \alpha_0 (W_0)^{\alpha_0 - 1}\}.$$

Since $(T + W_0)^{\alpha_0 - 1} < (W_0)^{\alpha_0 - 1}$ for $\alpha_0 \in (0, 1)$ and $T > 0$. Then we have $\frac{\partial \Delta F(W_0, T)}{\partial W_0} > 0$ for $\alpha_0 \in (0, 1)$ and $T > 0$. □

Lemma 3.1 shows that if $t_{s_0} \mid z, s_0$ follows a Weibull distribution with $\alpha_0 \in (0, 1)$, the delta effect in (3) can be calculated as

$$\Delta F(W_0, T) = e^{-\lambda_0 (T+W_0)^{\alpha_0} + \lambda_0 (W_0)^{\alpha_0}} - e^{-\lambda_1 T^{\alpha_1}}, \qquad (10)$$

and is increasing in W_0. This suggests we can bring more value to a user by sending a notification when the user has stayed in a state for a longer time. In other words, incorporating the delta effect into decision making reduces the frequency of sending a notification, because short intervals between notifications do not engage the user's attention effectively. The model we learned from data in Section 4 gives $\alpha \in (0, 1)$, which is in line with our conjecture.

Following (4), the likelihood function becomes

$$L = \prod_{i=1}^{n} \left(f_\epsilon \left(\frac{\log T_i - \boldsymbol{b} \boldsymbol{X}_i}{\sigma} \right) \right)^{\delta_i} \left(\left(1 - F_\epsilon \left(\frac{\log T_i - \boldsymbol{b} \boldsymbol{X}_i}{\sigma} \right) \right) \right)^{1 - \delta_i}, \qquad (11)$$

where f_ϵ and F_ϵ are from the extreme value distribution as in (8). Finally the parameters in AFT models $(\boldsymbol{b}, \sigma)$ can be estimated by maximizing the above likelihood function.

3.3 Calculation of the Delta Effect

Once we learn the parameter estimation $(\hat{\boldsymbol{b}}, \hat{\sigma})$ from model training, we can calculate the delta effect for user i at a given time as follows,

- Get all the features $\boldsymbol{X}_{0, i}$ including the state at the moment and the time since last state (i.e., badge count update) $W_{0, i}$ for member i.
- Derive the new features $\boldsymbol{X}_{1, i}$ given that a notification is sent at the moment, which updates the badge count as a state feature and state interaction features.
- According to (9), calculate

$$\hat{\lambda}_{0, i} = e^{-\hat{\boldsymbol{b}} \boldsymbol{X}_{0, i}/\hat{\sigma}}, \quad \hat{\alpha}_0 = 1/\hat{\sigma};$$
$$\hat{\lambda}_{1, i} = e^{-\hat{\boldsymbol{b}} \boldsymbol{X}_{1, i}/\hat{\sigma}}, \quad \hat{\alpha}_1 = 1/\hat{\sigma}.$$

- Apply the above values to (10), and calculate the delta effect

$$\Delta F_i(W_{0, i}, T) = e^{-\hat{\lambda}_{0, i}(T+W_{0, i})^{\hat{\alpha}_0} + \hat{\lambda}_{0, i}(W_{0, i})^{\hat{\alpha}_0}} - e^{-\hat{\lambda}_{1, i} T^{\hat{\alpha}_1}}. \qquad (12)$$

4 DATA COLLECTION AND MODEL TRAINING

Collecting large-scale unbiased training data is challenging, especially in the case of observational data. We collect data at LinkedIn from hundreds of millions of users for a given week including all badging notification events delivered to users and all user visit events. For each notification event, we include 3 broad categories of features in $\boldsymbol{X}_i$.

- user's profile features such as locale and network size.
- State features such as badge count.
- user's activity features such as user's last visit time, the number of site visits over the past week and the number of notifications received over the past week.

In addition, we also include interactions between the above features, such as interaction terms between the badge count and the

profile features so that we can learn different sensitivity to the badge count from different users.

To get the response T_i and censoring indicator δ_i, we sort notification events and visit events in the temporal order for each user so that we can get the next event type and next event time T_i. If the next event is a visit, then $\delta_i = 1$; otherwise $\delta_i = 0$. Note that the next event and next visit may extend beyond the given week, and thus the following week's data may be needed and joined accordingly. We then remove potential outliers by discarding records from users who receive too many notifications or visit too many times. Such records may come from erroneous tracking or abnormal accounts. Next, we split a week's notification data into training and test data with a ratio of 4:1. The test data are held out for evaluation in Section 6.

We train the AFT model with the training data on using Spark MLlib [24] and obtain $\hat{\boldsymbol{b}}$ and $\hat{\sigma}$. Parameters in the conditional Weibull distribution can be calculated as $\hat{\alpha} = 1/\hat{\sigma}$, $\hat{\lambda}_i = e^{-\hat{\boldsymbol{b}}X_i/\hat{\sigma}}$.

The model we learned from training data suggests very different feature importance from that of a notification CTR model. For example, the badge count is a strong predictor and most people are more sensitive to one badge count increase when the badge count is low and become indifferent when the badge count is high. On the other hand, the badge count, the time after the last notification sent are usually not strong signals for a notification CTR model based on our previous experience. The two models can be complementary to each other in a MOO setup described in Section 5.1, since they seem to capture different aspects of notifications.

The $\hat{\sigma}$ we learned is greater than 1, so we have $\hat{\alpha} \in (0, 1)$, suggesting that the $\Delta F_i(W_{0,i}, T)$ in (3) is increasing in W_0 according to Lemma 3.1. This is aligned with our intuition that the longer time spacing we have from the previous notification send time, the more incentive we have to send another notification.

The model we train also suggests that the marginal effect on user engagement diminishes as the badge count increases. The interaction between badge count and user features are significant, meaning different users have different sensitivity to badging.

5 APPLICATIONS AND THRESHOLDS

In this section, we show how our model can be leveraged by different notification decision systems.

5.1 Notification MOO Problems

The model works well with notification MOO problems as a utility function. Consider a typical example where we have notifications available to send to N users and we would like to maximize the total engagement gains while increasing the total notification clicks and controlling the send volume. Let y_i be the decision variable for notification M_i with 1 indicating send and 0 not send. $\Delta F_i(W_{0,i}, T)$ is the predicted session gain, where $W_{0,i}$ is the time since last badge update and T is the prediction time window we are interested in, e.g., the next 24 hours. Assuming we have another model that predicts the probability of a click $P_i(click)$ for a notification available for user i given it is sent, we can formulate a MOO problem,

$$
\begin{aligned}
\text{Maximize} \quad & \sum_{i=1}^{N} \Delta F_i(W_{0,i}, T) y_i \\
\text{subject to} \quad & \sum_{i=1}^{N} P_i(click) y_i \geq C_{click}, \\
& \sum_{i=1}^{N} y_i \leq C_{send}, \\
& 0 \leq y_i \leq 1.
\end{aligned}
\tag{13}
$$

The objective above is to maximize user visits due to notifications, which is quantified by $\Delta F_i(W_{0,i}, T)$ if notification i is sent. The first constraint requires the total number of clicks on notifications to be greater than or equal to C_{click}, thus ensuring that the notifications sent are relevant to users. The second requires the total number of notifications sent to be less than or equal to C_{send}, thus controlling the send volume to avoid notification overload.

By considering the duality of the linear programming problem, the resulting decision rule would be

$$
y_i = 1 \iff \Delta F_i(W_{0,i}, T) + \kappa_1 P_i(click) > \kappa_2,
\tag{14}
$$

where κ_1 and κ_2 correspond to dual variables for the first two constraints. The decision rule is a global threshold of κ_2 across all users on a linear combination of engagement effect $\Delta F_i(W_{0,i}, T)$ and notification quality $P_i(click)$. Similar volume optimization problems can be found in [11, 12] for emails.

5.2 Delivery Time Optimization (DTO)

Mobile notifications are time-sensitive. Sending notifications at a better timing may increase user engagement and improve user experience. The major advantage of our model is to add a utility to evaluate along the time dimension through two channels. The first one is real-time features in the model itself, such as current badge count. The other is the time since last badge update W, which would affect the calculation of $\Delta F_i(W_{0,i}, T)$. Under the Weibull distribution assumption, $\Delta F_i(W_{0,i}, T)$ is increasing in W according to Lemma 3.1, which means we have less incentive to send a notification if we already sent one shortly before and more if we have not sent one in a long time. This makes the model effective in DTO and notification spacing. A straightforward strategy to achieve this is to send a notification to a user i only when $\Delta F_i(W_{0,i}, T)$ is above a certain threshold,

$$
\Delta F_i(W_{0,i}, T) > \kappa.
\tag{15}
$$

In practice, we find that a modification below can improve the performance in some cases when we optimize user engagement,

$$
\frac{\Delta F_i(W_{0,i}, T)}{P_i(\text{visit}|\text{not send})} > \kappa.
\tag{16}
$$

where $P_i(\text{visit}|\text{not send})$ is defined in (2) for user i. The latter decision rule (16) can be viewed as a personalized version of (15), where the personalization is based on a per-member constraint on the number of notification sends.

6 OFFLINE EVALUATION

We compare our proposed survival-based approach with the conventional baseline logistic regression model. While there are potentially more accurate baseline models such as tree models and deep models, survival models can also be extended beyond a linear structure [13, 28]. Such a comparison isolates the impact of data censoring and the survival approach from that of feature engineering. For any given time frame T, we train a logistic regression with the same set of features including their interactions $X_{1,i}$ and a response of whether a user's visit occurs within T after the notification is delivered. One advantage of our formulation over a classification task is that the same model can be used to predict a user's probability of visiting given any time frame T through a Weibull distribution $F(T; \lambda, \alpha)$ in (7). Therefore, the same model can be deployed in different applications, where the prediction windows are chosen differently. On the other hand, we need to train an individual logistic regression model for each different T since the response variables are different.

To evaluate the prediction performance, we calculate the area under the receiver operating characteristic curve (AUC) for selected T values as binary classification problems. For the AFT model, we calculate $F(T; \hat{\lambda}, \hat{\alpha})$ in (7) to be used in the same way as the logistic prediction for the AUC. Figure 6 shows how our model compares with the baseline model in terms of the AUCs as a function of the prediction time window T. The shorter the time window is, the harder the prediction as a binary classification is, since the randomness of the users' engagement behavior tends to be dominating in the short term. At 4 hours prediction window, the model already gives a reasonable AUC of about 0.74 while the logistic regression model only gives 0.58. At 24 hours for daily engagement prediction, the model gives an AUC as high as 0.85 while the baseline model reaches 0.73. Interestingly, as we further increase the prediction window, the AUC of the logistic regression model starts to fall while our model reaches 0.89 at 48 hours. The decline of the logistic regression could come from bias introduced by attributing a visit event to multiple notification events within the time window T. This bias becomes more severe as the time window T increases and likely covers more notifications. The AFT model, on the other hand, avoids such bias by correctly incorporating information from both censored and uncensored observations.

The results show that handling censoring properly is very crucial to mobile notification data. In addition, our model achieved great flexibility in T and superior prediction power compared with the logistic classification models at every given T.

7 ONLINE USE CASE AND EXPERIMENT

In this section, we present a case study deployed at LinkedIn to show how we improve our decision making for mobile notifications with our model.

7.1 Delivery Time Optimization for Less Time-sensitive Notifications

Social network services send both time-sensitive and less-time-sensitive notifications to users. Time-sensitive ones are usually triggered by user-to-user messages or connections' activities such as sharing an article. These notifications need to be sent immediately when
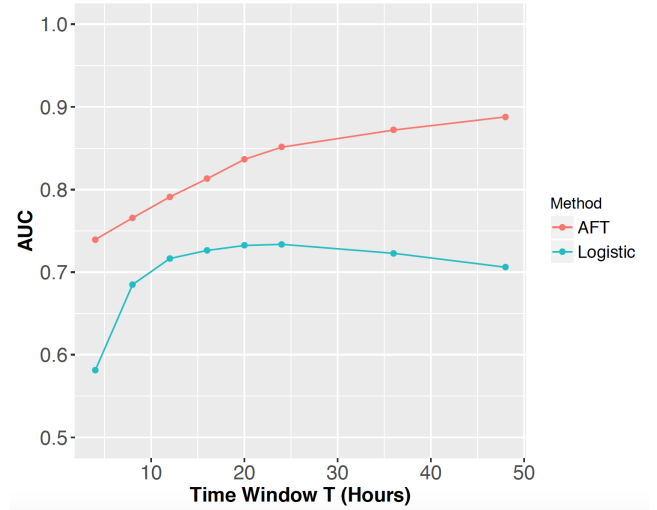

Figure 6: AUC as a function of T

triggered to keep users well-informed. There are also types of notifications which are less time-sensitive. For example, birthday notifications reminding a connection's upcoming birthday can be sent the day of the birthday or several days ahead. In this use case, the less-time-sensitive notifications are first filtered based on a click-through-rate (CTR) prediction model thus dropping notifications with low predicted CTR to ensure high notification quality. The filtered are queued to be then sent within a valid send time window for each individual notification. The send time window ranges from a few hours to a few days depending on the nature of the notification. In this application, "when to send" is decoupled from "whether to send" since the latter decision is already made at the filtering stage. This makes a good use case of delivery time optimization described in Section 5.2.

In this application, we apply the decision rule in (16), where T is set to be 4 hours and κ is chosen from offline analysis and online tuning to optimize the performance. For comparison, we set up a control treatment, in which notifications are sent immediately when available, and a baseline treatment, in which we send a notification to a user only if their badge counts are less than or equal to 1. For users who have notifications in the queue, we evaluate send decisions every 4 hours.

7.2 Online Experiment and Results

Table 1 shows the A/B test experiment results comparing the DTO based on our model with the control and baseline models described above. We are mostly interested in user engagement and notification interactions, which can be characterized by the following metrics.

- *Sessions*: A session is a collection of full page views made by a single user on the same device type. Two sessions are separated by 15 minutes of zero activity.
- *Engaged Feed Sessions*: This metric counts the number of sessions where the user engaged with the newsfeed (either by
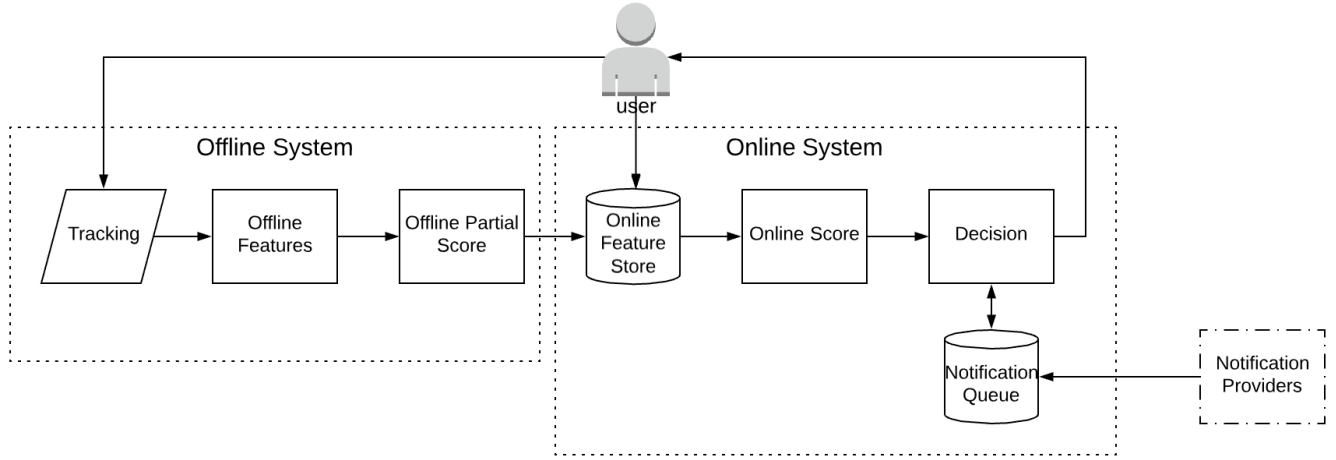
Figure 7: System architecture

interacting with feed updates, or by viewing at least 10 feed updates).

- *Notification Sessions*: This metric counts the number of sessions in which the user viewed or interacted with the notification page.
- *Notification Daily Unique Send CTR*: This metric measures the average click-through-rate of notifications sent to a user in a day.

The experiment was tested over a full week and the numbers in the table are all statistically significant. Compared with the control, which is basically no DTO, our model increased the total sessions by 1.86%, notification sessions by 6.19% and engaged feed sessions by 1.78%. The higher boost in notification sessions was expected since we are optimizing notification send time. The roughly proportional gain in engaged feed sessions suggests that the additional sessions are of similar quality to existing ones. In addition, the notification daily unique send CTR was increased by 2.51% against control, suggesting notifications were delivered at better timing resulting in increased user engagement. Compared with the baseline model, the proposed model showed healthy gains in all four metrics. One interesting observation is that the increase in notification daily unique send CTR (+4.48%) is higher than the comparison with the control (+2.51%). This suggests that although the badge count baseline model increases user engagement, it reduces the CTR compared with the control, implying that it may not be a desirable user experience.

7.3 System Architecture

We outline a design of a notification decision system using the state transition model in Figure 7. Since the model takes a few real-time features (e.g., current badge count, time since last badge count update) as important signals, having an online scoring system is ideal for model performance. To avoid maintaining all features in an online database, we include an offline component for more static

Table 1: Online A/B results for delivery time optimization

Metric	vs. Control	vs. Baseline
Sessions	+ 1.86%	+0.67%
Engaged Feed Sessions	+ 1.78%	+0.69%
Notification Sessions	+6.19%	+1.51%
Notification Daily Unique Send CTR	+2.51%	+4.48%

features, such as user profile features. In this offline component, offline features are retrieved from tracking data on our HDFS system and a partial score is calculated based on the trained model coefficients. We push the partial scores to an online feature store daily through Apache Kafka [19]. The online component maintains realtime features and make realtime decisions based on the real-time $\Delta F_i(W_0, i, T)$ score.

8 DISCUSSION

To our best knowledge, this is the first work on probabilistic modeling of interactions between mobile notifications and user engagement at scale. We develop a state transition model and derive a delta effect to measure the effectiveness of a notification. With a common existence of censoring in observational mobile notification data, we estimate the delta effect through an AFT regression with a Weibull distribution. The prediction from this survival regression is both flexible to apply and superior in prediction accuracy compared to baseline models with the same feature set.

Our state transition model is generalizable and can have broader applications. While we focus on modeling the badging notifications, our model is applicable for all types of mobile notifications. For example, UI push notifications can be modeled with a distribution possibly different from a Weibull distribution.

We consider a user's visit as a reward to a mobile notification. However, the reward can be generalized to a user's purchase event

for on-line shopping apps such as Amazon or a user's content creation event for question-and-answer apps such as Quora. In the cases where data censoring is a major concern for modeling mobile notifications, we provide a general framework to evaluate the effectiveness of a notification towards driving a reward.

ACKNOWLEDGEMENT

REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. 2011. Click shaping to optimize multiple objectives. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 132–140.

[2] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. 2012. Personalized click shaping through lagrangian duality for online recommendation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 485–494.

[3] Christy Ashley and Tracy Tuten. 2015. Creative strategies in social media marketing: An exploratory study of branded social content and consumer engagement. *Psychology & Marketing* 32, 1 (2015), 15–27.

[4] Josh Attenberg, Sandeep Pandey, and Torsten Suel. 2009. Modeling and predicting user behavior in sponsored search. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1067–1076.

[5] Jonathan Buckley and Ian James. 1979. Linear regression with censored data. *Biometrika* 66, 3 (1979), 429–436.

[6] David R Cox. 1992. Regression models and life-tables. In *Breakthroughs in statistics*. Springer, 527–541.

[7] Kushal Dave, Vasudeva Varma, et al. 2014. Computational advertising: Techniques for targeting relevant ads. *Foundations and Trends® in Information Retrieval* 8, 4–5 (2014), 263–418.

[8] Joel E Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. In *Proceedings of the 13th international conference on human computer interaction with mobile devices and services*. ACM, 181–190.

[9] Daniel G Goldstein, R Preston McAfee, and Siddharth Suri. 2013. The cost of annoying ads. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 459–470.

[10] Manuel Gomez-Rodriguez, Jure Leskovec, and Bernhard Schölkopf. 2013. Modeling information propagation with survival theory. In *International Conference on Machine Learning*. 666–674.

[11] Rupesh Gupta, Guanfeng Liang, and Romer Rosales. 2017. Optimizing Email Volume For Sitewide Engagement. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1947–1955.

[12] Rupesh Gupta, Guanfeng Liang, Hsiao-Ping Tseng, Ravi Kiran Holur Vijay, Xiaoyu Chen, and Romer Rosales. 2016. Email volume optimization at LinkedIn. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 97–106.

[13] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, Michael S Lauer, et al. 2008. Random survival forests. *The annals of applied statistics* 2, 3 (2008), 841–860.

[14] Ian R James and PJ Smith. 1984. Consistency results for linear regression with censored data. *The Annals of Statistics* (1984), 590–600.

[15] Komal Kapoor, Mingxuan Sun, Jaideep Srivastava, and Tao Ye. 2014. A hazard based approach to user return time prediction. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1719–1728.

[16] Niels Keiding, Per Kragh Andersen, and John P Klein. 1997. The role of frailty models and accelerated failure time models in describing heterogeneity due to omitted covariates. *Statistics in medicine* 16, 2 (1997), 215–224.

[17] M Laeeq Khan. 2017. Social media engagement: What motivates user participation and consumption on YouTube? *Computers in Human Behavior* 66 (2017), 236–247.

[18] John P Klein and Melvin L Moeschberger. 2005. *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media.

[19] Jay Kreps, Neha Narkhede, Jun Rao, et al. 2011. Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*. 1–7.

[20] Huayu Li, Yong Ge, Hengshu Zhu, Hui Xiong, and Hongke Zhao. 2017. Prospecting the career development of talents: A survival analysis perspective. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 917–925.

[21] Chao Liu, Ryen W White, and Susan Dumais. 2010. Understanding web browsing behaviors through Weibull analysis of dwell time. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 379–386.

[22] Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing content-driven intelligent notification mechanisms for mobile applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 813–824.

[23] Abhinav Mehrotra, Veljko Pejovic, and Mirco Musolesi. 2014. SenSocial: a middleware for integrating online social networks and mobile sensing data streams. In *Proceedings of the 15th International Middleware Conference*. ACM, 205–216.

[24] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, and Ameet Talwalkar. 2016. MLlib: Machine Learning in Apache Spark. *J. Mach. Learn. Res.* 17, 1 (Jan. 2016), 1235–1241. http://dl.acm.org/citation.cfm?id=2946645.2946679

[25] Martin Pielot, Bruno Cardoso, Kleomenis Katevas, Joan Serrà, Aleksandar Matic, and Nuria Oliver. 2017. Beyond interruptibility: Predicting opportune moments to engage mobile phone users. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 91.

[26] Martin Pielot, Karen Church, and Rodrigo De Oliveira. 2014. An in-situ study of mobile phone notifications. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. ACM, 233–242.

[27] Martin Pielot, Rodrigo de Oliveira, Haewoon Kwak, and Nuria Oliver. 2014. Didn't you see my message?: predicting attentiveness to mobile instant messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3319–3328.

[28] Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. 2016. Deep Survival Analysis. In *Proceedings of the 1st Machine Learning for Healthcare Conference (Proceedings of Machine Learning Research)*, Finale Doshi-Velez, Jim Fackler, David Kale, Byron Wallace, and Jenna Wiens (Eds.), Vol. 56. PMLR, Children's Hospital LA, Los Angeles, CA, USA, 101–114.

[29] Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. 2014. Large-scale assessment of mobile notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3055–3064.

[30] Theodore Vasiloudis, Hossein Vahabi, Ross Kravitz, and Valery Rashkov. 2017. Predicting Session Length in Media Streaming. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 977–980. https://doi.org/10.1145/3077136.3080695

[31] Taifeng Wang, Jiang Bian, Shusen Liu, Yuyu Zhang, and Tie-Yan Liu. 2013. Psychological advertising: exploring user psychology for click prediction in sponsored search. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 563–571.

[32] Lee-Jen Wei. 1992. The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis. *Statistics in medicine* 11, 14-15 (1992), 1871–1879.

[33] Ye Xu, Mu Lin, Hong Lu, Giuseppe Cardone, Nicholas Lane, Zhenyu Chen, Andrew Campbell, and Tanzeem Choudhury. 2013. Preference, context and communities: a multi-faceted approach to predicting smartphone app usage patterns. In *Proceedings of the 2013 International Symposium on Wearable Computers*. ACM, 69–76.

[34] Chan Yun Yoo and Kihan Kim. 2005. Processing of animation in online banner advertising: The roles of cognitive and emotional responses. *Journal of Interactive Marketing* 19, 4 (2005), 18–34.

[35] Linyun Yu, Peng Cui, Chaoming Song, Tianyang Zhang, and Shiqiang Yang. 2017. A Temporally Heterogeneous Survival Framework with Application to Social Behavior Dynamics. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1295–1304.

[36] Ke Zhou, Miriam Redi, Andrew Haines, and Mounia Lalmas. 2016. Predicting pre-click quality for native advertisements. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 299–310.