Journal of Data Science 22 (1), 56–76
January 2024

DOI: 10.6339/23-JDS1101
Statistical Data Science

A Time To Event Framework For Multi-touch Attribution

Dinah Shender[1], Ali Nasiri Amini[1], Xinlong Bao[1], Mert Dikmen[1], Jing Wang[1,*], and Amy Richardson Fricke[†]
[1] *Google, Mountain View, CA, USA*

Abstract

Multi-touch attribution (MTA) estimates the relative contributions of the multiple ads a user may see prior to any observed conversions. Increasingly, advertisers also want to base budget and bidding decisions on these attributions, spending more on ads that drive more conversions. We describe two requirements for an MTA system to be suitable for this application: First, it must be able to handle continuously updated and incomplete data. Second, it must be sufficiently flexible to capture that an ad's effect will change over time. We describe an MTA system, consisting of a model for user conversion behavior and a credit assignment algorithm, that satisfies these requirements. Our model for user conversion behavior treats conversions as occurrences in an inhomogeneous Poisson process, while our attribution algorithm is based on iteratively removing the last ad in the path.

Keywords *data driven attribution; poisson process*

1 Introduction

One of the promises of online advertising has been the ability to tie together ad clicks with actual outcomes (e.g., purchases, website visits, etc), also known as conversions. This gives advertisers the opportunity to understand both the overall effectiveness of their ads and the relative effectiveness of different types of ads. With the right data, multi-touch attribution (MTA) aids this second goal by estimating the relative contributions of the various ads a user may see prior to any observed conversions. This can then be incorporated into the advertiser's budget and bidding decisions, increasing spend on ads that drive more conversions.

We propose two key requirements, both related to the effects of time, that this type of attribution system should satisfy. The first is based on the need for an MTA system to be compatible with bidding models, which themselves require real-time data to respond to changes in ad effectiveness, business fluctuations, etc. This requires the MTA system to also ingest real-time data, which is by its nature incomplete, or more precisely, right-censored. We don't know how many users who saw ads yesterday will ultimately convert, only how many have converted so far. Therefore our first requirement is that our MTA system be capable of handling censored data.

A second requirement is that an MTA system be sufficiently flexible to capture that an ad's immediate effect on a user is likely to be different than its effect several weeks later. In practice, this tends to mean that conversions occurring soon after an ad are more likely to have been

[*] Corresponding author. Email: jiwang@google.com.
[†] Author was at Google when work was done.

Received October 3, 2022; Accepted April 22, 2023

caused by the ad (and therefore the ad tends to receive more credit) than conversions occuring long after the ad.

To handle both requirements, we propose an MTA system with two parts: a model for users' conversion behavior that treats conversions as occurrences in an inhomogenous Poisson process, and an attribution credit assignment algorithm that assigns credit according to how ads change the estimated intensity of this inhomogenous Poisson process at conversion time. In the case where a user can convert at most once, our model reduces to the classic Cox proportional hazards model with time-varying covariates (Andersen et al., 2012). More generally, these types of survival models, including variations allowing for multiple occurrence, are widely used, particularly in biostatistics, e.g., Cook and Lawless (2007), where they are also known as time-to-event models, inspiring the name TEDDA (Time to Event Data Driven Attribution) for our modelling approach. While we do not take a time series approach to modeling a user's propensity to convert, our approach to modeling the *change* in a user's propensity to convert bears some resemblance to interrupted time series methods. See Dugan (2011) for a discussion and comparison of the two methods in the context of criminology research.

The traditional approaches to understanding the relative effectiveness of different types of ads have focused on Media Mix Modeling (MMM), which typically fits time series models to a few years of aggregated conversion data in order to compare broad classes of ads, e.g., De Haan et al. (2016); Kireyev et al. (2016). While this can help in allocating overall marketing spend between channels, it cannot provide the kind of granular cross- and within-channel insight into specific types of ads that MTA obtains by leveraging individual user paths.

More recent lines of work focusing on modelling individual user paths can be divided into models that rely on the sequence of user events and those that incorporate the timestamps of these events. The former category includes Markov chain methods such as Li and Kannan (2014) and Anderl et al. (2014), which treat both conversions and each ad channel as states in a kth-order Markov chain and estimate the probability of a user moving from one state to the next. Attribution credit in these models is based on the change in conversion probabilities when a channel is removed from the chain. Dalessandro et al. (2012) and Shao and Li (2011) are similar in depending only on the sequence of ad events, but they each fit logistic regression models for a binary conversion outcome. They then use Shapley values to distribute attribution credit, using the probability of a conversion as the value function in the Shapley algorithm.

There are also several previous papers falling into the latter category, i.e. models that incorporate event timestamps to fit continuous time models for user conversion activity. A recent line of work, including Du et al. (2019), Kumar et al. (2020), and Yao et al. (2021) use recurrent neural networks combined with various attention mechanisms to estimate conversion probabilities, with various architectures to correct for possible confounding. Most then use the Shapley value algorithm to distribute attribution credit. Geng et al. (2021) discusses how to incorporate the attributions generated by Du et al. (2019) into a bidding system.

There is also a line of work using more traditional statistical methods, while still incorporating event time stamps. Lewis and Wong (2018) uses exponentially decaying ad effects to build a model to estimate the incremental effect of ads using experimental data. They use an attribution methodology similar to the one we will describe to then bid optimally based on that model. Finally, Zhang et al. (2014) and Xu et al. (2014) take a similar approach to our model by treating conversions as occurrences in a Poisson process. Both assume an exponentially decaying ad effect. Zhang et al. (2014) considers a more restricted setting where users can convert at most once, similar to traditional survival analysis, and then fits an additive model that assumes there are no interactions between ads. Xu et al. (2014) models both conversion events and ad events in different channels as a set of mutually exciting Poisson processes, allowing ad events to affect not just the conversion probability, but also the probability of future ads. However, they consider only the aggregate conversion credit over the data set, not the credit per ad.

Our model can be used with either observational data or data from randomized experiments set up to measure the causal (or incremental) effects of ads. Both types of data can be useful, depending on the goals and context. Observational data can be interpreted in terms of the correlation between showing ads and conversions, while data from randomized experiments allows for a causal estimate of the number of conversions caused by ads. The latter is the gold standard, but these experiments are often difficult to run, and may not be available on an ongoing basis. In these cases, advertisers may prefer to do MTA on correlational data over not having any sort of MTA, particularly if they have evidence, perhaps from previous randomized experiments, that the relative credit assigned to the ad types of interest is unchanged after accounting for incrementality, even if the absolute credit differs. These are choices that the modeller must make based on their knowledge of the application area, media types, and brands involved. Rather than discussing the pros and cons of observational and experimental data, this paper will focus on the overall system that can be used to model either type of data.

The remainder of this paper is organized as follows: Section 2 goes into further detail about the key issues an MTA system must solve. In Section 3 we present our system, describing both the model for conversion occurrences and the attribution credit assignment methodology. Section 4 discusses how to evaluate the quality and accuracy of the system. Section 5 presents a simulation study of our approach (with additional simulations in the supplementary material). Section 6 discusses possible future improvements to this system.

2 Attribution Requirements

We propose two key requirements we believe an MTA system should satisfy, both related to how the system should consider the effects of time.

2.1 Requirement #1

MTA provides critical insights about the relative value of ads. In the context of digital advertising, where advertisers compete in an auction to determine whose ad is shown, the fastest way to incorporate these insights is to allow them to affect the advertiser's bid. This requires an MTA system with nearly real-time (e.g., daily) updates, in order to allow the advertiser to bid in accordance with the attribution results without sacrificing their responsiveness to business fluctuations, changes in their ads' effectiveness, or other novel trends.

Real-time user path data is fundamentally incomplete: If an ad was shown this morning at 10am, and it's now 6pm and there was no conversion, that does not mean that this ad resulted in 0 conversions. Rather, it resulted in 0 conversions over the first 8 hours. We do not know what will happen over the next day or week. Instead of treating this observation as a 0 or negative response, we should treat this observation as being right-censored. Thus our first requirement is that our system be able to handle incomplete or censored data.

To illustrate the importance of this principle, consider the toy example in Figure 1. If our attribution system treats the data as being complete, so that the conversion outcome is binary for each user, with no censoring, and we run it now, it will likely find that having ad type 2 is not associated additional conversions: paths with and without ad type 2 convert at equal rates. However, in four hours, it would find that ad type 2 is associated with 50% more conversions.

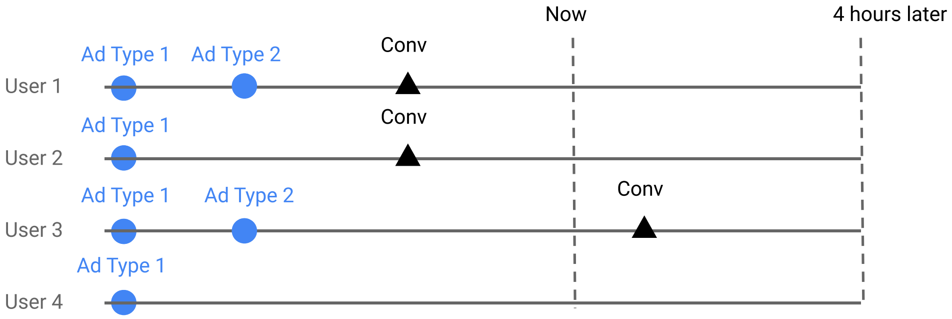

Figure 1: An example of the effects of incomplete or censored data.

Such strongly conflicting results can cause unstable and problematic behavior when used in bidding algorithms. To avoid this, our system needs to recognize that after 4 hours, we not only have new values for the response, but we also have more overall information about the rate of conversions because we have observed the user's post-ad paths for longer. In other words, we need to recognize that we have incomplete observations due to censoring.

This leads us to consider modeling the number of conversions over time, which would naturally capture that our estimates are more uncertain "now" than in 4 hours. The next requirement leads us further in this direction.

2.2 Requirement #2

An ad's contribution to any potential conversion should be allowed to differ depending on the time between the ad exposure and the conversion, typically decreasing as the time between the ad exposure and the conversion increases. Similarly, ad credit shouldn't just depend on the order of the ads in the path.

Many advertisers already recognize this in an informal way, setting "lookback windows" of $N = 7$ or 30 days and only distributing conversion credit to ads in the N days before the conversion. However, an ad's influence likely decays more continuously, so that even within the lookback window, ads further in time from the conversion deserve less credit than those close to it, if all else is equal. While this implies that the order in which ads occur should affect the relative conversion credit that ads receive, the order is not sufficient: if two identical ads occur within hours of each other, we'd expect them to receive similar levels of credit for any subsequent conversions, whereas if they are separated by 30 days, we'd expect the later ad to receive more credit. How fast or steeply this decay occurs will depend on the individual advertiser and the type of ad, and should be learned from the data, but our model should be flexible enough to treat ads differently based on not just their order, but when they occurred.

We illustrate this with another toy example in Figure 2. To distinguish this issue from Requirement #1, let's suppose that the data is complete in this case, e.g., this data is from long enough ago that no further conversions are possible/expected. If we look only at the order in which ads occur, we would learn that ad type 2 is associated with having 50% more conversions. But if we look at just users 1 and 2 and consider the timestamps, we will instead learn that the effect of ad type 2 decays fast enough that it deserves little credit for conversions far from it. However, looking at users 3 and 4, we will see that ad type 2 is actually associated with all the conversions that occur soon after it is shown. This is a simplified example, but it nevertheless illustrates that a model that looks only at the order of the ads and the binary conversion label

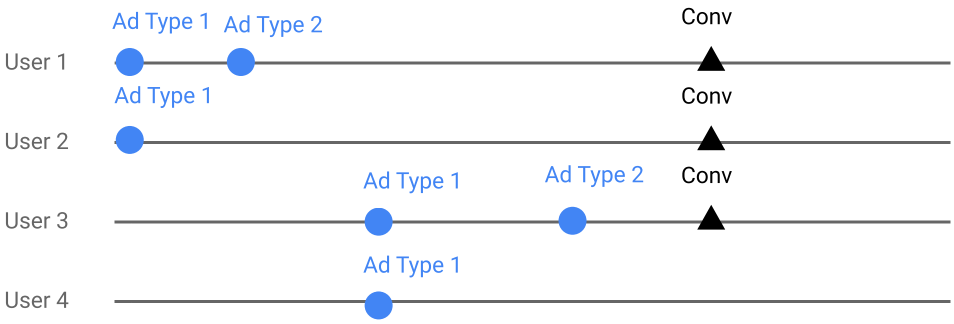

Figure 2: An example of the importance of considering event timestamps.

can reach a very different conclusion compared to a model that takes into account the times at which ads and conversions occur.

Together these two requirements imply that we should model not just the sequence of ad events and binary (or even integer) conversion outcomes for each path, but rather the times when conversions occur given when ads occur. A natural option for this type of modeling is survival analysis. In this framework, conversions are viewed as occurrences of a Poisson process. The conversion intensity function, also known as the instantaneous occurrence rate or the hazard rate or just the intensity function, is allowed to vary with time, and in our case depends on the times at which a user has previously seen ads. Attribution credit is then based on the effect of an ad on a user's instantaneous conversion rate at the time of the conversion.

In the next section we detail our model for conversions and how we use it for attribution.

3 Proposed Model

Our attribution system has two parts: a model for users' conversion behavior, and an attribution credit assignment algorithm that assigns credit in accordance with this model. We will focus the bulk of our exposition on the former. Given a model for user conversion behavior, there are many reasonable credit assignment algorithms. While we will discuss options and the algorithm that we use, the best choice depends on the goals of the system.

3.1 Modeling User Conversion Behavior

As discussed in the previous section, in order to capture how ad effects vary over time, as well as handle incomplete data, we will model conversions as a realization of a Poisson counting process with a time-varying intensity function, $\lambda(t)$. In particular, if we define $Y_i(t)$ as the number of occurrences (conversions) for user i up until time t, then

$$Y_i(t) - Y_i(s) \sim \text{Poisson}\left(\int_s^t \lambda(t)\mathrm{d}t\right)$$

for any $0 \leqslant s \leqslant t$. We will use a log-linear model for the intensity, and allow it to depend on user features (e.g., the country the user is located in), the time since previous ads were seen, as well as other ad features (e.g., format of the ad). We will start with an overly simplified model for $\lambda(t)$ and gradually add these complexities. This model formulation treats data from randomized experiments in the same way as observational data, but with an additional feature for treatment

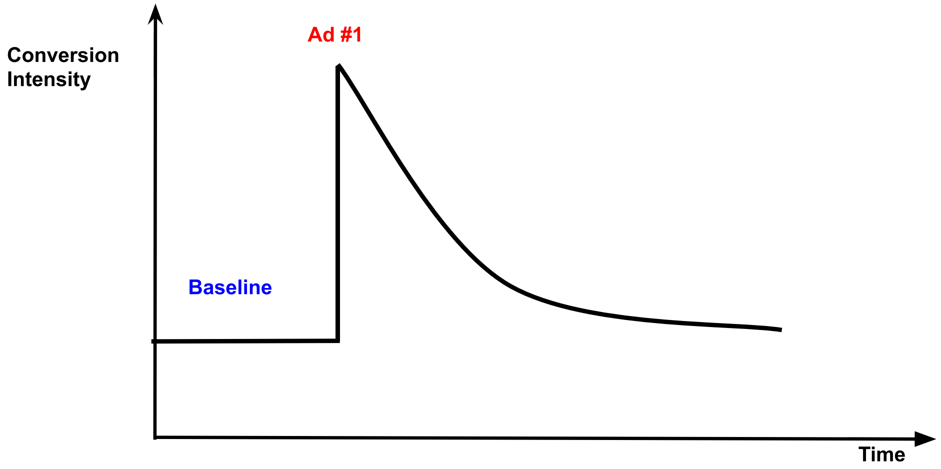

Figure 3: Conversion intensity over time for a user with a single ad event.

group assignment. We will discuss specifics in Section 3.1.5. We will end the conversion modeling section with a brief discussion on options for estimating this model.

3.1.1 No User Features, no Ad Features, Single Ad per User

To start with, consider a model that has no user features, no ad features (other than time), and where we assume that each user sees at most one ad. This is an overly simplified view of the world, particularly the assumption of just a single ad per user, but it is useful for exposition. Suppose that the single ad occurs at t_1; in principle this should also be indexed by user, but we will drop the user subscripts for brevity. In this set-up, our model becomes

$$\log(\lambda(t)) = \alpha_0 + f(t - t_1).$$

α_0 represents the log of the conversion rate before ads are shown, while $f(t - t_1)$ represents the effect of an ad on the user's conversion rate. A user's conversion intensity over time might then look as in Figure 3. The baseline rate before the ad is $\exp(\alpha_0)$, while after the ad the intensity is $\exp(\alpha_0) \times \exp(f(t - t_1))$. Since this model is log-linear, ads have a multiplicative effect on users' conversion rates.

In general, f is a function of time that we would like to estimate. We restrict ourselves to functions such that $f(x) = 0$ whenever $x \leqslant 0$, i.e. there is no ad effect before the ad is actually seen. There are many ways to parameterize f in order to do estimation; we will review a few concrete options, but many other parameterizations are also possible.

One option is to treat f as a continuous function. Since we often expect the ad effect to decay rapidly, modeling f as a mixture of exponentials is a natural choice. We can consider a linear combination of exponential decay functions $\{\exp(-\theta_l t)\}_{l=1,\ldots,L}$, where the choices of θ_l determine the span of the linear combinations and the speed of the decay. Then we can parameterize the intensity as

$$\log(\lambda(t)) = \alpha_0 + \sum_{l=1}^{L} \beta_l \exp(-\theta_l (t - t_1)),$$

where β_l are parameters to be estimated. The above equation is consistent with a proportional hazards model and essentially implies a doubly exponential decay for $\lambda(t)$. Lewis and Wong

(2018) and Zhang et al. (2014) proceed in a similar fashion, but set the right-hand side of the above equation equal to $\lambda(t)$ rather than $\log(\lambda(t))$, essentially assuming an additive hazards model.

Another option for estimating f as a continuous function is to use splines. Letting $b_1, \ldots, b_L$ be the functions in the spline basis, the model becomes

$$\log(\lambda(t)) = \alpha_0 + \sum_{l=1}^{L} \beta_l b_l(t - t_1).$$

We can then optimize for α_0 and $\beta_1, \ldots, \beta_L$, possibly with a regularization constraint or prior on the β_l values. Splines are commonly used for approximating continuous functions. However, if the true intensity does decay very rapidly, then for a fixed basis size, an exponential function "basis" might perform better than a spline basis. We do not make a specific recommendation here; rather, this is an area for future study.

A third option might be to approximate f as a step function, estimating a separate ad effect for each step. For example, we might choose to estimate the jump in conversions in the first 24 hours, the next 24 hours, and the subsequent 28 days. Then if t is measured in hours, and using I to denote indicator functions, the log-intensity becomes:

$$\log(\lambda(t)) = \alpha_0 + \beta_1 I\{t - t_1 \leqslant 24\} + \beta_2 I\{24 < t - t_1 \leqslant 48\} + \beta_3 I\{48 < t - t_1 \leqslant 24 \times 30\}. \quad (1)$$

Again, we can now optimize for α_0, β_1, β_2, and β_3, perhaps placing a prior on the coefficients or applying other types of regularization. As we add additional features to the model or if we wish to pool data across advertisers, treating these coefficients as random effects may also be attractive.

3.1.2 Ad Features

Staying for now in the single ad setting, we can also consider how to add ad features, such as the format of the ad, whether it was shown on a mobile device, and so on, to the model. For example, perhaps certain ad formats or campaigns are associated with a larger change in conversions than others. To capture this, we want to allow the ad effect to vary depending on these features. One way to do this with K total features is to write:

$$\log(\lambda(t)) = \alpha_0 + f(t - t_1) + \sum_{k=1}^{K} g_k(t - t_1, x_{1k}).$$

Here k indexes the features in our model and x_{1k} is the value of the kth feature for the first ad. Like f, g_k is a function of time to be estimated. One option is to constrain g_k to be constant over time, so that changing the feature value simply shifts the intercept for f. Conceptually, this corresponds to a change in an ad's initial effect, but not its decay rate. In the example above where f is a linear combination of spline basis functions, we could take g_k to be a linear combination of lower-order spline basis functions. This would change both the ad's initial effect and the effect's decay rate. While we would generally expect to choose g_k to be simpler than f, even this is not required.

As written, if for each level of each feature k we allow a non-zero value for g_k, the model is overparameterized. This can be handled in the usual ways, e.g., setting $g_k = 0$ for some reference level and interpreting f as the ad effect for when all ad features are at their reference level, or

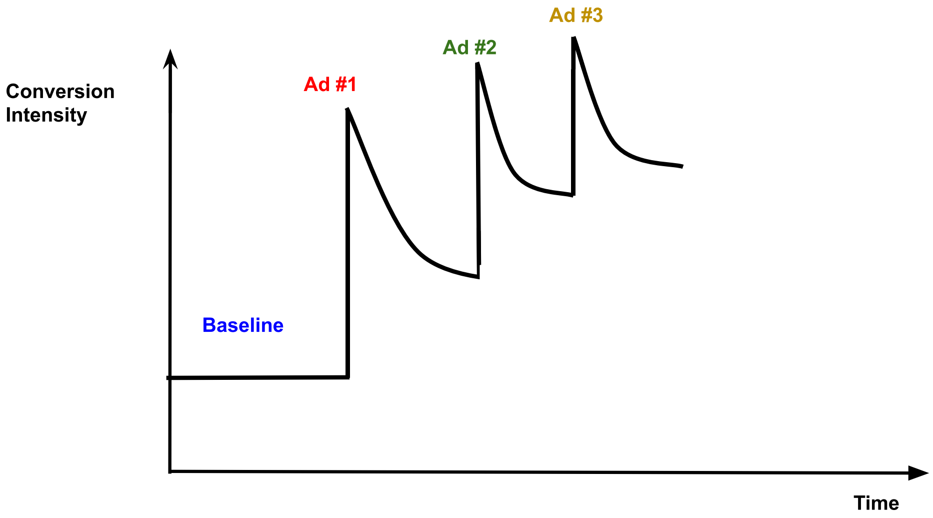

Figure 4: Conversion intensity over time for a user with multiple ad events.

by adding a constraint so that the g_k average to 0 for each k and interpreting f as the average ad effect. Alternatively, if we are applying regularization when we fit the model, then that may be enough for all the parameters to be well-defined. Our model is flexible with respect to these choices and so we leave the notation general.

3.1.3 Multiple Ads

The motivation for MTA is the case where users see multiple ads, making the models so far of expository, rather than practical, interest. Suppose now that a user sees multiple ads at times $t_1, \dots, t_J$. As a starting point, we consider the following model:

$$\log(\lambda(t)) = \alpha_0 + \sum_j f(t - t_j) + \sum_{j,k} g_k(t - t_j, x_{jk}),$$

where x_{jk} is the feature value for the kth feature for the jth ad. At a high level, this model says that ads with identical features have the same decay curve, but ads with different feature values may still have different decay curves. This is illustrated in Figure 4, which shows what an intensity curve for a user who sees three ads, each with different features values, might look like.

While this is not easily illustrated in a figure, notice that our model formulation also encodes that the effects of multiple ads are multiplicative (and therefore additive on a log-scale). This does not entirely match with our intuition: if showing a user an ad doubles their instantaneous conversion rate in the short-term, we wouldn't expect showing them 50 ads within 10 minutes (perhaps with multiple ads per site) to increase their conversion rate by a factor of 2^{50}. Simultaneously, it's intuitive that there might be a synergy between ads, such that seeing ad A might increase conversions by 2x, seeing ad B might increase conversions by 1.5x, but that seeing both ads increases conversions by 5x rather than 3x. In other words, interactions between ads are possible and our model needs to be flexible enough for the user to fit interaction effects, should they choose to.

One way to accomplish this is to generalize our notation slightly and allow x_{jk} to depend not only on ad j, but also on ads $j' < j$, as well as the subscript value j itself. In other words, the features can depend not only on the current ad, but on previous ads and the ad index.

This allows us to add an effect for being the jth ad. For example, assuming for simplicity that we have no other ad features, we could define a single ad feature $g_1(t-t_j, j) = f_j(t-t_j) - f(t-t_j)$, where f_j is the effect function for the jth ad and f is the "default" or average ad effect function. Then our model formulation is equivalent to letting

$$\log(\lambda(t)) = \alpha_0 + \sum_j f_j(t - t_j), \tag{2}$$

i.e. a model where the ad effect differs for each ad.

Allowing the feature vector for an ad, $x_j = (x_{j1}, \ldots, x_{jK})$, to depend on previous ads also allows us to encode interactions between different ad formats or the timing of different ads. For example, we could add a feature $I(t_j - t_{j'} < \Delta \text{ for } j' < j)$ as an indicator for whether there was a preceding ad in the previous Δ hours. This can be desirable if we only want to consider ad interactions if the two ads are within Δ hours of each other. Assuming for notational simplicity that this is the only feature we model, we have

$$\log(\lambda(t)) = \alpha_0 + \sum_j f(t - t_j) + \sum_j g_1(t - t_j) I(t_j - t_{j'} < \Delta \text{ for } j' < j), \tag{3}$$

where, without loss of generality, we have set $I(t_j - t_{j'} < \Delta \text{ for } j' < j) = 0$ as the "default" or reference level.

The complexity of the model and any included interactions are necessarily application-dependent. In the case of an advertiser with a large amount of data and many ads shown per user, we may be able to make detailed estimates of the marginal effect of successive ads, as well as their interactions. On the other hand, when data is scarcer, it may be more practical to assume all ads have the same effect. The key point here is that this framework is flexible enough to allow for many different specifications of the factors that affect the conversion rate.

3.1.4 User Features

User features that affect conversion rates can be handled in one of two ways. If the feature only changes the user's overall conversion rate, but not how they react to ads, then we can treat it as a shift in α_0. Taking as an example the case of a model with a single user feature, a user's bucketized age, we write

$$\log(\lambda(t)) = \alpha_0 + \alpha_{\text{age}} + \sum_j f(t - t_j) + \sum_{j,k} g_k(t - t_j, x_{jk}).$$

The standard identifiability restrictions for a regression with both an intercept and one or more categorical variables apply: we can either drop one of the coefficients for the age buckets, so that the interpretation of α_0 changes to be the average conversion intensity before any ads are shown for the dropped bucket or equivalently, drop α_0. Alternatively, we can require that $\sum \alpha_{\text{age}} = 0$, where the sum is over all possible age buckets, and the interpretation of α_{age} changes to be the deviation, for that age bucket, from the average conversion intensity before ads are shown.

If instead we believe that the user feature changes the ad effect, then we can incorporate it into the model in the same way as any other ad feature, x_{jk}, thinking of the feature as "age when this ad was shown." Of course, unlike other ad features, which may vary amongst ads on the same path, this one will not, but that does not change how we write our model. As long as this feature varies across users, it will still be estimable.

3.1.5 Experimental Data

So far we have considered a general model, which is applicable to all data and which, absent additional assumptions or prior experimental data, estimates the correlational effect of ads on conversion intensity. However, as mentioned in the introduction, with experimental data we can measure the causal effect of ads on conversion rates. These can be incorporated into our model.

The design of experiments measuring the causal effects of ads is highly dependent on the details of the media type and ad serving environment. For our discussion, we assume a generic design in which some ads are shown (“exposed”), while others are withheld (“unexposed”), but where we still log when an advertiser’s ad would have been shown had we not withheld it. We call the event where we receive a request for an ad a query event, and an event where the ad is actually returned an ad event. Thus when ads are shown, there are two simultaneous events, an ad event and a query event, while when ads are not shown there is a single query event.

Then, letting $B(j)$ be an indicator function that is 1 when ad j is shown and 0 otherwise, consider the model

$$\begin{aligned}\log(\lambda(t)) = \alpha_0 + \alpha_{\text{age}} &+ \sum_j f(t-t_j)B(j) + \sum_{j,k} g_k(t-t_j, x_{jk})B(j) \\ &+ \sum_j m(t-t_j) + \sum_{j,k} n_k(t-t_j, x_{jk}),\end{aligned} \tag{4}$$

where, analogously to f_j, $m_j(t-t_j) = m(t-t_j) + \sum_k n_k(t-t_j, x_{jk})$ represents the observed change in a user’s conversion rate (on a log scale) after the jth query. Note that as with the previous observational data, this query effect is not necessarily a causal effect: being targeted for an ad does not lead to you then making a purchase. If you use a search engine to search for “sneakers”, you’re more likely to buy sneakers after the query than a user who doesn’t do that search, but it probably wasn’t the query that caused that difference. Rather, you were interested in sneakers, and then you did the search.

The ad effect, separate from any query effects, for the jth ad is given by $f_j(t-t_j) = f(t-t_j) + \sum_k g_k(t-t_j, x_{jk})$. This is the additional increase (on a log scale) in a user’s conversion intensity if the ad is actually shown. Figure 5 shows the intensity curves for a user who saw ads (solid line) and a user who had the same queries but for whom ads were withheld. Their difference would represent the ad effect.

Whether our experiment is such that f_j, the ad effect for a user’s jth query, is causally identifiable depends on the details of our experiment, as well as how ads are served in our setting. In the generic design described above, the overall causal effect of all the ads subject to ablation, i.e. the number of incremental conversions, is clearly identifiable. However, if there are multiple types of ads (e.g., comparing effects of first and second ad or search ads and display ads), then either changes in the experimental design (e.g., randomly ablating subsets of ads for each user) or additional assumptions (e.g., order of ads doesn’t matter) may be needed for all the effects to be causally identifiable. The details of this kind of experimental design are out of scope for this paper. However, in our experience, quite often some function of the f_j has a valid causal estimator that can be derived from our model.

3.1.6 Further Refinements

So far the model has assumed that only ads, queries, and user features change the conversion rate. However, there are other factors which can also change conversion rates. In general, our model is flexible enough to accommodate many of these.

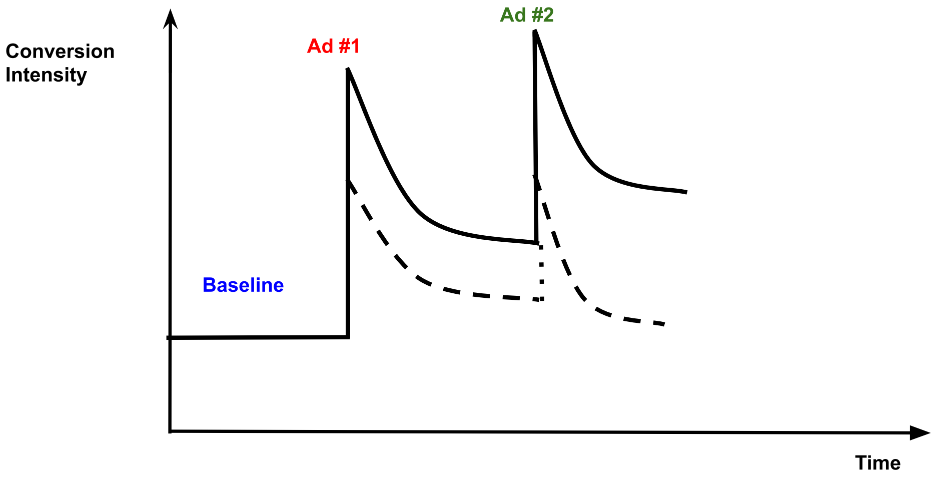

Figure 5: Conversion intensity over time for a user who saw ads (solid line) vs a user with the same queries for whom ads were held back (dashed line).

One common factor is seasonal effects, such as holidays, or more granular effects due to time of day or day of week. Modeling these can be as simple as adding indicators for each hour of day, or as complex as fitting additional spline functions to capture the pattern.

Another factor is that conversions themselves affect future rates of conversion. At one extreme are cases where only a single conversion is possible for each user, such as if the conversion represents signing up for a service or downloading an app. In those cases we can treat the model as a single-occurrence survival model and estimate it accordingly. In less extreme cases a conversion might decrease the likelihood of future conversions (e.g., someone buying a vacation package is probably less likely to buy a second one right away), but it can also increase the likelihood of a conversion (e.g., someone buying clothes from a retailer might be more likely to purchase from them again in future). Thus treating conversions themselves as events that can change the post-conversion intensity may improve the model. This is done by Xu et al. (2014), which models both ad clicks and conversions as mutually exciting Poisson processes, where the intensity of each process depends on both itself and the other processes. While we may not want to model the ad clicks explicitly in our setting, we may consider a similar approach of letting the conversion intensity depend on the number or timing of past conversions.

3.1.7 Estimation

There are different ways to estimate the intensity function of a Poisson process. If f, g_k are either piecewise constant or can be approximated as such, then by breaking each user's path into intervals where the intensity, $\lambda(t)$, is constant, we can treat each interval as an observation in a Poisson regression problem. The number of conversions in the interval is then the response, and the length of the interval is the offset.

There is a large literature on fitting these types of large-scale regression problems, Poisson and otherwise. The details are outside the scope of this paper and we will not attempt to survey the existing technologies, but point to Johnson et al. (2016) as one example of a practical solution. There the authors place a prior on the parameters and use Bayesian machine learning methods to estimate the parameters and their prior variances, essentially treating the parameters as random effects.

If we do not want to assume that f, g_k are (approximately) piecewise constant, then we can use the likelihood for an inhomogeneous Poisson process directly. Suppose we observe users for the time interval $[0, \tau]$. Let $\lambda_i(t)$ be the conversion intensity for user i at time t, and let T_{ij}, $j = 1, \ldots, C_i$ be the conversion times for user i. The log-likelihood is

$$\sum_{i=1}^{N} \left[-\int_0^\tau \lambda_i(t)\mathrm{d}t + \sum_{j=1}^{C_i} \log(\lambda_i(T_{ij})) \right].$$

See Andersen et al. (2012) for a detailed derivation. In the case where f, g_k (and therefore λ) are piecewise constant, this reduces to the log-likelihood for the Poisson regression approach above. Without the piecewise constant assumption, one could instead try standard optimization techniques, such as gradient descent, to estimate the parameters for f and g_k.

3.2 Credit Assignment Algorithm

Even given a model for conversions, there are still many ways to distribute credit for the observed conversions to preceding ads. Different methodologies with different properties may be desirable depending on the context. The method we propose, which we call backwards elimination, differs from existing methods in how it distributes credit for conversions (or changes in conversion intensity) that only occur because users were shown multiple ads. Our algorithm tends to give this credit to later ads, while Shapley value-based methods, commonly used in the existing literature, divide this credit evenly amongst the ads. Both methods are reasonable, but they solve different problems, as we will discuss later.

We focus on introducing backwards elimination in detail, including considerations for attribution with experimental data. We leave a more detailed examination of how backwards elimination distributes credit due to synergies between ads and comparison to Shapley values to Appendix B.

3.2.1 Backwards Elimination

To illustrate the backwards elimination algorithm, consider a user path with three ads followed by a conversion at time t^*, and whose estimated conversion intensity is plotted in Figure 6.

We define the contribution from the last ad before the conversion to be the difference in the estimated conversion intensity at t^* with all ads minus the estimated intensity at t^* if the last ad is dropped. The contribution from the second-to-last ad is the difference in conversion intensity at time t^* if the last ad is dropped minus the intensity if the last two ads are dropped. More generally, we proceed backwards through the path, removing an additional ad and attributing credit to the removed ad in proportion to the resulting change in intensity.

More formally, let $\mathcal{A}(n) = \{(X_j, t_j) : j = 1, \ldots, n\}$ denote the first n ads on user i's path, where we have dropped the index i for convenience, and where $X_j = (x_{j1}, \ldots, x_{jk}, \ldots, x_{jK})$ is the vector of ad features in our model. Without loss of generality, suppose that $t_n < t^* < t_{n+1}$, i.e. that the conversion of interest occurs after the nth ad but before the $n+1^{st}$ ad. Let $\hat{\lambda}(t, \mathcal{A}(j))$ be the estimated conversion intensity at time t for a user who sees the ads in $\mathcal{A}(j)$. Let $C_{\text{raw}}(j)$ denote the raw credit from our algorithm. This raw credit equals

$$C_{\text{raw}}(j) = \hat{\lambda}(t^*, \mathcal{A}(j)) - \hat{\lambda}(t^*, \mathcal{A}(j-1)),$$

where we let $\mathcal{A}(0) = \emptyset$. We can also define the baseline credit as $C_{\text{raw}}(baseline) = \hat{\lambda}(t^*, \emptyset)$. Notice that the total raw credit given to ads equals $\hat{\lambda}(t^*, \mathcal{A}(n)) - \hat{\lambda}(t^*, \emptyset))$, which in turn equals

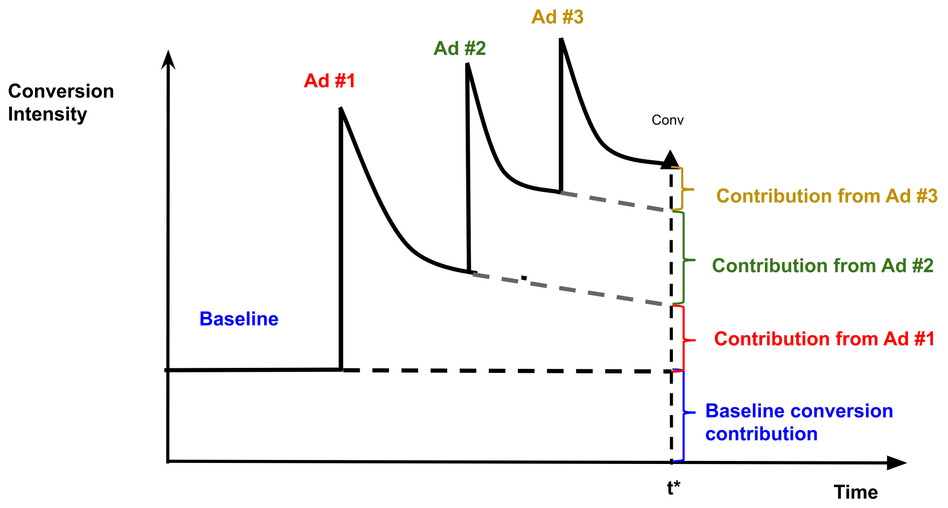

Figure 6: Attributed contribution of each ad to a user conversion occurring at t^*.

the difference in the instantaneous conversion rate at t^* if all ads are dropped. This follows straightforwardly from the telescoping nature of the formula above.

As an example, suppose the estimated model for $\hat{\lambda}(t)$ is of the form

$$\log(\hat{\lambda}(t)) = \hat{\alpha}_0 + \hat{\alpha}_{\text{age}} + \sum_j \hat{f}(t - t_j) + \sum_{j,k} \hat{g}_k(t - t_j, x_{jk}).$$

Then for the l^{th} ad, we have

$$\log(\hat{\lambda}(t^*, \mathcal{A}(l))) = \hat{\alpha}_0 + \hat{\alpha}_{\text{age}} + \sum_{j=1}^{l} \hat{f}(t^* - t_j) + \sum_{j=1}^{l} \sum_k \hat{g}_k(t^* - t_j, x_{jk}).$$

With this model for successive ads $\hat{\lambda}(t^*, \mathcal{A}(l))$ will differ from $\hat{\lambda}(t^*, \mathcal{A}(l-1))$ by a multiplicative factor of $\exp(\hat{f}(t^* - t_l) + \sum_k \hat{g}_k(t^* - t_l, x_{lk}))$ so that

$$\begin{aligned} C_{\text{raw}}(l) &= \hat{\lambda}(t^*, \mathcal{A}(l)) - \hat{\lambda}(t^*, \mathcal{A}(l-1)) \\ &= \hat{\lambda}(t^*, \mathcal{A}(l-1)) \left(\exp\left(\hat{f}(t^* - t_l) + \sum_k \hat{g}_k(t^* - t_l, x_{lk}) \right) - 1 \right). \end{aligned}$$

There are two natural ways to normalize the raw credit. We can consider either a version that normalizes by the total credit

$$C_{\text{norm}}(j) = \frac{\hat{\lambda}(t^*, \mathcal{A}(j)) - \hat{\lambda}(t^*, \mathcal{A}(j-1))}{\hat{\lambda}(t^*, \mathcal{A}(n))}$$

or a version that normalizes only by the ad credit

$$C'_{\text{norm}}(j) = \frac{\hat{\lambda}(t^*, \mathcal{A}(j)) - \hat{\lambda}(t^*, \mathcal{A}(j-1))}{\hat{\lambda}(t^*, \mathcal{A}(n)) - \hat{\lambda}(t^*, \emptyset)}.$$

If we use $C_{\text{norm}}(j)$, and normalize $C_{\text{raw}}(baseline)$ in a similar way, then the total ad credit plus the baseline credit will equal the number of conversions. If we instead use $C'_{\text{norm}}(j)$, then the total ad credit will equal the total number of conversions.

It can be shown that, assuming our estimated intensity is correct, the total expected credit from all conversion occurrences in the path is

$$E[C_{\text{norm}}(j)] = \int \Big(\hat{\lambda}(t, \mathcal{A}(j)) - \hat{\lambda}(t, \mathcal{A}(j-1))\Big)\,\mathrm{d}t \tag{5}$$

or the difference in the expected number of conversions for a user seeing the ads in $\mathcal{A}(j)$ versus a user seeing the ads in $\mathcal{A}(j-1)$. See Appendix A for details. Thus the expected ad credit for ad j is the expected number of additional conversions gained when it was added to the end of the path, without considering gains due to combining ad j with later ads.

One implication of this is that for a path with multiple ads, credit for "extra" conversions that occur because the ads are shown to the same user, rather than each ad being shown to different users, goes to the later ads, at least in expectation. In fact, this holds not just in expectation and is actually a property of the raw credit, not just the normalized credit. The details are in Appendix B, where we give an example and compare how backwards elimination and Shapley values divide these "extra" conversions.

3.2.2 Incremental Attribution

Suppose that we have built a model for $\hat{\lambda}(t)$ with experimental data and separate ad and query effects. In this case we want to attribute credit to ads based only on the increase in conversion intensity caused by ads and not the query effect. For concreteness, suppose the model is the same as equation (4), i.e.:

$$\begin{aligned}\log(\lambda(t)) = \alpha_0 + \alpha_{\text{age}} &+ \sum_j f(t-t_j)B(j) + \sum_{j,k} g_k(t-t_j, x_{jk})B(j)\\ &+ \sum_j m(t-t_j) + \sum_{j,k} n_k(t-t_j, x_{jk}).\end{aligned}$$

We define the attribution credit in the same way as before, removing a single ad at a time and observing the change in $\hat{\lambda}(t^*)$, but now we keep all the query effects throughout. Assuming, as before, that there are exactly n ads before the conversion, we have

$$\begin{aligned}\log(\hat{\lambda}(t^*, \mathcal{A}(l)) = \hat{\alpha}_0 + \hat{\alpha}_{\text{age}} &+ \sum_{j=1}^{l} \hat{f}(t-t_j)B(j) + \sum_{j=1}^{l}\sum_k \hat{g}_k(t-t_j, x_{jk})B(j)\\ &+ \sum_{j=1}^{n} \hat{m}(t-t_j) + \sum_{j=1}^{n}\sum_k \hat{n}_k(t-t_j, x_{jk}).\end{aligned}$$

That is, the summations for $\hat{f}$ and $\hat{g}_k$, the ad effects, consider only the ads in $\mathcal{A}(l)$, while the summations for $\hat{m}$ and $\hat{n}_k$, the query effects, consider all of the n events preceding the conversion. The overall raw credit given to ads is therefore the difference between the estimated conversion intensity at t^* and the estimated conversion intensity for a counterfactual path with the same queries, but where ads were always withheld. As with the non-incremental credit, it can be shown that the expected value of the total normalized ad credit for each user equals the expected number of incremental conversions (i.e. the expected difference in conversions with and without ads) for that path. See Appendix A for details. Thus while it may seem strange to have an ad's credit depend on features of later queries, this actually leads to the most intuitive value of the individual and total ad credit.

4 Evaluation

To evaluate this system, we use standard model fit metrics for a Poisson regression, such as the log-likelihood or Poisson loss. There is also the prediction bias: predicted conversions / observed conversions – 1. We can also consider sliced versions of these metrics to aid in model comparison for feature selection.

With experimental data, we can estimate the ground truth number of incremental conversions that are caused by ads without needing a model by simply comparing the number of conversions in the exposed and unexposed groups. To evaluate our model, which is fit using both the exposed and unexposed users, we can compare this ground truth estimate to the predicted incremental conversions obtained by comparing the model's predicted number of conversions in the exposed and unexposed groups. If our goal is to correctly model incrementality, comparing the ground truth and predicted versions of incremental conversions (i.e the difference between groups) may be of greater interest than comparing the actual and predicted conversions in each group. As a way of evaluating both our model and credit assignment methodology, we can also compare the ground truth incremental conversions to the normalized version of the ad credit. To justify this comparison, recall that as discussed in Section 3.2.2, the expected value of the ad credit should equal the expected number of incremental conversions, assuming the model estimates are correct.

The exact estimators for these metrics depends on the details of how the experiment is run, which in turn depend on the details of the media type and ad serving environment. For simplicity and specificity, suppose again that the experiment splits users for the duration of the experiment into either an exposed group, which sees ads as normal, or an unexposed group, for whom ads are withheld, but that we are able to observe the conversions of both groups. Then we can consider the following ground truth metric, which we call incremental conversions per user:

$$\text{ICPU} = \frac{\text{conversions in exposed group}}{\text{users in exposed group}} - \frac{\text{conversions in unexposed group}}{\text{users in unexposed group}}.$$

This is essentially the observed difference in the average conversion rate between exposed and unexposed groups. A closely related metric is the rate of incremental conversions per unit time:

$$\begin{aligned}\text{ICPT} = {}& \frac{\text{conversions in exposed group}}{\text{total observation time for users in exposed group}} \\ &- \frac{\text{conversions in unexposed group}}{\text{total observation time for users in unexposed group}},\end{aligned}$$

where the total observation time is defined as

$$\begin{aligned}&\text{total observation time for users in } X \text{ group} = \\ &\qquad\qquad \sum_{i:i\in X} \text{total time that user } i \text{ is observed in the experiment.}\end{aligned}$$

If all users are observed for the same length of time in the experiment, then ICPT differs from ICPU by a constant factor. However, if the average time that each user is observed for differs between exposed and unexposed, then the difference is more complicated and ICPT may be a more useful metric.

We can also consider ICPE or incremental conversions per exposed conversion:

$$\text{ICPE} = \text{ICPU} \times \frac{\text{users in exposed group}}{\text{conversions in exposed group}}.$$

This represents the proportion of conversions in the exposed group that are incremental, after normalizing for any differences in the sizes of the exposed and unexposed groups. As with ICPT compared to ICPU, if the average time that each user is observed for differs between exposed and unexposed, we may prefer a version of ICPE that accounts for this:

$$\text{ICPE'} = \text{ICPT} \times \frac{\text{total observation time for users in exposed group}}{\text{conversions in exposed group}}.$$

We can compare each of these with their counterparts predicted by the model, which we call PICPU (predicted incremental conversions per user) and PICPPE (predicted incremental conversions per predicted exposed conversion). Their definitions are the same, but with "conversions" replaced by "predicted conversions". We can also evaluate our attribution methodology by considering the attribution credit, normalized by $\hat{\lambda}(t^*)$, given to ads for conversions that occur in the exposed group. This metric, which we call AICPE (attributed incremental conversions per exposed conversion), represents the proportion of credit for exposed conversions that is given to ads and is comparable to PICPPE and ICPE. Comparing PICPPE and AICPE against ICPE is therefore an appropriate validation metric for our model and attribution methodology, respectively. Confidence intervals for these metrics can be obtained by bootstrapping over users. Alternatively, we could incorporate the uncertainty over the model itself by bootstrapping or using a block jackknife to refit the model and compute the evaluation metrics per block and then compute the variability of this estimate over the blocks.

For feature selection, it can be helpful to compare these metrics on various slices of users, however we must be cautious in choosing features to slice on. In particular, we must avoid any confounding between a user's treatment assignment, slice value, and conversion outcome. For example, if ad exposure increases the length of a user's path, perhaps because seeing ads leads to increased activity, then slicing by the total number of queries in a path leads to incomparable sets of exposed and unexposed users. However slicing on user features that are unaffected by the experiment, such as city or gender, is still valid and can be useful both for feature selection and understanding user behavior.

5 Simulation Study

We simulate user paths and show that our system performs well on this data. In general, simulating user paths and conversion behavior requires strong assumptions about the generating process for both ad exposures and conversions. These assumptions are necessarily much simpler than actual user behavior. Nevertheless, simulations can still provide confidence by showing that the model is working as expected in these cases. We will focus on a single scenario here, leaving the remainder to the supplementary materials.

In each scenario, we simulate 500 distinct data sets ("advertisers"), each with 1 million users and 30 days of data per user. For each dataset, we also create 1000 bootstrap replicates and fit the model on both the original and replicated datasets. We compute the "basic" or reverse percentile bootstrap interval (Davison and Hinkley, 1997) for the model coefficients:

$$(2\hat{\theta} - \hat{\theta}^*_{1-\alpha/2}, 2\hat{\theta} - \hat{\theta}^*_{\alpha/2}),$$

where $\hat{\theta}$ is the estimate of the parameter using the original dataset and $\hat{\theta}^*_\alpha$ is the α quantile of the bootstrap estimates for the parameter. We take $\alpha = 0.05$ and also compute the coverage of the resulting CIs over the 500 distinct datasets.

5.1 Scenario 1

We will consider two possible types of ads, each with a different effect on conversions, and simulate each user as having exactly one ad of each type (for a total of two ads). We simulate the ad occurrence times as uniform in our 30 day observation window, i.e. [0, 30.0] and independent of each other. The first ad type doubles the conversion rate on the first day after the ad, increases it by a factor of 1.5 on the second day, and increases it by a factor of 1.2 after that. We refer to these as the short, medium, and long term effects, respectively. The second ad type increases the conversion intensity by a factor of 1.5 on the first day, 1.2 on the second day and 1.0 (or no increase) after that.

We can then simulate a user's conversion events as a Poisson process where the intensity is

$$\begin{aligned}\log(\lambda(t)) = \alpha_0 &+ \beta_1 I\{0 < t - t_{\text{ad type 1}} \leqslant 1\} + \beta_2 I\{1 < t - t_{\text{ad type 1}} \leqslant 2\} + \beta_3 I\{2 < t - t_{\text{ad type 1}} \leqslant 30\} \\ &+ \beta_4 I\{0 < t - t_{\text{ad type 2}} \leqslant 1\} + \beta_5 I\{1 < t - t_{\text{ad type 2}} \leqslant 2\} + \beta_6 I\{2 < t - t_{\text{ad type 2}} \leqslant 30\},\end{aligned}$$

where $t_{\text{ad type i}}$ is the occurrence time for the ad of type i for this user and the β_i's are as described above; the values are also listed in Table 1. There are a number of ways to simulate this Poisson process; we follow Algorithm 5 in Pasupathy (2010). This algorithm becomes particularly straightforward in our case because the intensities are piecewise constant. An example of a simulated dataset for 1 advertiser with 1 million users is included in the supplementary materials.

Table 1 shows the ground truth parameters, an example estimate and bootstrap 95% CI for a single advertiser (to help with the reader's intuition about the typical width of the CI's), as well as the coverage percentage of the 95% CI's over the 500 datasets. As we can see, we get reasonably close to the nominal 95% coverage.

We can use our attribution methodology and normalize by $\hat{\lambda}(t^*)$ to get AICPE. This model was not fit on simulated experimental data and no query effect was fit, so there is technically no ICPE to compare this to. However, we can compare it to what the ICPE would be if we assumed there was no query effect and we simulated additional users who have an ad query but don't see an ad, i.e. users whose conversion intensity is equal to the baseline for the entire observation window. This is equivalent to an experimental model where there is no query effect: i.e. where the observed and incremental effect of ads is the same. While computing E[ICPE] and E[AICPE] in this scenario is somewhat involved, by leveraging equation (5), it can be shown that under very mild assumptions $E[\text{ICPE}] = E[\text{AICPE}]$. Thus we define the additive error as AICPE − ICPE and the relative error as AIPCE/ICPE − 1 and consider whether their CI's cover 0. The results are in Table 2 and show that we achieve the nominal coverage.

6 Discussion

We have presented a data-driven attribution system based on estimating the effect of ads on a user's conversion rate per unit of time. This system satisfies our previously outlined requirements,

Table 1: Ground truth, estimated model coefficients, and coverage % for Scenario 1.

		Ground Truth	Example Estimate [Example CI]	95% CI Coverage (over 500 datasets)
Baseline (per day) $[\exp(\alpha_0)]$		0.0333	0.0333 [0.0332, 0.0334]	95.4%
Ad Type 1	Short term $[\exp(\beta_1)]$	2.0	2.006 [1.990, 2.023]	94.4%
	Medium term $[\exp(\beta_2)]$	1.5	1.512 [1.498, 1.526]	92.8%
	Long term $[\exp(\beta_3)]$	1.2	1.199 [1.194, 1.203]	95.2%
Ad Type 2	Short term $[\exp(\beta_4)]$	1.5	1.504 [1.492, 1.517]	93.8%
	Medium term $[\exp(\beta_5)]$	1.2	1.207 [1.196, 1.219]	94.8%
	Long term $[\exp(\beta_6)]$	1.0	1.000 [1.000, 1.003]	91.6%

Table 2: Additive and relative error of AICPE for Scenario 1.

	Example Estimate [Example CI]	95% CI Coverage (over 500 datasets)
Additive Error $\times 1e-2$	0.05 $[-0.25, 0.35]$	94.8%
Relative Error $\times 1e-2$	0.34 $[-1.81, 2.49]$	94.8%

namely that it can handle incomplete or censored data as well as take into account the times as which ads occur, not just their order, when assigning credit. These features of our system make it appropriate for use in real-time bidding, although the details are beyond the scope of this paper given the many application-specific considerations. We have also discussed some examples of how to use covariates to model the conversion intensity over time, although again the detailed choice of covariates is highly application-dependent. In this section, we will discuss two areas of potential future work. The first relates to the relationship between ads, while the second concerns the effect of modeling assumptions on attribution outcomes.

Thus far, we have assumed that ads are independent of each other, in addition to the standard Poisson process assumption that conversions are independent of each other. However, for some types of media seeing an ad can lead to a user seeing more ads in the future. This can be a direct result of a user's interactions with an ad, such a click on a display ad that then leads to the user seeing further related ads. It can also occur as a result of more indirect interactions, such as when seeing an ad prompts a user to search for a related term, leading to them seeing additional related search ads. In these cases, to the extent that the later ads are caused by the earlier ads, some of the credit for the later ads should arguably be redistributed to the earlier ads. Our system does not currently account for these effects when allocating credit. One way to remedy this might be by having a multi-stage model, where we first use all the ads to predict conversions. Consider the earlier example where display ads can cause search ads. Suppose that in our original model, the search ad gets 0.6 credit and the display ad gets 0.4. We can fit a second model where we use display ads as the events and search ads (rather than conversions) as the response. We can then allocate credit for the search ad between the preceding display ads and a baseline. If, for example, in this second model 30% of the credit for the search ad occurring goes to the display ad, with the remaining 70% going to the baseline, then we can reallocate 30% of the conversion credit attributed to the search ad in the first model to the display ad, keeping 70% of the credit with the search ad. As a result, the total credit for the display ad would be $0.4 + 0.3 \times 0.6 = 0.58$ and the credit for the search ad would be $0.7 \times 0.6 = 0.42$. In practice however, this requires a rich dataset in order to be able to detect these interactions and thus may not always be possible.

While we have discussed a few examples of features and structure for modeling $\lambda(t)$, we have not discussed in detail how the model structure influences attribution results. Consider a simpler version of the model from our simulation, where all ads have he same effect but where the user may have many ads:

$$\begin{aligned}\log(\lambda(t)) = \alpha_0 &+ \beta_1 I\{0 < t - t_i \leqslant 1 \text{ for some i}\}\\ &+ \beta_2 I\{1 < t - t_i \leqslant 2 \text{ for some i}\}\\ &+ \beta_3 I\{2 < t - t_i \leqslant 30 \text{ for some i}\}.\end{aligned}$$

If there were two ads in the 24 hours before a conversion, the second ad would not change the estimated conversion intensity at conversion time and would therefore get 0 credit, i.e. if we assume there are only two ads in the path, $\hat{\lambda}(t^*, \mathcal{A}(2)) - \hat{\lambda}(t^*, \mathcal{A}(1)) = 0$. If we find it unrealistic to distribute all the credit to the first ad rather than the second, then in this particular case we can simply model the marginal effects of an additional ad in each time bucket (this is similar to simulation Scenario 4, considered in the supplementary materials):

$$\begin{aligned}\log(\lambda(t)) = \alpha_0 &+ \beta_1 I\{0 < t - t_i \leqslant 1\} + \beta_2 I\{1 < t - t_i \leqslant 2\} + \beta_3 I\{0 < t - t_i \leqslant 30\}\\ &+ \beta_4 I\{0 < t - t_i \leqslant 1 \text{ for two distinct values of i}\}\\ &+ \beta_5 I\{1 < t - t_i \leqslant 2 \text{ for two distinct values of i}\}\\ &+ \beta_6 I\{0 < t - t_i \leqslant 30 \text{ for two distinct values of i}\}.\end{aligned}$$

More generally, we can consider a model where we estimate a separate ad decay function for each ad as in Equation (2):

$$\log(\lambda(t)) = \alpha_0 + \sum_j f_j(t - t_j).$$

However, we cannot model an infinite number of ads in each term. At some point either data sparsity or regularization will lead to some of the later ad effects being estimated as 0. As a result, in some cases the last ads before a conversion may get 0 credit. One alternative to avoid this is to pool data and estimate some ads as having the same effect. This can be done by assuming that all ads $j > j'$ have the same effect:

$$\log(\lambda(t)) = \alpha_0 + \sum_{j=1}^{j'} f_j(t - t_j) + \sum_{j=j'}^{J} f_{j'}(t - t_j).$$

Or we could pool in a more sophisticated way, by e.g., supposing that an ad with no other ads in the preceding X days "resets" the counter and has the same effect as the very first ad. This pooling makes it more likely that the last ads before a conversion will get non-zero attribution credit.

The larger point here is that the model structure and the way in which we pool data to estimate the different ad decay curves has repercussions on the results of the attribution algorithm. We can of course use overall model fit metrics to guide our choices, but this points to the necessity of considering a wide range of models. Care may also be needed, since attribution results may differ most for long paths, where we most care about MTA, while overall model fit may be comparable, if long paths are relatively rare. As with most modeling choices, the right approach depends on the context and requires careful consideration from the modeller.

Supplementary Material

The supplementary material contains three files. The first is a PDF with additional technical material: In Appendix A we prove a result about $E[NormalizedCredit(j)]$ mentioned in section 3.2.1 of the main text. In Appendix B we give a detailed discussion of the similarities and differences between Backwards Elimination (our proposed attribution method) and Shapley Values, another commonly used method. In Appendix C, we present additional simulation scenarios and their results.

The other two files are a data file, data.csv, with sample data used in our simulations, and a README file with a detailed description of the data. We do not include code to replicate the simulation results. While our method can be applied using standard Poisson regression methods, our simulation framework is tightly integrated with our production environment and our proprietary data format. This makes sharing the code impractical.

Acknowledgement

The authors would like to thank both past and present members of the TEDDA team at Google for their contributions to developing this methodology, as well as members from other Ads teams for their tremendous support.

References

Anderl E, Becker I, Wangenheim FV, Schumann JH (2014). Mapping the customer journey: A graph-based framework for online attribution modeling. Available at SSRN 2343077.

Andersen PK, Borgan O, Gill RD, Keiding N (2012). *Statistical Models Based on Counting Processes*. Springer Science & Business Media.

Cook RJ, Lawless J (2007). *The Statistical Analysis of Recurrent Events*. Springer Science & Business Media.

Dalessandro B, Perlich C, Stitelman O, Provost F (2012). Causally motivated attribution for online advertising. In: *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*, 1–9.

Davison AC, Hinkley DV (1997). *Bootstrap Methods and Their Application. 1*. Cambridge university press.

De Haan E, Wiesel T, Pauwels K (2016). The effectiveness of different forms of online advertising for purchase conversion in a multiple-channel attribution framework. *International Journal of Research in Marketing*, 33(3): 491–507. https://doi.org/10.1016/j.ijresmar.2015.12.001

Du R, Zhong Y, Nair H, Cui B, Shou R (2019). Causally driven incremental multi touch attribution using a recurrent neural network.

Dugan L (2011). The series hazard model: An alternative to time series for event data. *Journal of Quantitative Criminology*, 27: 379–402. https://doi.org/10.1007/s10940-010-9127-1

Geng T, Sun F, Wu D, Zhou W, Nair H, Lin Z (2021). Automated bidding and budget optimization for performance advertising campaigns. Available at SSRN 3913039.

Johnson NA, Kuehnel FO, Amini AN (2016). A scalable blocked gibbs sampling algorithm for gaussian and poisson regression models.

Kireyev P, Pauwels K, Gupta S (2016). Do display ads influence search? attribution and dynamics in online advertising. *International Journal of Research in Marketing*, 33(3): 475–490. https://doi.org/10.1016/j.ijresmar.2015.09.007

Kumar S, Gupta G, Prasad R, Chatterjee A, Vig L, Shroff G (2020). Camta: Causal attention model for multi-touch attribution. In: *2020 International Conference on Data Mining Workshops (ICDMW)*, 79–86. IEEE.

Lewis RA, Wong J (2018). Incrementality bidding & attribution. Available at SSRN: https://ssrn.com/abstract=3129350 or http://dx.doi.org/10.2139/ssrn.3129350.

Li H, Kannan P (2014). Attributing conversions in a multichannel online marketing environment: An empirical model and a field experiment. *Journal of Marketing Research*, 51(1): 40–56. https://doi.org/10.1509/jmr.13.0050

Pasupathy R (2010). Generating homogeneous poisson processes. *Wiley Encyclopedia of Operations Research and Management Science.*

Shao X, Li L (2011). Data-driven multi-touch attribution models. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 258–264.

Xu L, Duan JA, Whinston A (2014). Path to purchase: A mutually exciting point process model for online advertising and conversion. *Management Science*, 60(6): 1392–1412. https://doi.org/10.1287/mnsc.2014.1952

Yao D, Gong C, Zhang L, Chen S, Bi J (2021). Causalmta: Eliminating the user confounding bias for causal multi-touch attribution. arXiv preprint: https://arxiv.org/abs/2201.00689.

Zhang Y, Wei Y, Ren J (2014). Multi-touch attribution in online advertising with survival theory. In: *2014 IEEE International Conference on Data Mining*, 687–696. IEEE.

Zhao K, Mahboobi SH, Bagheri SR (2018). Shapley value methods for attribution modeling in online advertising.