Attribution Evaluation with User Matched Paths

Gil Tabak[1], Jon Vaver[1], and Qixuan Feng[2]

[1]Google, Inc.
[2]Booking.com

June 15, 2021

Abstract

Many digital advertisers continue to rely on attribution models to estimate the effectiveness of their marketing spend, allocate budget, and guide bidding decisions for real time auctions. The work described in this paper builds on previous efforts to better understand the capabilities and limitations of attribution models using simulated path data with experiment-based ground truth. While previous efforts were based on a generic specification of user path characteristics (e.g., ad channels considered, observed events included, and the transition rates between observed events), here we generalize the process to include a pre-analysis optimization step that matches the characteristics of the simulated path data with a set of reference path data from a particular advertiser. An attribution model analysis conducted with path-matched data is more relevant and applicable to an advertiser than generic path data. We demonstrate this path-fitting process using data from Booking.com. The simulated matched paths are used to demonstrate a few key capabilities and limitations for several position-based attribution models.

1 Introduction

1.1 Need for measurement based attribution

As the options for online advertising continue to grow, it has become increasingly important for advertisers to have practical and reliable approaches for measuring the effectiveness of their advertising spend. This information is needed to inform tactical and strategic decisions regarding future ad spend. Multi-touch attribution (MTA) models are a relatively simple, cheap, automated, and continuous approach for estimating ad effectiveness. However, systematically assessing the accuracy of these models is not straightforward.

MTA models use observational user-level path data to assign credit for conversions back to the marketing events that users encountered prior to converting. These attribution credits are aggregated across all of the individual marketing events to determine the overall value of each marketing channel.

In practice, assessing the accuracy of MTA models can be problematic because (1) attribution credit is not well defined relative to the incremental impact of advertising, (2) the same attribution model may be used to inform multiple decisions that require different incremental measurement (e.g., ad channel prioritization, budget allocation across channels, and real-time bidding), and (3) there is no generally accepted method for validating MTA models for any application. Because they are observational, MTA models make assumptions about how advertising impacts user behavior. These assumptions are usually not stated and/or understood by advertisers, which could lead to misconceptions about MTA efficacy [7].

If it is effective, an MTA model should provide information about the incremental impact of each marketing channel (e.g., the number of conversions that would be lost if all marketing spend for an ad channel was turned off). Because MTA models rely on observational data, they are susceptible to biases that would not be present in a well designed randomized experiment. Unfortunately, experiments are not always a practical alternative to MTA. For example, it is usually not practical to design and execute the full factorial experiment needed to account for the primary and synergistic effects of multiple ad channels that span multiple media and perhaps multiple publishers (See Appendix A in [4]). Advertisers may also be reluctant to use experiments as a continuous form of measurement because experiments require reducing ad spend to support a control group and this can result in a loss of revenue.

1.2 Relationship to previous work

Previously, an ad system simulation (DASS) was used to design 'unit tests' for evaluating MTA models [8]. The simulation is a non-stationary Markov model that specifies the browsing activities of users (e.g., search, visit to a third party website, or watch a video) and the probabilities that determine how users transition from one activity to the next. It also allows the injection of ads that can modify these transition probabilities over the course of a user's path of activities (e.g., a search ad gives a user the opportunity to visit the advertiser's website via a search ad click, or a display ad makes it more likely that a user will directly navigate to the advertiser's website). This framework generates user-level path data that is used as input for MTA models. At the same time, it also makes it possible to run experiments in the simulator environment that deliberately turn ad activity on or off to measure the true impact of ads on conversion volume.

If an MTA model is a perfect substitute for the measurement provided by experiments, then it will be able to accurately estimate the effectiveness of every ad channel in every possible advertising scenario and for every potential impact of ads on user behavior (e.g., search ads that have an impression value, video ads that drive users to make advertiser-related searches, and targeted or retargeted display ads that inspire users to navigate to the advertiser's website). To test this supposition, an array of 'unit tests' with known ground truth for ad effectiveness was created. The application of these tests can help us better understand the capabilities and limitations of MTA models as described in [9].

1.3 Need for current work

Simulation-based analysis of attribution models is more likely to be useful to advertisers if the simulated data closely resemble their path data (e.g., similar ad channels, observed events, event-to-event transition probabilities). The information learned from generic attribution analysis is most useful for specific applications like comparing and developing new attribution models and understanding how they perform under different data conditions. Parametrizing the simulated scenarios to a specific advertiser provides results that are more immediately relevant to that advertiser and their measurement concerns and needs.

The main goal of this paper is to extend the previous attribution model analysis work (see Section 1.2) by first fitting a model that can generate path data that are similar to the path data associated with a particular advertiser. The approach described below makes use of an intermediate state-space model during the parameter fitting optimization stage of the analysis. This model is analogous to the non-stationary Markov model used in past work [8]. After this fitting, the results are translated into a DASS model parameterization that can be used to perform MTA model evaluations. The evaluation stage is analogous to that of prior analysis work – for each unit test, we make an assumption about how ads impact user behavior, specify a magnitude of ad impact, generate an associated set of simulated path data, apply MTA models to the path data, and compare MTA model results with experiment-based ground truth.

1.4 Organization of this paper

This paper is organized as follows. Section 2 contains a description of the target path data from observations of real advertiser data from Booking.com. Section 3 includes an overview of the fitting process used to generate simulated path data that matches this target path data. More detailed modeling choices related to fitting are described in Section 4 and fitting results are shown in Section 5. In Section 6, the matched path data is used in comparing ad effectiveness ground truth from simulator-based experiments to the attribution credit assigned by several position-based attribution models. Finally, Section 7 includes a summary and areas of future research.

2 Target Path Data

In this section we describe some key characteristics of user level path data. While our focus is path data from Booking.com, we expect some of these characteristics to be similar for many other advertisers as well.

2.1 Events and scope

User level path data consists of an ordered sequence of events such as a search, an ad impression, an ad click, or a conversion. In general, events are also timestamped, which we currently ignore for simplicity and because many widely used attribution models do not use these as input. Attribution models do not have visibility to all user-level events. We use the term *in-scope* to refer to the subset of events that are included as input to the MTA model. In-scope events generally include conversions and clicks, but not ad impressions or third-party site visits. For the path data used in this paper, the in-scope events include (1) conversions, (2) paid clicks for ad channels including search, display, and other, and (3) unpaid

search clicks, which we denote SEO (search-engine optimization). In addition, special events are used to denote the start and end of a path.

Explicitly, the in-scope events are:

- conversion
- search clicks (possibly multiple types)
- display clicks (possibly multiple types)
- other clicks (possibly multiple types)
- SEO clicks, i.e. unpaid visit after a search
- start of path
- end of path

Out-of-scope events include paid ad impressions (although in other applications of MTA these could be in-scope), user searches and search ad impressions, third party site visits, and any other possible user states. These out-of-scope events play an important role in path generation and the path fitting process. The specific structuring of out-of-scope events relies on modeling choices with assumptions about their relationship to observable user behavior. We elaborate on these choices for out-of-scope events in Section 4.

2.2 Transition probabilities

The aggregate statistic used for path-fitting is based on bigram frequencies of in-scope events; the prevalence of transitioning from in-scope event A to in-scope event B. These in-scope events include the start and end of path. The target path data have several important characteristics that helped inform the model choice:

1. Some events are orders of magnitude more common than others.
2. Repeated in-scope events occur frequently.
3. Transitions are approximately symmetric.

The relative rarity of some bigrams (item 1 above) is illustrated in a histogram of all (row-normalized) bigrams in Figure 1.

In Section 3.2.4 we discuss how item 2 affects our modelling choices.

2.3 Fabricated Data

To avoid sharing proprietary information related to Booking.com path data, we generate fabricated target bigram frequency data with similar overall characteristics for the purpose of fully illustrating the path fitting process. The method for generating fabricated data is described in Appendix A, and a related set of analysis results is described in Appendix B.

2.4 Other path characteristics

Bigram frequencies do not completely characterize a set of user path data. For example, in our target path data, the path lengths have a 'long-tailed' behavior. Some users have very short paths (often as short as a single event), while other paths are very long. Since we only use bigram frequencies in the fitting process, we do not expect to fully capture this behavior. However, this is probably not the most important data feature to capture in an initial evaluation of MTA models. We discuss possible extensions to our work to overcome issues of this kind in Section 7.3.

3 Path-Fitting Methodology

Our path-fitting goal is to assign values to the parameters of a model of the kind discussed in [8] to generate paths with aggregated statistics that match those of a target set of paths. This is one version of path-fitting that we expect will capture much of the information specific to a given advertiser's path data. Of course, it is possible to consider more generalized forms of path fitting (see Section 7.3). These are not considered here.

To fully capture the range of possible impacts that advertising can have on user behavior, we need an ad system simulation model that allows ads to change user behavior at the point of ad serving (e.g., a click on a search ad impression) and downstream of ad serving (e.g., a future direct visit to the advertiser's website). For the purposes of path fitting, we use a simplified model that only captures that former type of ad impact and not the latter. This simplification does not limit subsequent MTA evaluation because the path fitted parameters can be mapped to DASS model parameters, which allows downstream ad impact to be considered in MTA evaluation, as illustrated in Section 6.

In this section we discuss in detail how we construct this 'short-term effect' Markov model and the path fitting procedure. In Section 4 we discuss additional modeling choices needed for the model to be useful.

3.1 Alternative approaches

The most direct approach to path fitting is to fit DASS parameters directly using a black-box optimization algorithm. This does not require any modification or simplification of the DASS model. However, it has severe shortcomings because the optimization procedure does not have access to analytically specified gradients. Gradients must be approximated with multiple simulation runs, each of which

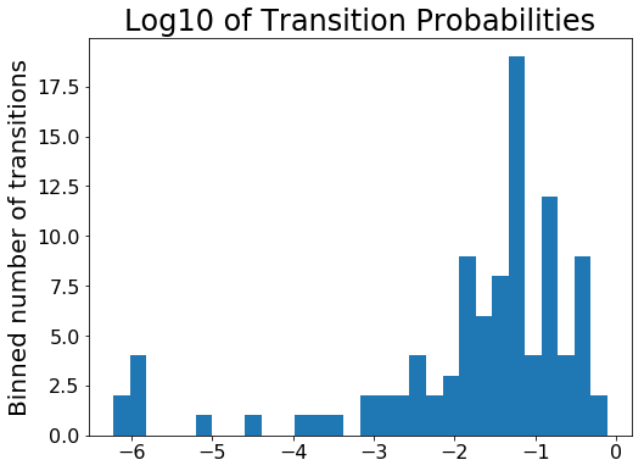


Figure 1: Transition probabilities observed in the Booking.com data. We see a wide spread of transition probabilities, spanning several orders of magnitude.

requires nontrivial computation and run-time. As a result, this approach does not scale well, especially when there are many parameters to fit. For a case with even a moderate number of user groups, the path fitting process would be required to fit several hundred parameters which could take many hours to run. A more practical alternative is needed.

3.2 Scoped Markov model

The main idea for path-fitting is to use an auxiliary ad system model that does not allow ads to have a long term impact on user behavior. A Markov model is used to model user behavior, which describes transitions among both scoped and unscoped events. This model has a specified set of constrained transitions (e.g., a click must follow an impression). On top of the Markov model, we include scoping (i.e., event-hiding), and event grouping (discussed below).

Importantly, our 'scoped' Markov model should not be confused with a hidden Markov model (HMM). In an HMM, each hidden event in a path is associated with an observable event, which usually come from a different and distinct set of events. In this application, scoping refers to removing hidden/unobservable events from the path. The length of the observable path is shorter than the full path after unobservable events are removed.

3.2.1 Setup of the scoped Markov model

Each event, which may be in or out of scope, is represented as a state of a Markov chain. Explicitly, suppose there are k unobservable states $u_1, ..., u_k$ and n observable states $o_1, ..., o_n$. Consider the state-space to be the ordered set of events

$$S = \{u_1, ..., u_k, o_1, ..., o_n\}. \tag{1}$$

A user path always starts with a 'start of path' event, which is observable, and transitions at each discrete step to another state. The user path ends when the user reaches an 'end of path' state. Notice that each state is either observable or unobservable. The transition probabilities from each state are specified by a row of the *full transition matrix* [1] M. Specifically, if a user is at a state with index i, then row i of M indicates the probability distribution of the events in the next step. Let

$$M = \left[\begin{array}{c|c} M_{UU} & M_{UO} \\ \hline M_{OU} & M_{OO} \end{array}\right]. \tag{2}$$

Here we separate the transition matrix into four block components M_{UU}, M_{UO} M_{OU}, and M_{OO}, representing the transitions from unobservable states to other unobservable states, from unobservable states to observable states, and so on.

Next, we would like to find the transition matrix that describes user dynamics when applying the event scoping (i.e., unobservable events are removed from the path), which we will call the *observable transition matrix* $M_{\text{observable}}$. This is the first step we need to apply to the full model dynamics to arrive at statistics which are computed from only observable events.

[1] The 'end of path' is an absorbing state, and M excludes transitions to or from it (i.e., M represents the transient component, and is not a stochastic matrix). M is 'full' in the sense that it represents transitions including both observable and unobservable events.

To derive $M_{\text{observable}}$, we consider the possible ways to transition from one observable event to the next observable event. One way is to transition directly, while another is to transition through one or more unobservable events. The number of unobservable events is unknown, so we end up with an infinite sequence:

$$M_{\text{observable}} = \underbrace{M_{OO}}_{\substack{\text{direct}\\ \text{observable}\\ \text{transition}}} + \tag{3}$$

$$\underbrace{M_{OU}}_{\substack{\text{first}\\ \text{observable}\\ \text{to first}\\ \text{unobservable}}} \underbrace{(I + M_{UU} + M_{UU}^2 + ...)}_{\substack{\text{unobservable}\\ \text{transitions}}} \underbrace{M_{UO}}_{\substack{\text{last}\\ \text{unobservable}\\ \text{to next}\\ \text{observable}}} . \tag{4}$$

Summing the infinite series, we get

$$M_{\text{observable}} = M_{OU}(I - M_{UU})^{-1}M_{UO} + M_{OO}. \tag{5}$$

3.2.2 Constrained transitions and tunability

Here we describe a few implementation details that are helpful for fitting the model.

Our model configuration allows for some transitions to be 'tunable' and some to be 'fixed'. A tunable transition probability may change over the course of the optimization used in the path-fitting process, while a fixed transition does not. The initial value of each transition can be specified or chosen at random.

Tunable transitions are represented by their corresponding logit values to avoid imposing the positivity constraint directly. A row-normalization operation is also needed to ensure that the probabilities across each row of M always sum to 1 as tunable parameters change. Fixed transitions are removed from the normalization procedure to ensure they remain truly fixed.

3.2.3 Example with one user group

With the scoping operation, we are already equipped to set up an example with a single user group. The flowchart in Figure 2 represents the possible transitions among events for a single user group case. The observable events are shown in white nodes, and unobservable events in gray. While the full graph represents all transitions in the Markov chain, it can be transformed into another Markov chain with only observable events using the scoping operation. The scoped model will only have visible events, but importantly will still be parameterized using transitions from the full model.

3.2.4 Multiple User Groups

We have not been able to find a good model fit for a model with a single user group discussed in Section 3.2.3 (using our optimization process discussed below). When we compared the bigrams from the best-fit model with the original input data, the main source of mismatch seemed to be the repeated bigrams. One way to circumvent this would be adding additional transitions (e.g., from a click back to a visit of the same channel, representing repeated interaction with a channel after a click).

However, we find it more plausible that different users are simply more prone to interact more with specific channels. To capture this inherent difference in behavior across users, our model uses multiple user groups that are each allowed to have different underlying behavior prior to ad intervention, as described in Section 3.2.5. In reality, every user has a unique underlying behavior. So, this modeling choice is well justified.

3.2.5 Event grouping

In the auxiliary ad system model, it is necessary to combine multiple observable events before comparing bigram frequencies with the target. More precisely, suppose that two observable events o_i and o_j have transitions specified by distinct rows in M. It's possible that their observations in the data should be regarded as equivalent for the purposes of path matching.

For example, suppose the auxiliary ad system model tracks two separate sets of users, each of which is allowed to have a different baseline browsing behavior and/or a different response to ad interventions. While distinct 'search' and 'search click' states are used across these two user groups, the observable 'search click' states are not distinct when it comes to path matching comparisons with the target path data. The target path data does not differentiate search clicks across users. So, these search click events are grouped together before path matching comparisons are made. Analogous statements are true for the 'conversion' events across these two user groups.

One way to achieve this event grouping is to generate a large number of paths, generate groups of equivalent events, and then perform the required path matching comparison. While straightforward, the need to generate paths is computationally expensive, and does not allow us to easily compute gradients. It is more efficient to achieve this grouping directly from the transition matrix.

Applying the grouping operation for aggregating bigram statistics requires the relative weights of each

of the events that are to be combined (i.e., the overall proportion each constituent event contributes to the aggregate). One way to compute these probabilities is to find the fundamental matrix of the Markov chain with the 'end of path' event as an absorbing state:

$$F = (1 - M_{\text{observable}})^{-1}. \tag{6}$$

The ith row of F corresponds to the expected number of visits to each event starting from state i (see for example chapter 11.2 in [3]). Since the starting event is known, we can use the corresponding row of F, which we denote by F_{start}, to find the expected number of visits to each constituent state. Notice that, due to computational considerations, it may be easier to find the fundamental matrix of the full transition matrix M first, and then truncate the result to the observable state.

Suppose that after the grouping operation there are m remaining observable events. Define the grouping operator G as the $n \times m$ matrix whose rows one-hot encode how each observable event is mapped to a grouped event. Then the grouping operation is represented by

$$M_{\text{grouped}} = \text{norm}\left(G^T \text{diag}(F_{\text{start}}) M_{\text{observable}} G\right). \tag{7}$$

Here diag(.) is the diagonal matrix whose entries are the values in its input. The final row-normalization denoted by norm(.) is needed because the row F_{start} provides unigram expectations that have not been normalized.

3.3 Loss function

After grouping and row normalization, the observable transition matrix is ready to be compared to the target bigram rates. We use a slight variation of the Hellinger distance (see [6]) for matrices as a natural metric to construct our loss function, although a wide range of statistical distances may be reasonable choices. Specifically, we use

$$\begin{aligned} D_{\text{Hellinger}}(A, B) =& \|A^{\circ\frac{1}{2}} - B^{\circ\frac{1}{2}}\|_F \\ =& \sqrt{\sum_{i,j} (\sqrt{A_{i,j}} - \sqrt{B_{i,j}})^2}. \end{aligned} \tag{8}$$

Here the $\|.\|_F$ represents the Frobenius norm and $(.)^{\circ\frac{1}{2}}$ is the Hadamard (i.e., elementwise) square root.

3.4 Optimization algorithm

We wish to match the observed statistics (specifically bigrams in our case) to the ones generated by the model with parameter set θ. Denoting M_{data} as the bigrams from the dataset, and using M_{grouped} and $D_{\text{Hellinger}}$ from Eqs. 7 and 8, the optimization problem is

$$\underset{\theta}{\text{minimize}}\ D_{\text{Hellinger}}(M_{\text{data}}, M_{\text{grouped}}(\theta)). \tag{9}$$

The auxiliary ad system model is implemented in Tensorflow [1] for speed and access to well-known gradient-based optimization algorithms. Using our formulation, the gradient of the objective function can be found in Tensorflow using automatic differentiation. The Adam optimization [5] works very well, especially when it is used in multiple optimization stages with a decreasing learning rate. Typically, only a few hundred steps are enough to converge. This takes roughly a minute with around ten user groups and hundreds of model parameters to fit. The efficiency of this implementation could be improved further by utilizing the sparsity in M using a sparse solver (see Section 7.3).

4 Modeling Choices

For the auxiliary ad system model to be useful, it needs to have sensible structure and behavior. This includes making reasonable choices for out-of-scope events and permissible transitions.

4.1 Paid Channels on Third-party Site visits

Each non-search ad channel (i.e., any display or other ad channel) has its own unobservable state. A user can transition among these unobservable states or initiate searches. Each visit to one of these states activates a probability of being served an ad for the corresponding ad channel. This is reasonable because, in real life, two different display ad channels might each show ads on distinct sets of third party websites. There is also a separate unobservable impression state for each ad channel. Once an impression is served, the user also has the opportunity to click on the ad and transition to the advertiser's website, which generates an observable click state.

If the user does not click on the ad impression, then the user sees the same set of transition probabilities that would have been present without an ad impression. This behavior is achieved with an unobservable auxiliary state as depicted in the display mechanism component of Figure 2.

This modeling choice also allows us to directly map parameters to DASS, where base transitions in the absence of ads and ad effects are specified separately. The probability of an ad being impressed is

the 'share of voice' of the ad, and the probability of clicking an impressed ad is the 'click-through rate'. When an ad is impressed, the new output click event is added to the output distribution, and the remaining output probabilities are normalized. To see the correspondence between the two models, notice that in DASS the output probabilities given a served ad conditioned on not clicking the ad is equal to the original output probabilities. In the scoped Markov model these two output distributions (no impression, or impression conditioned on no click) are tied using the auxiliary state.

4.2 Search states

Search states are similar to third-party visit states, except that the user can also make a transition to an unpaid visit (SEO). See the search mechanism component of Figure 2.

4.3 Conversions

A conversion can be reached either immediately after the start of the path (either a user navigated in an unknown way to the advertiser's website, or the initial part of their path was truncated), after a paid or unpaid click, or after a previous conversion.

4.4 Consistent user behavior

The behavior of a user should not drastically and arbitrarily change over the course of his/her path. For example, if the transitions from the 'start of path' event and the transitions from an advertiser site visit event are independent then user behavior may be very different at the very beginning of a path than it is later in the path[2]. To prevent erratic changes in user behavior, we need to link the transitions from states we expect should be similar. We do this by including an unobserved generic 'website_visits' state that is reached both after 'start of path' as well as after a user reaches the site via either a paid or unpaid click, or after a conversion (see Figure 2).

4.5 Paid ad click-through rates

The paid ad click-through rate (CTR) is a fixed parameter in the model. This choice is justified since CTR can be an observable quantity for real ad campaigns. For the results shown below, the search CTR is set to 0.1, which is high relative to most real campaigns. If the CTR is set too high, an experiment where the channel is turned off may amplify other small output transition probabilities resulting in unreasonable channel interaction.

4.6 Channel decoupling

Running a simulator experiment that removes one ad channel can have unintended consequences on another ad channel. For example, consider the case with two channels 'A' and 'B', each having a probability of ending the path. If we remove 'A', we may see more instances of 'B' in each given path to compensate for the loss of a chance to end the path from 'A'. Similarly, including an ad effect that increases the number of unobserved 'third party visits' associated with one ad channel with a higher 'end of path' transition probability may result in a decrease in the number of 'third party visits' associated with other ad channels. Removing or altering a particular channel's effectiveness should have little direct consequences to other unobserved user activities. The fundamental disconnect between the model configuration and reality in these cases is that the user spends the majority of time disengaged from the advertiser.

We avoid this modeling issue by introducing an 'disengaged' state that can transition to or from third party visits and searches. Importantly, the only way to reach the 'end of path' is through an disengaged state. The probability of transitioning from a third party site visit or a search to a disengaged state is, somewhat arbitrarily, fixed at 0.5 during the model fitting step. For scenarios in which the impact of one ad channel increases the activity in another (e.g., a display impression increasing the probability of a search), the overall transition probabilities to the disengaged state naturally decrease since the output probabilities from each event must be normalized. The net result is that this additional state eliminates the undesirable properties described above, since increased activity in affected ad channels does not compete with other ad channels.

4.7 Summary of events

For this particular application, the out-of-scope events are

- website visits
- Third-party (non-advertiser) site visits
- Ad impressions
- The auxiliary state associated with each paid ad channel
- Disengaged state

[2]This situation differs from the modeling of long-lasting effects of advertising, which could change user behavior across a path but is not part of the auxiliary model.

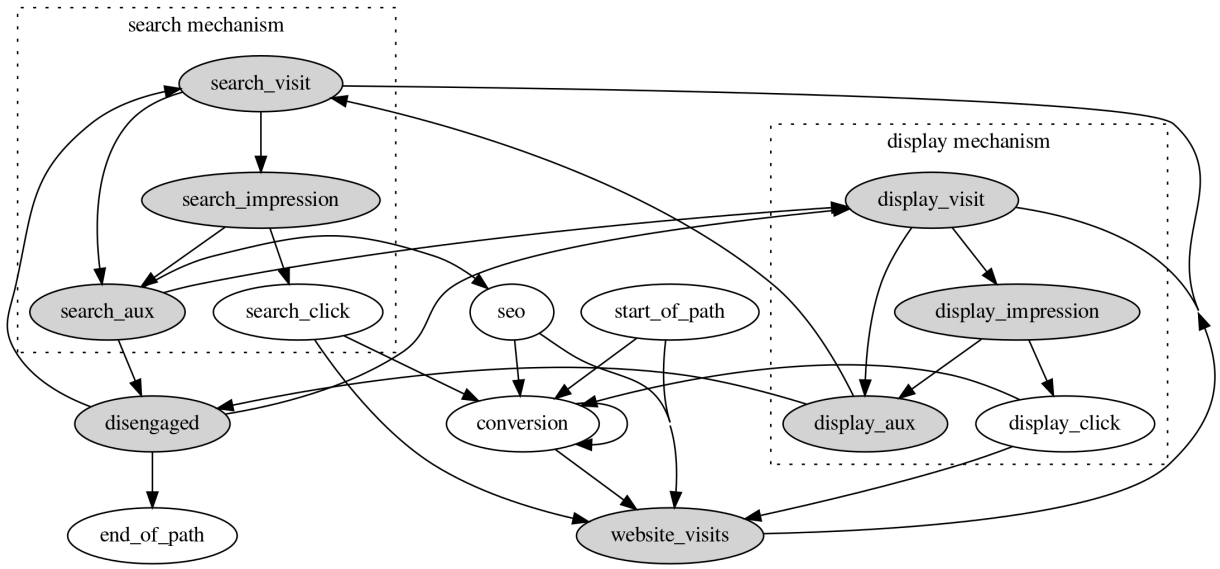


Figure 2: This flowchart represents the allowed transitions for a single user in the ad system model. The grayed-out nodes represent unobservable events. Representative search and display channels are shown. Advertiser path data may include multiple search, display, and other ad channels (not shown), which are modeled similarly.

Figure 2 depicts the model configuration used to generate the results shared in the following sections. To simplify the presentation, only a single search and a single display channel are included in this flowchart.

4.8 Multiple user groups

As discussed in Section 2.2, the model that we use for path fitting includes multiple groups of users. Each user group is allowed to have its own set of baseline behavior. To easily fit these into our framework, we use a single 'start of path' state that can transition to a separate 'start of path' state for each user group. These secondary 'start of path' transition probabilities correspond to the mixture probabilities (i.e., the proportions of each user group) and are determined as part of the path matching process. Aside from this common starting event, the user groups are disjoint copies with different randomly initialized transition probabilities.

5 Path-Fitting Results

This section describes the quality of fit for Booking.com path data. Appendix A describes the quality of fit for a set of fabricated data that includes more explicit bigram matching results.

5.1 Increasing the number of user groups

The ad system model allows for the use of multiple user groups. This generalization is necessary because a single user group does not provide a quality path match when there is a high prevalence of repeated bigrams, as is the case for the target path data (see Section 2.2). As the number of user groups increases, the fitting error decreases rapidly before eventually plateauing (see Figure 3). For our set of target path data, nine user groups is sufficient to allow for a quality path match.

Increasing the number of user groups provides the ad system model with additional degrees of freedom. As noted previously, each real path is generated by a unique user. So, allowing for nine different types of users is quite reasonable. While it's not surprising that more user groups leads to better path matching, adding arbitrary flexibility isn't necessarily helpful and it provides an additional burden on the optimization to fit additional parameters.

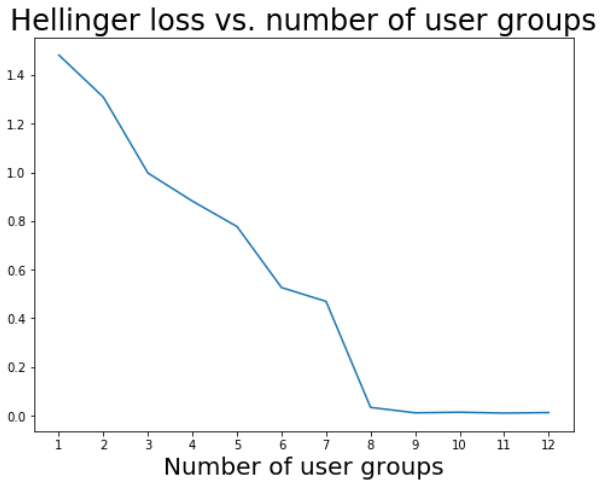


Figure 3: The loss of the fitted model as a function of the number of user groups decreases until the number of user groups is sufficiently large, at which point it plateaus at a small value. Intuitively, the 'knee' of this plot corresponds to the number of user groups needed to adequately represent the diversity of user behavior.

5.2 Bigram match error

The path matching results for Booking.com path data are shown in Figure 4. Figure 4a shows the distribution of absolute errors for the observable bigrams. These errors are the absolute value of the difference between each bigram transition rate generated by the fitted ad system model and the bigrams in the target path data. All of the transition probability errors are reasonable small.

Figure 4b shows the relative error for the fit and observed transition probabilities. The outliers in the plot correspond to transitions in a rarely visited channel, which is of no consequence for the post-matching analysis described below.

Note that to avoid sharing proprietary information related to Booking.com path data, these fitted results do not show the bigram probability fit more directly. See Appendix B to see this comparison for fabricated target data.

5.3 Path lengths

Because the path length is a function of the bigrams being fit (the aggregation step preserves expected path lengths), the average path length closely matches the real data (relative error of less than 0.1%). However, since the objective function only includes bigram rate matching, the model does not fully characterize the distribution of path lengths, especially the long-tailed behavior of the path data.

6 Attribution Analysis with Matched Paths

In this section we use the fitted model to generate several attribution model unit tests, which we apply to three position-based models; last interaction - all credit for a conversion is assigned to the event that immediately precedes the conversion, first interaction - all credit is assigned to the earliest event in the conversion path, and linear - credit is divided equally among the events in the converting path. The analysis is completely analogous to the one described in [9].

A 'scenario' is defined as the specification of a single set of DASS parameters. In our case, this is the fitted model. A 'scenario family' corresponds to multiple sets of closely related parameter specifications. These specifications differ in the value assigned to a single parameter that changes the magnitude of the impact of a single ad channel. Most often, the parameter that is varied changes the magnitude of the ad. A scenario family can be used to determine an attribution model's ability to detect changes in ad effectiveness for different mechanisms of ad impact.

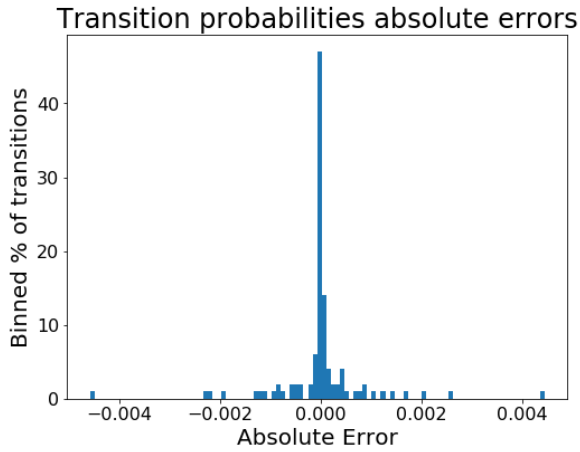
For each scenario in a scenario family, the ground truth impact of an ad channel on the conversion volume is found using an experiment that is run within the ad system simulator. The credit attributed by an MTA model to the target ad channel is compared to this ground truth, as shown in Figure 5. The experiment involves comparing the number of conversions with the ad channel turned on versus off. Turning off the ad doesn't mean removing the underlying third party of search state – instead it means setting the transition to the corresponding impressed state to zero.

Recall that the fitted model does not include ad effects that last beyond the ad impression by construction as a trade-off for efficient model fitting. However, after fitting the model and mapping the parameter values to the DASS model parameters, it is possible to introduce downstream ad effects without limitation.

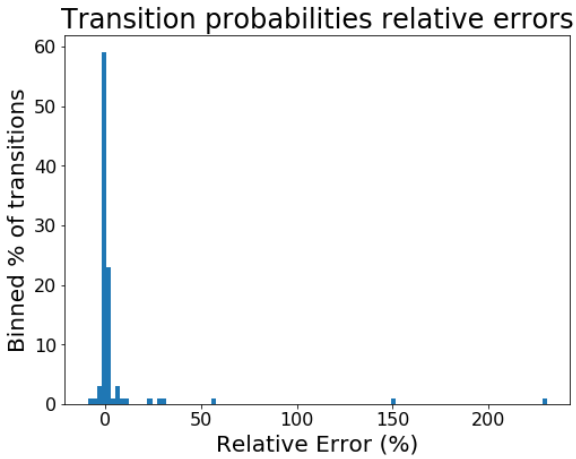
6.1 Search click-through rate

In the unit test corresponding to this first family of scenarios, the click-through rate (CTR) of a search channel is varied across the scenarios. A search ad impacts user behavior by increasing the probability of a visit to the advertiser's website by providing the opportunity for a paid click.

To implement this scenario, the transition probability from an impressed state to its corresponding click is increased. Recall the impression state may transition to either a click state or an auxiliary state



(a) The absolute errors are defined as the difference $T_{\mathrm{m}} - T_{\mathrm{o}}$ between each transition probability generated by our model T_{m} in the fitting process and the corresponding observed transition rate T_{o}. All transition probability errors are all reasonably small.



(b) Each relative error is defined as $\frac{T_{\mathrm{m}} - T_{\mathrm{o}}}{T_{\mathrm{o}}}$ for observed and modeled transition probabilities T_{o} and T_{m} respectively. There are some outliers (relative error greater than 10% or so). These outliers all associated with transitions to an extremely rare channel and are not important for our application.

Figure 4: The absolute and relative errors for the matched path data. In general, the choice of loss function will determine the importance of relative versus absolute errors.

for the corresponding channel (Figure 2). As a result, the transition probability to the auxiliary state must be decreased.

The last interaction attribution model does very well in this situation (see Figure 5a). The corresponding curve nearly matches the ground truth curve. This is expected, since the implicit assumptions of this model are satisfied: ad impact (generation of an incremental site visit) can only occur with a click and there is no effect beyond the point of the click, the click is observable (in-scope) for the path data, and no observable event can appear between the click and a conversion generated by the click. The first interaction and linear models aren't as effective in recognizing or responding to changes in magnitude (CTR) for this type of ad impact. However, they still track the ground truth curve reasonably well. These results are consistent with the findings in Figure 5 of [8].

6.2 Permanent click effect

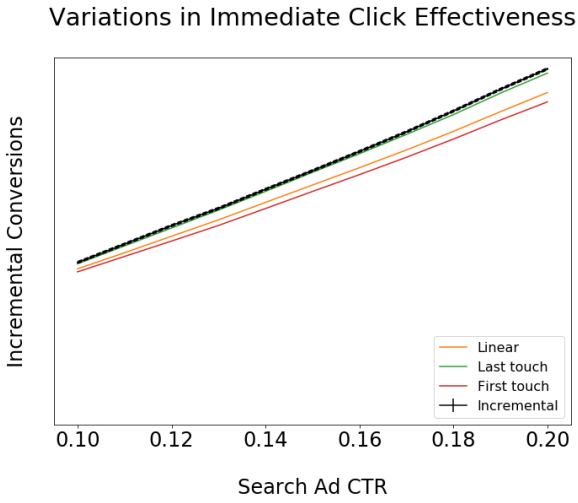
The purpose of this next family of scenarios is to understand how MTA models identify the downstream impact of a click on a search ad. As was the case for the first scenario family, a search ad impacts user behavior by increasing the probability of a visit to the advertiser's website by providing the opportunity for a paid click. Additionally, a paid search click for one of the search channels results in an increased downstream probability of another search ad click, as well as an increased probability of an unpaid click (SEO). So, a search ad can modify the behavior of a user beyond the click-through visit to the advertiser's website. This type of ad effectiveness is relevant to so called 'generic search ads' that generate awareness and sustained interest in an advertiser without immediately generating a conversion.

To implement this scenario, a user's probability of transitioning to a search or SEO event is scaled by a factor of $1 + \epsilon$ after each occurrence of a click on the relevant search channel. Here, ϵ is the parameter that scales ad effectiveness (i.e., the x-axis in Figure 5b). In a follow-up step, each row of the transition matrix is re-normalized. This scaling and re-normalization remains in effect for all subsequent event generation for this user.

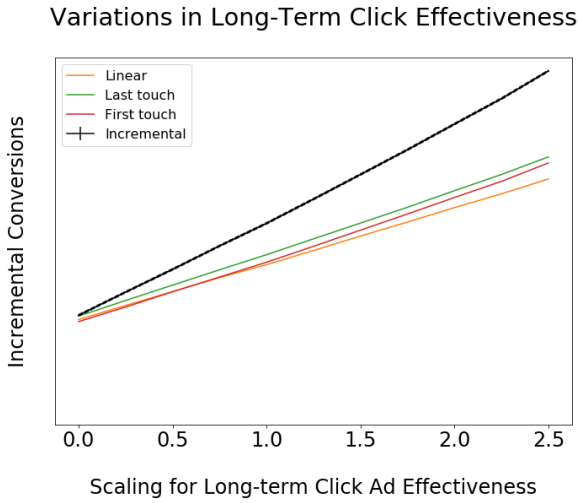
Figure 5b shows results for this family of scenarios. The difference between the results of these attribution models and the ground truth for incremental conversions grows as the downstream impact from a search ad click grows. None of these simple attribution models are able to effectively interpret this type of ad effectiveness.

6.3 Display impression effect on search

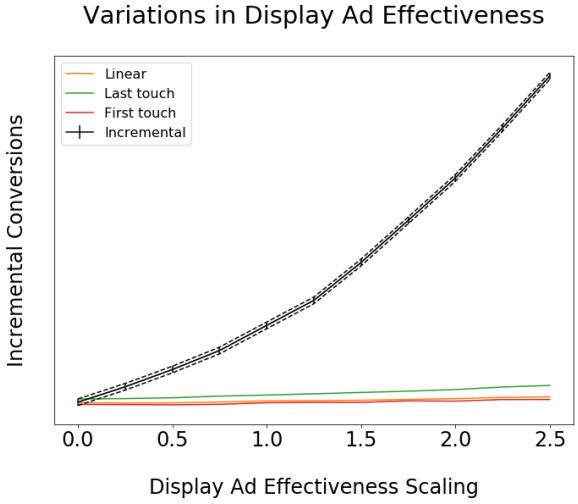
In this family of scenarios a display impression has a downstream impact on search activity. Specifically,



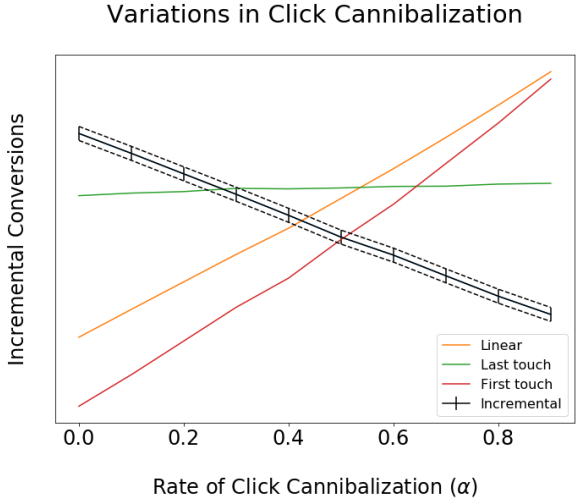
(a) A search ad impacts user behavior by increasing the probability of a visit to the advertiser's website via a paid click. The click-through rate (CTR) of a search channel is varied from 0.1 to 0.2. The last interaction attribution model is the most successful of the three models at matching the ground truth incremental conversions generated by this search channel.



(b) A search ad impressions gives the user an opportunity to visit the advertiser's website via a search click, and the search click increases the probability that the user will do an advertiser-related search downstream. All three of the attribution models struggle to recognize the full value of the downstream ad impact.



(c) Exposure to an display ad impression increases the probability that the user will do an advertiser-related search downstream. None of the attribution models are able to recognize this type of ad effectiveness because the path data they use as input does not include display ad impressions



(d) The probability of an unpaid search click (SEO) is decreased conditional on a search ad impression. This models a situation in which unpaid search clicks are cannibalized by paid search ad clicks (see Section 6.4 for full details). The volume of incremental clicks generated by a paid channel decreases as the rate of cannibalization increases.

Figure 5: Attribution unit test results for four different type of change in ad effectiveness. For each degree of change in ad effectiveness, the ground truth is found by running an experiment in the simulated environment in which the target ad channel is on versus off.

an impression generated by one of the display channels increases the probability that the user exposed to the ad will do an advertiser-related search in their downstream activity

This ad impact is implemented by introducing the scaling factor $1 + \epsilon$ where $\epsilon > 0$. When a user visits a third party search state and is exposed to a corresponding display ad, every transition probability to a search state for that user is scaled by this factor. In a follow-up step, each row of the transition matrix is re-normalized. This scaling and re-normalization remains in effect for all subsequent event generation for this user.

None of the three attribution models are able to recognize this type of ad effectiveness. All three models are completely insensitive to the magnitude of ad effectiveness (see Figure 5c). The primary reason for this failure is that display impressions are out-of-scope, i.e., they are not included in the observable path data. In this situation, the path data provided to the attribution models does not include the information needed to accurately estimate ad impact. No attribution will perform well with this observable path data. This result is very similar to the one shown in Figure 6 of [8] in which out-of-scope search ad impressions have a downstream impact on user behavior.

6.4 Paid search cannibalization

The potential cannibalization of unpaid search clicks by paid search clicks is a common concern for advertisers. This occurs when a user does a search and the results page includes the opportunity to click on a paid search ad and the opportunity to click on an unpaid search result. A user might click on the paid search ad even though he/she would have clicked on the unpaid search result had the search ad not been shown. In this situation, the advertiser pays for a visit that it could have been obtained for free.

In this scenario, we simulate the effect of cannibalization by decreasing the probability of transitioning from a search state to an unpaid search click (SEO) when a search ad impression is shown. That is, we hypothesize that some fraction of the observed paid clicks were really replacements for unpaid clicks that would have occurred if the search ad had not been shown. The cannibalization rate is adjusted by varying a single parameter, α, across all user groups from 0 (no cannibalization) to 1 (maximum cannibalization). More specifically, α is used across each user group in one of two ways:

- If the original probability of an unpaid click, U, is larger than that of a paid click, P, the maximum cannibalization rate (i.e., loss of unpaid click probability) is P. More generally, the probability of an unpaid click after a search ad impression is $U - \alpha P$, which allows the cannibalized unpaid click probability to vary from U (no cannibalization) to $U - P$ (maximum cannibalization).
- If U is smaller than P, the maximum cannibalization rate is the entirety of the unpaid click probability U. In this case, the probability of an unpaid click after a search ad impression is $(1-\alpha)U$, which can vary from U (no cannibalization) to 0 (maximum cannibalization).

Figure 5d shows the number of incremental conversions generated by paid search as a function of α. As the cannibalization rate increases, the number of ground truth incremental conversions decreases. Note that the CTR for paid visits does not change across scenarios in this plot. As a result, the volume of paid visits remains relatively constant. However, as cannibalization increases, more and more of these paid visits are actually replacements for visits via SEO clicks. So, when search ads are turned off to measure their incrementality, fewer and fewer site visits and conversion opportunities are lost.

In this scenario, no attribution model matches the decrease in incremental conversions of the ground truth curve. The last event curve remains relatively flat with increasing cannibalization. This happens because the main impact of cannibalization in this scenario family is to decrease the number of unpaid visits. The number of paid visits does not change, and therefore the number of paths with paid visits that are immediately followed by a conversion does not change. So, the last event allocation of conversion credit does not change either.

On the other hand, the first event and linear models give progressively more credit to the paid search channel. Paid and unpaid search clicks are strongly correlated in the observed bigrams of the target path data. There is also a degree of asymmetry – the transition probability from unpaid to paid clicks is significantly higher than vice versa. So, when increasing cannibalization removes unpaid clicks that are upstream in these converting paths (and moreover this effect is stronger than the reverse effect, switching paid and unpaid clicks), more credit is allocated to the paid ad channel.

This idea is supported by a plot of the conversion credit attributed to the unpaid click channel (see Figure 6). When the cannibalization rate is zero, the linear and first-touch attribution models assign more credit to the unpaid click channel. As the cannibalization rate increases, the removal of unpaid events

from converting paths affects the credit assigned by the first event and linear models much more than the last event model.

While it is not possible to determine the cannibalization rate in general without an experiment due to unknown covariates, an attribution model should minimally be able to compare unpaid click rates with a search impression versus without.

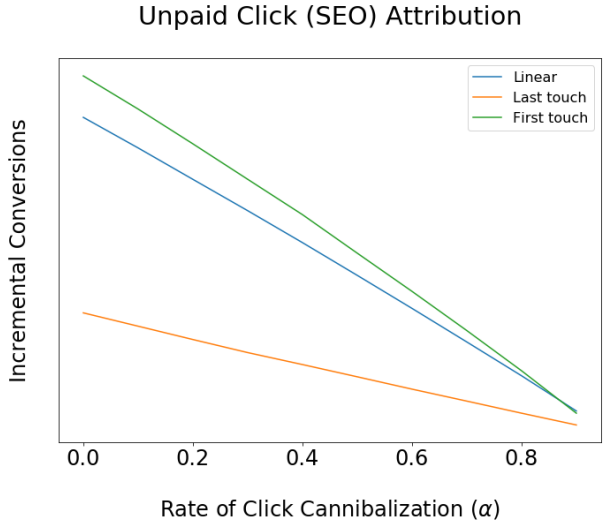


Figure 6: The conversion credit attributed to the unpaid search channel across cannibalization rates for real data.

7 Conclusion

7.1 Summary

In this paper, we describe and demonstrate a process for generating user level path data that matches the characteristics of path data for a particular advertiser. Here, the objective is to match the channels, observable events, and the frequencies of event bigrams. Although, in principle, it's possible to generalize this notion of matching to include additional characteristics (e.g., trigrams or the distribution of path lengths). Path matching is achieved by solving an optimization that identifies the input parameters for an ad system simulation. This minimizes the difference between the bigram frequencies for path data generated by a candidate set of input parameters and the target bigram frequencies from the advertiser path data.

Once the appropriate parameters have been identified for the ad system simulation, it is possible to generate unit tests that can be used to better understand the capabilities and limitations of MTA models. Because the simulated data are more specific to the advertiser than generic path data, the results of these unit tests are more relevant and actionable to the advertiser. We explored a few example scenarios that are illustrative of situations that may be of concern to advertisers that use MTA models.

7.2 Limitations

The path matching process described in this paper has several limitations. First, and most importantly, the matching process cannot be used to directly determine the level of ad effectiveness of any ad channel from the advertiser path data. The observable data is not sufficient to do this. Many user covariates aren't observable, many complexities aren't represented in the ad system model, and the components and true functional form of ad effectiveness are unknown.

The deficiency of information in the user level path data leads to another limitation. There is more than one set of ad system parameters that match the path data. In the optimization process, a different initial guess can lead to a different path matching solution. Although our testing found that the unit test results are generally robust to different initial guesses, there is no guarantee that this will always be the case.

Finally, only the base level of ad effectiveness (i.e., the leftmost data point in Figure 5) corresponds to simulated path data that matches the target path data. This is partly because it is not possible to include long-term ad effects in the auxiliary ad system model used to match the advertiser path data. A more general auxiliary model would not have this limitation (see Section 7.3).

7.3 Future work

In this section we discuss how the work in this paper can be extended and how shortcomings can be addressed.

7.3.1 Long-term ad effects

One very notable limitation of the approach described above is that the model fitting step assumes stationary Markovian behavior. This does not allow ads to impact the downstream behavior of users beyond the action that immediately follows a search impression. It would be useful to fit an ad system model with fixed non-stationary behavior instead of adding this behavior as a separate follow-up step. For example, this capability would be useful in studying the extent to which ad effects are identifiable for a particular set of path data. That is, we want to know if many different types and magnitudes of ad effectiveness can lead to the same observable path data by changing unobservable user behavior.

7.3.2 Time component

In [9] a method is introduced to include the decay of ad impact across time without drastically increasing the size of the state-space. Specifically, a 'unengaged' state is introduced, and the passage of time is captured by transitions between engaged and disengaged states. Such a method can be incorporated into the current methodology as well. So, along with the inclusion of long-term ad impact, the fitting process could also incorporate a long-term decay of ad impact, as long as the target path data includes time stamps.

7.3.3 Sparsity

When the ad system model includes multiple user groups the full Markov model transition matrix is very sparse. Performance can likely be improved by performing the scoping and grouping using efficient sparse operators. In particular, the matrix inverses in Eq. 5 and Eq. 6 do not have to be generated explicitly. Instead, it is possible to solve a number of linear equations using methods like GMRES or BiCGSTAB (for example, see chapter 11 in [2]).

7.3.4 Other objectives

If the target data includes additional aggregate path statistics then objectives beyond bigram frequency can be used in the matching optimization. For example, matching trigram frequency may better capture the evolution of user behavior. Also, matching the distribution of path lengths would better capture the long-tail of user behavior.

For fitting in the most general case in which the target data includes a complete set of user level path data, it should be possible to use a maximum likelihood estimate (MLE) to find an optimal set of ad system parameters. This possibility is appealing, but the use of individual user paths would require a step to preserve user privacy (e.g., removing low volume paths from the analysis).

Acknowledgments

We would like to thank our colleagues at Google who helped improve this work. These include Mike Perry, Jim Koehler, Penny Chu, Tony Fagan, Georg Goerg, and David Chan.

Appendix A

Generation of Fabricated Target Data

The goal is to generate bigram data that is qualitatively similar to that of real user path data, including data from Booking.com. The main features of the data are:

- High probabilities for repeated events (diagonal terms in the transition matrix).
- A significant degree of symmetry in the transition matrix.
- High imbalance across the frequencies of events.
- High probability of EOP ('end of path') after most events.

To duplicate these characteristics we use the following process. For clarity, we use an index for the D matrices, but in practice updates can be done in place:

1. Generate two matrices D_{sym} and D_{asym} by sampling elements from the uniform distribution $U(\alpha_{\text{floor}}, 1)$ and compute

$$D_1 = \frac{D_{\text{sym}} + D_{\text{asym}}}{2} + \alpha_{\text{asym}} D_{\text{asym}}. \quad (10)$$

   The parameter $\alpha_{\text{asym}} \geq 0$ controls the degree of asymmetry, and the parameter $0 \leq \alpha_{\text{floor}} \leq 1$ controls how widely the distribution can vary.

2. Multiply the columns of D_1 by a power distribution with parameter $\alpha_{\text{power}} > 0$, forming D_2. This ensures some events are much more likely to occur than others.

3. Compute $D_3 = D_2 + \alpha_{\text{diag}} \text{diag}(V_d)$ where V_d is a vector with elements sampled from $U(\alpha_{\text{floor}}, 1)$ and $\alpha_{\text{diag}} \geq 0$. This ensures diagonal terms are significantly larger than non-diagonal terms.

4. Add $\alpha_{\text{EOP}} \geq 0$ to the column of D_3 representing transitions to the EOP, forming D_4. This makes events more likely to transition to EOP.

5. Set specific entries of D_4 to zero, forming D_5:
   - Start of path to EOP (this would be an empty path).
   - All event transitions to the 'start of path' event.

- EOP to all other events.

6. Normalize the rows of D_5 to form D. This step ensures that D is a valid transition matrix.

Typically, we want specific types of events to occur more or less frequently than others (e.g., conversion is a low frequency event). To meet this requirement, unigram frequencies are computed using D and then the events associated with the columns of D are reassigned accordingly.

Appendix B

Fabricated Data Results

This appendix includes path matching and attribution analysis results generated using bigram data fabricated using the process described in Appendix A. This fabricated bigram data is analogous to the real data from Booking.com. The target data includes 8 observable events (not including the start and end of path). The parameters used to generate the fabricated data are:

Parameter	Value
α_{power}	0.6
α_{diag}	3.0
α_{EOP}	1.0
α_{asym}	0.3
α_{floor}	0.2

Figure 7 contains the distribution of transitions in the fabricated data generated using the process described in Appendix A. This distribution is qualitatively similar to that of the Booking.com data (compare to Figure 1).

Figure 8 shows the distribution of relative and absolute errors for the transition probabilities of the fitted model compared with the fabricated target transitions. The errors are somewhat larger than those of the fit generated using the target data from Booking.com with the exception of the outliers in the relative errors in Figure 8 (which are due to the presence of a rarely visited event in the Booking.com data). These errors do not decrease significantly with an increasing number of user groups, which suggests that other characteristics of the underlying model are the limiting factor.

Figure 9a shows the distribution of unigrams in the path matched data generated using the fabricated target data. The unigrams are computed from the transition probabilities in each case using the fundamental matrix of the (observable) Markov transitions, and compared to the original fabricated transitions D. Figure 9b shows the distribution of bigram transitions from one sample observable event to the other observable events. Notice the procedure outlined in Appendix A produces a large proportion of repeated bigrams (in this sample, transitions from dsp2 to dsp2), as desired. In addition, there is a large difference between the frequency of some unigrams and bigrams, which also matches the desired behavior.

Figure 10 shows the results of the attribution unit tests. The procedure applied to the fabricated data was exactly analogous to the one applied to the path data matched to the Booking.com target. Both sets of results are qualitatively similar in the ability of these attribution models to track the ground truth curve, although there are a few differences. For example, in Figure 10a the curve corresponding to last-touch attribution does not match the ground truth curve as closely as it does in Figure 5a. The slopes are very similar, but there is an offset. This discrepancy is most likely due to channel interaction that is enhanced with this particular set of target data.

In Figure 10d the first touch and linear attribution models are relatively insensitive to the cannibalization of unpaid clicks, whereas in Figure 5d these models assign progressively more credit to the paid search channel. The first touch and linear attribution models also assign less credit to unpaid clicks than last touch before including cannibalization, so there is less credit to shift from unpaid to paid clicks with increasing cannibalization (compare Figure 6 and Figure 11). This is partly due to the unpaid and paid click channels being less strongly correlated and occurring less frequently together in converting paths. Moreover, the transition rates from paid to unpaid click events versus unpaid to paid click events are much closer to equal (see the discussion in Section 6.4). We intentionally made the fabricated data bigrams approximately symmetric, since we observed some general symmetry in the real data, but the transitions between paid and unpaid clicks in this case were less symmetric in the real data than the fabricated data.

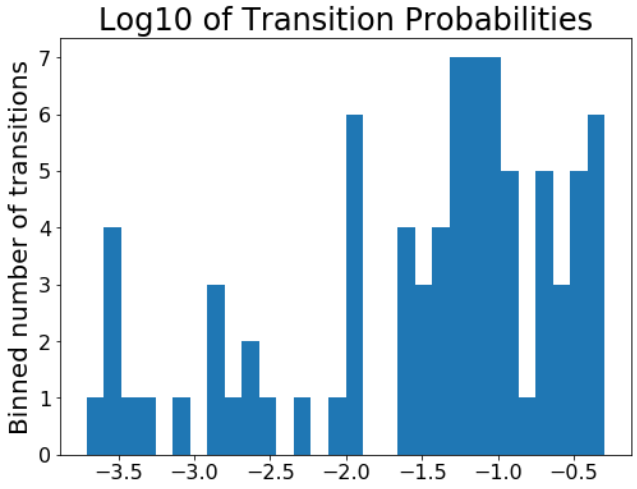
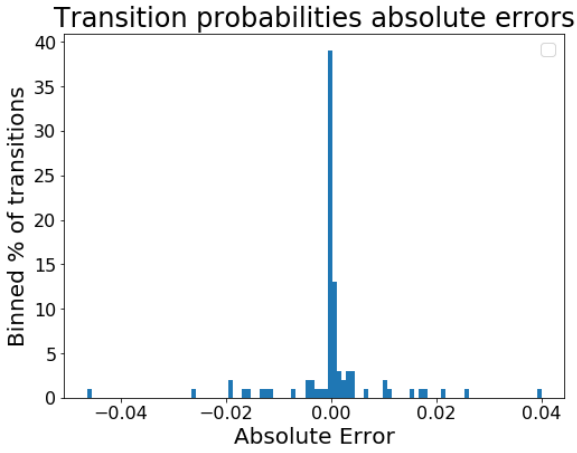
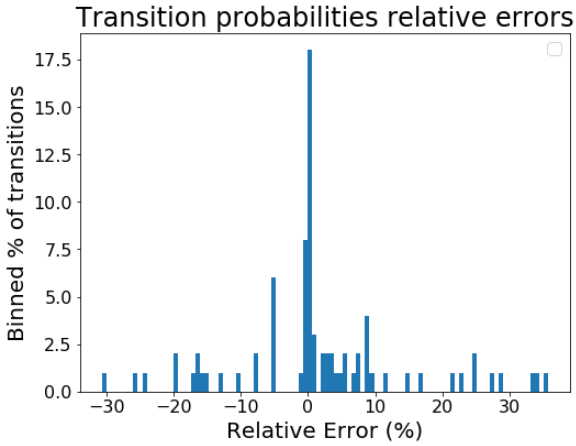


Figure 7: The transition rates in the fabricated target data are similar to those of the real data (compare to Figure 1) in that they have a 'tail' of low-probability transitions. In the fabricated data, this is a result of using a power distribution for the overall prevalence of particular observable events.

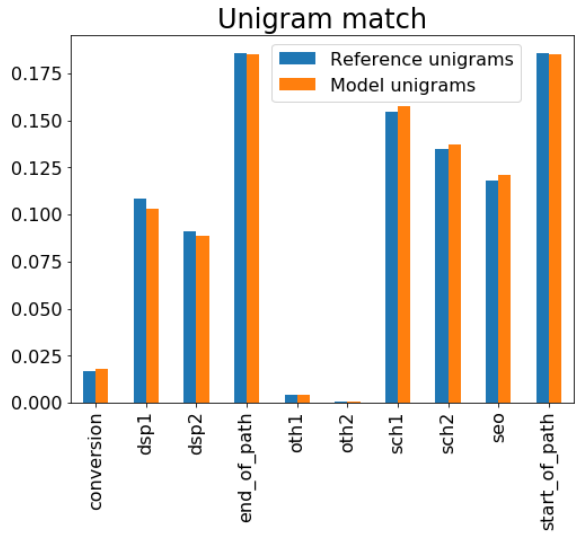


(a) The absolute errors of the transition probabilities for the matched path data are generally small, although not as small as those from matching to the Booking.com data (compare to Figure 4a).

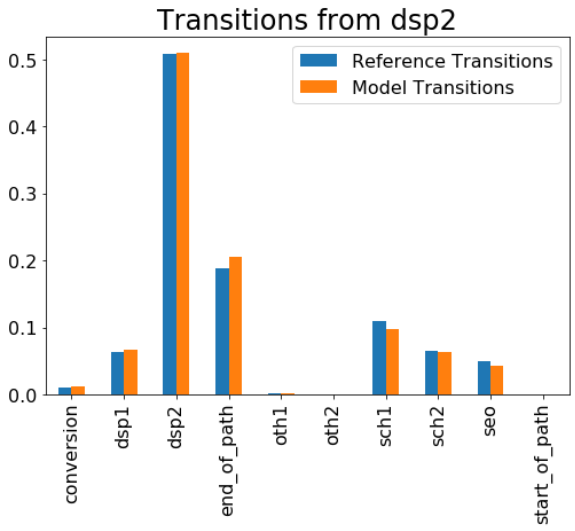


(b) The relative transition errors of the transition probabilities for the matched path data are generally small (with some exceptions). These errors do not include the large outliers observed in the match to Booking.com data (compare to Figure 4b). This may be because the rare events were less rare in the fabricated data (compare Figures 1 and 7).

Figure 8: The absolute and relative errors in the transition rates of the matched path data compared to the fabricated target transition rates.



(a) The unigram rates for the fabricated target data and the matched path data.



(b) The reference transition probabilities from one sample observable event (dsp2, i.e., an ad click for the dsp2 channel) to every other observable event for the fabricated target data and the matched path data.

Figure 9: Unigram rates and sample bigram transition probabilities. We show both the target results for the fabricated target bigrams and the matched path data. The events shown are the observable (i.e., in-scope) events: 'start of path', 'end of path', conversion, and click events for each paid channel.

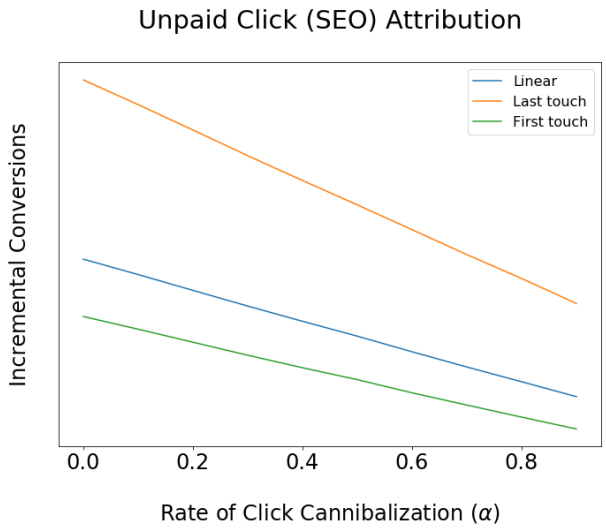


Figure 11: The conversion credit attributed to the unpaid search channel across cannibalization rates for fabricated data.

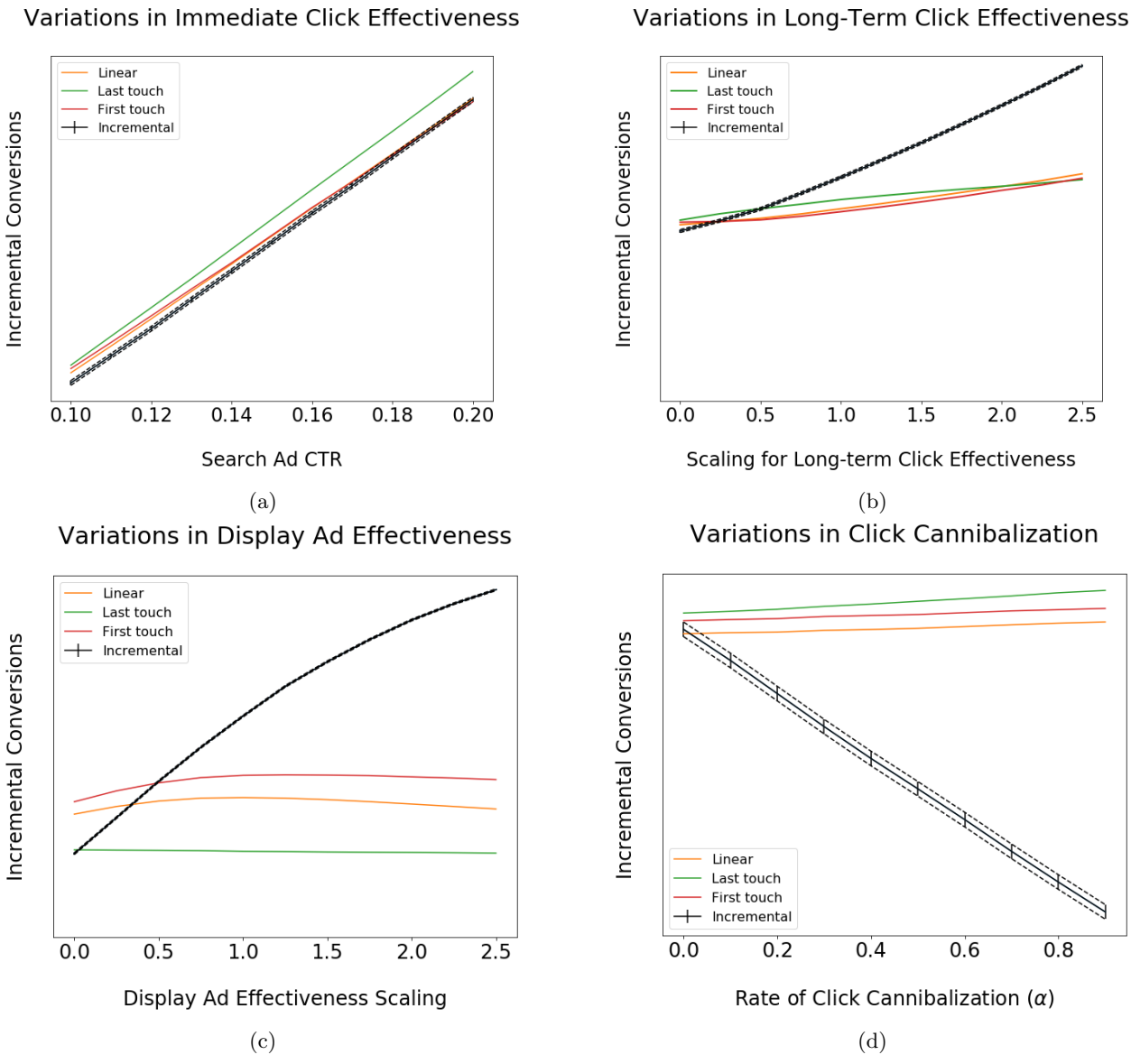


Figure 10: Attribution unit test results using path data matched to the fabricated target data. These results are analogous to those for the Booking.com data (shown in Figure 5).

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] G Golub and C Van Loan. Matrix computations, 4th edn. johns hopkins, 2013.

[3] Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012.

[4] Joseph Kelly, Jon Vaver, and Jim Koehler. A causal framework for digital attribution. Technical report, Google LLC, 2018. `https://research.google/pubs/pub46905/`.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] Mikhail S Nikulin et al. Hellinger distance. *Encyclopedia of mathematics*, 78, 2001.

[7] Stephanie Sapp and Jon Vaver. Toward improving digital attribution model accuracy. Technical report, Google Inc., 2016. `https://research.google/pubs/pub45766/`.

[8] Stephanie Sapp, Jon Vaver, Minghui Shi, and Neil Bathia. Dass: Digital advertising system simulation. Technical report, Google Inc., 2016. `https://research.google/pubs/pub45331/`.

[9] Kyra Singh, Jon Vaver, Richard E. Little, and Rachel Fan. Attribution model evaluation. Technical report, Google LLC, 2018. `https://research.google/pubs/pub46901/`.