A stochastic approach to multi channel attribution in online advertising

Authors
Venkatesh Sreedharan (venkateshw@freshworks.com)
Swaminathan Padmanabhan (swami@freshworks.com)
Machine Learning @Freshworks

Abstract

In digital advertising, Multi-channel attribution is the process of determining the set of events that have a causal influence on a user's conversion, e.g. signing up for an online service, or performing an e-commerce transaction. It also entails allocation of proportionate credit to each event along the funnel leading up to conversion. Attributing value to only a few of these customer interactions (events) can result in disproportionate allocation of ad campaign budgets, wherein the online marketer might de-prioritize channels that do not have obvious causal value. As per a recent Forrester report[1] on cross channel attribution, around 84% of the marketers feel the need to adapt their media buying patterns based on insights from multi-channel attribution. Wikipedia defines channel attribution as 'the identification of a set of user actions ("events" or "touch points") that contribute in some manner to a desired outcome, and then the assignment of a value to each of these events.' In this white paper, we describe an algorithm for Multi channel attribution that was conceived and developed at Freshworks.

Freshworks offers a portfolio of SaaS products that help businesses optimize their customer engagement workflows; encompassing customer support, marketing, sales and workforce management. Freshdesk, which is our flagship customer support product, leverages multiple online advertising channels to reach out to prospective business customers. These include Google/Bing search, Facebook, LinkedIn, Display ads on targeted sites (e.g. helpdesk review sites), re-targeting and mobile. A prospective Freshdesk user can therefore experience different forms of advertising over a period of time, as he/she moves down our conversion funnel. For instance, consider the following path followed by user *A* leading upto conversion, i.e. signing up for a Freshdesk account.

[1] Forrester report on cross channel attribution:
https://kenshoo.com/wp-content/uploads/2014/09/Cross-Channel-Attribution-Must-Convert-Insight-Into-Action-Forrester-Consulting-TLP.pdf

A path describes the sequence of advertising interactions[2] experienced by the user, without reference to the time interval between events (or touchpoints).

A: (start)
→ Google_adclick (search_term=”helpdesk software”)
→ Bing_organic (search_term=”freshdesk reviews”)
→ Display_retargeting_adclick (Yahoo! news)
→ Direct web visit
→ (Conversion)

In this scenario, user A first conducts a Google web search using a search term that matches one of Freshdesk’s campaign[3] keywords, clicks on our sponsored search result (ad) and visits the corresponding ad landing page (e.g. www.freshdesk.com/signup), but does not sign up for a trial account. At a later point in time, the same user comes back to the Freshdesk domain via Bing Search, however this time by clicking on an organic (non-sponsored) search result shown by Bing. But once again, the user leaves without signing up. Soon thereafter, the user is shown a Freshdesk banner advertisement when he is browsing through Yahoo! News. A few days later, the user visits the domain *freshdesk.com* directly by entering the URL into his/her web browser, presumably because he/she is by now familiar with Freshdesk, given all the previous visits and ad experiences. And this time, the user signs up for a free trial account.

This journey leaves us with an interesting question, “Which advertising/organic channel(s) should be credited for the user’s conversion and by how much”? While it is impossible to precisely answer this question for every user, we can derive an *impact score* for each ad campaign by collectively examining paths taken by multiple users. This process is known as Attribution modeling.

[2] Without loss of generality, the sequence of events can be extended to include non-advertising interactions as well, e.g. an action performed by the user on www.freshdesk.com

[3] By “campaign”, we refer to a unit of targeted advertising, i.e a well-defined target supply/user segment, a customizable set of ad creatives and a bidding strategy. Other nomenclatures such as “ad group” are analogous in this context.

The answer to this question would help the Freshworks marketing team allocate budgets to different ad campaigns in an appropriate way to achieve[4] their business objectives.

Heuristic approaches

Going back to the above example, the following are heuristics that can be used to answer this question.

1. Assign maximum credit to the Google campaign since it is responsible for introducing the brand (Freshdesk) to the user -- this is called *"First touch attribution"*
2. Assign maximum credit to the last ad experience (Display retargeting on Y! News). The rationale being that it was the one that triggered the conversion -- this is called *"Last touch attribution"*
3. Assign equal value to all touch points in the path

The Freshworks Attribution model

The heuristic approaches have multiple shortcomings. For instance, they would assign the least amount of credit to the Bing search event in the above example. However, it is possible that individual touchpoints are highly influenced by each other (they are not i.i.d). In the above example, the display retargeting impression was triggered because of the Bing search that was performed earlier. If the user had not searched on Bing, the display impression would have not been shown to him at all. This implies that the Bing search touchpoint deserves higher credit than what any of the above approaches would assign to it.

As another example, let us consider the following path.

B: (start)
→ Google_adclick (search_term="customer support software")
→ Direct web visit
→ Google_adclick (search_term="Freshdesk signup")
→ (Conversion)

User B conducts a *generic* requirement search to begin with, indicating that he is in the market for customer support software. He sees a Freshdesk ad and interacts with it, visits our landing page but does not sign up on this occasion. The user then makes a direct visit presumably to look for further details about the Freshdesk product, and in the last hop performs a direct brand search ("Freshdesk signup") to visit us again and signs up for the trail account. Over here, it is intuitive to see that if the Freshdesk ad had not shown up during the *generic* search, the eventual brand search would not have

[4] The objective of the marketing team can be to achieve growth (higher spend and #conversions) or efficiency (higher #conversions with the same budget). Budget allocation can be framed as a constraint optimization problem in either regard.

occurred. This example is a predominant case in reality, given that it closely mirrors a user's conversion cycle (scout, research, rank and convert).

We have adopted the approach of modeling such user paths as a Markovian state transition graph. Each node or state in the graph corresponds to a distinct user interaction or event, i.e. in this case, each ad campaign is a graph vertex. Transition probabilities between states are learnt using historical data. We perform several random walk simulations to determine the value of each distinct event in the graph. The next sections describe the different processes involved in constructing the model and the insights that can be derived from it.

Input data

We record all events (touchpoints and actions) for individual users and also capture their end state, i.e. whether the user converted or not.

- Each user is labelled with an anonymous and unique visitor-id. Visitor-ids are persisted using cookies on the user's device.
- A user can visit one of our web or mobile pages, e.g. *www.freshdesk.com/support* and perform a set of actions such as clicks and page visits. Each such visit constitutes a session.
- The session also keeps track of the source from which the user came from, e.g. Direct or Google-adclick.
- The user's journey till conversion (or lack thereof) is thus captured as a path.
- The path is tagged as *non-conversion*, if the user becomes inactive for a period of 90 days[5]

Visitor grouping

Given that Freshdesk is a software product for businesses, we realised that a typical path to conversion involves multiple users from a company (i.e. a prospective customer of Freshdesk), analyzing different aspects of the Freshdesk product. These users visit the Freshdesk domain using different devices. We were able to better model the conversion journey of the company by combining all devices from the same IP network as one user. This bundling of devices at IP level is called Visitor grouping.

Visitor grouping also helps in the case wherein a single user engages with Freshdesk using multiple devices. Consider the scenario where user *U* first visits the Freshdesk domain via a Facebook ad shown on his/her mobile device. The user does not convert on this occasion but in due course of time, he/she performs a Google search for Freshdesk on their business computer, clicks on the sponsored result and signs up for a Freshdesk account. The following are two different manifestations of the user depending on whether we do visitor grouping or not.

With Visitor grouping

[5] The period of inactivity to identify non-conversion should be derived based on observed conversion dynamics (mean time to conversion, etc.) of the product that is being sold.

U: Facebook (Mobile) > Google (keyword='freshdesk') > (conversion)

Without Visitor grouping
U1: Facebook (Mobile) > (no conversion)
U2: Google (keyword='freshdesk') > (conversion)

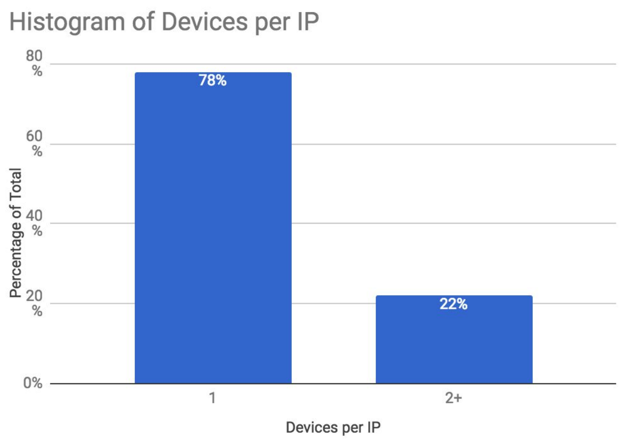

Figure 1: Histogram showing percentage of devices per IP.
Multiple devices visit from the same IP address ~22% of the time.

Without visitor grouping, we will end up attributing no credit to the Facebook mobile campaign for this conversion. Also, we will incorrectly classify the Google brand campaign (covering direct searches for the Freshdesk brand) as a potential first touchpoint, i.e. as a campaign that a user first interacts with, in his search for helpdesk software. Such a classification is counter-intuitive and does not conform to the typical conversion cycle (scout, research, rank and convert).

Estimating transition probabilities

The Markovian graph is constructed by adding each campaign (or touchpoint) as a vertex or state. The following probability expressions are then learnt using historical path data of Freshdesk users, as observed over a finite time frame. More details on training and validation timeframes can be found in the section on *Model Validation*.

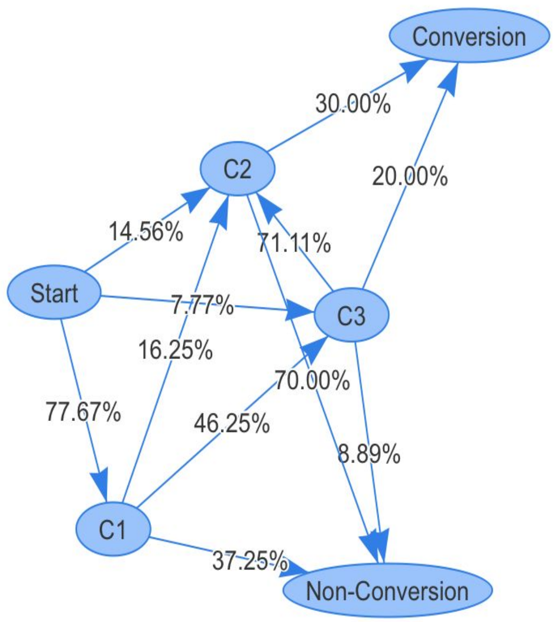

Figure 2: An example transition graph

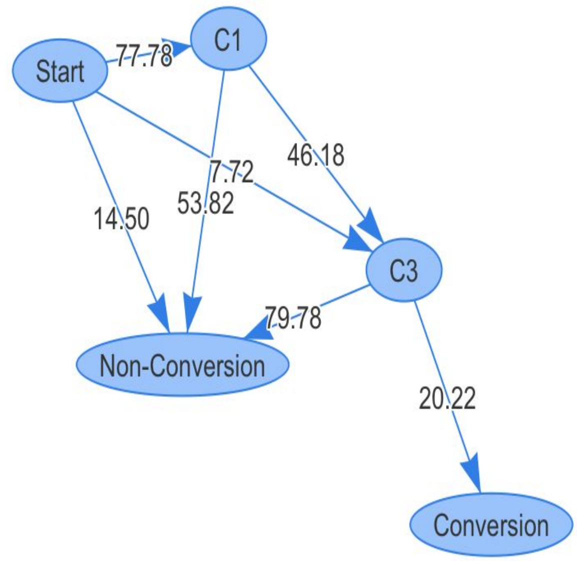

Figure 3: An example transition graph with C2 removed

The probability that the state of the user at time hop t (S_t) is i_t, i.e. the user is exposed to campaign i_t at time t, is given by

$$P(S_t = i_t) \quad = \quad P(S_t = i_t \mid S_{t-1} = i_{t-1}, S_{t-2} = i_{t-2}, S_{t-3} = i_{t-3}, \ldots, S_0 = i_0) \\ * \quad P(S_{t-1} = i_{t-1}, S_{t-2} = i_{t-2}, S_{t-3} = i_{t-3}, \ldots, S_0 = i_0)$$

Which can be recursively decomposed using Bayes rule as

$$\begin{aligned} P(S_t = i_t) \quad = \quad & P(S_t = i_t \mid S_{t-1} = i_{t-1}, S_{t-2} = i_{t-2}, S_{t-3} = i_{t-3}, \ldots, S_0 = i_0) \\ & * P(S_{t-1} = i_{t-1} \mid S_{t-2} = i_{t-2}, S_{t-3} = i_{t-3}, \ldots, S_0 = i_0) \\ & * \ldots.. \\ & * P(S_0 = i_0) \end{aligned}$$

where $P(S_0)$ is the distribution of starting states, i.e. potential first touchpoints for a user. By the order k markovian assumption, we simplify each transition (probability) expression as follows.

$$\begin{aligned} & P(S_t = i_t \mid S_{t-1} = i_{t-1}, S_{t-2} = i_{t-2}, S_{t-3} = i_{t-3}, \ldots, S_0 = i_0) \\ & \quad = \quad P(S_t = i_t \mid S_{t-1} = i_{t-1}, S_{t-2} = i_{t-2}, \ldots, S_{t-k} = i_{t-k}) \end{aligned}$$

In a simpler version of the model, the transition probabilities are defined as the chance of seeing a certain state i_t at time t, given the states observed in the previous k time intervals. As an extension to the basic model, we model the transition probability as a function of the currently active bid (and budget remaining) of the target state in addition to the k previous states in the path. The above expression then changes as follows.

$$P(S_t = i_t) \quad = \quad f\,(S_{t-1} = i_{t-1},\ S_{t-2} = i_{t-2},...,\ S_{t-k} = i_{t-k},\ bid(i_t),\ budget(i_t)\ )$$

This expression is learnt using a suitable regression algorithm such as random forests. Relationship between channels is thus captured by this approach, wherein a high transition probability between two channels (at a point in time) indicates a strong association between them.

Random walk simulations

To simulate users, we perform random walks on the Markovian graph using the following procedure.

1. We draw from the distribution of initial states $P(S_0)$ to get the first touchpoint of the (simulated) user. Let this state be $S_0 = i_0$.
2. Set the time step $t = 1$
3. Draw the next state $S_t = i_t$ from the distribution $f\,(S_{t-1} = i_{t-1},\ S_{t-2} = i_{t-2},..,\ S_{t-k} = i_{t-k},\ bid(i_t),\ budget(i_t))$
4. If it belongs to {conversion, non-conversion}, i.e. sink states which denote the end of the user's path
    a. Terminate the simulation
    b. Return the list of traversed states $(S_0,\ S_1,..,\ S_t)$ as the output of the random walk
5. Increment t and go to step 3.

Simulations are performed so as to walk imaginary users along the model generated paths. Each user begins at the '(start)' state and walks till either one of '(conversion)' or '(non-conversion)' states is reached. At each state, a random number is generated and based on the cumulative probability interval in which the number falls, the next state is decided. Relationship between channels is modeled well by this approach, a higher transition probability between two channels indicates their reliance on each other.

Model Validation

The validation process concerns with determining how well the model is able to project paths (generalize) for users in future, i.e. for a fixed period after the training timeframe. The following is an outline of the validation process.

1. Train the attribution model M using data of users whose first appearance time falls in the window (t_i, t_{i+T}), where T is the training timeframe.

2. Let $(t_{i+T},\ t_{i+T+V})$ be the corresponding validation time frame. The validation dataset would include all users who first appeared in this time window. Let the number of such users be *k*, and their set of paths be P_{act}
3. Draw *k* samples[6] by running simulations over the model *M*, let the sampled set of paths be P_{sim}
4. Compute the following metrics to determine how close P_{sim} is to P_{act}
    a. *MAPE* error in conversion rate, i.e. the fraction of users who converted in actual vs simulation
    b. *KL divergence* between the path length (and transition probability) distributions of actual and simulated users
5. Repeat the above steps for multiple (t_i,t_{i+T}) training timeframes and average the validation metrics.

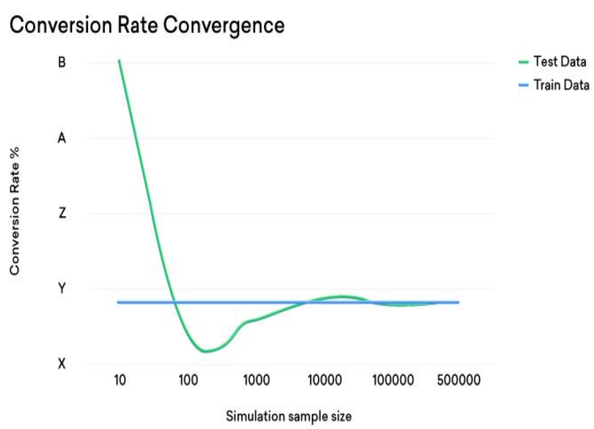

Figure 4: Conversion rate of test data converges towards training conversion rates with increasing sample size

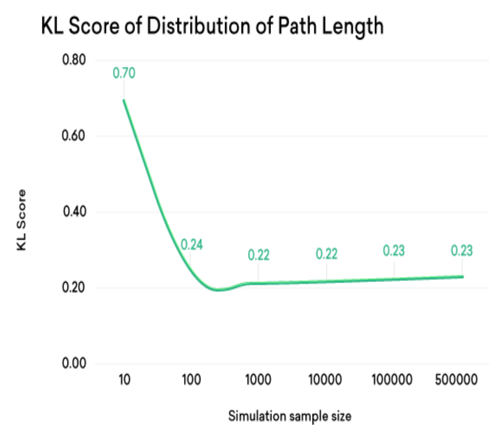

Figure 5: Reduction in KL-divergence with increasing #samples

This process is repeated for multiple (overlapping) training timeframes, and the results are averaged. The different hyperparameters of the model, including the number of simulations to calculate each removal effect and the order of the markovian assumption, are configured so to optimize for these target metrics.

Inferences from the model

Removal effects

[6] The choice of k should be made so as to minimize the model error metrics. It is a hyperparameter.

Removal effect of a campaign is the number of conversions that would have been lost had the campaign not been active in the first place. This corresponds to the *true value* or the quantum of credit that should be allocated to the campaign for all ensued conversions. The process of computing the removal effects is as follows.

1. Train the attribution model M using data of users whose first appearance time falls in the window (t_i, t_{i+T}), where T is the training timeframe.
2. Draw k samples by running simulations over the model M. Let the number of converting paths be z. The estimated conversion rate is $CR_{ov}=(z/p)$.
3. For each campaign C present in the model M,
    a. Deactivate campaign C in M, i.e. set all inbound transition probabilities into C to zero.
    b. Again, draw p samples from M and estimate the conversion rate CR_{-C}.
    c. Removal effect of C is estimated as $(CR_{ov} - CR_{-C})$

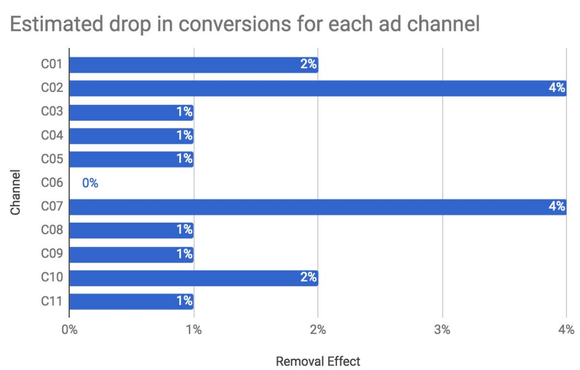

Figure 6: Removal effects computed for one campaign at a time.

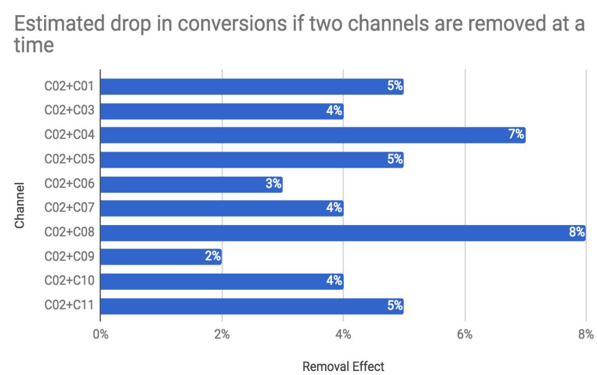

Figure 7: Removal effects computed for two campaigns at a time

If the removal effect for a campaign is low, then its spend (budget) can be reduced. Likewise, campaigns with significant removal effects should be considered as being effective, and the corresponding budgets can be increased. By the same approach, the effects of stopping multiple campaigns simultaneously can also be estimated.

Future work

Marketing teams can make changes to a campaign based on these estimated removal effects. As an extension to this exercise, we are working on the ability to evaluate the effects of changing a campaign's velocity, i.e by adjusting its constituent parameters such as bid and target audience. This would help determine the *extent* to which a campaign's velocity should be altered. Our vision is to eventually build a cross-channel bidding engine that can be configured to achieve ROI targets, given constraints around geographical reach and budget limits.