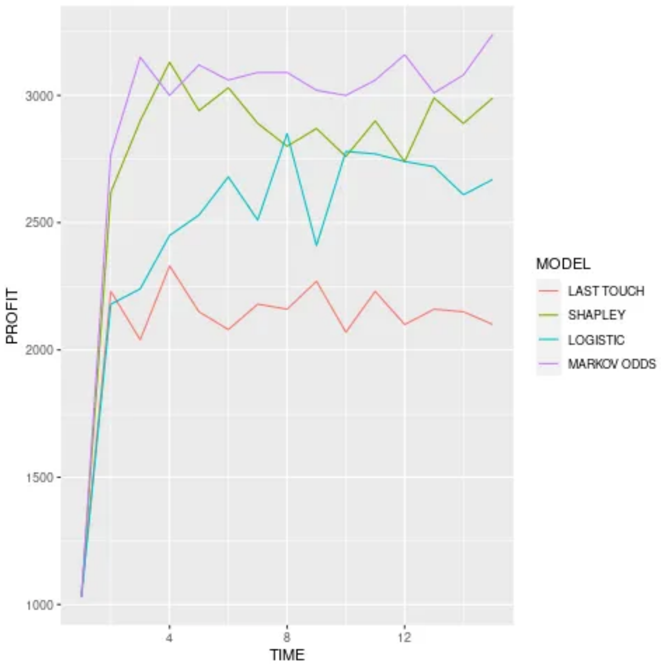
Multi-touch attribution and budget allocation

Comparison between different multi-touch attribution models in a budget allocation problem

Davide Altomare
16 min read · Mar 25, 2022

52 4



Budget allocation in digital marketing is nowadays widely based on multi-touch attribution and return on investment (ROI). Multi-touch attribution often relies on heuristic methodologies (e.g. last-touch, first-touch, linear, and weighted attribution) despite advanced methods being available (e.g. logistic regression, Shapley value, and Markov model). Heuristics are easy to understand and implement while advanced methodologies instead are not so immediate and often require more computations. In this paper, we will go in-depth on multi-touch attribution and budget allocation when high-frequency and/or dependent touch-points are present and we will present a simulation use case in which different attribution methodologies will be compared in terms of generated profit. We will show that if attribution is performed using odds calculated from Markov model then a greater profit can be reached with respect to other models.

*

1. Introduction

Multi-touch attribution evaluates the contribution each digital touch-point has in making conversions using customer journeys recorded through web cookies.

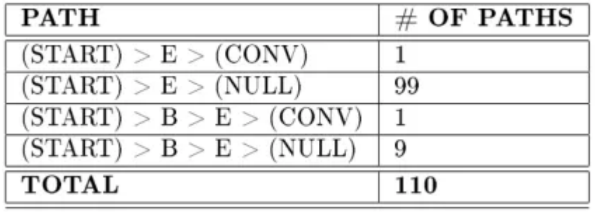
Multi-touch attribution has been introduced after cookies started to be used in digital marketing to track users. Cookies were created in 1994 to allow people to store their items in shopping carts. Within a year, the existing advertising companies have started to track users and follow them around with ad campaigns.

At the beginning of the tracking era, it was natural to start using simple heuristic methodologies to quantify the importance each touch-point has in making conversions. At the beginning of the 2010s other methodologies, such as logistic regression[2] and Shapley value[4] and Markov model[1], were introduced in the attribution arena. In 2015 the open-source library ChannelAttribution was published letting feasible and fast the application of

Markov models to real use cases. But despite this, many companies continue to use attribution using heuristic methodologies, last-touch overall.

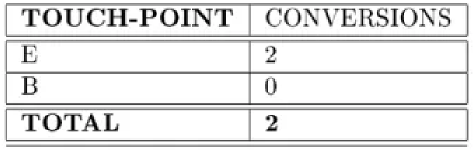
2. Multi-touch attribution

Suppose we have two touch-points: email (*E*) and banner (*B*). (*E*) is a high-frequency touch-point thus it is highly probable that it is present in a customer journey without any contribution to making conversions. Suppose we observe the following customer journeys (paths):



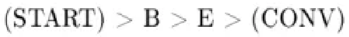
PATH	# OF PATHS
(START) > E > (CONV)	1
(START) > E > (NULL)	99
(START) > B > E > (CONV)	1
(START) > B > E > (NULL)	9
TOTAL	**110**

If we perform attribution using last-touch we get:



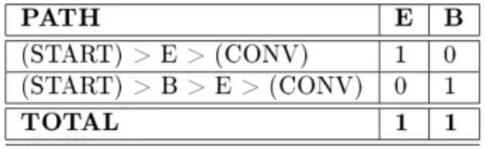
TOUCH-POINT	CONVERSIONS
E	2
B	0
TOTAL	**2**

And all the conversions are attributed to (*E*) despite it being clear that the contribution of (*E*) in path



(START) > B > E > (CONV)

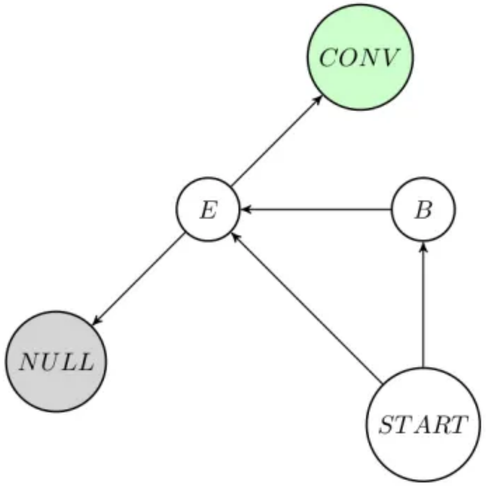
is almost null. Instead what we would have expected in this example is the following attribution:



PATH	E	B
(START) > E > (CONV)	1	0
(START) > B > E > (CONV)	0	1
TOTAL	**1**	**1**

and last-touch is clearly far from this result.

Now we try to perform attribution using a first-order Markov model. First, we need to build a Markov graph from the customer journeys we have observed:



where

$$p(\mathcal{S} \to E) = \frac{100}{110}$$
$$p(\mathcal{S} \to B) = \frac{10}{110}$$
$$p(E \to \mathcal{C}) = \frac{2}{110}$$
$$p(E \to \mathcal{N}) = \frac{108}{110}$$
$$p(B \to E) = \frac{10}{10}$$

The overall conversion probability can be calculated considering the probability of all the paths that reach the conversion state.

$$\begin{aligned} p_{\mathcal{C}}(\mathcal{G}) &= p(\mathcal{S},E) \times p(E,\mathcal{C}) + p(\mathcal{S},B) \times p(B,E) \times p(E,\mathcal{C}) = \\ &= e_{E}^{\mathcal{C}} + e_{BE}^{\mathcal{C}} = \frac{182}{1,000} \end{aligned}$$

Attribution through Markov model is usually performed using removal effects. Removal effects for channel α are the difference between the conversion probability of the full graph minus the conversion probability between the graph obtained removing channel α. In our example, the probabilities of the graphs obtained removing each touch-point once at a time are:

$$\begin{aligned} p_{\mathcal{C}}(\mathcal{G}/E) &= 0 \\ p_{\mathcal{C}}(\mathcal{G}/B) &= e_{E}^{\mathcal{C}} = \frac{165}{1,000} \end{aligned}$$

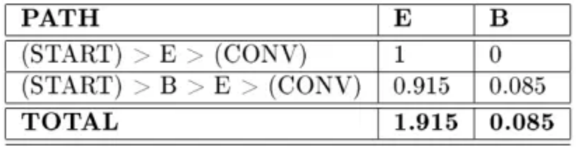
Thus removal effects for *E* and *B* are:

$$\begin{aligned} re(E) = p_{\mathcal{C}}(\mathcal{G}) - p_{\mathcal{C}}(\mathcal{G}/E) = \frac{182}{1,000} \\ re(B) = p_{\mathcal{C}}(\mathcal{G}) - p_{\mathcal{C}}(\mathcal{G}/B) = \frac{17}{1,000} \end{aligned}$$

Then removal effects are normalized:

$$\begin{aligned} w(E) &= \frac{re(E)}{re(E)+re(B)} = \frac{915}{1,000} \\ w(B) &= \frac{re(B)}{re(E)+re(B)} = \frac{85}{1,000} \end{aligned}$$

and used to perform path-level attribution:



PATH	E	B
(START) > E > (CONV)	1	0
(START) > B > E > (CONV)	0.915	0.085
TOTAL	**1.915**	**0.085**

Markov model with removal effects attributes 0.085 conversions to *B* instead of 0 as last-touch did. And also in this case the value is highly underestimated since we know that *E* is almost non-influential in the conversion process.

Hence we would provide an alternative measure that could assign a reasonable value of conversions to *B*. Let P[*C*](*G*|α) be the conversion probability if touch-point α has been observed and P[*N*](*G*|α) be the null probability if touch-point α has been observed. We can define:

$$O(\alpha) = \frac{P_{\mathcal{C}}(\mathcal{G}|\alpha)}{P_{\mathcal{N}}(\mathcal{G}|\alpha)}$$

O(α) is the odds for touch-point α.

Now we calculate odds for touch-point *E* and *B*:

$$O(E) = \frac{p(\mathcal{S},E) \times p(E,\mathcal{C}) + p(\mathcal{S},B) \times p(B,E) \times p(E,\mathcal{C})}{p(\mathcal{S},E) \times p(E,\mathcal{N}) + p(\mathcal{S},B) \times p(B,E) \times p(E,\mathcal{N})} = \frac{19}{1,000}$$

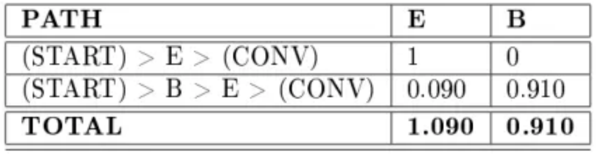
$$O(B) = \frac{p(\mathcal{S},B) \times p(B,E) \times p(E,\mathcal{C})}{p(\mathcal{S},B) \times p(B,E) \times p(E,\mathcal{N})} = \frac{185}{1,000}$$

Odds can be normalized:

$$w(E) = \frac{O(E)}{O(E)+O(B)} = \frac{9}{100}$$

$$w(B) = \frac{O(B)}{O(E)+O(B)} = \frac{91}{100}$$

We can apply these weights to each path:



PATH	E	B
(START) > E > (CONV)	1	0
(START) > B > E > (CONV)	0.090	0.910
TOTAL	**1.090**	**0.910**

Ethan Du

And we see that first-order Markov model with odds increases the conversion attributed to B to 0.910 which is very near to 1, the expected value for B. Thus we can conclude that *odds is the natural quantity to perform path-level attribution with Markov model*.

In this paragraph, we have seen that last-touch approach can easily bring to an incorrect attribution when high-frequency touch-points are present. Instead, Markov model with odds is able to find a more accurate path-level attribution.

3. Budget allocation

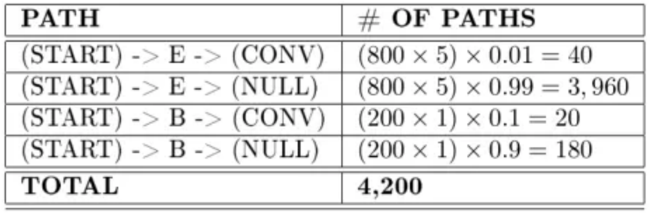
In the last paragraph, we have seen that a non-probabilistic approach such as last-touch can lead to a non-realistic attribution in case there are high-frequency touch-points in customer journeys. That is because last-touch attribution is strongly influenced by the frequency of touch-points in customer journeys. Markov model with removal effects as weights for attribution partially suffers the same issue if it is used for path-level attribution. But we have seen that it is sufficient to replace removal effects with odds to solve the problem. We say *realistic* because we do not know the real generation process of our data. We have only observed some paths and we have inferred that *E* is not relevant in making conversions. That has been possible considering the conversion capacity of each touch-point. Thus considering only the converting paths is not enough, as heuristic methodologies do. Non-converting paths are equally important in the attribution process and must be considered.
After attribution is made, the natural question is how it can be used to allocate budget among touch-points. The obvious answer is through **ROI** (return on investment). ROI for touch-point α is defined as:

$$\mathrm{ROI}(\alpha) = \frac{\text{total revenue attributed to } \alpha}{\text{total cost of } \alpha}$$

Now consider the following example. We have two touch-points: email (*E*) and banner (*B*). Moreover, the following assumptions are valid:

1. conversion rates are constant over time hence there are no external factors with any influence;
2. total budget is fixed over time;
3. all the customer journeys contain only one touch-point;
4. conversion rates are respectively 0.01 for *E* and 0.1 for *B*;
5. each 1 euro invested in *E* will always lead to 5 touches to *E* while each 1 euro invested in *B* will always lead to 1 touch to *B*;
6. At time t=0 we have invested 800 euros in *E* and 200 euros in *B*.
7. Each conversion is valued 20 euros.

These assumptions imply that the system evolves in a deterministic way over time. Thus at t=0 we got the following paths:

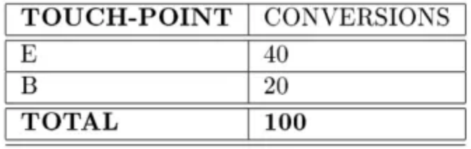


PATH	**# OF PATHS**
(START) -> E -> (CONV)	$(800 \times 5) \times 0.01 = 40$
(START) -> E -> (NULL)	$(800 \times 5) \times 0.99 = 3,960$
(START) -> B -> (CONV)	$(200 \times 1) \times 0.1 = 20$
(START) -> B -> (NULL)	$(200 \times 1) \times 0.9 = 180$
TOTAL	**4,200**

We have observed 4,200 distinct paths and 60 conversions with a total profit of (60 x 20) -1,000 = 200 euros.

In this simple example, the attribution process is trivial since we only have mono paths. We see that 40 conversions are due to *E* while 20 to *B*.

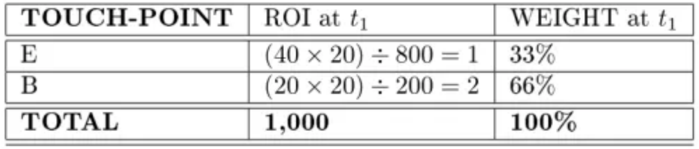
At t=1 we use last-touch approach to perform attribution:



TOUCH-POINT	CONVERSIONS
E	40
B	20
TOTAL	**100**

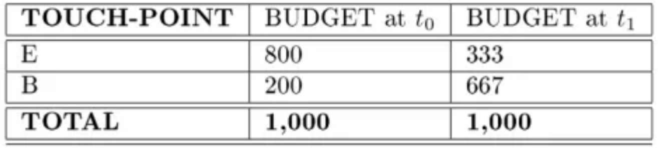
In fact, we see that last-touch correctly assigned 40 conversions to *E* and 20 conversions to *B*. Moreover, we note that in this simple use case, all the heuristic methodologies (last-touch, first-touch, linear touch, and exponential time decay) lead to the same results.

But what happens if we use results from last-touch to allocate budget? A strategy we could follow is calculating the ROI for each touch-point:



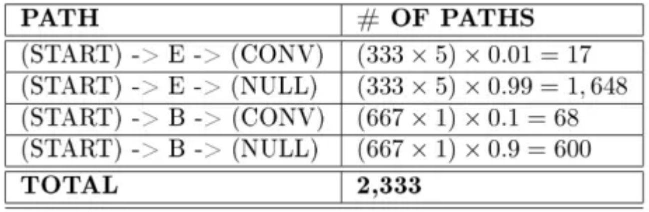
TOUCH-POINT	ROI at t_1	WEIGHT at t_1
E	$(40 \times 20) \div 800 = 1$	33%
B	$(20 \times 20) \div 200 = 2$	66%
TOTAL	**1,000**	**100%**

and then use those ROIs to split budget among touch-points:



TOUCH-POINT	BUDGET at t_0	BUDGET at t_1
E	800	333
B	200	667
TOTAL	**1,000**	**1,000**

Since initial assumptions are valid, at time t=2 we observed the following paths:

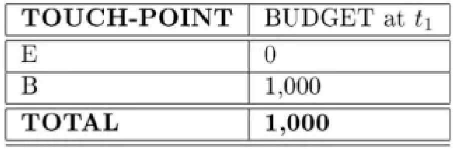


PATH	# OF PATHS
(START) -> E -> (CONV)	$(333 \times 5) \times 0.01 = 17$
(START) -> E -> (NULL)	$(333 \times 5) \times 0.99 = 1,648$
(START) -> B -> (CONV)	$(667 \times 1) \times 0.1 = 68$
(START) -> B -> (NULL)	$(667 \times 1) \times 0.9 = 600$
TOTAL	**2,333**

We had 2,333 paths and 85 conversions. Thus our new budget allocation has increased conversion from 60 to 85 with a total profit of (85 x 20)-1,000 = 700 euros.

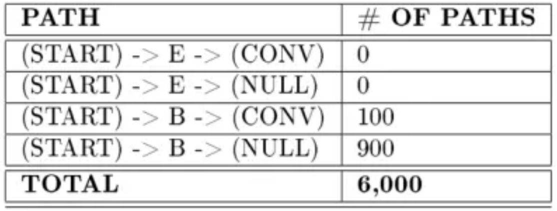
It is not difficult to show that path-level attribution with Markov model and removal effects (or odds) leads to the same result as well because the two touch-points are independent.

But is this the optimal allocation? The answer is no. It is not difficult to find that the optimal allocation takes place when all the budget is assigned to *B*:



TOUCH-POINT	BUDGET at t_1
E	0
B	1,000
TOTAL	**1,000**

Since initial assumptions are valid, at time t=2, adopting this allocation at time t=2 we get:



PATH	# OF PATHS
(START) -> E -> (CONV)	0
(START) -> E -> (NULL)	0
(START) -> B -> (CONV)	100
(START) -> B -> (NULL)	900
TOTAL	**6,000**

obtaining 6,000 paths and 100 conversion with a profit of (100 x 20) -1,000 = 1,000 euros .

It is interesting to notice that despite all the attribution models (heuristics methodologies and Markov model) leading to a correct attribution, the budget allocation they suggest through the ROI is not optimal in terms of profit. ROI leads to a sub-optimal allocation, reaching a profit that is 15% lower than the maximum value reachable. But also conversion rates lead to a sub-optimal allocation despite the profit reached being 8 % higher with respect to that reached by last-touch or by Markov model.

For this reason, some authors[4] argue that attribution should not be used to allocate budget. For sure attribution and budget allocation are conceptually different. And we have seen that if attribution is used for budget optimization it leads to a sub-optimal result in terms of profit. In the example presented there is a linear relationship between the budget allocated on touch-points and traffic generated on that touch-point. Moreover, touch-points are also independent. Thus giving all the budget to that touch-point with the higher conversion rate ensures the optimal allocation. But the reality is always more complex.
If we decide to abandon attribution and ROI, a global budget optimization strategy is anyhow hard to define. Because we would need to model a complex system that involves many variables and needs many assumptions. Thus finding the optimal allocation in real use-cases is utopian. Considering that, ROI and attribution remain important tools in budget allocation strategy. But if ROI is useful to perform budget allocation then making a correct attribution is crucial to get a greater profit. And a better attribution can be made using a probabilistic approach like Markov model, as we have seen before.

4. Comparison between different path-level attribution models

Suppose we have K touch-points and we need to allocate budget among them for T time instants. Let B(k,t) the budget allocated to touch-point k at time t, V(k,t) the conversion value attributed to touch-point k at time t, and X(k,t) be the cost associated with touch-point k at time t and B*(t) the total available budget at time t. At each t, B*(t) is split using the following strategy:

$$(\mathrm{ROI})_{k,t} = \frac{V_{k,t}}{X_{k,t}}$$

$$\begin{aligned}\mathcal{B}_{k,t} = p_1\mathcal{B}^*_t + &\left\{(1-Kp_1)p_2\frac{\mathcal{B}_{k,t-1}}{\sum_{j=1}^{K}\mathcal{B}_{j,t-1}}\right\}\mathcal{B}^*_t + \\ + &\left\{(1-Kp_1)(1-p_2)\frac{(\mathrm{ROI})_{k,t}}{\sum_{j=1}^{K}(\mathrm{ROI})_{j,t}}\right\}\mathcal{B}^*_t\end{aligned}$$

$$k = 1,\ldots,K \quad t = 1,\ldots,T \qquad p_1, p_2 \in [0,1]$$

Thus for each t, the budget allocated to a touch-point can not be less than [100 x p(1)]% of the total allocated budget. Moreover, for each t, we only vary the {100 x [1-Kp(1)] x [1-p(2)]}% of the total available budget in proportion to ROIs, while the remaining {100 x [1-Kp(1)] x p(2)}% will be allocated in proportion to the allocation made at t-1.

In the following we will present a simulation study on three touch-points: *E* (email), *S* (social), and *B* (banner) where:

1. Total budget B*(t) is equal to 3,000 euros and it remains constant over time;
2. p(1)=1% and p(2)=10% thus at each t: the budget allocated to a touch-point can not be less than 1% of the total allocated budget while we only vary the 9.7% of the total allocated budget using ROIs;
3. Each conversion is valued 30 euros;
4. E costs 0.1 eur/touch, *S* and *B* cost 1 eur/touch;

For generating paths and conversions we split the generation process into three steps:

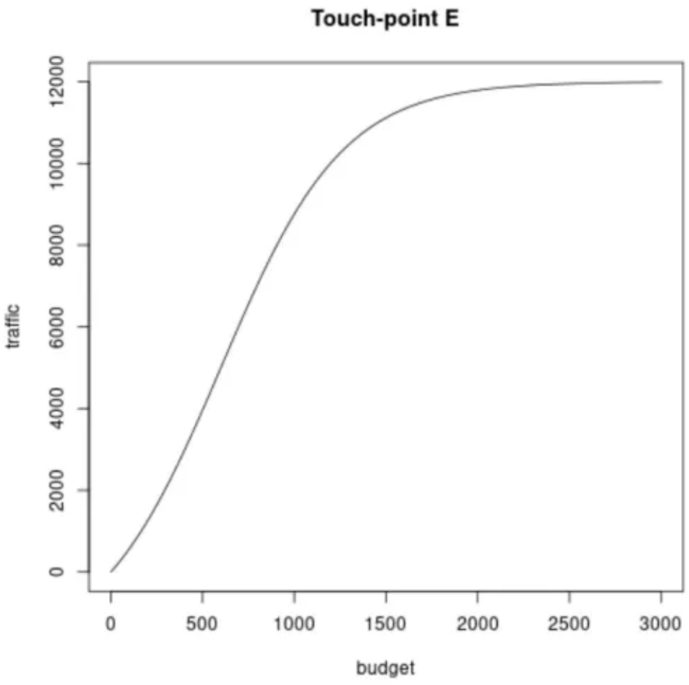
1. Define a model that quantifies how much traffic is generated on each touch-point given the budget allocated;
2. Define a model that generates customer journeys for a given traffic level;
3. Define a model that for each generated path decides if it ends in a conversion or not.

First of all, we need to define a function that links the budget allocated to each touch-point, with the traffic generated on that touch-point. For simplicity, we will assume that there is a deterministic relationship between them that follows a logistic curve:

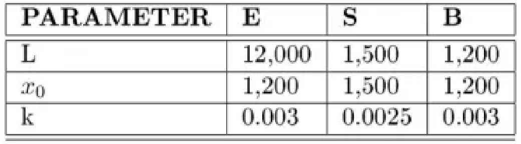
$$\text{traffic}_{\alpha} = \frac{L_{\alpha}}{1 + \exp(-k_{\alpha} \times (\text{budget}_{\alpha} - x_{\alpha}))}$$

$$\alpha \in \{\text{E ,S ,B}\}$$

where L(α) is the maximum level of traffic reachable, x(α), the x-value of the sigmoid’s midpoint, and k(α) the logistic growth rate or steepness of the curve. The curve is also been rescaled to let it intersect (0,0). Below the curve for touch-point *E* is shown:



A logistic relationship is more realistic than a linear one because we can not expect that increasing the budget will always increase traffic at the same rate. The following table shows the parameters adopted for each touch-point:



PARAMETER	E	S	B
L	12,000	1,500	1,200
x_0	1,200	1,500	1,200
k	0.003	0.0025	0.003

Now for a given budget allocation, we have the traffic generated on each touch-point. Traffic indicates the maximum number of touches we could have for each touch-point.
With this information, we can generate customer journeys. Paths are generated once at the time with the following probability function:

$$P(\text{path}) = \begin{cases} 1/53 & \text{if path} = \text{"(START)} > \text{E"} \\ 1/53 & \text{if path} = \text{"(START)} > \text{B"} \\ 10/53 & \text{if path} = \text{"(START)} > \text{E} > \text{B"} \\ 10/53 & \text{if path} = \text{"(START)} > \text{B} > \text{E"} \\ 10/53 & \text{if path} = \text{"(START)} > \text{S} > \text{B} > \text{E"} \\ 10/53 & \text{if path} = \text{"(START)} > \text{E} > \text{S} > \text{B"} \\ 10/53 & \text{if path} = \text{"(START)} > \text{S} > \text{E} > \text{B"} \\ 1/53 & \text{if path} = \text{"(START)} > \text{S} > \text{B"} \end{cases}$$

After each generation, the available traffic for each touch-point is decreased. When available traffic for touch-point α reaches 0 then all probabilities associated with paths that include α are set to 0. Thus if, for example, traffic of *S* reaches 0 then the probability function becomes:

$$P(\text{path}) = \begin{cases} 1/22 & \text{if path} = \text{"(START)} > \text{E"} \\ 1/22 & \text{if path} = \text{"(START)} > \text{B"} \\ 10/22 & \text{if path} = \text{"(START)} > \text{E} > \text{B"} \\ 10/22 & \text{if path} = \text{"(START)} > \text{B} > \text{E"} \end{cases}$$

Once a path is generated we have to decide if it ends with a conversion or not. We use the following probability function:

$$P(\text{conv}|\text{path}) = \begin{cases} 30/100 & \text{if S and B are in path} \\ 15/100 & \text{if S is in path and B is not in path} \\ 1/100 & \text{if S and B are not in path and E is in path} \end{cases}$$

Now we want to evaluate the performance of the following path-level attribution models in terms of generated profit:

roi-last-touch

Attribution is made at path-level using last-touch approach, then budget is split through ROI.

This is the well-known last-touch approach that we have discussed in section 2.

roi-shapley

Attribution is performed at path-level using Shapley value, then budget is splitted through ROI.
Shapley value for a touch-point k is defined as:

$$
\begin{aligned}
\phi_k &= \sum_{S \subseteq \mathcal{K} \setminus \{k\}} \frac{|S|!(K-|S|-1)!}{K!}[\nu(S \cup \{k\}) - \nu(S)] \\
k &= 1, \ldots, K
\end{aligned}
$$

where *K*={1,...,K} is the set of touch-points, S a subset of *K*, |*S*| the cardinality of *S* and ν(*S*) the contribution of *S*.
In our case ν(S) is the conversion rate of *S*.

roi-logistic

Attribution is made at path-level using logistic regression, then budget is split using ROI.
Logistic regression is implemented using the following formulation:

$$
\begin{aligned}
\mathrm{P}(Y_i = 1) &= \Lambda\left(\beta_0 + \sum_{k=1}^{K} \beta_k C_{i,k} + \sum_{k \neq h} \beta_{k,h} C_{i,kh}\right) \\
\Lambda(x) &= \frac{1}{1+\exp(-x)}
\end{aligned}
$$

where Y(i) is equal to 1 if path i converts, C(i, k) is equal to 1 if touch-point k belongs to path i and C(i, kh) is equal to 1 if both touch-point k and touch-point h belong to path i.

roi-markov-re

Attribution is performed at path-level using Markov model and removal effects, then budget is split using ROI.
In this case, we use the removal effects calculated from Markov model to perform path-level attribution as we did in section 2. This method is widely used in practice despite it leads to an incorrect attribution when there are high-frequency touch-points.

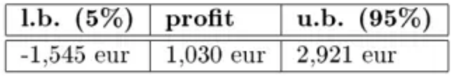
roi-markov-re-corr

Attribution is performed at path-level using Markov model and corrected removal effects, then budget is split using ROI.
In this case, removal effects from Markov model are used at path-level before they are iteratively corrected to let path-level attribution match global attribution.

roi-markov-odds

Attribution is performed at path-level using Markov model and odds, then budget is split through ROI.
In this case, odds are calculated from Markov model and used for path-level attribution, as we did in section 2.

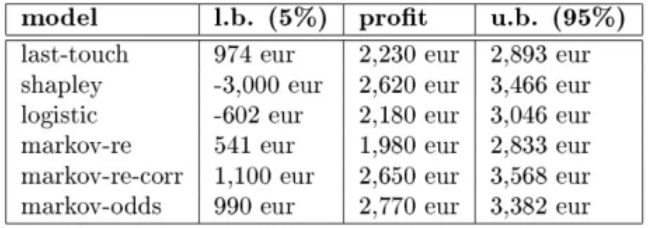
The experiment has been repeated 30 times r=1,…,30 and each time, budget allocation and profit calculation were made for 15 time instances t=1,…,15. For each r at t=1 a random budget allocation is generated. Using all 30 replications we found the following empirical confidence interval for the observed profit:



l.b. (5%)	profit	u.b. (95%)
-1,545 eur	1,030 eur	2,921 eur

where profit is the 50th percentile of the empirical distribution of the profits calculated considering all the 30 customer-journey datasets generated, l.b. is the 5th percentile, and u.b. is the 95th percentile. We see that if a random budget allocation is adopted, then we expect a median profit of 1,030 euros.

For each r, at t=2 each model proposes a new budget allocation based on ROI. Then customer journeys are generated and profit is calculated. We obtained the following results:

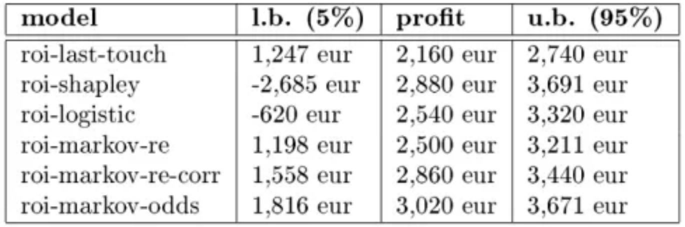


model	l.b. (5%)	profit	u.b. (95%)
last-touch	974 eur	2,230 eur	2,893 eur
shapley	-3,000 eur	2,620 eur	3,466 eur
logistic	-602 eur	2,180 eur	3,046 eur
markov-re	541 eur	1,980 eur	2,833 eur
markov-re-corr	1,100 eur	2,650 eur	3,568 eur
markov-odds	990 eur	2,770 eur	3,382 eur

We see that Markov with odds and Markov with corrected removal effects reaches the higher profit while Markov that uses the overall removal effects generates a lower one. This confirms that the practice of using overall removal effects in path-level attribution is not optimal.

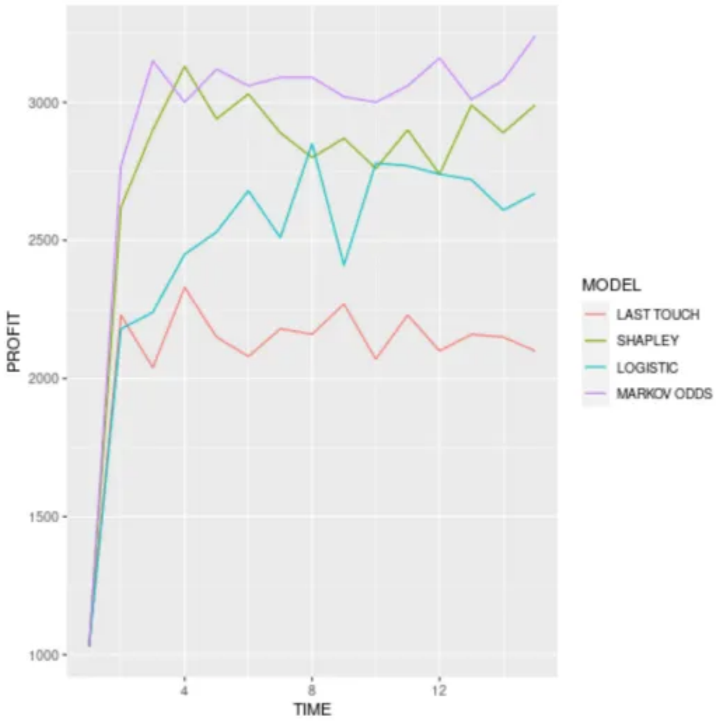
t=2 is quite important because it is the first budget allocation after a random allocation. Our use-case is simple and there are no external factors that affect the system. Thus the capacity to generate profit starting from a random budget allocation can be viewed as the capacity of generating profit when exogenous unpredictable external factors are present. And it seems that Markov model has this capacity much more than the other models.

The overall results, considering all the 30x15 profits generated for each model are:



model	l.b. (5%)	profit	u.b. (95%)
roi-last-touch	1,247 eur	2,160 eur	2,740 eur
roi-shapley	-2,685 eur	2,880 eur	3,691 eur
roi-logistic	-620 eur	2,540 eur	3,320 eur
roi-markov-re	1,198 eur	2,500 eur	3,211 eur
roi-markov-re-corr	1,558 eur	2,860 eur	3,440 eur
roi-markov-odds	1,816 eur	3,020 eur	3,671 eur

We can also plot the median profit over time for each model:



These results confirm that *Markov model with odds is a valid methodology for path-level attribution when high-frequency and dependent touch-points are present, as it usually happens in real use-cases*.

6. Path-level attribution with ChannelAttributionPro

In the following, it will be shown how easy is to perform transaction-level attribution using *ChannelAttributionPro* with R or Python.

For any detail about how to install *ChannelAttributioPro* and to get a password visit channelattribution.io.

R

```
library(ChannelAttributionPro)

##########
#Load Data
##########

data(PathData)

password="youpassword"

######
#Train
######

res=markov_model(Data, var_path="path",var_conv="total_conversions",
var_value="total_conversion_value",var_null="total_null",order=1,
sep=">",ncore=1,out_more=TRUE,verbose=TRUE,type="odds",password=passw
ord)

#####################
#Save path attribution
#####################

res_path_attr=res$path_attribution
```

Python

```
from ChannelAttributionPro import *

password=”youpassword”

##############
#Download data
##############

Data=pd.read_csv(“https://channelattribution.io/csv/Data.csv",sep=";"
)

######
#Train
######

res=markov_model(Data, var_path=”path”,var_conv=”total_conversions”,
var_value=”total_conversion_value”,var_null=”total_null”, order=1,
sep=”>”,ncore=1,out_more=True,verbose=True,type=”odds”,password=passw
ord)
```

```
######################
#Save path attribution
######################

res_path_attr=res[‘path_attribution’]
```

5. Conclusions

Nowadays budget allocation in digital marketing is heavily based on multi-touch attribution and return on investment (ROI) calculation. Attribution and budget allocation are two deeply different concepts and some authors suggest avoiding using attribution for budget allocation purposes. Despite using ROI for budget allocation is not optimal, considering a more complex approach does not ensure that an optimal allocation can be found in any case. That is because real use cases are complex to be modeled and require many assumptions. For this reason, attribution remains crucial in budget allocation strategy.

ROI is an effective measure in budget allocation only if conversion value is correctly attributed to each touch-point. When high-frequency and/or dependent touch-points are present, as usually happens in real use-cases, then non-probabilistic approaches can perform significantly worse than probabilistic ones. Through a simple but realistic simulation study, we showed that Markov model when used in ROI calculation is able to generate more profit than last-touch. Markov model also outperforms Shapley value and logistic regression.

Path-level attribution with Markov model is usually implemented using removal effects as weights. We showed that this choice is not able to correctly manage high-frequency touch-points. Hence we have proposed odds as an effective and alternative measure to removal effects. In the use-case presented we have seen that, if path-level attribution is performed using Markov model with odds, better performance in terms of profit can be reached.

ChannelAttributionPro is the professional library available for R and Python intended for companies who want to make a more accurate attribution and budget allocation for increasing their profit. It lets to calculate path-level attribution with Markov model and odds easily and fast. If you need the code to replicate the simulation study presented and/or other information about *ChannelAttributionPro*, visit channelattribution.io.

6. References

[1] Anderl E. et al., *Mapping the Customer Journey: A Graph-Based Framework for Online Attribution Modeling*, 2014, doi

[2] Xuhui S. and Lexin L., *Data-driven Multi-touch Attribution Models,* 2011, doi

[3] Danaher P. and Heerde H., Delusion in Attribution: *Caveats in Using Attribution for Multimedia Budget Allocation*, 2018, doi

[4] Zhao K., Mahboobi S.H., Bagheri S.R., *Shapley Value Methods for Attribution Modeling in Online Advertising, 2012,* doi

Digital Marketing

Markov Models

Multi Touch Attribution

Budget Allocation

Data Science

Written by Davide Altomare

Edit profile

31 Followers

Machine Learning Developer | Statistician | ChannelAttribution Author