About Helge

Data Scientist

OFFER ZEN

Applied Mathematics
OfferZen’s mission

- Talent
- Opportunity
The master plan

OfferZen → Root → Source → Make

Get devs jobs → Build tools that gives more power → Help devs learn faster → Get more people building software
HOW TO GET AHEAD IN ADVERTISING

THE CAREER WHERE TWO HEADS ARE BETTER THAN ONE.

HANDMADE FILMS PRESENTS
"HOW TO GET AHEAD IN ADVERTISING"
STARRING RICHARD E. GRANT, RACHEL WARD
ORIGINAL MUSIC BY DAVID DUNN AND RICK WENTWORTH
PHOTOGRAPHED BY PETER HARRIAN CO-PRODUCER RAY COOPER
EXECUTIVE PRODUCERS: GEORGE HARRISON AND DENNIS O'BRIEN
PRODUCED BY DAVID WIMBURY WRITTEN AND DIRECTED BY BRUCE ROBINSON
In this talk

- What is the marketing attribution problem?
In this talk

- What is the marketing attribution problem?
- Models and methods
In this talk

- What is the marketing attribution problem?
- Models and methods
- Practical lessons learned
What is the marketing channel attribution problem?
Marketing channel attribution

I MEAN

HOW HARD CAN IT BE?

quickmeme.com
Marketing channel attribution

Who Gets The Credit??

Affiliate - Organic - Remarketing - Direct - Direct
Social - Direct - Direct - Paid - Remarketing
Paid - Paid - Organic - Direct - Email
Marketing channel attribution

Awareness → Consideration → Purchase
Marketing channel attribution

This is the original link


These are the UTM parameters
Models and methods
Models and methods

- Last Click
- First Click
- Linear
- Position-based
- Time Decay
- Data-Driven
“all models are wrong, but some are useful”

George Box
1919 - 2013
Models

Markov Chain Attribution
Markov Chain Attribution

Diagram of a Markov Chain:
- States: S1, S2, S3, S4
- Transition Probabilities:
  - S1 to S2: 1
  - S1 to S3: 0.5
  - S2 to S3: 0.2
  - S2 to S4: 0.3
  - S3 to S4: 0.1
  - S3 to S1: 0.8
  - S4 to S2: 0.2

Game Theory Attribution
Game Theory Attribution
Shapley Values

\[ \phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \, (n - |S| - 1)!}{n!} \left( v(S \cup \{i\}) - v(S) \right) \]
Markov Chain Attribution

In [1]:
library(ChannelAttribution)
library(reshape)
library(ggplot2)

Toy data

In [2]:
df1 <- data.frame(sequence = c('c1 > c2 > c3', 'c1', 'c2 > c3'), conv = c(1, 0, 0), no_conv = c(0, 1, 1))
df1

<table>
<thead>
<tr>
<th>sequence</th>
<th>conv</th>
<th>no_conv</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1 &gt; c2 &gt; c3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>c1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>c2 &gt; c3</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Define the model

In [3]:
model1 <- markov_model(df1,
                        var_path = 'sequence',
                        var_conv = 'conv',
                        var_null = 'no_conv',
                        out_more = TRUE)

Get the results

In [4]:
df_result_1 <- model1$result
df_result_1

<table>
<thead>
<tr>
<th>channel_name</th>
<th>total_conversions</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>0.2002886</td>
</tr>
<tr>
<td>c2</td>
<td>0.3998557</td>
</tr>
<tr>
<td>c3</td>
<td>0.3998557</td>
</tr>
</tbody>
</table>

Removal effects
In [5]:

removal_effects = model1$removal_effects
removal_effects

<table>
<thead>
<tr>
<th>channel_name</th>
<th>removal_effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>0.5009023</td>
</tr>
<tr>
<td>c2</td>
<td>1.0000000</td>
</tr>
<tr>
<td>c3</td>
<td>1.0000000</td>
</tr>
</tbody>
</table>

In [6]:

# 1. Probability of conversion

\[0.667 \times 0.5 \times 1 \times 0.5 + 0.333 \times 1 \times 0.5\]

0.33325
In [7]:
# 2. Probability of conversion
0.333 * 1 * 0.5
0.1665

In [8]:
# Removal effect
1 - 0.167/0.333
0.498498498498498

In [9]:
removel_effects = removal_effects$removal_effects
data.frame(removel_effects/sum(removel_effects))

<table>
<thead>
<tr>
<th>removel_effects.sum.removel_effects.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2002886</td>
</tr>
<tr>
<td>0.3998557</td>
</tr>
<tr>
<td>0.3998557</td>
</tr>
</tbody>
</table>

Game Theory Attribution

In [10]:
library('GameTheoryAllocation')
Loading required package: e1071
Loading required package: lpSolveAPI

Coalitions
In [11]:

```r
df_B1 = data.frame(coalitions(3)$Binary)
names(df_B1) <- c('c1', 'c2', 'c3')
df_B1
```

<table>
<thead>
<tr>
<th>c1</th>
<th>c2</th>
<th>c3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

In [12]:

```r
2**3
```

8

Characteristic function

In [13]:

```r
characteristic_function <- c(0,7,4,6,7,15,9,19)
```

In [14]:

```r
df_B1$conversions <- characteristic_function
df_B1
```

<table>
<thead>
<tr>
<th>c1</th>
<th>c2</th>
<th>c3</th>
<th>conversions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>19</td>
</tr>
</tbody>
</table>

Shapely values
In [15]:
```r
shapley_values <- Shapley_value(characteristic_function, game="profit")
```
```
[1] "Shapley Value"
```
In [16]:
```r
#shapley_values = data.frame(shapley_values)
data.frame("channel_name" = c('c1', 'c2', 'c3'), "shapley_values" = c(shapley_values))
```
<table>
<thead>
<tr>
<th>channel_name</th>
<th>shapley_values</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>7.666667</td>
</tr>
<tr>
<td>c2</td>
<td>3.166667</td>
</tr>
<tr>
<td>c3</td>
<td>8.166667</td>
</tr>
</tbody>
</table>

Marginal values

**c1-c3-c2** -> 7 + 8 + 4  
**c1-c2-c3** -> 7 + 0 + 12  
**c2-c3-c1** -> 4 + 5 + 10  
**c2-c1-c3** -> 4 + 3 + 12  
**c3-c1-c2** -> 6 + 9 + 4  
**c3-c2-c1** -> 6 + 3 + 10  

In [17]:
```r
factorial(3)
```
```
6
```

Shapley values

In [18]:
```r
# c1
(7 + 7 + 10 + 3 + 9 + 10)/6
```
```
7.66666666666667
```
In [19]:
```r
# c2
(4 + 0 + 4 + 4 + 4 + 3)/6
```
```
3.16666666666667
```
In [20]:
# c3
(8 + 12 + 5 + 12 + 6 + 6)/6

8.16666666666667

In [21]:
shapley_values

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.66667</td>
<td>3.16667</td>
<td>8.16667</td>
</tr>
</tbody>
</table>

In [22]:

shapley_values/sum(shapley_values)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.4035088</td>
<td>0.166667</td>
<td>0.4298246</td>
</tr>
</tbody>
</table>
Practical lessons learned

- Marketing attribution is an important and hard problem
Practical lessons learned

- Marketing attribution is an important and challenging problem.
- No best model or method (might not matter too much)
Marketing attribution is an important and challenging problem.

- No best model or method (might not matter too much)

- Good data quality + simple model -> often sufficient
“The career where two heads are better than one.”
Questions?