User Configuration Manual

How Cooperative Game Theory can be utilised to enhance marketing analytics attribution.

Master of Science in Data Analytics
James Matthews, X13109260, James.Matthews1@student.ncirl.ie
## Contents

1. User Configuration Overview ................................................................. 3

2. Prerequisite Google Analytics Configuration ........................................ 3

   2.1 Configuration Goals ......................................................................... 3

      2.1.1 Destination Goal Count for Completed Conversions .................. 3

      2.1.2 Destination Goal for Count Abandoned (null) Conversions ........... 3

      2.1.3 Smart Goal to Capture Transaction Value ................................... 3

   2.2 Configuration Custom Variables ....................................................... 4

      2.2.1 Custom Variable to Record Conversion Path ............................. 4

      2.2 Google Analytics Oauth token authentication ............................... 4

3. Mockaroo Dataset Generation ................................................................. 4

   3.1 Defining Mockaroo Data Schema ...................................................... 4

   3.2 Generating & Downloading Mockaroo Dataset .................................. 6

4. Core Solution Framework Elements ....................................................... 6

   4.1 Real-time Element for Polling Data .................................................. 7

      4.4.1 Scheduling R Script ................................................................. 7

      4.4.1 Functionality Google_Analytics_Final.r .................................... 8

   4.2 Attribution Modelling Overview ....................................................... 9

      4.2.1 Last Click Attribution Model ..................................................... 10

      4.2.1 Cooperative Attribution Model ............................................... 11

   4.3 Running Attribution Model ............................................................. 12

6 References ............................................................................................ 13

7 Appendix ............................................................................................... 13
1. User Configuration Overview

The solution framework assembled by the author is targeted towards enterprise and small medium businesses that are utilising the Google Technology stack. The solution can be replicated and adapted to work against instances of Google Analytics that have completed the perquisites customisation steps outlined in section 2.

This user configuration manual details each element of the framework and the necessary steps involved in conducting the attribution benchmark experiment to address How Cooperative Game Theory can be utilised to enhance marketing analytics attribution.

2. Prerequisite Google Analytics Configuration

This section outlines the prerequisite steps that must be completed prior to running the solution framework.

2.1 Configuration Goals

The following goals must be configured in the instance of Google Analytics where the solution framework will be in use. Google Analytics has three different types of goals from which to choose from, please refer to the steps outlined in the Google Analytics - Create, edit, and share goals guide in setting up the following Goals (Support.google.com, 2016).

2.1.1 Destination Goal Count for Completed Conversions

Destination Goal type, counts the number of successful online transactions and sessions that has been completed on the business website, to be set-up to trigger on displaying of the payment confirmation webpage.

2.1.2 Destination Goal for Count Abandoned (null) Conversions

Destination Goal type, that counts the number of abandoned transactions and sessions that has been completed on the business website, to be set-up to trigger on webpages prior to the payment confirmation.

2.1.3 Smart Goal to Capture Transaction Value

Smart Goal type that captures the total currency value of online transactions that have been completed on the business website, to be set-up to trigger on displaying of the payment confirmation webpage.
2.2 Configuration Custom Variables
The following custom variable must be configured in the instance of Google Analytics where the solution framework will be in use. Use of custom variables gives websites the ability to include & format non-standard data in their instance of Google Analytics. Please refer to the steps outlined in the Google Analytics - Custom dimensions & metrics guide in setting up the following custom variable (Support.google.com, 2016).

2.2.1 Custom Variable to Record Conversion Path
Session based Custom Variable that captures the conversion path flow of customers; custom variable is to be placed on website landing pages that are being used in Online Marketing activities alongside the website checkout ecommerce pages. Custom Variable is to capture the source/medium values of all active sessions on the website; source/medium will show the marketing channel name and its source of the conversion path a user has taken. For example Display GDN > Organic Google > PPC > Direct > Purchase.

2.2 Google Analytics Oauth token authentication
In order to use the Real-time Element for Polling Data from Google Analytics, a Oauth authentication token must be obtained. To generate the Google Analytics Oauth authentication token please refer to the steps outlined in the Google Analytics - Using OAuth 2.0 to Access Google APIs guide (Google Developers, 2016). Once the Client ID and Client Secret tokens have been generated, replace the placeholder tokens within the script Google_Analytics_Final.r outlined in see section 4.1.

3. Mockaroo Dataset Generation
Mockaroo is an online platform that allows for the generation en masse of synthesized and realistic datasets for use in modelling, testing and demoing purposes. This section outlines the steps involved in generating a synthesized dataset for use in the attribution benchmark experiments (Mockaroo, 2016).

3.1 Defining Mockaroo Data Schema
Mockaroo generates datasets by utilising a data schema, which is based upon an existing underlying data structure predefined by the user. The Mockaroo data schema has two components in the form of a Field Name and Field Type.
The Field Name denotes an attribute name in a given dataset, whilst Field Type being associated with a Field Name contains either a numerical or categorical value. Field Types if numerical are assigned a numeric range that matches the underlying data structure. Moreover, categorical Field Types are assigned all possible characteristic values again based upon the underlying data structure. The data schema is inputted and defined in the Mockaroo platform, in company with the number of required rows needed in the dataset.

![Mockaroo Data Schema](image)

**Figure 1 - Example Data Schema**

Figure 1 illustrates an example data schema that will generate conversion path data that can be utilised in the attribution model. The schema contains a total of five Field Names alongside their respective Field Types.

- ID generates a unique identification number for each dataset entry.
• Conversion Path field type will randomly assign a conversion path characteristic value from the categorical value list. Which is made up of the Marketing channel names.
• Sum of Conversions and Sum of null conversions assigns a value from a numeric range 0 to 3891.
• Sum of conversions assigns a value from a numeric range of 1.1 through to 12,757.

3.2 Generating & Downloading Mockaroo Dataset
To generate a Mocaroo dataset follow these steps:
1. Go to [www.mockaroo.com](http://www.mockaroo.com)
2. Register for an account by clicking ‘sign in’ and follow the on screen prompts.
3. Login to your mockaroo account.
4. Click on data schema.
5. Enter each field name and select a corresponding Field type. Repeat this step until you have the number of desired Field Names & Field Types inputted.
6. Once the schema has been defined in Mockaroo the required number of rows needed is entered into the count rows field and a format for the file type is selected
7. ‘Click Download Data’

Upon clicking the “Download Data” button Mockaroo generates the dataset based upon the data schema creating the Field Names first and then randomly assigning the categorical Field types with their associated numeric range or characteristic values. The generated dataset is then downloaded and saved locally to the user’s machine.

4. Core Solution Framework Elements
This section of the user configuration manual outlines the core elements of the solution framework. The framework runs on the statistical programming language called R, which can be downloaded and installed for free from this [link](http://example.com). Both files should be copied and save into a working directory on your local PC, all Google Analytics Conversion path data and attribution modelling output will be written to this directory.
4.1 Real-time Element for Polling Data.
The real-time element of the solution framework is an R programming script titled Google_Analytics_Final.r. This script utilises the RGoogleAnalytics package and the code within is based upon examples provided in the RGoogleAnalytics Reference manual (Pearmain, 2016).

4.4.1 Scheduling R Script
The script Google_Analytics_Final.r is set to run every 120 minutes via a schedule task set up in windows task scheduler.

To create a schedule task follow these steps:

1. Open the Windows Task Scheduler Application
2. Click Create Task
3. Assign the Task the name of Google Analytics Data Downloader.
4. Click Triggers Tab > New > Daily > Repeat Task Every:
5. Check Repeat Task Every and input 120 minutes, set duration of indefinitely.
6. Click Actions Tab > New > Start a Program
7. From program/script Browse to where you have R installed on your PC.
   Typically the installation path will be on your C drive “C:\Program Files\R\R-3.3.1\bin”
8. Enter “Google_Analytics_Final.r” as the add arguments. Click OK
9. The create task action window, should be similar to Figure 2 below.
10. Click Ok and Ok again.
11. The data polling element of the framework will now run every 120 minutes.

![Figure 2 - Scheduled Task Actions Tab for Google_Analytics_Final.r]
4.4.1 Functionality Google_Analytics_Final.r
The script Google_Analytics_Final.r is set to run every 120 minutes via windows task scheduler. The script will automatically save a CSV file titled “Google_Analytics_Conversion_Path_Data.csv” of conversion path information to your local PC.

Figure 3 below shows the Google API Query Parameters, to fully configure the Google_Analytics_Final.r script, to suit your environment follow these steps to update the query parameters:

1. Install the RGoogleAnalytics package by running the command install.packages("RGoogleAnalytics") within your R console.
2. Change the Set working file path to match the location of where you have the R scripts saved on your PC.
3. Load RGoogleAnalytics Package – Highlight Line 14 > Click the run command.
   *This command will load the needed package to connect to the Google Analytics API.*
4. Replace the oauth token client.id and client secret id with your newly generated credentials from step 2.2 above.
5. Set Query Start Date to the desired date.
6. Set Query End Date; note this can be set to a future date eliminating the need to set the date each time.
7. Check that the newly created goals and custom variable sequence numbers match that of the script. To note if you have any previous goals created in your Google Analytics account then the number sequences will be different, goal sequence numbers are detailed in your Google Analytics view.
   a. ga:customVarValue1
   b. ga:customVarValue2
   c. ga:goal1Completions
   d. ga:goal2Completions
   e. ga:goal3Completions
8. Change the table.id to the Google Analytics View where the query should be run against.
9. Set-up complete.
4.2 Attribution Modelling Overview.
The attribution modelling element of the solution framework is a R programming script titled Attribution_model_Final.r. This script utilises the ChannelAttribution package and the code within is based upon examples provided in the ChannelAttribution Reference manual (Altomare, 2016).

To configure the attribution modelling script follow these steps:

1. Install the RGoogleAnalytics package by running the command `install.packages("ChannelAttribution")` within your R console.
2. Change the Set working file path to match the location of where you have the R scripts saved on your PC.
3. Check you have latest conversion path CSV file from Google Analytics, titled Google_Analytics_Conversion_Path_Data.csv
4.2.1 Last Click Attribution Model

Figure 4 below shows the Last Click attribution model code in situ.

```r
# LAST CLICK ATTRIBUTION MODEL
#Run Last Click Attribution Model
#Get Last Click volume of Conversions
heuristic_models(data, "Conversion_Path", "sum_of_conversions")
#Get last click volume of conversions
heuristic_models(data, "Conversion_Path", "sum_of_conversions", var_value = "sum_of_conversion_value")
#Write Results to CSV
writeResultsCSV <- capture.output(heuristic_models(data, "Conversion_Path", "Sum_of_Conversions", var_value = "Sum_of_Conversion_Value"))
cat("Last Click", writeResultsCSV, file="Last Click Attribution Model.csv", sep="", append=TRUE)
#Clear R Console
cat("\n")
```

**Figure 4 – Attribution_model_Final.r - Last Click Attribution Model Functionality**

To run the Last Click Attribution Model follow these steps:

1. Load ChannelAttribution Package – Highlight Line 14 > Click the run command.  
   *This command will load the needed package to run the attribution model.*
2. Import Dataset – Highlight Lines 20 & 21 > Click the run command.  
   *This command will import the latest conversion path information from Google_Analytics_Conversion_Path_Data.csv file.*
3. Run Last Click Model > Calculate Last Click Volume of Conversions > Highlight Line 28 > Click the run command.  
   *Outputted Last Click Volume of Conversions results can be interpreted onscreen in R.*

```r
> heuristic_models(Data,"Conversion_Path","Sum_of_Conversions")
   channel_name first_touch last_touch linear_touch
   1 PPC 24331 15062 16667.63929
   2 Organic-Google 14415 25272 21309.49286
   3 Social-Twitter 215 731 1437.5924
   4 Display-Quantcast 3945 6082 4633.35833
   5 Direct 17027 28230 22117.41548
   6 Social-Facebook 533 3581 2462.22413
   7 Display-GDN 22311 5596 14479.62738
   8 Display-AdRoll 7110 4553 6179.56310
   9 Email 914 470 818.10595
   10 Social-LinkedIn 517 1723 1200.97619
   11 organic-Yahoo 14 14 17.33333
   12 Organic-Bing 6 24 14.67143
```

4. Run Last Click Model > Calculate Last Click Value of Conversions > Highlight Line 31 > Click the run command.  
   *Outputted Last Click Value of Conversions results can be interpreted onscreen in R.*
5. Write Last Click Attribution Results to CSV file > Highlight line 34 to 36 > Click the run command. This command will save to CSV file the outputted results in terms of conversion value volume to a CSV file titled: *Last Click Attribution Model.csv*

6. Process Complete

### 4.2.1 Cooperative Attribution Model

Figure 5 below shows the Cooperative attribution model code in situ,

```r
#-------------------------
# COOPERATIVE Attribution Model
#-------------------------
# Run Cooperative Attribution Model
#Calculate Cooperative volume of Conversions
markov_model(data, "conversion_path", "sum_of_conversions")

#Calculate cooperative value of conversions
markov_model(data, "conversion_path", "sum_of_conversions", var_value="sum_of_conversion_value")

#Calculate contributions of individual marketing channel
markov_model(data,"conversion_path","sum_of_conversions", var_value="sum_of_conversion_value", var_null="sum_of_null_conversions")

#Write Attribution Model Results to CSV
writeResultsCSV <- capture.output(markov_model(data,"conversion_path","Sum_of_Conversions", var_value="sum_of_conversion_value", var_null="sum_of_null_conversions")
cat("Cooperative", writeResultsCSV, file="Cooperative Attribution Model.csv", sep="n", append=TRUE)

#Clear R Console
cat("\n\n")
```

Figure 5 Attribution_model_Final.r - Cooperative Attribution Model Functionality

To run the Last Cooperative follow these steps:

1. Load ChannelAttribution Package – Highlight Line 14 > Click the run command.
   *This command will load the need package to run the attribution model.*

2. Import Dataset – Highlight Lines 20 & 21 > Click the run command
   *This command will import the latest version of the Google_Analytics_Conversion_Path_Data.csv file.*

User Configuration Manual
3. Run Cooperative Model > Calculate Cooperative Volume of Conversions > Highlight Line 45 > Click the run command

Outputted Cooperative Volume of Conversions results can be interpreted onscreen in R.

4. Run Cooperative Model > Calculate Cooperative Value of Conversions > Highlight Line 48 > Click the run command

Outputted Cooperative Value of Conversions results can be interpreted onscreen in R.

5. Write Cooperative attribution results to CSV file > Highlight line 58 to 61 > Click the run command.

This command will save to CSV file the outputted results in terms of conversion value volume to a CSV file titled: Cooperative Attribution Model.csv


4.3 Attribution Model Results.
Results from the attribution model can be interpreted onscreen or exported to a CSV file for expanded analysis.
6 References


7 Appendix

Solution Frame is available for download via Google Drive
library(GoogleAnalytics)
# set the working directory
setwd("C:/Users/Name/Google Drive/College lecture notes - R/Learning/Google Analytics/Model/Final")

# generate the user token object
auth.token <- OAuthClient(id = "151656294-xxxxxxxxxxxxx.apps.googleusercontent.com",
            client_secret = "4T8000000000000000"
)

# save the token object to a file
save(auth.token, file = "auth_token.r"
)

# load the token object
load(auth_token.r"

# create a list of query parameters
query.list <- list(start_date = "2014-07-01",
            end.date = "2015-07-31",
            sort = "ga.data", return Returned results by data
            differentData = "ga.data", ga:session, ga:customvarValue", set set dimensions
            metrics = ga:sessions, ga:pageviews, ga:customvarValue, ga:goalCompletions, ga:goal2Completions, ga: pageviews,
            max-results = 10000, sort: "time"
)

# create the query object
ga.query <- QueryBuilder(query.list)

# execute the query
ga.cf <- GetExportData(query, auth_token)

# write the data to an Excel file
write.csv(ga.cf, file = "Google_Analytics_Conversion_Percentage_Data.csv")