

# Package ‘ChannelAttribution’

July 21, 2025

**Type** Package

**Title** Markov Model for Online Multi-Channel Attribution

**Version** 2.0.7

**Date** 2023-05-17

**Maintainer** Davide Altomare <info@channelattribution.io>

**Description** Advertisers use a variety of online marketing channels to reach consumers and they want to know the degree each channel contributes to their marketing success. This is called online multi-channel attribution problem. This package contains a probabilistic algorithm for the attribution problem. The model uses a k-order Markov representation to identify structural correlations in the customer journey data. The package also contains three heuristic algorithms (first-touch, last-touch and linear-touch approach) for the same problem. The algorithms are implemented in C++.

**License** GPL-3 | file LICENSE

**URL** <https://channelattribution.io>

**LinkingTo** Rcpp, RcppArmadillo

**Imports** Rcpp

**NeedsCompilation** yes

**Author** Davide Altomare [cre, aut],  
David Loris [aut]

**Repository** CRAN

**Date/Publication** 2023-05-17 19:20:02 UTC

## Contents

|                            |   |
|----------------------------|---|
| ChannelAttribution-package | 2 |
| auto_markov_model          | 3 |
| choose_order               | 4 |
| Data                       | 5 |
| heuristic_models           | 6 |
| markov_model               | 7 |
| transition_matrix          | 8 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>10</b> |
|--------------|-----------|

ChannelAttribution-package

*Markov Model for Online Multi-Channel Attribution*

---

**Description**

Advertisers use a variety of online marketing channels to reach consumers and they want to know the degree each channel contributes to their marketing success. This is called online multi-channel attribution problem. In many cases, advertisers approach this problem through some simple heuristics methods that do not take into account any customer interactions and often tend to underestimate the importance of small channels in marketing contribution. This package provides a function that approaches the attribution problem in a probabilistic way. It uses a k-order Markov representation to identify structural correlations in the customer journey data. This would allow advertisers to give a more reliable assessment of the marketing contribution of each channel. The approach basically follows the one presented in Eva Anderl, Ingo Becker, Florian v. Wangenheim, Jan H. Schumann (2014). Differently for them, we solved the estimation process using stochastic simulations. In this way it is also possible to take into account conversion values and their variability in the computation of the channel importance. The package also contains a function that estimates three heuristic models (first-touch, last-touch and linear-touch approach) for the same problem.

**Details**

Package: ChannelAttribution  
Type: Package  
Version: 2.0.7  
Date: 2023-05-17  
License: GPL (>= 2)

Package contains functions for channel attribution in web marketing.

**Author(s)**

Davide Altomare, David Loris

Maintainer Davide Altomare <info@channelattribution.io>

**References**

ChannelAttribution Official Website: <https://channelattribution.io>

Eva Anderl, Ingo Becker, Florian v. Wangenheim, Jan H. Schumann: Mapping the Customer Journey, 2014, doi: [10.2139/ssrn.2343077](https://doi.org/10.2139/ssrn.2343077)

---

|                   |                                |
|-------------------|--------------------------------|
| auto_markov_model | <i>Automatic Markov Model.</i> |
|-------------------|--------------------------------|

---

## Description

Estimate a Markov model from customer journey data after automatically choosing a suitable order. It requires paths that do not lead to conversion as input.

## Usage

```
auto_markov_model(Data, var_path, var_conv, var_null, var_value=NULL,
                  max_order=10, roc_npt=100, plot=FALSE, nsim_start=1e5,
                  max_step=NULL, out_more=FALSE, sep=">",
                  ncore=1, nfold=10, seed=0, conv_par=0.05, rate_step_sim=1.5,
                  verbose=TRUE, flg_adv=TRUE)
```

## Arguments

|               |                                                                                                                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data          | data.frame containing customer journeys data.                                                                                                                                       |
| var_path      | column name containing paths.                                                                                                                                                       |
| var_conv      | column name containing total conversions.                                                                                                                                           |
| var_null      | column name containing total paths that do not lead to conversions.                                                                                                                 |
| var_value     | column name containing total conversion value.                                                                                                                                      |
| max_order     | maximum Markov Model order considered.                                                                                                                                              |
| roc_npt       | number of points used for approximating roc and auc.                                                                                                                                |
| plot          | if TRUE, a plot with penalized auc with respect to order will be displayed.                                                                                                         |
| nsim_start    | minimum number of simulations used in computation.                                                                                                                                  |
| max_step      | maximum number of steps for a single simulated path. if NULL, it is the maximum number of steps found into Data.                                                                    |
| out_more      | if TRUE, transition probabilities between channels and removal effects will be shown.                                                                                               |
| sep           | separator between the channels.                                                                                                                                                     |
| ncore         | number of threads used in computation.                                                                                                                                              |
| nfold         | how many repetitions are used to verify if convergence is reached at each iteration.                                                                                                |
| seed          | random seed. Giving this parameter the same value over different runs guarantees that results will not vary.                                                                        |
| conv_par      | convergence parameter for the algorithm. The estimation process ends when the percentage of variation of the results over different repetitions is less than convergence parameter. |
| rate_step_sim | number of simulations used at each iteration is equal to the number of simulations used at previous iteration multiplied by rate_step_sim.                                          |
| verbose       | if TRUE, additional information about process convergence will be shown.                                                                                                            |
| flg_adv       | if TRUE, ChannelAttribution Pro banner is printed.                                                                                                                                  |

**Value**

An object of class `data.frame` with the estimated number of conversions and the estimated conversion value attributed to each channel.

**Author(s)**

Davide Altomare (<info@channelattribution.io>).

**Examples**

```
## Not run:

library(ChannelAttribution)

data(PathData)

auto_markov_model(Data, "path", "total_conversions", "total_null")

## End(Not run)
```

---

choose\_order

*Choose order for Markov model.*

---

**Description**

Find the minimum Markov Model order that gives a good representation of customers' behaviour for data considered. It requires paths that do not lead to conversion as input. Minimum order is found maximizing a penalized area under ROC curve.

**Usage**

```
choose_order(Data, var_path, var_conv, var_null, max_order=10, sep=">",
             ncore=1, roc_npt=100, plot=TRUE, flg_adv=TRUE)
```

**Arguments**

|                        |                                                                            |
|------------------------|----------------------------------------------------------------------------|
| <code>Data</code>      | data.frame containing customer journeys.                                   |
| <code>var_path</code>  | column name of Data containing paths.                                      |
| <code>var_conv</code>  | column name of Data containing total conversions.                          |
| <code>var_null</code>  | column name of Data containing total paths that do not lead to conversion. |
| <code>max_order</code> | maximum Markov Model order considered.                                     |
| <code>sep</code>       | separator between channels.                                                |
| <code>ncore</code>     | number of threads used in computation.                                     |

|         |                                                                             |
|---------|-----------------------------------------------------------------------------|
| roc_npt | number of points used for approximating roc and auc.                        |
| plot    | if TRUE, a plot with penalized auc with respect to order will be displayed. |
| flg_adv | if TRUE, ChannelAttribution Pro banner is printed.                          |

**Value**

An object of class `List` with the estimated roc, auc and penalized auc.

**Author(s)**

Davide Altomare (<info@channelattribution.io>).

**Examples**

```
## Not run:

library(ChannelAttribution)

data(PathData)

res=choose_order(Data, var_path="path", var_conv="total_conversions",
                 var_null="total_null")

#plot auc and penalized auc

plot(res$auc$order, res$auc$auc, type="l", xlab="order", ylab="pauc", main="AUC")
lines(res$auc$order, res$auc$pauc, col="red")
legend("right", legend=c("auc", "penalized auc"),
      col=c("black", "red"), lty=1)

## End(Not run)
```

---

|      |                                |
|------|--------------------------------|
| Data | <i>Customer journeys data.</i> |
|------|--------------------------------|

---

**Description**

Example dataset.

**Usage**

```
data(PathData)
```

**Format**

Data is a `data.frame` with 10.000 rows and 4 columns: "path" containing customer paths, "total\_conversions" containing total number of conversions, "total\_conversion\_value" containing total conversion value and "total\_null" containing total number of paths that do not lead to conversion.

---

|                  |                                                             |
|------------------|-------------------------------------------------------------|
| heuristic_models | <i>Heuristic models for the online attribution problem.</i> |
|------------------|-------------------------------------------------------------|

---

**Description**

Estimate three heuristic models (first-touch, last-touch and linear) from customer journey data.

**Usage**

```
heuristic_models(Data, var_path, var_conv, var_value=NULL, sep=">", flg_adv=TRUE)
```

**Arguments**

|           |                                                    |
|-----------|----------------------------------------------------|
| Data      | data.frame containing paths and conversions.       |
| var_path  | column name containing paths.                      |
| var_conv  | column name containing total conversions.          |
| var_value | column name containing total conversion value.     |
| sep       | separator between the channels.                    |
| flg_adv   | if TRUE, ChannelAttribution Pro banner is printed. |

**Value**

An object of class data.frame with the estimated number of conversions and the estimated conversion value attributed to each channel for each model.

**Author(s)**

Davide Altomare (<info@channelattribution.io>).

**Examples**

```
## Not run:

library(ChannelAttribution)

data(PathData)

heuristic_models(Data,"path","total_conversions")
heuristic_models(Data,"path","total_conversions",var_value="total_conversion_value")

## End(Not run)
```

markov\_model

*Markov model for the online attribution problem.***Description**

Estimate a k-order Markov model from customer journey data. Differently from markov\_model, this function iterates estimation until convergence is reached and enables multiprocessing.

**Usage**

```
markov_model(Data, var_path, var_conv, var_value=NULL, var_null=NULL,
             order=1, nsim_start=1e5, max_step=NULL, out_more=FALSE, sep=">",
             ncore=1, nfold=10, seed=0, conv_par=0.05, rate_step_sim=1.5,
             verbose=TRUE, flg_adv=TRUE)
```

**Arguments**

|               |                                                                                                                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data          | data.frame containing customer journeys data.                                                                                                                                       |
| var_path      | column name containing paths.                                                                                                                                                       |
| var_conv      | column name containing total conversions.                                                                                                                                           |
| var_value     | column name containing total conversion value.                                                                                                                                      |
| var_null      | column name containing total paths that do not lead to conversions.                                                                                                                 |
| order         | Markov Model order.                                                                                                                                                                 |
| nsim_start    | minimum number of simulations used in computation.                                                                                                                                  |
| max_step      | maximum number of steps for a single simulated path. if NULL, it is the maximum number of steps found into Data.                                                                    |
| out_more      | if TRUE, transition probabilities between channels and removal effects will be returned.                                                                                            |
| sep           | separator between the channels.                                                                                                                                                     |
| ncore         | number of threads used in computation.                                                                                                                                              |
| nfold         | how many repetitions are used to verify if convergence has been reached at each iteration.                                                                                          |
| seed          | random seed. Giving this parameter the same value over different runs guarantees that results will not vary.                                                                        |
| conv_par      | convergence parameter for the algorithm. The estimation process ends when the percentage of variation of the results over different repetitions is less than convergence parameter. |
| rate_step_sim | number of simulations used at each iteration is equal to the number of simulations used at previous iteration multiplied by rate_step_sim.                                          |
| verbose       | if TRUE, additional information about process convergence will be shown.                                                                                                            |
| flg_adv       | if TRUE, ChannelAttribution Pro banner is printed.                                                                                                                                  |

**Value**

An object of class `data.frame` with the estimated number of conversions and the estimated conversion value attributed to each channel.

**Author(s)**

Davide Altomare (<info@channelattribution.io>).

**Examples**

```
## Not run:

library(ChannelAttribution)

data(PathData)

#Estimate a Makov model using total conversions
markov_model(Data, var_path="path", "total_conversions")

#Estimate a Makov model using total conversions and revenues
markov_model(Data, "path", "total_conversions",
var_value="total_conversion_value")

#Estimate a Makov model using total conversions, revenues and paths that do not lead to conversions
markov_model(Data, "path", "total_conversions",
var_value="total_conversion_value", var_null="total_null")

#Estimate a Makov model returning transition matrix and removal effects
markov_model(Data, "path", "total_conversions",
var_value="total_conversion_value", var_null="total_null", out_more=TRUE)

#Estimate a Markov model using 4 threads
markov_model(Data, "path", "total_conversions",
var_value="total_conversion_value", ncore=4)

## End(Not run)
```

---

|                   |                           |
|-------------------|---------------------------|
| transition_matrix | <i>Transition matrix.</i> |
|-------------------|---------------------------|

---

**Description**

Estimate a k-order transition matrix from customer journey data.

**Usage**

```
transition_matrix(Data, var_path, var_conv, var_null, order=1, sep=">",
flg_equal=TRUE, flg_adv=TRUE)
```

**Arguments**

|           |                                                                   |
|-----------|-------------------------------------------------------------------|
| Data      | data.frame containing customer journeys data.                     |
| var_path  | column name containing paths.                                     |
| var_conv  | column name containing total conversions.                         |
| var_null  | column name containing paths that do not lead to conversions.     |
| order     | Markov Model order.                                               |
| sep       | separator between the channels.                                   |
| flg_equal | if TRUE, transitions from a channel to itself will be considered. |
| flg_adv   | if TRUE, ChannelAttribution Pro banner is printed.                |

**Value**

An object of class List containing a dataframe with channel names and a dataframe with the estimated transition matrix.

**Author(s)**

Davide Altomare (<info@channelattribution.io>).

**Examples**

```
## Not run:

library(ChannelAttribution)

data(PathData)

transition_matrix(Data, var_path="path", var_conv="total_conversions",
                  var_null="total_null", order=1, sep=">", flg_equal=TRUE)

transition_matrix(Data, var_path="path", var_conv="total_conversions",
                  var_null="total_null", order=3, sep=">", flg_equal=TRUE)

## End(Not run)
```

# Index

- \* **channel attribution**
  - ChannelAttribution-package, [2](#)
- \* **channel marketing**
  - ChannelAttribution-package, [2](#)
- \* **choose markov graph order**
  - choose\_order, [4](#)
- \* **choose markov model order**
  - choose\_order, [4](#)
- \* **customer journey dataset**
  - Data, [5](#)
- \* **customer journey**
  - ChannelAttribution-package, [2](#)
- \* **customer path data**
  - Data, [5](#)
- \* **dataset**
  - Data, [5](#)
- \* **first touch**
  - heuristic\_models, [6](#)
- \* **last touch**
  - heuristic\_models, [6](#)
- \* **linear touch**
  - heuristic\_models, [6](#)
- \* **marketing attribution**
  - ChannelAttribution-package, [2](#)
- \* **markov graph**
  - auto\_markov\_model, [3](#)
  - markov\_model, [7](#)
  - transition\_matrix, [8](#)
- \* **markov model**
  - auto\_markov\_model, [3](#)
  - markov\_model, [7](#)
  - transition\_matrix, [8](#)
- \* **multi channel funnel**
  - ChannelAttribution-package, [2](#)
- \* **multi channel marketing**
  - ChannelAttribution-package, [2](#)
- \* **online attribution**
  - ChannelAttribution-package, [2](#)
- \* **web marketing**
  - ChannelAttribution-package, [2](#)
- \* **web statistics**
  - ChannelAttribution-package, [2](#)
- auto\_markov\_model, [3](#)
- ChannelAttribution
  - (ChannelAttribution-package), [2](#)
  - ChannelAttribution-package, [2](#)
  - choose\_order, [4](#)
- Data, [5](#)
- heuristic\_models, [6](#)
- markov\_model, [7](#)
- transition\_matrix, [8](#)