ChannelAttribution Pro 3 Handbook

Davide Altomare and David Loris

2023-10-19

channelattribution.io

What's ChannelAttribution Pro 3

ChannelAttribution Pro 3 is a machine learning library for data-driven marketing attribution from customer-journey data. It is an **R package** and a **Python library** available for the main Operative Systems (**Linux**, **Windows** and **Mac**). We can also provide a preconfigured **Docker container** with RStudio or Jupyter and *ChannelAttribution Pro* installed.

ChannelAttribution Pro 3 is installed locally and all the elaborations are made on the local system where it is installed. It means that no data are transferred outside your organization.

What's in ChannelAttribution Pro 3

ChannelAttribution Pro 3 improves the open-source library *ChannelAttribution* by offering the following additional features:

FEATURE	VALUE	FUNCTION
Transaction-level attribution with heuristic models, Markov model and Shapley value	• Monitor ROI for each channel at path-level and for aggregation of paths at time intervals	heuristic_models markov_model shapley new_paths_attribution combine_mta_mmm
Real-time attribution with Markov model and Shapley value	• Save computational time. Train the model on huge amount of customer journeys, store the model parameters and then use it for performing attribution on new customer-journeys	new_paths_attribution

Markov model and Shapley value with odds	• More accurate attribution at path-level	markov_model shapley
Combine results from Media-mix Model and Multi-touch attribution models at path-level	• More accurate attribution at path-level bringing results from a Media-mix model at path-level	combine_mta_mmm
Out-of-sample validation algorithm for choosing the best Markov model order	• More accurate attribution with Markov models • Choose the best order also for highly imbalanced data using precision-recall curve instead of roc curve	choose_order
Simplified Shapley value formula	• Classical Shapley value formula limit the use of Shapley value to problems with less than 10 channels, while simplified Shapley value can be used also with thousands of channels	shapley
Multiprocessing	• Faster execution of Markov model when huge amounts of customer journeys are elaborated	markov_model
Read customer journeys directly from CSV files	• Process huge amount of customer journeys avoiding out-of-memory issues	heuristic_models markov_model shapley
Perform budget allocation with Markov model	• Improve your budget allocation increasing your ROI	markov_budget_allocation
Predict next best action with Markov model	• Guide customers along journeys to maximize the conversion probability	next_best_action

ChannelAttribution Pro 3 VS ChannelAttribution Pro 2

ChannelAttribution Pro 3 improves *ChannelAttribution Pro 2* by offering the following additional features:

- Transaction-level attribution and real-time attribution with classical and simplified **Shapley** value formula and odds

- Perform attribution with the **Hidden Touch Attribution model** using aggregated traffic data from digital and traditional channels
- Out-of-sample validation algorithm for choosing the best Markov model order for **highly imbalanced data** using precision-recall curve
- Read customer journeys directly from **CSV files** avoiding out-of memory issues
- Improve your budget allocation using Markov model
- Predict next best action with Markov model

1 Installation

1.1 Installation / Updating

This section will show how *ChannelAttribution Pro 3* can be installed or updating.

Python

Download and run the following script into a new python session:

Python installation script

R

Download and run the following script into a new R session:

R installation script

1.2 Require a password

A password can be obtained filling in the following form

1.3 Testing

This section will show how you can test the correct installation of *ChannelAttribution Pro 3*.

Python

Download the following script:

Python test script

Replace:

password="..."
with your password. (Ask for a password to info@channelattribution.io)
Run it into a python session.

R

Download the following script:

R test script

Replace:

password="..."

with your password. (Ask for a password to info@channelattribution.io)

Run it into an R session.

2 Heuristic models

Transaction-level attribution with heuristic models (last touch, first touch and linear touch) can be performed with function *heuristic_ models*.

2.1 Function *heuristic_ models*

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
Data	data.frame/str		data.frame or a file address where customer journeys are stored.
var_path	str		name of the column containing paths.
var_conv	str		name of the column containing total conversions.
var_value	str	None	name of the column containing total conversion value.
var_null	str	None	name of the column containing total paths that do not lead to conversion.
row_sep	str	","	if *Data* is a file address then *row _sep* is the line separator.
cha_sep	str	">"	separator between channels.
flg_write_nulls	bool	True	If *True* then non converting paths will be returned in path attribution output.
flg_write_paths	bool	False	If *True* then paths will be returned in the path attribution output.
file_output	str	None	file address where path attribution will be written.
server	str	"app.channelattribution.net"	address of the server where password will be checked to authorize the execution of the function.
password	str	None	user password.

Output

OUTPUT	TYPE	DESCRIPTION
attribution	data.frame	path-level attribution.

2.2 Examples

2.2.1 Documentation

3 Markov Model

Transaction-level attribution with Markov models can be performed with function *markov_ model* [1].

3.1 Function *markov_ model*

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
Data	data.frame/str		data.frame or a file address where customer journeys are stored.
var_path	str		name of the column containing paths.
var_conv	str		name of the column containing total conversions.
var_value	str	None	name of the column containing total conversion value.
var_null	str	None	name of the column containing total paths that do not lead to conversion.
row_sep	str	","	if *Data* is a file address then *row _sep* is the line separator.
cha_sep	str	">"	separator between channels.
type	str	"odds"	type of quantity used for attribution. It can be equal to "re" (removal effect), "cr" (conversion rate), "odds" (classical odds), "diff_odds" (differential odds) or "exp_odds" (exponential odds).
order	int	1	Markov model order.
nsim_start	int	1e5	minimum number of simulations to be used in computation.
max_step	int	None	maximum number of length for a single simulated path. if *None*, it is the maximum length for a path belonging to Data.
ncore	int	1	number of threads to be used in computation.
nfold	int	10	how many repetitions to be used to verify if convergence has been reached at each iteration.
seed	int	1234567	random seed. Giving this parameter the same value over different runs guarantees that results will not vary.
conv_par	double	0.05	convergence parameter for the algorithm. The estimation process ends when the percentage of variation of the results over different repetions is less than convergence parameter.
rate_step_sim	double	1.5	number of simulations used at each iteration is equal to the number of simulations used at previous iteration multiplied by *rate_step_sim*.
verbose	bool	True	if *True*, additional information during the execution will be shown.
flg_out_tran_mtx	bool	False	if True, only transition matrix will be returned.
file_output	str	None	file address where path attribution will be written.
flg_write_nulls	bool	True	If *True* then non converting paths will be returned in path attribution output.
flg_write_paths	bool	False	If *True* then paths will be returned in the path attribution output.
server	str	"app.channelattribution.net"	address of the server where password will be checked to authorize the execution of the function.
password	str	None	user password.

Output

OUTPUT	TYPE	DESCRIPTION
parameters	data.frame	parameters for transaction-level attribution.
attribution	data.frame	transaction-level attribution.
transition_matrix	data.frame	transition matrix built from paths belonging to *Data*.

3.2 Examples

3.2.1 Documentation

4 Shapley value

Transaction-level attribution with Shapley value can be performed with function *shapley* [3].

4.1 Function *shapley*

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
Data	data.frame/str		data.frame or a file address where customer journeys are stored.
var_path	str		name of the column containing paths.
var_conv	str		name of the column containing total conversions.
var_value	str	None	name of the column containing total conversion value.
var_null	str	None	name of the column containing total paths that do not lead to conversion.
row_sep	str	","	if *Data* is a file address then *row_sep* is the line separator.
cha_sep	str	">"	separator between channels.
flg_simplified	bool	True	if *True* then simplified formula for Shapley value will be used.
type_worth	str	"odds"	type of quantity used for attribution. It can be equal to "sum" (sum of conversions), "cr" (conversion rate), "odds" (classical odds), "diff_odds" (differential odds) or "exp_odds" (exponential odds) .
verbose	bool	True	if *True*, additional information during the execution will be shown.
file_output	str	None	file address where path attribution will be written.
flg_write_nulls	bool	True	If *True* then non converting paths will be returned in path attribution output.
flg_write_paths	bool	False	If *True* then paths will be returned in the path attribution output.
server	str	"app.channelattribution.net"	address of the server where password will be checked to authorize the execution of the function.
password	str	None	user password.

Output

OUTPUT	TYPE	DESCRIPTION
parameters	data.frame	parameters for path-level attribution.
attribution	data.frame	path-level attribution.

4.2 Examples

4.2.1 Documentation

5 Selecting the best Markov model order

ChannelAttribution Pro 3 includes an *out-of-sample* algorithm for choosing the best Markov model order. First, the data are split into a train set and a test set. Using the train set a Markov model is estimated for each considered order. Each Markov model is used to predict the end state (conversion/no conversion) for each customer journey on the test set. For each Markov model, a ROC curve is defined and the area under the curve is calculated (AUC). The procedure is repeated on multiple test sets which are randomly chosen from the full data set (cross-validation procedure). For each order, an average AUC over all the test sets considered is calculated. The order with the maximum average AUC is finally chosen.

Best Markov model order in *ChannelAttribution Pro 3* can be choosen through *choose_order* function which incorporate the *out-of-sample* procedure procedure explained above.

5.1 Function *choose_order*

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
Data	data.frame/str		data.frame or a file address where customer journeys are stored.
var_path	str		name of the column containing paths.
var_conv	str		name of the column containing total conversions.
var_value	str	None	name of the column containing total conversion value.
var_null	str	None	name of the column containing total paths that do not lead to conversion.
row_sep	str	","	if *Data* is a file address then *row_sep* is the line separator.
cha_sep	str	">"	separator between channels.
roc_npt	int	100	number of points in ROC.
max_order	int	10	maximum Markov model order to be considered.
nfold	int	10	how many repetitions to be used to verify if convergence has been reached at each iteration.
perc_test	double	0.3	percentage of customer journeys that will be included in the test set.
seed	int	1234567	random seed. Giving this parameter the same value over different runs guarantees that results will not vary.
perc_tol	double	0.01	percentage of tolerance. If order o has an AUC(o) which is greater than (1-perc_tol) x AUC(o+1) then order o is consider better than *o+1*.
plot	bool	True	if *True*, a plot with auc will be displayed.
type	str	"auc-roc"	if *"auc-roc"*, area under ROC curve will be calculated, if *"auc-prerec"* area under Precision-Recall curve will be calculated.
verbose	bool	True	if *True*, additional information during the computation will be shown
server	str	"app.channelattribution.net"	address of the server where password will be checked to authorize the execution of the function.
password	str	None	user password.

Output

OUTPUT	TYPE	DESCRIPTION
auc	data.frame	AUC for each analyzed order.
best_order	int	best oreder selected by the procedure.

5.2 Examples

5.2.1 Documentation

6 Transaction-level attribution on new paths

ChannelAttribution Pro 3 lets you train a Markov or Shapley model and then apply it to new customer journeys. So it is easy and fast to make transaction-level attribution in a real-time context.

6.1 Function *new_paths_attribution*

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
Data	data.frame/str		data.frame or a file address where customer journeys are stored.
var_path	str		name of the column containing paths.
var_conv	str		name of the column containing total conversions.
Dparams	str	None	name of the column containing total paths that do not lead to conversion.
var_value	str	None	name of the column containing total conversion value.
row_sep	str	","	if *Data* is a file address then *row _sep* is the line separator.
cha_sep	str	">"	separator between channels.
file_output	str	None	file address where path attribution will be written.
flg_write_nulls	bool	True	If *True* then non converting paths will be returned in path attribution output.
flg_write_paths	bool	False	If *True* then paths will be returned in the path attribution output.
verbose	bool	True	if True, additional information during the execution will be shown.
server	str	"app.channelattribution.net"	address of the server where password will be checked to authorize the execution of the function.
password	str	None	user password.

Output

OUTPUT	TYPE	DESCRIPTION
attribution	data.frame	transction level attribution

6.2 Examples

6.2.1 Documentation

7 Combining attribution from media-mix model and multi-touch model

ChannelAttribution Pro 3 lets to combine results from attribution performed by a multi-touch model and that by a media-mix model.
Function *combine_mta_mmm* lets to combine transaction-level attribution of a multi-touch model with global attribution performed using a media-mix model, producing a new transaction-level attribution that combines both.

7.1 Function *combine_mta_mmm*

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
mta_path_attribution	data.frame		data.frame containing transaction-level MTA.
mmm_attribution	data.frame		data.frame containing global MMA.
prior_weights_mta	data.frame	None	data.frame containing subjective relative weights for each channel. Each weight can reduce or increase the weight of MTA with respect to MMA for each channel, in the final combined attribution.
max_steps	int	100	maximum number of iterations of the optimization process.
conv_rate	str	0.01	convergence rate of the optimization process.
verbose	bool	True	if *True*, additional information during the computation will be shown
server	str	"app.channelattribution.net"	address of the server where password will be checked to authorize the execution of the function.
password	str	None	user password.

Output

OUTPUT	TYPE	DESCRIPTION
attribution	data.frame	path-level attribution.

7.2 Algorithm

If K is the number of the available digital channels, N is the number of the observed customer journeys and C is the total number of conversions observed. Let

$$\text{GMTA} = (\hat{\theta}_1^{GMTA}C, \ldots, \hat{\theta}_K^{GMTA}C)$$

the global multi-touch attribution where $\hat{\theta}_k$ is the relative weight estimated for channel k and

$$\text{GMMA} = (\hat{\theta}_1^{GMMA}C, \ldots, \hat{\theta}_K^{GMMA}C)$$

the global media-mix attribution. If

$$w = (w_1, \ldots, w_K)$$

is a vector of prior weights for MTA then the final global attribution can be defined as:

$$\text{gFA} = (w)\text{GMTA} + (1 - w)\text{GMMA}$$

and we have that:

$$\text{GFA} = (\hat{\theta}_1^{GFA}C, \ldots, \hat{\theta}_K^{GFA}C)$$

Now we need to estimate:

$$\theta^{PFA} = (\theta_1^{PFA}, \ldots, \theta_K^{PFA})$$

which is the vector of the relative weights for transaction-level attribution of the final model. This can be done by solving the following optimization problem:

$$\hat{\theta}^{PFA} : \|GFA(\hat{\theta}^{GFA}) - PFA(\hat{\theta}^{PFA})\| = \min_{\theta^{PFA}} \|GFA(\hat{\theta}^{GFA}) - PFA(\theta^{PFA})\|$$

7.3 Examples

7.3.1 Documentation

8 Budget allocation with Markov model

Allocate your budget to your marketing channels using the attribution performed through Markov model. Budget allocation with Markov model can be performed with function *markov_budget_allocation*.

8.1 Function *markov_budget_allocation*

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
res_markov	list		list of data.frame containing the output of a markov_model function.
total_budget_new	double		overall budget you want allocate.
tab_costs	data.frame	None	data.frame containing the spend for each channel in the time period when customer journeys have been observed. The parameter is optional and can be set to None if spends are not known.
perc_reall	double	0.1	percentage of the overall budget that will be reallocate. (1-perc_reall) is the percentage of the overall budget that will allocated as in the last allocation. Since the allocation algorithm is a local optimization algorithm we suggest allocating a small percentage of the overall budget each time.
min_perc_budget	double	0.01	percentage of the overall budget that will be allocated equally through all the channels involved. This avoids that for one or more channels the allocation can be 0.

Output

OUTPUT	TYPE	DESCRIPTION
allocation	data.frame	suggested budget allocation.

8.2 Examples

8.2.1 Documentation

9 Next best action

9.1 Function *next_best_action_train*

Train a list of Markov models to predict the next best action in a customer journey.

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
Data	data.frame/str		data.frame or a file address where customer journeys are stored.
var_path	str		name of the column containing paths.
var_conv	str		name of the column containing total conversions.
var_null	str	None	name of the column containing total paths that do not lead to conversion.
row_sep	str	","	if *Data* is a file address then *row _sep* is the line separator.
cha_sep	str	">"	separator between channels.
max_order	int	3	maximum Markov model order to be considered .
nsim_start	int	1e5	minimum number of simulations to be used in computation.
max_step	int	None	maximum number of length for a single simulated path. if *None*, it is the maximum length for a path belonging to Data.
ncore	int	1	number of threads to be used in computation.
nfold	int	10	how many repetitions to be used to verify if convergence has been reached at each iteration.
seed	int	1234567	random seed. Giving this parameter the same value over different runs guarantees that results will not vary.
conv_par	double	0.05	convergence parameter for the algorithm. The estimation process ends when the percentage of variation of the results over different repetions is less than convergence parameter.
rate_step_sim	double	1.5	number of simulations used at each iteration is equal to the number of simulations used at previous iteration multiplied by *rate_ step_ sim*.
verbose	bool	True	if *True*, additional information during the execution will be shown.
server	str	"app.channelattribution.net"	address of the server where password will be checked to authorize the execution of the function.
password	str	None	user password.

Output

OUTPUT	TYPE	DESCRIPTION
channels	vector	channel names.
conversion_rates	data.frame	conversion rates.

9.2 Function *next_ best_ action*

Predict the next best action in a customer journey using the trained models with next_best_action_train.

Parameters

PARAMETER	TYPE	DEFAULT	DESCRIPTION
new_path	str		customer journey.
Params	list		output from next_best_action_train.
sep	str	">"	separator between channels.
server	str	"app.channelattribution.net"	address of the server where password will be checked to authorize the execution of the function.
password	str	None	user password.

Output

OUTPUT	TYPE	DESCRIPTION
suggested_action	str	next suggested channel to visit.
suggested_action_conversion_rate	double	conversion rate for the next suggested channel to visit.
actions	data.frame	conversion rate for each available channel.

9.3 Examples

9.3.1 Documentation

References

[1] Altomare and Loris (2022), Multi-touch attribution and budget allocation.

[2] Anderl E. et al. (2014), Mapping the Customer Journey: A Graph-Based Framework for Online Attribution Modeling.

[3] Zhao K. et al. (2012), Shapley Value Methods for Attribution Modeling in Online Advertising.