Análisis y evaluación de la sensibilidad al precio utilizando el paquete *partykit*

Jorge Martín Arevalillo

Departamento de Estadística e Investigación Operativa. UNED. Madrid. Spain

jmartin@ccia.uned.es

Noviembre 2019

Outline

A quick glance at my profile

- Research in both the theoretical and applied sides of statistical learning and machine learning. A wholehearted defender of the mathematical rigour to support applications.
- Author or coauthor of applied papers in different fields: biomedicine, biotechnology, environmental sciences, finance...
- Strongly committed with the transfer of mathematics to the industry.
- Leadership for several data science transfer projects in banking.
- Awarded by Santander Bank with the Research and Transfer prize in 2018 as a recognition for such commitment.

A quick glance at my profile

- Research in both the theoretical and applied sides of statistical learning and machine learning. A wholehearted defender of the mathematical rigour to support applications.
- Author or coauthor of applied papers in different fields: biomedicine, biotechnology, environmental sciences, finance...
- Strongly committed with the transfer of mathematics to the industry.
- Leadership for several data science transfer projects in banking.
- Awarded by Santander Bank with the Research and Transfer prize in 2018 as a recognition for such commitment.
- An enthusiastic of R !

Introductory highlights

- Price sensitivity (PS) is concerned with the sensitivity of customers to the price when they receive a bid to buy a product.

Introductory highlights

- Price sensitivity (PS) is concerned with the sensitivity of customers to the price when they receive a bid to buy a product.
- A valuable business goal is the assessment of PS in order to identify groups with differential PS so that customized pricing strategies can be deployed accordingly.

Introductory highlights

- Price sensitivity (PS) is concerned with the sensitivity of customers to the price when they receive a bid to buy a product.
- A valuable business goal is the assessment of PS in order to identify groups with differential PS so that customized pricing strategies can be deployed accordingly.
- Machine learning provides useful tools to achieve this goal: Decision trees and model based (MOB) recursive partitioning are well suited methods to address this problem.

Introductory highlights

- Price sensitivity (PS) is concerned with the sensitivity of customers to the price when they receive a bid to buy a product.
- A valuable business goal is the assessment of PS in order to identify groups with differential PS so that customized pricing strategies can be deployed accordingly.
- Machine learning provides useful tools to achieve this goal: Decision trees and model based (MOB) recursive partitioning are well suited methods to address this problem.
- The *partykit* package has an implementation of MOB.

Introductory highlights

- Price sensitivity (PS) is concerned with the sensitivity of customers to the price when they receive a bid to buy a product.
- A valuable business goal is the assessment of PS in order to identify groups with differential PS so that customized pricing strategies can be deployed accordingly.
- Machine learning provides useful tools to achieve this goal: Decision trees and model based (MOB) recursive partitioning are well suited methods to address this problem.
- The *partykit* package has an implementation of MOB.
- When applied to the business case of an auto lending company, taking the interest rate of the loan as the price variable, the results reveal the existence of customer groups that exhibit differential PS. This finding highlights useful business insights to guide pricing decision making.

CART algorithm (Breiman et.al. 1984)

The decision tree is a data-driven method for the recursive partitioning of a database by the search of optimal splitting variables within a set of segmentation variables from an input vector $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_q)$.

CART algorithm (Breiman et.al. 1984)

The decision tree is a data-driven method for the recursive partitioning of a database by the search of optimal splitting variables within a set of segmentation variables from an input vector $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_q)$.

- CART algorithm accomplishes the recursive partitioning through binary splits that generate a tree structure. The splits are obtained by assessing the impurity of an outcome variable Y at parent and descendant nodes using impurity measures like Gini or Entropy.
- CART explores the set of input variables and looks for the variable and splitting point that maximizes the impurity decrease.
- CART trees are grown in a recursive way until a large tree structure is obtained. Then the large tree is pruned by means of an intelligent strategy that eliminates uninformative branches to avoid overfitting.
- An implementation of CART is provided by the *rpart* R package.

CTREE algorithm (Hothorn, Hornik and Zeileis, 2006)

- One weakness of CART is its bias towards the selection of splitting variables with many categories. The CTREE algorithm provides an alternative approach to overcome this bias problem.
- It takes the *p*-value from permutation tests, defined by function-based statistics of the inputs, as a criterion to find the best splitting variable and cutoff point.
- CTREE can control the splitting bias but it has no pruning strategy like CART; so in principle the stopping rule must be set in advance.
- The rule may consist of a significance level threshold for the aforementioned tests, above which a node is declared as terminal (default $\alpha = 0{,}05$), or a minimum size for the descendant nodes.
- The *party* package contains an easy to use implementation of CTREE.

Model based partitioning

- Let $\mathbf{X} = (X_1, X_2, \ldots, X_p)$ be a vector of covariables and let us denote by $\mathcal{M}(\mathbf{X}, Y; \Theta)$ a parametric model that explains the input-outcome association.

Model based partitioning

- Let $\mathbf{X} = (X_1, X_2, \ldots, X_p)$ be a vector of covariables and let us denote by $\mathcal{M}(\mathbf{X}, Y; \Theta)$ a parametric model that explains the input-outcome association.
- Let us assume that the model $\mathcal{M}$ is fitted by optimizing an objective function $\Psi(\mathbf{X}, Y; \Theta)$. Typical objective functions are those ones defined by the ordinary least squares (OLS) or the maximum likelihood (ML) methods.

Model based partitioning

- Let $\mathbf{X} = (X_1, X_2, \ldots, X_p)$ be a vector of covariables and let us denote by $\mathcal{M}(\mathbf{X}, Y; \Theta)$ a parametric model that explains the input-outcome association.
- Let us assume that the model $\mathcal{M}$ is fitted by optimizing an objective function $\Psi(\mathbf{X}, Y; \Theta)$. Typical objective functions are those ones defined by the ordinary least squares (OLS) or the maximum likelihood (ML) methods.
- MOB rests upon the idea of the recursive partitioning of data, by means of segmentation variables from an input vector $\mathbf{Z}$, which are selected by the algorithm on the basis of fitting parametric models. The building block of MOB algorithm is the decision tree method.

MOB algorithm (Zeileis, Hothorn and Hornik, 2008)

- The goal is the search of non-overlapping groups in data, defined by the segmentation variables, such that the model $\mathcal{M}(\mathbf{X}, Y; \Theta)$ exhibits differential fits on each group. This is accomplished by assessing the stability of the parametric model through fluctuation tests, which are well-established inferential tools for testing parameter stability.

MOB algorithm (Zeileis, Hothorn and Hornik, 2008)

- The goal is the search of non-overlapping groups in data, defined by the segmentation variables, such that the model $\mathcal{M}(\mathbf{X}, Y; \Theta)$ exhibits differential fits on each group. This is accomplished by assessing the stability of the parametric model through fluctuation tests, which are well-established inferential tools for testing parameter stability.
- The MOB method is implemented by a greedy search algorithm that finds the segmentation variable yielding the highest instability, as assessed by the significance level of the fluctuation test.

MOB algorithm (Zeileis, Hothorn and Hornik, 2008)

- The goal is the search of non-overlapping groups in data, defined by the segmentation variables, such that the model $\mathcal{M}(\mathbf{X}, Y; \Theta)$ exhibits differential fits on each group. This is accomplished by assessing the stability of the parametric model through fluctuation tests, which are well-established inferential tools for testing parameter stability.
- The MOB method is implemented by a greedy search algorithm that finds the segmentation variable yielding the highest instability, as assessed by the significance level of the fluctuation test.

MOB algorithm (Zeileis, Hothorn and Hornik, 2008)

- The goal is the search of non-overlapping groups in data, defined by the segmentation variables, such that the model $\mathcal{M}(\mathbf{X}, Y; \Theta)$ exhibits differential fits on each group. This is accomplished by assessing the stability of the parametric model through fluctuation tests, which are well-established inferential tools for testing parameter stability.
- The MOB method is implemented by a greedy search algorithm that finds the segmentation variable yielding the highest instability, as assessed by the significance level of the fluctuation test.

MOB recursive partitioning method

Set the outcome Y, the vector of covariables $\mathbf{X}$ and the input vector $\mathbf{Z}$ containing the candidate variables for segmentation
Set the significance level threshold to assess parameter instability (default $\alpha = 0{,}05$)

Step 1. Fit the parametric model $\mathcal{M}(\mathbf{X}, Y; \Theta)$ to the data

Step 2. Test parameter stability for the variables in the input vector $\mathbf{Z}$

Step 3. Find the most significant variable, say Z_l.
If its significance is higher than α then stop and declare the node as terminal
else split in Z_l, by finding the cutoff point that locally optimizes Ψ, in order to get descendant nodes

Step 4. Go to step 1 and repeat the procedure for each one of the descendant nodes

MOB for price sensitivity assessment

- Jorge M Arevalillo (2019) *A machine learning approach to assess price sensitivity with application to automobile loan segmentation*. Applied Soft Computing. 76. 390-399.

MOB for price sensitivity assessment

- Jorge M Arevalillo (2019) *A machine learning approach to assess price sensitivity with application to automobile loan segmentation*. Applied Soft Computing. 76. 390-399.
- Within the PS scenario, the outcome is a binary variable Y such that $Y = 1$ if a customer accepted the bid and $Y = 0$ otherwise. Here, we consider only one covariable, X, which is the price variable.

MOB for price sensitivity assessment

- Jorge M Arevalillo (2019) *A machine learning approach to assess price sensitivity with application to automobile loan segmentation*. Applied Soft Computing. 76. 390-399.
- Within the PS scenario, the outcome is a binary variable Y such that $Y = 1$ if a customer accepted the bid and $Y = 0$ otherwise. Here, we consider only one covariable, X, which is the price variable.
- In this case the general model $\mathcal{M}$ is set to the *Logit*

$$\log \frac{P(Y = 1|X)}{P(Y = 0|X)} = \theta_0 + \theta_1 X. \tag{1}$$

MOB for price sensitivity assessment

- Jorge M Arevalillo (2019) *A machine learning approach to assess price sensitivity with application to automobile loan segmentation*. Applied Soft Computing. 76. 390-399.
- Within the PS scenario, the outcome is a binary variable Y such that $Y = 1$ if a customer accepted the bid and $Y = 0$ otherwise. Here, we consider only one covariable, X, which is the price variable.
- In this case the general model $\mathcal{M}$ is set to the *Logit*

$$\log \frac{P(Y = 1|X)}{P(Y = 0|X)} = \theta_0 + \theta_1 X. \tag{1}$$

- The coefficient θ_1 can be interpreted as a PS measure with the more negative values for θ_1 corresponding to the higher sensitivities.

MOB for price sensitivity assessment

- Jorge M Arevalillo (2019) *A machine learning approach to assess price sensitivity with application to automobile loan segmentation*. Applied Soft Computing. 76. 390-399.
- Within the PS scenario, the outcome is a binary variable Y such that $Y = 1$ if a customer accepted the bid and $Y = 0$ otherwise. Here, we consider only one covariable, X, which is the price variable.
- In this case the general model $\mathcal{M}$ is set to the *Logit*

$$\log \frac{P(Y = 1|X)}{P(Y = 0|X)} = \theta_0 + \theta_1 X. \tag{1}$$

- The coefficient θ_1 can be interpreted as a PS measure with the more negative values for θ_1 corresponding to the higher sensitivities.
- Hence, the MOB tree provides a data partition in non-overlapping groups that exhibit differential PS, as assessed by the differential fits of the model in equation (1).

Implementation with *partykit*

Achim Zeileis, Torsten Hothorn (2015) *Parties , Models , Mobsters : A New Implementation of Model-Based Recursive Partitioning in R*.
The MOB method is implemented by the simple interface function:

```
mob(formula, data, subset, na.action, weights, offset,fit,
            control = mob_control(), ...)
```

Implementation with *partykit*

Achim Zeileis, Torsten Hothorn (2015) *Parties , Models , Mobsters : A New Implementation of Model-Based Recursive Partitioning in R*.
The MOB method is implemented by the simple interface function:

```
mob(formula, data, subset, na.action, weights, offset,fit,
        control = mob_control(), ...)
```

- The formula contains three types of variables: the outcome Y, the vector of covariables $\mathbf{X}$ and the input vector $\mathbf{Z}$ containing the partitioning variables. They are incorporated to the model as follows:

$$Y \sim X_1 + X_2 + \cdots X_p | Z_1 + Z_2 + \cdots Z_q$$

Implementation with *partykit*

Achim Zeileis, Torsten Hothorn (2015) *Parties , Models , Mobsters : A New Implementation of Model-Based Recursive Partitioning in R*.
The MOB method is implemented by the simple interface function:

```
mob(formula, data, subset, na.action, weights, offset,fit,
                control = mob_control(), ...)
```

- The formula contains three types of variables: the outcome Y, the vector of covariables $\mathbf{X}$ and the input vector $\mathbf{Z}$ containing the partitioning variables. They are incorporated to the model as follows:

$$Y \sim X_1 + X_2 + \cdots X_p | Z_1 + Z_2 + \cdots Z_q$$

- The arguments `subset`, `na.action`, `weights` and `offset` are standards for data processing; they are passed to the `fit` argument.

The MOB implementation in *partykit* (cont)

```
mob(formula, data, subset, na.action, weights, offset,fit,
        control = mob_control(), ...)
```

- The `fit` argument supplies the parametric model to be fitted so that parameter stability and data partitioning can be carried out.

The MOB implementation in *partykit* (cont)

```
mob(formula, data, subset, na.action, weights, offset,fit,
        control = mob_control(), ...)
```

- The `fit` argument supplies the parametric model to be fitted so that parameter stability and data partitioning can be carried out.
- The argument `mob_control()` allows to set a list of control options for tree growing and stopping, like the significance level for parameter stability, the minimal node size and the maximal depth of the tree, as well as for post-pruning using information criteria like AIC or BIC.

The MOB implementation in *partykit* (cont)

```
mob(formula, data, subset, na.action, weights, offset,fit,
        control = mob_control(), ...)
```

- The `fit` argument supplies the parametric model to be fitted so that parameter stability and data partitioning can be carried out.
- The argument `mob_control()` allows to set a list of control options for tree growing and stopping, like the significance level for parameter stability, the minimal node size and the maximal depth of the tree, as well as for post-pruning using information criteria like AIC or BIC.
- The functions `lmtree` and `glmtree` provide shortcut user interfaces for recursive partitioning that simplify the computations and provide fancy functionalities for further visualization and prediction methods.

Business case for an auto lending company

- Historical data collected from 208085 loan applications. Filters: 1) 47210 refinancing applications were removed. 2) Only applications that received approval at least 45 days prior to the investigation end date are considered.

Business case for an auto lending company

- Historical data collected from 208085 loan applications. Filters: 1) 47210 refinancing applications were removed. 2) Only applications that received approval at least 45 days prior to the investigation end date are considered.
- We obtain a data set containing 152965 cases with 26323 funded applications; so the response rate to the bid is around 17,2 %.

Business case for an auto lending company

- Historical data collected from 208085 loan applications. Filters: 1) 47210 refinancing applications were removed. 2) Only applications that received approval at least 45 days prior to the investigation end date are considered.
- We obtain a data set containing 152965 cases with 26323 funded applications; so the response rate to the bid is around 17,2 %.
- Several business inputs were measured by the variables in the Table. The price variable is defined by the interest rate.

Business case for an auto lending company

- Historical data collected from 208085 loan applications. Filters: 1) 47210 refinancing applications were removed. 2) Only applications that received approval at least 45 days prior to the investigation end date are considered.
- We obtain a data set containing 152965 cases with 26323 funded applications; so the response rate to the bid is around 17,2 %.
- Several business inputs were measured by the variables in the Table. The price variable is defined by the interest rate.

Business case for an auto lending company

- Historical data collected from 208085 loan applications. Filters: 1) 47210 refinancing applications were removed. 2) Only applications that received approval at least 45 days prior to the investigation end date are considered.
- We obtain a data set containing 152965 cases with 26323 funded applications; so the response rate to the bid is around 17,2 %.
- Several business inputs were measured by the variables in the Table. The price variable is defined by the interest rate.

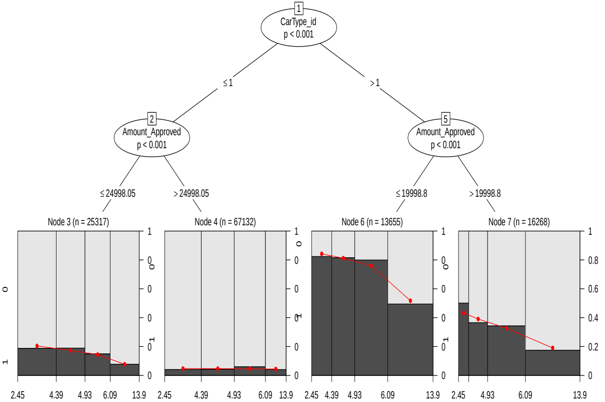
Set of variables for MOB modeling	
Tier	Classification of applicants based on FICO scores
Primary FICO	FICO score quantifying the applicant's risk in the range $[594, 854]$
Term	Loan term in months
Amount Approved	Amount of the loan in the range $[5, 100000]$
Competition rate	Interest rate of competitor
Car Type id	Type of car: new (1) or used (2)
term class	Four level segmentation of the Term variable
partnerbin	Segmentation based on partners (1: Direct. 2: Partner A. 3: Other partners)
rate	Interest rate of the application
onemonth	Prime rate
apply	Binary outcome variable with the purchase decision

Application using *partykit* functionalities

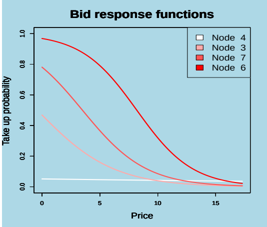
A MOB tree is fit on a training data set. The algorithm is parameterized with the default significance level, $\alpha = 0{,}05$, minimum node size 5 % of the training data and the *Logit* based tree approach to set the `fit` argument.

Application using *partykit* functionalities

A MOB tree is fit on a training data set. The algorithm is parameterized with the default significance level, $\alpha = 0{,}05$, minimum node size 5 % of the training data and the *Logit* based tree approach to set the `fit` argument.

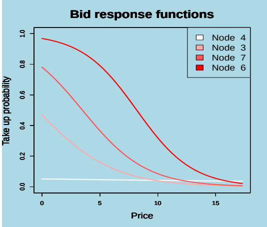


Results and business insights



Node ID	Logit equation	Node description
3	$-0{,}13 - 0{,}31X$	New auto and amount $<$ \$25000
4	$-2{,}92 - 0{,}02X$	New auto and amount $\geq$ \$25000
6	$3{,}42 - 0{,}42X$	Used auto and amount $<$ \$20000
7	$1{,}27 - 0{,}36X$	Used auto and amount $\geq$ \$20000

Results and business insights



Node ID	Logit equation	Node description
3	$-0{,}13 - 0{,}31X$	New auto and amount $<$ \$25000
4	$-2{,}92 - 0{,}02X$	New auto and amount $\geq$ \$25000
6	$3{,}42 - 0{,}42X$	Used auto and amount $<$ \$20000
7	$1{,}27 - 0{,}36X$	Used auto and amount $\geq$ \$20000

Details about optimal price allocation can be seen in the recent paper: Jorge M Arevalillo (2019) *Model Based Recursive Partitioning for Customized Price Optimization Analytics*. Pattern Recognition and Image Analysis. IbPRIA 2019. Lecture Notes in Computer Science, vol 11867. 113-124. Springer.

This work has been funded by *UNED - Santander Bank Research and Transfer 2018 Award* with reference 2018V/PREMIO/07 under grant 18CF05.

Dr. Jorge Martín Arevalillo
Profesor Contratado Doctor UNED
Data Scientist

Tel. +34 91 398 72 64
Tel móvil. 616 87 13 53
U.N.E.D
jmartin@ccia.uned.es

Gracias por su atención