Classification and regression trees

Natalia Levshina © 2019

University of Tallinn, May 2019

Outline

1. The main principles of CART
2. Conditional inference trees
3. A case study: a multivariate analysis of nerd and geek

CART

- Allow to model the relationships between a response variable and different predictors
  - E.g. Income depending on education, gender, work experience, etc.
  - Reaction times based on word length and frequency
  - The use of near-synonyms (e.g. nerd and geek) depending on contextual factors (evaluation, number, subcorpus, etc.)
- One can have categorical response (classification) or numeric (regression) and any kind of predictors (numeric and categorical).

Recursive binary partitioning

Step 1. The algorithm starts with the entire data set and picks up the predictor the most strongly associated with the response.

Step 2. The algorithm splits the data into two data sets (binary) according to the values of that predictor (partitioning).

Steps 1 and 2 are repeated again and again until a certain criterion is met (recursive).

Outline

1. The main principles of CART
2. Conditional inference trees
3. A case study: a multivariate analysis of nerd and geek

Conditional inference trees

- A specific algorithm for CART
- Works well with correlated predictors
- Can deal with missing values in predictors
- Uses and returns *p*-values:
  - the splits are made in the predictor whose association/correlation with the response yields the smallest p-value.
  - stops when no more associations/correlations between the response and predictors smaller than the significance level (0.05) can be found.

Outline

1. The main principles of CART
2. Conditional inference trees
3. A case study: a multivariate analysis of nerd and geek

Nerd and geek revisited

- We’ve tested pairwise associations between the use of nerd or geek in COCA and different contextual variables.
- But how to take into account many variables at once?
- We need multivariate statistics, such as CART.
- The noun (nerd or geek) will be the response variable, and the other variables will be tested as predictors that enable us to predict the speaker’s choice between nerd and geek.

Why multivariate?

- Indeed, why not test the associations pairwise?
- Imagine that the XXI century subcorpus may contain more positive contexts in general. That could explain why there are more uses of geek (which is more frequently used in positive contexts) in the XXI century data.
- We need to measure the effect of century while controlling for evaluation, and the effect of evaluation while controlling for century.

Another example: help + (to) Inf

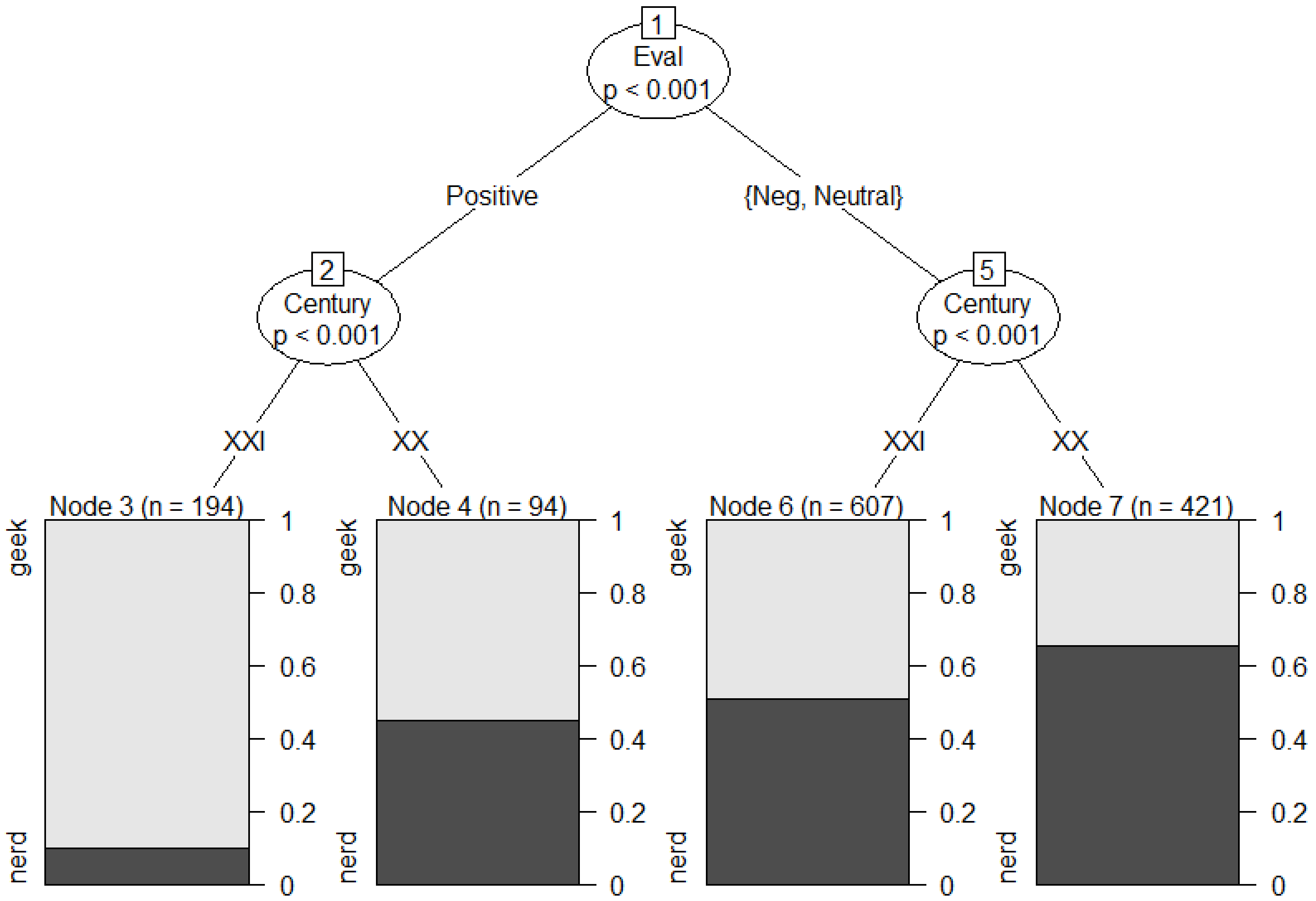
- E.g. Mary helped John (to) cook the dinner.
- The frequency of help + bare infinitive has been increasing over the last two centuries, and the proportion of help + to-infinitive has been decreasing.
- help + bare infinitive is also considered to be more informal than help + (to) infinitive.
- English language use becomes more informal over time (Americanization?).
- Therefore, one may think that the increase of help + bare infinitive is due to the increase of informality, and there is no language change (only the more general change in style).
- In order to capture the separate effects of informality and time, we would need to test both simultaneously in a multivariate model.

Dataset nerd revisited

```
> colnames(nerd)
1] "Noun"        "Num"          "Century"
"Register" "Eval"
```

Creating a tree

```
> library(party)
> nerd.ctree <- ctree(Noun ~ Num + Century
+ Register + Eval, data = nerd, controls =
ctree_control(testtype = "MonteCarlo"))
#based on 9999 permutations
> plot(nerd.ctree)
```


1
Eval
p < 0.001
Positive
{Neg, Neutral}
2
Century
p < 0.001
5
Century
p < 0.001
XXI
XX
XXI
XX
Node 3 (n = 194)
Node 4 (n = 94)
Node 6 (n = 607)
Node 7 (n = 421)
geek
nerd
1
0.8
0.6
0.4
0.2
0

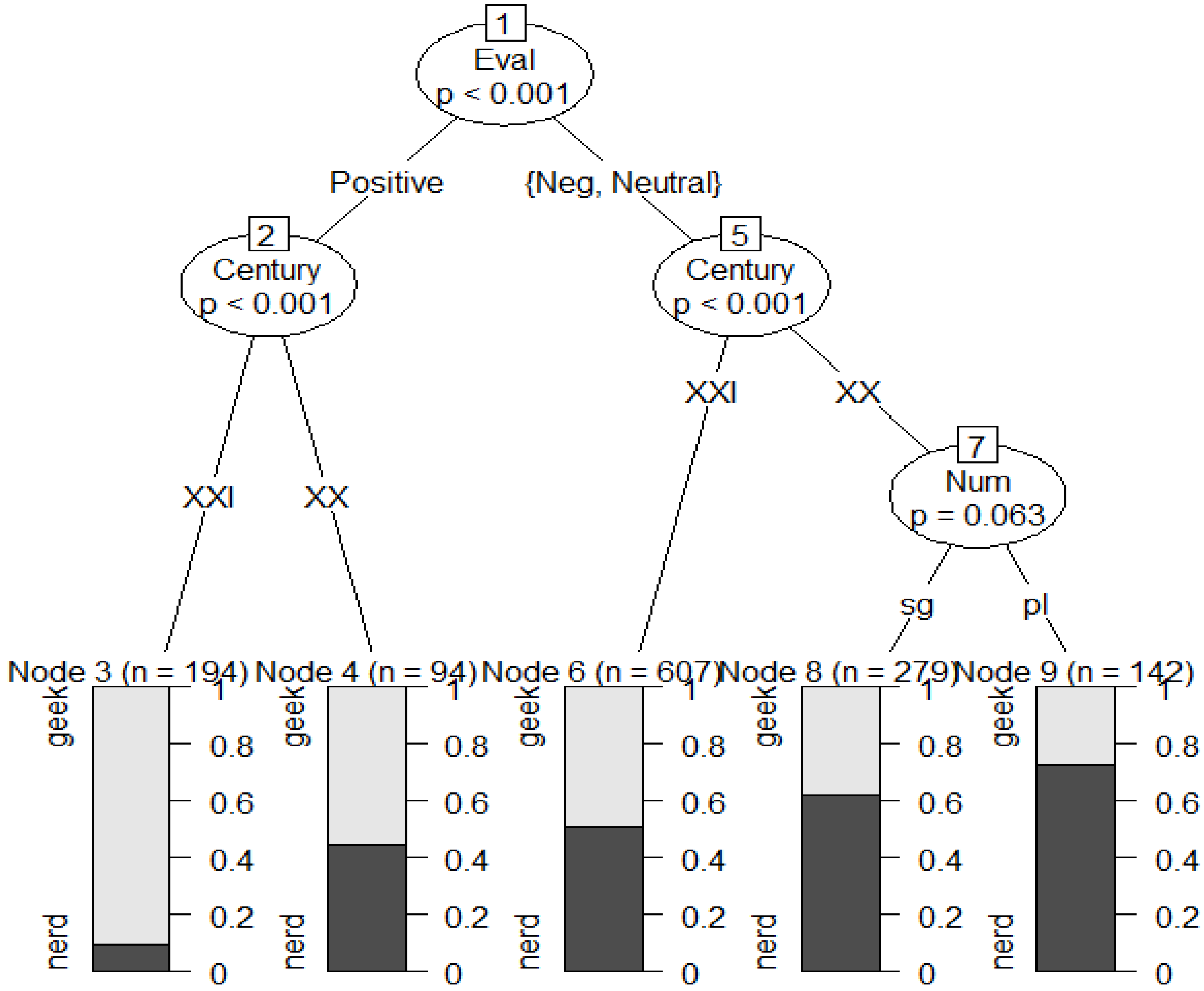
Default settings

- mincriterion = 0.95 (the p-values should be 0.05 and less)
- minsplit = 20 (the minimum number of observations to be considered for splitting)
- minbucket = 7 (the minimum number of observations in a terminal node)
- nresample = 9999 (number of samples for permutation)

How to change the default settings

```
> nerd.ctree1 <- ctree(Noun ~ Num +
Century + Register + Eval, data = nerd,
controls = ctree_control(testtype =
"MonteCarlo", mincriterion = 0.9)) #change
the significance level to p = 0.1 for
exploratory purposes

> plot(nerd.ctree1)
```


1
Eval
p < 0.001
Positive
{Neg, Neutral}
2
Century
p < 0.001
5
Century
p < 0.001
XXI
XX
XXI
XX
7
Num
p = 0.063
sg
pl
Node 3 (n = 194)
Node 4 (n = 94)
Node 6 (n = 607)
Node 8 (n = 279)
Node 9 (n = 142)
geek
nerd
1
0.8
0.6
0.4
0.2
0