Initiation à la construction d'arbres dans R avec 'rpart' et 'party'

Gilbert Ritschard

Département d'économétrie et Laboratoire de Démographie, Université de Genève
http://mephisto.unige.ch/biomining

Séminaire Statistique Sciences sociales et démographie
19 mars 2009

Plan

1. Induction Trees
2. Initiation to the practice of decision trees with `party`
3. Références

Outline

1. Induction Trees
2. Initiation to the practice of decision trees with party
3. Références

Section outline

1. Induction Trees
   - Introduction
   - Supervised learning
   - Tree Growing Principle
   - The criteria

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES ÉCONOMIQUES ET SOCIALES
Département d'économétrie

Induction Trees : Introduction (1)

- Trees induced from data.
- Recursive partitioning, segmentation, ...
- Most often used for classification : classification tree, when target is a categorical variable.
- Regression tree, when response variable is measurable at interval or ratio scale.
- Objective : Partition data according to explanatory factors (attributes, predictors, covariates) so that the distribution of the response variable (dependent variable to be predicted) :
  - is the purest possible in each class
    (maximize class homogeneity = minimize within class differences)
  - differs as much as possible from one class to the other
    (maximize between class differences) ;

Induction Trees : Introduction (2)

Singles out interactions of covariates in their effect on the response variable

Results :

- visual (a tree) ;
- no coefficients measuring the effect of covariates ;
- classification rules.

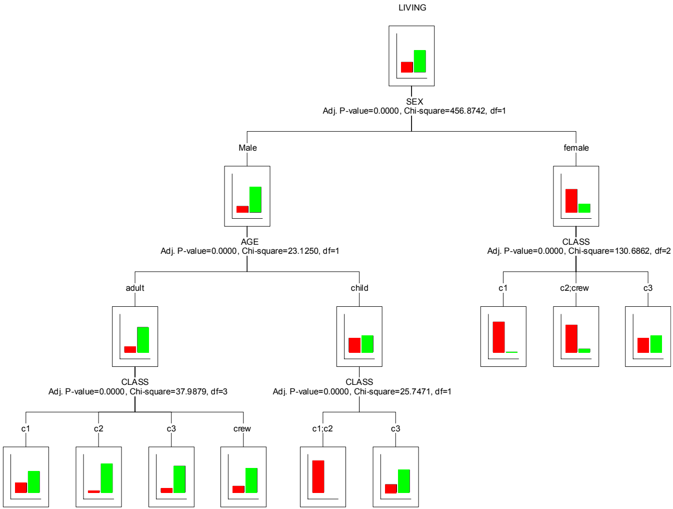
Induction Trees : Introduction (2)

Singles out interactions of covariates in their effect on the response variable

Results :

- visual (a tree) ;
- no coefficients measuring the effect of covariates ;
- classification rules.

Illustration : Titanic


LIVING
SEX
Adj. P-value=0.0000, Chi-square=456.8742, df=1
Male
female
AGE
Adj. P-value=0.0000, Chi-square=23.1250, df=1
CLASS
Adj. P-value=0.0000, Chi-square=130.6862, df=2
adult
child
c1
c2;crew
c3
CLASS
Adj. P-value=0.0000, Chi-square=37.9879, df=3
CLASS
Adj. P-value=0.0000, Chi-square=25.7471, df=1
c1
c2
c3
crew
c1;c2
c3

Section outline

Supervised learning

- Based on a learning sample $\{(\mathbf{x}_\alpha, y_\alpha)\}_{\alpha=1,\ldots,n}$,
  - where y_α is the value (class) of the response (dependent, ...) variable for case α,
  - and $\mathbf{x}_\alpha = (x_{\alpha 1}, \ldots, x_{\alpha p})$ is the profile of α in terms of the covariates.
- Build a predictive function (or classification function)

$$y = f(\mathbf{x})$$

  with which we can predict the value or class y when only the profile $\mathbf{x}$ is known.
- Example : predict whether a passenger of the Titanic survives from the sole knowledge of sex, age (child/adult) and navigation class.

Section outline

1. Induction Trees
   - Introduction
   - Supervised learning
   - Tree Growing Principle
   - The criteria

Target Table

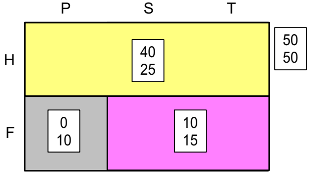
- Assuming all variables are categorical, we can represent the data with a contingency table that cross tabulates the response variable with a composite variable defined by the cross tabulation of all covariates.
- Example of a target contingency table **T**.
- Response variable is marital status, predictors are sex and sector of activity

	man			woman			
married	primary	secondary	tertiary	primary	secondary	tertiary	total
no	11	14	15	0	5	5	50
yes	8	8	9	10	7	8	50
total	19	22	24	10	12	13	100

Constructing the rules

An induction tree (like a logistic regression) determines the rule $f(x)$ in two steps

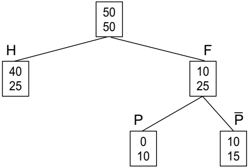
1. Determine a partition of the possible profiles $\mathbf{x}$ such that the distribution p_y of the response Y is as different as possible from one class to the other.



2. The rule consists then in assigning to each case the most frequently observed value y in the class defined by the values of x.

$$\hat{y} = f(\mathbf{x}) = \arg\max_i \hat{p}_i(\mathbf{x})$$

Induced Tree



- Partitions are determined by successive splits of nodes.
- Starting with the root node (formed by the set of all cases), we seek the covariate that permits the better split according to a given criterion (greatest entropy reduction, strongest association with the response.)
- Operation is repeated at each new obtained node until fulfilment of some stopping criterion (a minimal node size or a minimal gain in the criterion).

Section outline

1. Induction Trees
   - Introduction
   - Supervised learning
   - Tree Growing Principle
   - The criteria

Splitting criteria

Criteria from

- Information Theory : Entropies (uncertainty) prediction made from the resulting distribution

  Shannon's entropy : $h_S(p) = -\sum_{i=1}^{c} p_i \log_2 p_i$

  Quadratic entropy (Gini) : $h_Q(p) = \sum_{i=1}^{c} p_i(1-p_i) = 1 - \sum_{i=1}^{c} p_i^2$

  $\Rightarrow$ maximizing entropy reduction

  (maximizing within leaves homogeneity)

- Statistical associations : Pearson's Chi-square, measures of association.

  $\Rightarrow$ maximizing association,

  minimizing the p-value of the no-association test.

  (maximizing diversity between leaves)

Gain of information (1)

- Splitting the root node by sex, we get two nodes.
- The distribution in each node is that of the corresponding column of Table below

Marital status by sex

age	man	woman	total
married	40	10	50
not married	25	25	50
total	65	35	100

- What information brings "sex"?

Gain of information (2)

- Gain = reduction of uncertainty
- Uncertainty : Shannon's entropy

$$
\begin{aligned}
H(\text{marital status}) &= -\sum_{i=1}^{c} p_i \log_2 p_i \\
&= -\left(\frac{50}{100} \log_2\left(\frac{50}{100}\right) + \frac{50}{100} \log_2\left(\frac{50}{100}\right) \right) = \boxed{1} \\
H(\text{marital status}|\text{man}) &= -\left(\frac{40}{65} \log_2\left(\frac{40}{65}\right) + \frac{25}{65} \log_2\left(\frac{25}{65}\right) \right) = \boxed{.961} \\
H(\text{marital status}|\text{woman}) &= -\left(\frac{10}{35} \log_2\left(\frac{10}{35}\right) + \frac{25}{35} \log_2\left(\frac{25}{35}\right) \right) = \boxed{.863} \\
H(\text{marital status}|\text{sex}) &= (65/100)0.961 + (35/100)0.863 = \boxed{0.927} \\
\text{Gain}(\text{sex}) &= H(\text{marital status}) - H(\text{marital status}|\text{sex}) \\
&= 1 - 0.927 = \boxed{0.073}
\end{aligned}
$$

Most popular tree growing methods

- CHAID, CHi-square based Automatic Interaction Detection (?Biggs et al., 1991) : n-ary trees, criterion based on Bonferroni adjusted *p*-values of independence tests.
  - CHAID is an extension of an earlier regression tree method called AID (Morgan and Sonquist, 1963)
- CART, Classification and Regression Tree (Breiman et al., 1984) : binary trees, criterion is maximizing decrease of Gini purity measure, pruning, surrogate splits in case of missing values.
- C4.5 (Quinlan, 1993) : binary trees, criterion is Information Gain, the reduction in Shannon's entropy standardized by the entropy of the predictor.
- C4.5 was designed in a less statistical and more IA perspective.

Most popular tree growing methods

- CHAID, CHi-square based Automatic Interaction Detection (?Biggs et al., 1991) : n-ary trees, criterion based on Bonferroni adjusted *p*-values of independence tests.
  - CHAID is an extension of an earlier regression tree method called AID (Morgan and Sonquist, 1963)
- CART, Classification and Regression Tree (Breiman et al., 1984) : binary trees, criterion is maximizing decrease of Gini purity measure, pruning, surrogate splits in case of missing values.
- C4.5 (Quinlan, 1993) : binary trees, criterion is Information Gain, the reduction in Shannon's entropy standardized by the entropy of the predictor.
- C4.5 was designed in a less statistical and more IA perspective.

Most popular tree growing methods

- CHAID, CHi-square based Automatic Interaction Detection (?Biggs et al., 1991) : n-ary trees, criterion based on Bonferroni adjusted *p*-values of independence tests.
  - CHAID is an extension of an earlier regression tree method called AID (Morgan and Sonquist, 1963)
- CART, Classification and Regression Tree (Breiman et al., 1984) : binary trees, criterion is maximizing decrease of Gini purity measure, pruning, surrogate splits in case of missing values.
- C4.5 (Quinlan, 1993) : binary trees, criterion is Information Gain, the reduction in Shannon's entropy standardized by the entropy of the predictor.
- C4.5 was designed in a less statistical and more IA perspective.

Most popular tree growing methods (2)

- CART and C4.5 were designed for prediction purposes (prediction error is a primary concern).
- CHAID and AID primary aim is interaction detection. Their aim is primary description, rather than prediction.

Outline

1 Induction Trees

2 Initiation to the practice of decision trees with party

3 Références

Section outline

2 Initiation to the practice of decision trees with party

- rpart and party
- Now building a mobility tree

rpart and party

- At least two R-packages for growing (binary) trees :
  - rpart (?) : recursive partitioning
    CART, Relative risk trees,
  - party (Hothorn et al., 2006) : conditional partitioning
    Based on a statistical conditional inference method (permutation tests)
- We propose here a short introduction to these packages
  - rpart Essentially Cart + extension for relative risk trees
  - party much more powerful and flexible.
  - better visual rendering (Plots distributions inside the nodes)

rpart and party

- At least two R-packages for growing (binary) trees :
  - rpart (**?**) : recursive partitioning
    CART, Relative risk trees,
  - party (Hothorn et al., 2006) : conditional partitioning
    Based on a statistical conditional inference method
    (permutation tests)
- We propose here a short introduction to these packages
  - rpart Essentially Cart + extension for relative risk trees
  - party much more powerful and flexible.
  - better visual rendering (Plots distributions inside the nodes)

rpart and party

- At least two R-packages for growing (binary) trees :
  - rpart (?) : recursive partitioning
    CART, Relative risk trees,
  - party (Hothorn et al., 2006) : conditional partitioning
    Based on a statistical conditional inference method
    (permutation tests)
- We propose here a short introduction to these packages
  - rpart Essentially Cart + extension for relative risk trees
  - party much more powerful and flexible.
  - better visual rendering (Plots distributions inside the nodes)

rpart and party

- At least two R-packages for growing (binary) trees :
    - rpart (**?**) : recursive partitioning
      CART, Relative risk trees,
    - party (Hothorn et al., 2006) : conditional partitioning
      Based on a statistical conditional inference method (permutation tests)
- We propose here a short introduction to these packages
    - rpart Essentially Cart + extension for relative risk trees
    - party much more powerful and flexible.
    - better visual rendering (Plots distributions inside the nodes)

party principle

- party selects each split in two steps (to avoid bias in favor of predictors with many different values) :
    - First, selects the predictor with strongest association with target,
    - Then, selects the best binary split for selected predictor.

Linear statistic and permutation test

- Both steps are based on the conditional distribution of linear statistics in a permutation test framework.
  - Linear statistic is :

$$\mathbf{T}_j = \text{vec}\left(\sum_{i=1}^{n} w_i g_j(X_{ji}) h(\mathbf{Y}_i, (\mathbf{Y}_1, \ldots, \mathbf{Y}_n))^T \right) \in \mathbb{R}^{p_j q}$$

    where $g_j(X_{ji})$ is a transformation of X_{ji}, and $h()$ an influence function.
  - $\mathbf{T}_j$ is computed for each permutation of the $\mathbf{Y}$ values among cases, and results characterize its conditional independence distribution.
  - the variable and split selection is then based on the p-value of the observed $\mathbf{t}$ under this conditional independence distribution.

Creating or reading a data set in R

- You can either create a `data.frame` within R

```
# creating data set in R
marr <- rbind(
  data.frame(married="yes",sex="man",  activity="primary",  weight=11),
  data.frame(married="yes",sex="man",  activity="secondary",weight=14),
  data.frame(married="yes",sex="man",  activity="tertiary", weight=15),
  data.frame(married="yes",sex="woman",activity="primary",  weight=0),
  data.frame(married="yes",sex="woman",activity="secondary",weight=5),
  data.frame(married="yes",sex="woman",activity="tertiary", weight=5),
  data.frame(married="no", sex="man",  activity="primary",  weight=8),
  data.frame(married="no", sex="man",  activity="secondary",weight=8),
  data.frame(married="no", sex="man",  activity="tertiary", weight=9),
  data.frame(married="no", sex="woman",activity="primary",  weight=10),
  data.frame(married="no", sex="woman",activity="secondary",weight=7),
  data.frame(married="no", sex="woman",activity="tertiary", weight=8) )
marr # displays content of marr
```

- It is however more convenient to read a file, for instance a csv file

```
marr <- read.csv(file="C:/data/lund/exple_married_sex_sector.csv",header=TRUE)
```

Creating or reading a data set in R

- You can either create a `data.frame` within R

```
# creating data set in R
marr <- rbind(
  data.frame(married="yes",sex="man",  activity="primary",  weight=11),
  data.frame(married="yes",sex="man",  activity="secondary",weight=14),
  data.frame(married="yes",sex="man",  activity="tertiary", weight=15),
  data.frame(married="yes",sex="woman",activity="primary",  weight=0),
  data.frame(married="yes",sex="woman",activity="secondary",weight=5),
  data.frame(married="yes",sex="woman",activity="tertiary", weight=5),
  data.frame(married="no", sex="man",  activity="primary",  weight=8),
  data.frame(married="no", sex="man",  activity="secondary",weight=8),
  data.frame(married="no", sex="man",  activity="tertiary", weight=9),
  data.frame(married="no", sex="woman",activity="primary",  weight=10),
  data.frame(married="no", sex="woman",activity="secondary",weight=7),
  data.frame(married="no", sex="woman",activity="tertiary", weight=8) )
marr # displays content of marr
```

- It is however more convenient to read a file, for instance a csv file

```
marr <- read.csv(file="C:/data/lund/exple_married_sex_sector.csv",header=TRUE)
```

A R script for generating a tree

- You grow the tree with the ctree command

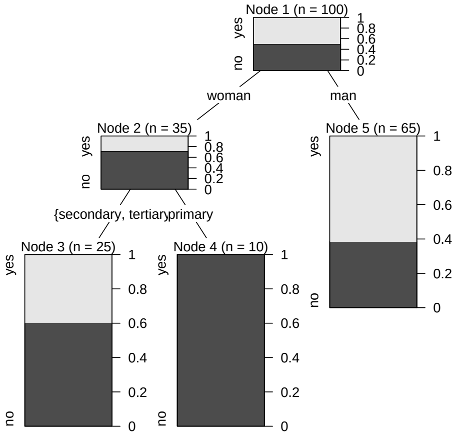
```
#loading party
library(party)

marrtree <- ctree(married ~ ., data=marr[,1:3],
    controls=ctree_control(mincriterion=.50,minsplit=0),
    weights=marr$weight)
marrtree # dispays info on tree

plot(marrtree)  # plots the tree

# Plotting same tree using some controls.
plot(marrtree,drop_terminal=F,inner_panel=node_barplot)
```

Output in R console

```
> marrtree

         Conditional inference tree with 4 terminal nodes

Response:  married
Inputs:  sex, activity
Number of observations:  12

1) sex == {woman}; criterion = 0.996, statistic = 9.791
  2) activity == {secondary, tertiary}; criterion = 0.874, statistic = 5.471
    3)*  weights = 25
  2) activity == {primary}
    4)*  weights = 10
1) sex == {man}
  5)*  weights = 65
```

Here is the first plotted tree

Response variable is : "married"



Section outline

2 Initiation to the practice of decision trees with party

- rpart and party
- Now building a mobility tree

Mobility tree on the 3 generations mobility data

```
## Mobility tree example with data from marriage acts of 19th Century Geneva

library(foreign) # library for importing data from various sources
sm_data <- read.spss(file="C:/data/lund/mobility/par_enf_tree_267.sav",to.data.frame=T)
sm_data$NC1_ST3 <- factor(sm_data$NC1_ST3) # to remove deceased category

# ordering and renaming state variables
seqs <- data.frame(GdFather=sm_data$NG1ST_P3, Father_his_M = sm_data$NP1_ST3,
       Father_son_M = sm_data$NC1ST_P3, Son_M=sm_data$NC1_ST3)

# Growing mob tree with ctree (party package)

library(party)

cl_tree <- ctree(seqs$Son_M ~ seqs$Father_son_M + seqs$Father_his_M + seqs$GdFather +
       sm_data$C1LIEU11)
plot(cl_tree)

# you may control the tree with ctree_control()

control <- ctree_control(testtype="Univariate",mincriterion=.9,minsplit=20,minbucket=10)
cl_tree <- ctree(seqs$Son_M ~ seqs$Father_son_M + seqs$Father_his_M + seqs$GdFather +
   sm_data$C1LIEU11,controls=control)
plot(cl_tree,drop_terminal=F)
```

State variables

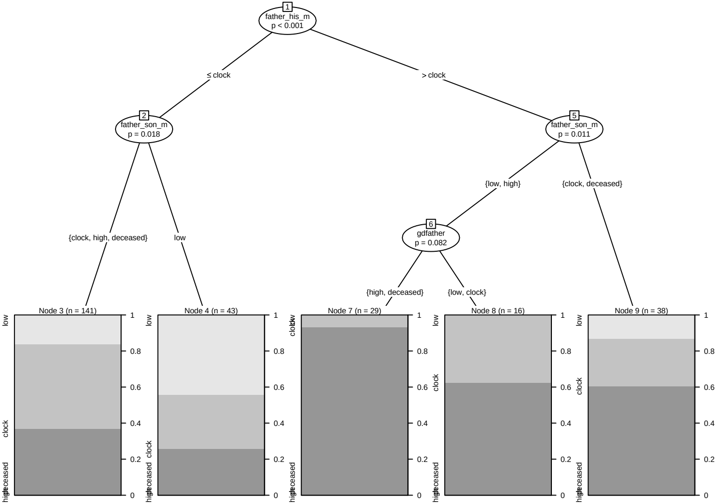
- Variables are

variable	label
GdFather	'Status Grd-father, 3 categories'
Father_his_M	'Status Father (his marr.), 3 categories'
Father_son_M	'Status Father (son”s marr.), 3 categories'
Son_M	'Status Son (his marr.), 3 categories'

Text output

```
         Conditional inference tree with 8 terminal nodes

Response:  seqs$Son_M
Inputs:  seqs$Father_son_M, seqs$Father_his_M, seqs$GdFather, sm_data$C1LIEU11
Number of observations:  267

1) seqs$Father_his_M == {high}; criterion = 1, statistic = 48.744
  2) seqs$Father_son_M == {clock, deceased}; criterion = 0.948, statistic = 12.494
    3)*  weights = 38
  2) seqs$Father_son_M == {low, high}
    4) seqs$GdFather == {low, clock}; criterion = 0.918, statistic = 6.709
      5)*  weights = 16
    4) seqs$GdFather == {high, deceased}
      6)*  weights = 29
1) seqs$Father_his_M == {low, clock}
  7) seqs$Father_son_M == {clock, high, deceased}; criterion = 0.998, statistic = 20.864
    8) seqs$Father_his_M == {low}; criterion = 0.897, statistic = 13.387
      9) seqs$GdFather == {clock, high}; criterion = 0.992, statistic = 17.472
        10)*  weights = 16
      9) seqs$GdFather == {low, deceased}
        11) seqs$GdFather == {low}; criterion = 0.808, statistic = 8.461
          12)*  weights = 24
        11) seqs$GdFather == {deceased}
          13)*  weights = 25
    8) seqs$Father_his_M == {clock}
      14)*  weights = 76
  7) seqs$Father_son_M == {low}
    15)*  weights = 43
```

And here is the induced tree



Transition rates

- You may get transition rates with TraMineR

```
> library(TraMineR)
> seqtrate(seqs)
 Computing transition rates between states clock deceased high low, please wait
              [-> clock] [-> deceased] [-> high]  [-> low]
[clock ->]     0.5062500     0.1562500 0.2625000 0.0750000
[deceased ->]  0.3333333     0.0000000 0.3607306 0.3059361
[high ->]      0.1641791     0.1492537 0.5621891 0.1243781
[low ->]       0.1357466     0.2352941 0.1764706 0.4524887
>
```

Outline

1. Induction Trees
2. Initiation to the practice of decision trees with party
3. Références

Références I

Abbott, A. and A. Tsay (2000). Sequence analysis and optimal matching methods in sociology, Review and prospect. *Sociological Methods and Research 29*(1), 3–33. (With discussion, pp 34-76).

Biggs, D., B. De Ville, and E. Suen (1991). A method of choosing multiway partitions for classification and decision trees. *Journal of Applied Statistics 18*(1), 49–62.

Billari, F. C. (2001). The analysis of early life courses: Complex description of the transition to adulthood. *Journal of Population Research 18*(2), 119–142.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification And Regression Trees*. New York: Chapman and Hall.

Gabadinho, A., G. Ritschard, M. Studer, and N. S. Müller (2008). Mining sequence data in R with the TraMineR package: A user's guide. Technical report, Department of Econometrics and Laboratory of Demography, University of Geneva, Geneva.

Hothorn, T., K. Hornik, and A. Zeileis (2006). party: A laboratory for recursive part(y)itioning. User's manual.

Références II

McVicar, D. and M. Anyadike-Danes (2002). Predicting successful and unsuccessful transitions from school to work using sequence methods. *Journal of the Royal Statistical Society A 165*(2), 317–334.

Morgan, J. N. and J. A. Sonquist (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association 58*, 415–434.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann.