Decision tree learning algorithms

CHAID – CART – C4.5 and the others…

Ricco RAKOTOMALALA

Main issues of the decision tree learning

Choosing the splitting criterion

- Impurity based criteria
- Information gain
- Statistical measures of association…

Binary or multiway splits

- Multiway split: 1 value of the splitting attribute = 1 leaf
- Binary split: finding the best binary grouping
- Grouping only the leaves which are similar regarding the classes distribution

Finding the right sized tree

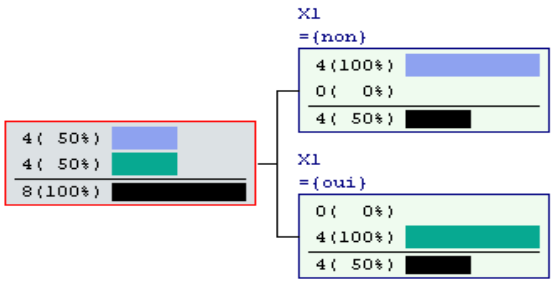
- Pre-pruning
- Post-pruning

Other challenges: decision graph, oblique tree, etc.

Splitting measures

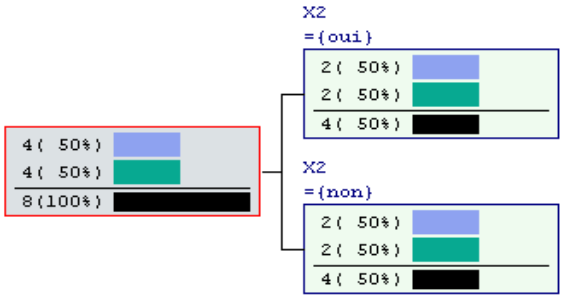
Splitting criterion

Main properties



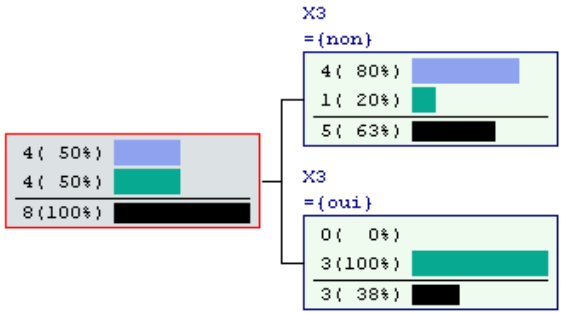
S1: Maximum

The leaves are homogenous.



S2: Minimum

Conditional distributions are the same.



S3 : Intermediate situation

The leaves are more homogeneous regarding Y.
X provides information about Y.

Splitting criterion

Chi-square test statistic for independence and its variants

Contingency table

Cross tabulation between Y and X

Y / X	x_1	x_l	x_L	Σ
y_1				
		$\vdots$		
y_k	$\cdots$	n_{kl}	$\cdots$	$n_{k.}$
		$\vdots$		
y_K				
Σ		$n_{.l}$		n

Measures of association

Comparing the observed and theoretical frequencies
(under the null hypothesis : Y and X are independent)

$$\chi^2 = \sum_{k=1}^{K}\sum_{l=1}^{L} \frac{\left(n_{kl} - \frac{n_{k.} \times n_{.l}}{n}\right)^2}{\frac{n_{k.} \times n_{.l}}{n}}$$

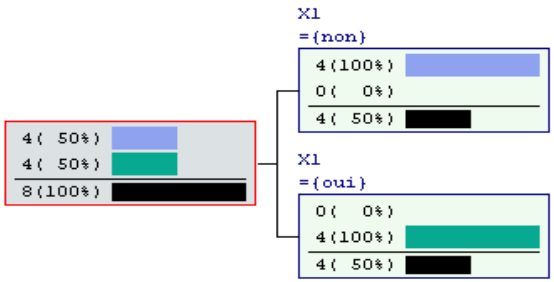
Tschuprow's t

Allows comparing splits with different number of leaves

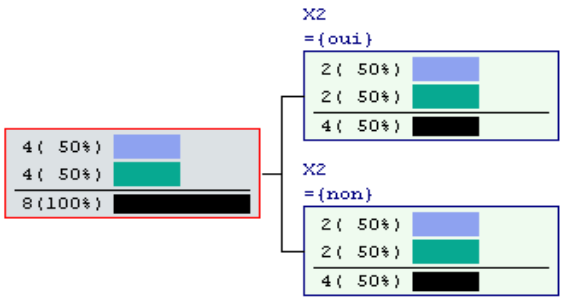
$$t^2 = \frac{\chi^2}{n \times \sqrt{(K-1) \times (L-1)}}$$

Splitting criterion

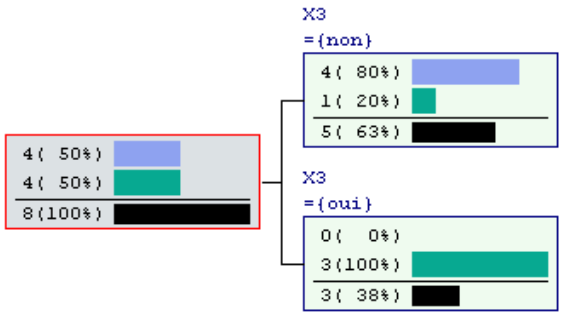
For “Improved” CHAID (SIPINA software)



$S1 : 1.0$



$S2 : 0.0$



$0.0 \leq S3 : 0.7746 \leq 1.0$

Splitting criterion

Information Gain – Gain ratio (C4.5)

Shannon entropy

Measure of uncertainty

$$E(Y) = -\sum_{k=1}^{K} \frac{n_{k.}}{n} \times \log_2\left(\frac{n_{k.}}{n}\right)$$

Condition entropy

Expected entropy of Y knowing the values of X

$$E(Y/X) = -\sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} \times \log_2\left(\frac{n_{kl}}{n_{.l}}\right)$$

Information gain

Reduction of uncertainty

$$G(Y/X) = E(Y) - E(Y/X)$$

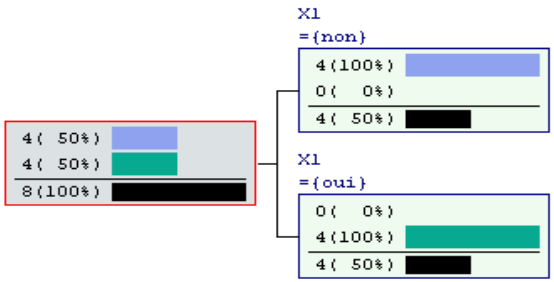
(Information) Gain ratio

Favors the splits with low number of leaves

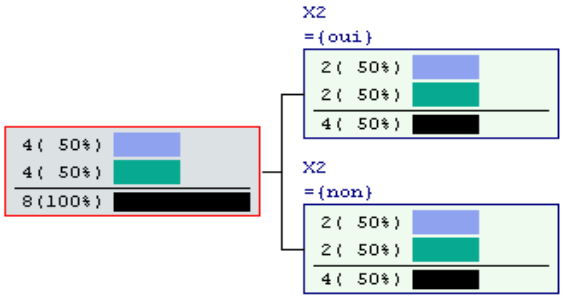
$$GR(Y/X) = \frac{E(Y) - E(Y/X)}{E(X)}$$

Splitting criterion

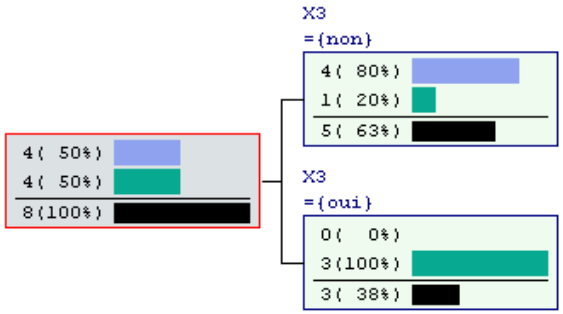
For C4.5 (SIPINA software)



$S1 : 1.0$



$S2 : 0.0$



$0.0 \leq S3 : 0.5750 \leq 1.0$

Splitting criterion

Gini impurity (CART)

Gini index

Measure of impurity

$$I(Y) = -\sum_{k=1}^{K} \frac{n_{k.}}{n} \times \left(1 - \frac{n_{k.}}{n}\right)$$

Conditional impurity

Average impurity of Y conditionally to X

$$I(Y/X) = -\sum_{l=1}^{L} \frac{n_{.l}}{n} \sum_{k=1}^{K} \frac{n_{kl}}{n_{.l}} \times \left(1 - \frac{n_{kl}}{n_{.l}}\right)$$

Gain

$$D(Y/X) = I(Y) - I(Y/X)$$

Gini index = Viewed as an entropy (cf. Daroczy)

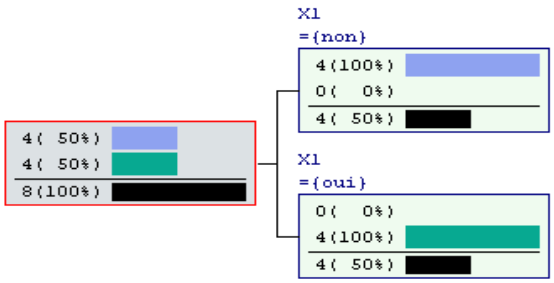
D can be viewed as a kind of information gain

Gini index = Viewed as a variance for categorical variable

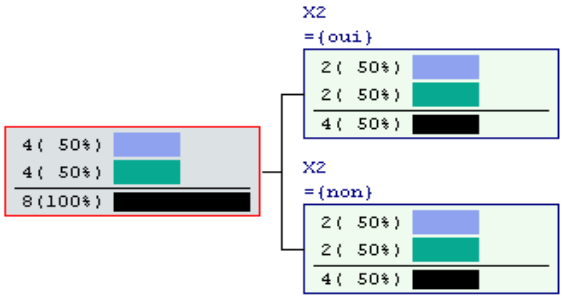
CATANOVA (analysis of variance for categorical data) → D = variance between groups

Splitting criterion

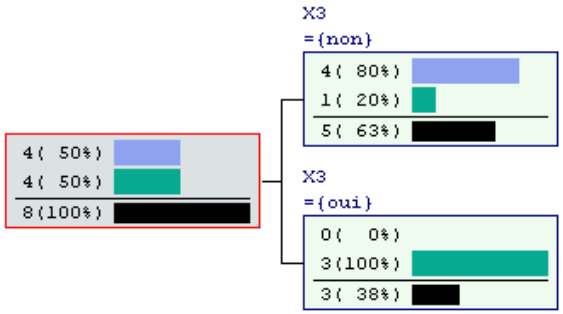
For C&RT (Tanagra software)



$S1 : 0.5$



$S2 : 0.0$



$0.0 \leq S3 : 0.3 \leq 1.0$

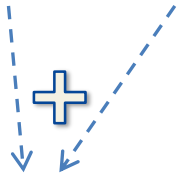
Using unbiased measure

… allows to alleviate the data fragmentation problem

Splitting into 4 subsets using the X1 attribute

Y / X1	A1	B1	C1	D1	Total
positif	2	3	6	3	14
négatif	4	4	8	0	16
Total	6	7	14	3	30

CHI-2	3.9796
T Tschuprow	0.0766



Y / X2	A2	B2	D2	Total
positif	2	9	3	14
négatif	4	12	0	16
Total	6	21	3	30

CHI-2	3.9796
T Tschuprow	**0.0938**

X2 is better than X1

Splitting into 3 subsets using the X2 attribute

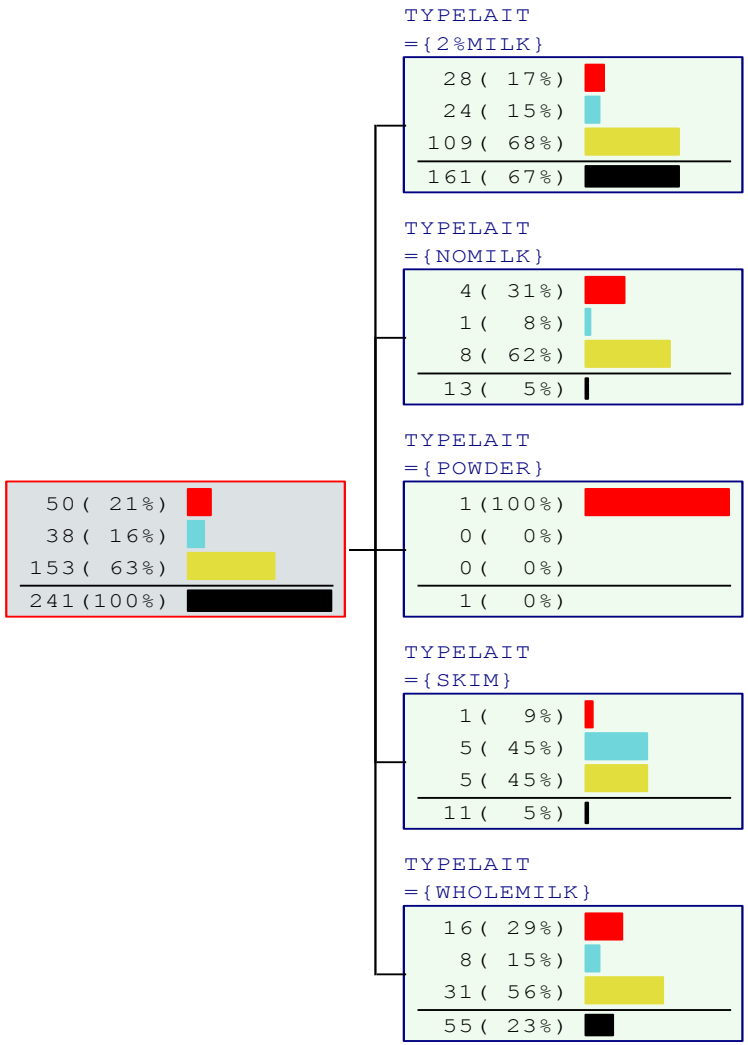
- Tschuprow's t corrects the bias of the chi-square measure
- Gain Ratio corrects the bias of the information gain
- The Gini reduction in impurity is biased in favor of variables with more levels

(but the CART algorithm constructs necessarily a binary decision tree)

Merging process

Multiway splitting (C4.5)

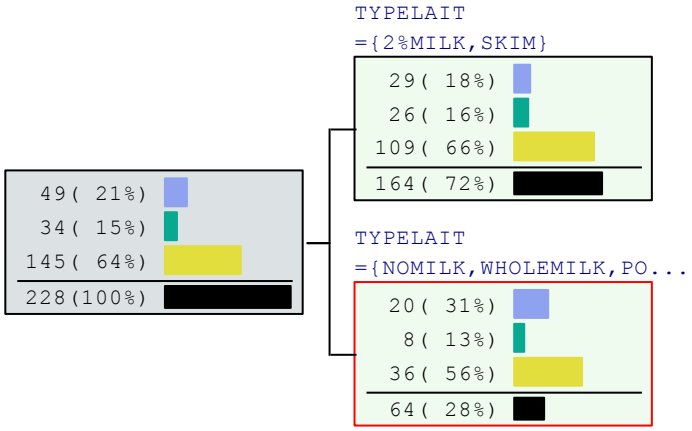
1 level = 1 leaf in the splitting process



- The prediction rules are easy to read
- Data fragmentation problem, especially for small dataset
- "Large" decision tree with a high number of leaves
- "Low depth" decision tree

Binary splitting (CART)

Detecting the best combination in two subsets

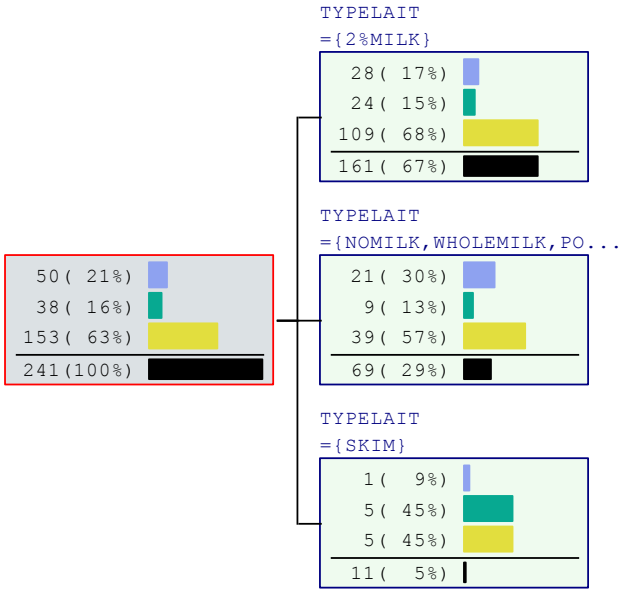


- This grouping allows to overcome the bias of the splitting measure used
- The data fragmentation problem is alleviated
- "High depth" decision tree (CART uses a post pruning process for remedy to this)
- Merging into two groups is not always relevant !

Merging approach (CHAID)

Merging the similar leaves according the classes distribution

- Alleviate the data fragmentation problem
- Choosing the alpha level for merging is not obvious



Principle: Iterative merging if the distributions are not significantly different into leaves (bottom-up strategy)

	NoMilk, Powder	WholeMilk
High	5	16
Low	1	8
Normal	8	31
Total	14	55

$$\chi^2 = 14 \times 55 \times \left[\frac{(5/14 - 16/55)^2}{5+16} + \frac{(1/14 - 8/55)^2}{1+8} + \frac{(8/14 - 31/55)^2}{8+31} \right]$$

$$= 0.6309$$

$$\text{p} - \text{value}_{\chi^2[(3-1)\times(2-1)]} = 0.73$$

Merging if (p-value > alpha level for merging)

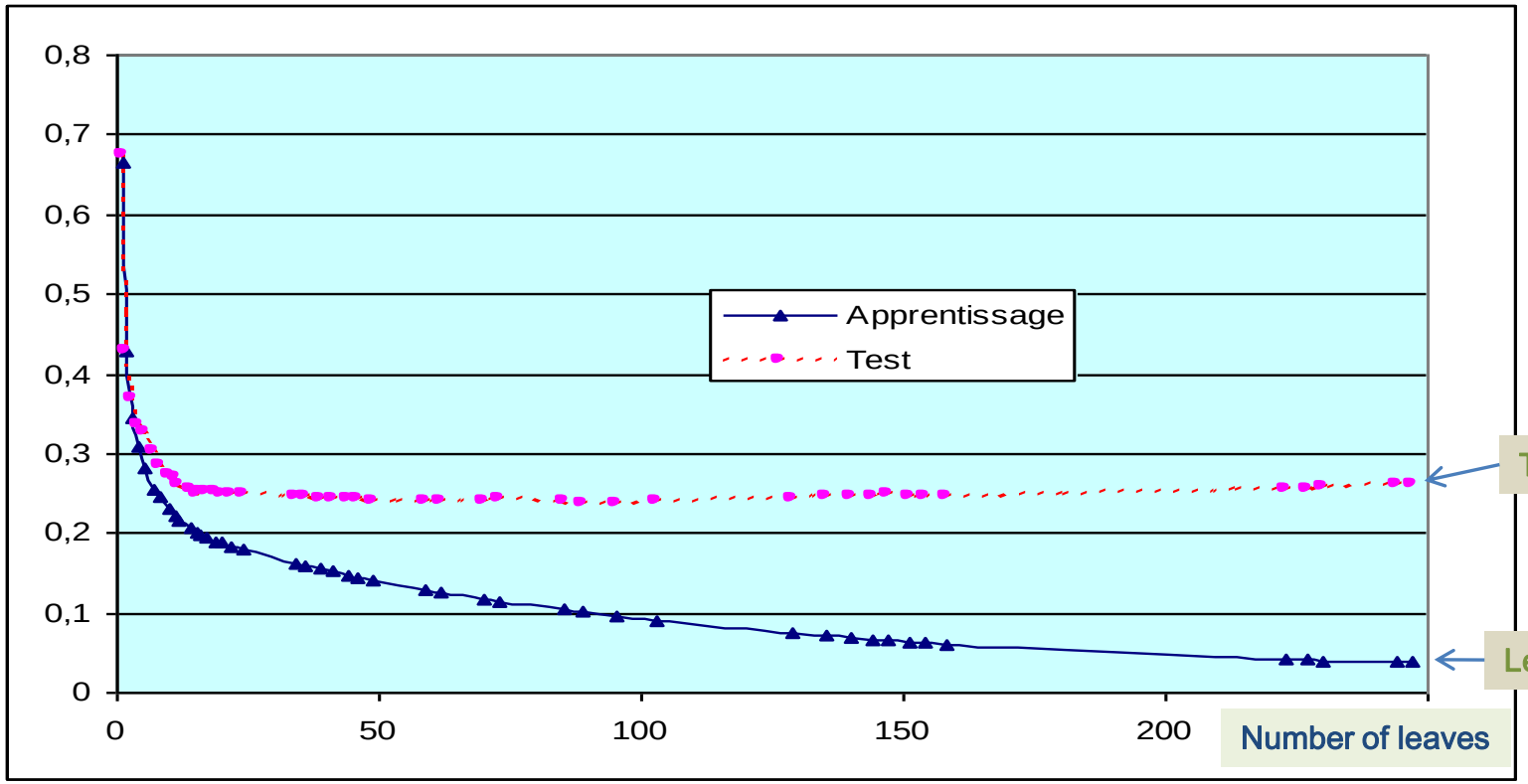
Selecting the right sized tree

Bias variance tradeoff

... according the tree complexity

Bias ~ how powerful is the model

Variance ~ how sensitive the model is to the training set



Underfitting
The tree is too small

"Optimal" tree size

Overfitting
The tree is too large

Pre-pruning

Stopping the growing process

Confidence and support criteria

- Group purity: confidence threshold
- Support criterion: min. size node to split, min. instances in leaves

+ Easy to understand and easy to use
- The right thresholds for a given problem is not obvious

Statistical approach (CHAID)

- Chi-squared test for independence

- The right alpha-level for the splitting test is very hard to determine

But in practice, this approach is often used because :

- the obtained tree reaches a good performance, the area for the "optimal" error rate is large
- it is fast (the growing is stopped earlier, no additional calculations for the post pruning)
- it is preferred at least in the exploratory phase

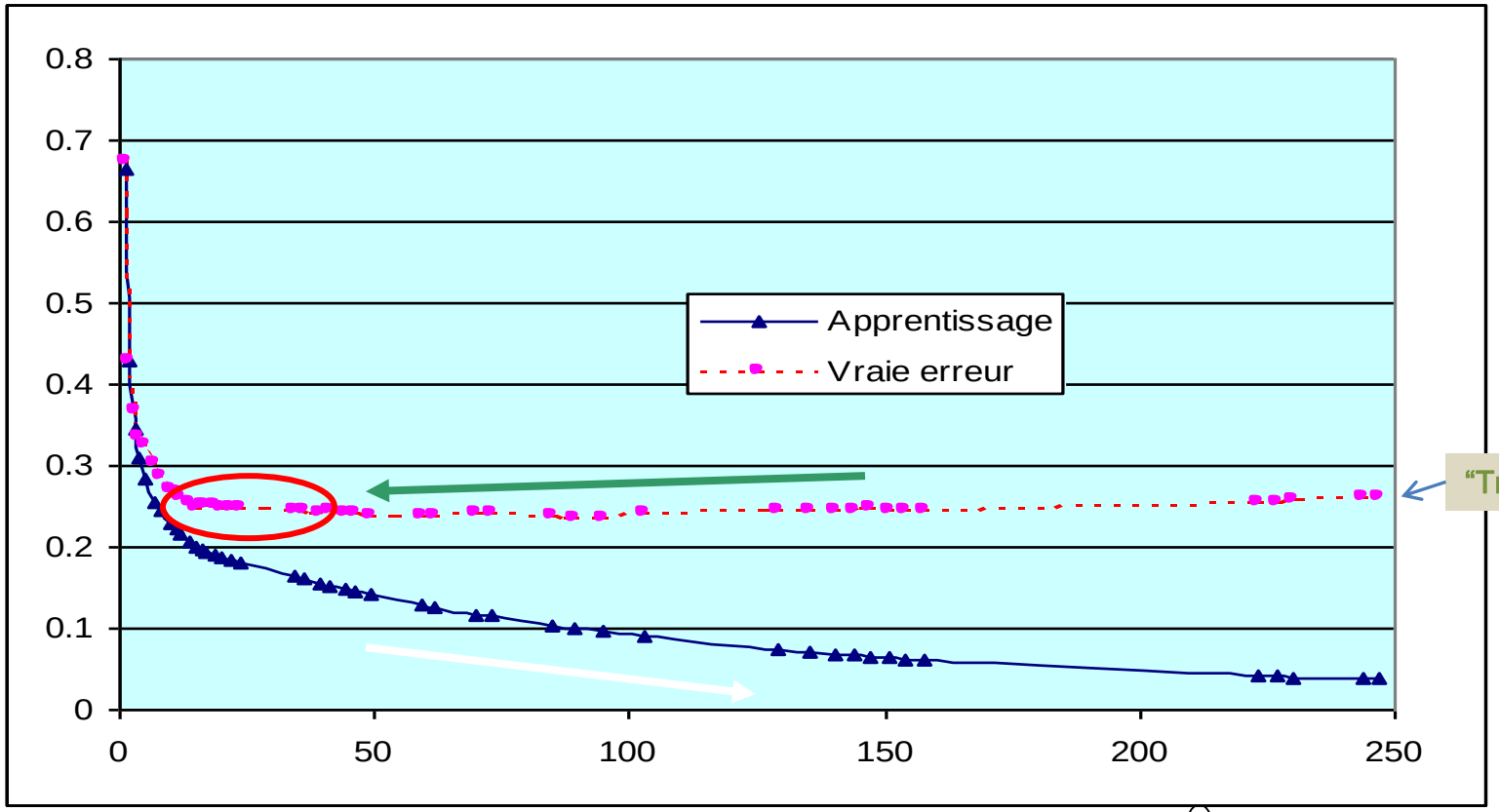
Post-pruning

An additional step in order to avoid the over-dependence to the growing sample

Two steps for the decision tree learning

(1) Growing phase → maximizing the purity of the leaves

(2) (Post) pruning phase → minimizing the "true" error rate



How to obtain a good estimation of the "true" error rate ?

Post-pruning

CART approach with a "pruning set" ("validation sample" in some tools)

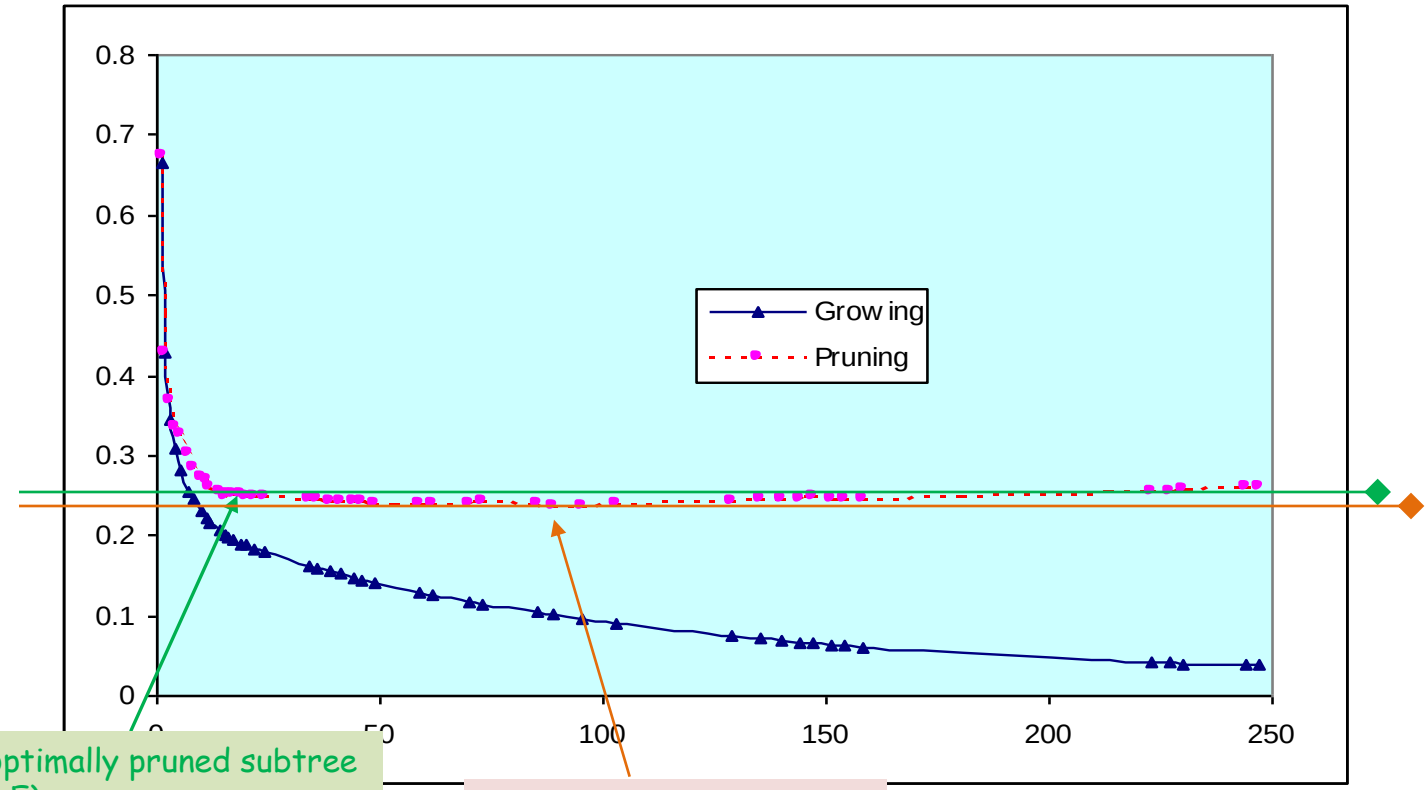
Partitioning the learning sample in two parts
(1) Growing set (#67%)
(2) Pruning set (#33%)

To obtain an honest estimation of the error rate

Cost-complexity pruning process

$$E_\alpha(T) = E(T) + \alpha \times |T|$$

To avoid the overdependence to the pruning sample



Smallest optimally pruned subtree (1-SE RULE)

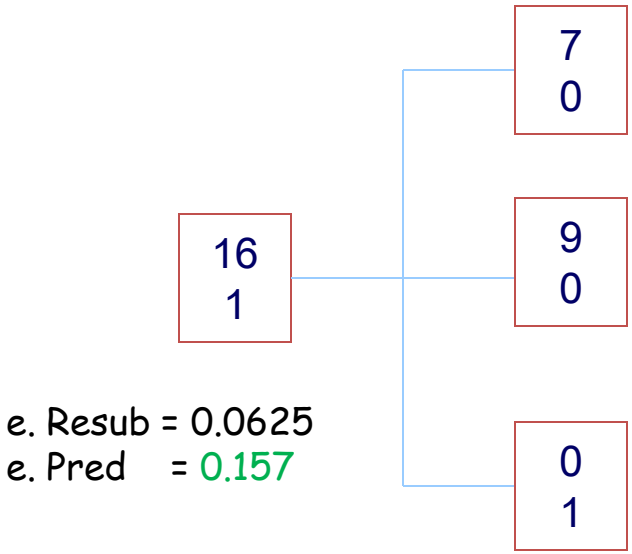
Optimally pruned subtree

Post-pruning

C4.5 approach – Error-based pruning procedure (a variant of the pessimistic pruning)

Predicted (pessimistic) error rate = upper bound of the confidence limit of the resubstitution error rate

➔ the penalty is even stronger that the node size is low



e. Resub = 0.0625
e. Pred = 0.157

e. Resub = 0.0
e. Pred = 0.206

e. Resub = 0.0
e. Pred = 0.143

e. Resub = 0.0
e. Pred = 0.750

Procedure: bottom-up traversal over all nodes i.e. start from the bottom of the tree and examine each non leaf subtree.

Post-pruning because: 0.157 < (7 x 0.206 + 9 x 0.143 + 1 x 0.750)/17 = 0.2174

Summary

Method	CHAID	CART	C4.5
Splitting criterion	Chi-square statistic (Tschuprow's t)	Gini index	Information Gain (Gain Ratio)
Merging process	"Optimal" grouping Test for similarity	Binary grouping	No merging 1 value = 1 leaf
Determining the right sized tree (overall)	Min. size to node split Min. instances in leaves Confidence threshold Tree depth		
Determining the right sized tree (specific)	Pre-pruning Chi-square test for independence	Post-pruning Cost complexity pruning	Post-pruning Error based pruning
Recommended when…	Exploratory phase Handling very large dataset	Classification performance - Reliability No complicated settings	Small dataset Not sensitive to the settings
Not recommended when…	Difficult to set the right settings Tree size is very sensitive to the settings Classification performance	Small dataset Binary tree is not always suitable	Bad behavior of the post-pruning process on very large dataset Tree size increases with the dataset size

Other refinements

Misclassification costs

Considering the misclassification costs in the CART post-pruning

In real problem solving, the misclassification costs are not symmetrical (e.g. cancer prediction)

How to handle the costs during the tree learning ?

Observé \ Prédiction	Cancer	Non-Cancer
Cancer	0	5
Non-Cancer	1	0

Cancer : 10 (33%)
Non-Cancer : 20 (67%)

Not considering the costs

E (cancer) = 20/30 = 0.67
E (non-cancer) = 10/30 = 0.33

Decision = non-cancer → E (Leaf) = 0.33

Considering the costs

C (cancer) = 10/30 x 0 + 20/30 x 1 = 0.67
C (non-cancer) = 10/30 x 5 + 20/30 x 0 = 1.67

Decision = cancer → C (Leaf) = 0.67

CART approach:

(1) Define the sequence of pruned subtree
(2) Select the tree which minimizes the classification cost

$$C_\alpha(T) = C(T) + \alpha \times |T|$$

Misclassification costs

CART : “PIMA Indians Diabetes” dataset

Standard CART

Predicted

Actual

Confusion matrix

	positive	negative	Sum
positive	176	92	268
negative	73	427	500
Sum	249	519	768

Decision tree

- plasma < 144.5000
  - bodymass < 26.3000 then diabete = **negative** (98.10 % of 105 examples)
  - bodymass >= 26.3000
    - age < 24.5000 then diabete = **negative** (91.36 % of 81 examples)
    - age >= 24.5000
      - plasma < 103.5000 then diabete = **negative** (77.61 % of 67 examples)
      - plasma >= 103.5000
        - age < 59.0000
          - pedigree < 0.5285
            - bodymass < 42.2500 then diabete = **negative** (62.03 % of 79 examples)
            - bodymass >= 42.2500 then diabete = **positive** (85.71 % of 7 examples)
          - pedigree >= 0.5285 then diabete = **positive** (64.58 % of 48 examples)
        - age >= 59.0000 then diabete = **negative** (100.00 % of 8 examples)
- plasma >= 144.5000 then diabete = **positive** (69.75 % of 119 examples)

Cost sensitive CART

(FP cost : 1 ; FN cost = 4)

Confusion matrix

	positive	negative	Sum
positive	263	5	268
negative	356	144	500
Sum	619	149	768

Decision tree

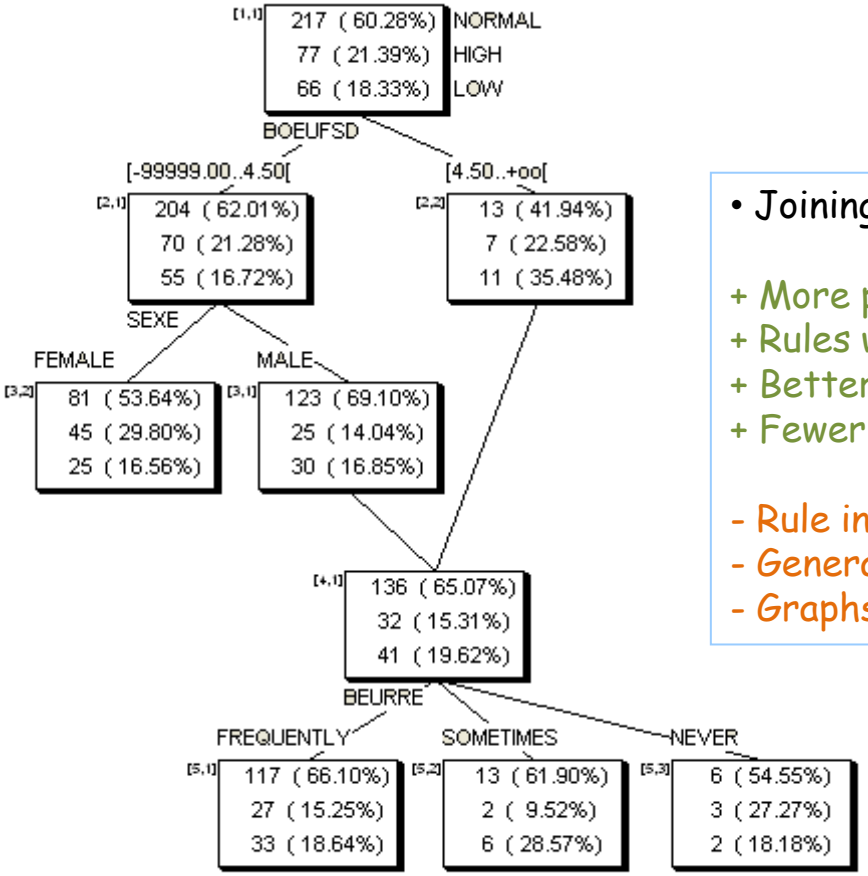
- plasma < 144.5000
  - bodymass < 26.3000 then diabete = **negative** (98.10 % of 105 examples)
  - bodymass >= 26.3000 then diabete = **positive** (30.69 % of 290 examples)
- plasma >= 144.5000 then diabete = **positive** (69.75 % of 119 examples)

The number of false negative is highly decreased

(because the false negative cost is increased)

Induction graphs (Decision graphs)

Generalizing the merging process on any nodes



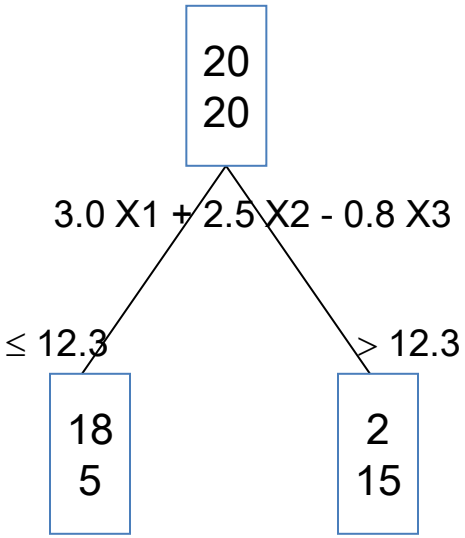
- Joining two or more paths

\+ More powerful representation
\+ Rules with disjunctions (OR)
\+ Better use of the small dataset
\+ Fewer leaves than tree

\- Rule interpretation is not obvious (several paths to one leaf)
\- Generalization performance are not better than tree
\- Graphs with unnecessary joining on noisy dataset

Oblique tree

Using linear combination of variables to split a node



+ More powerful representation
+ The decision tree is shorter

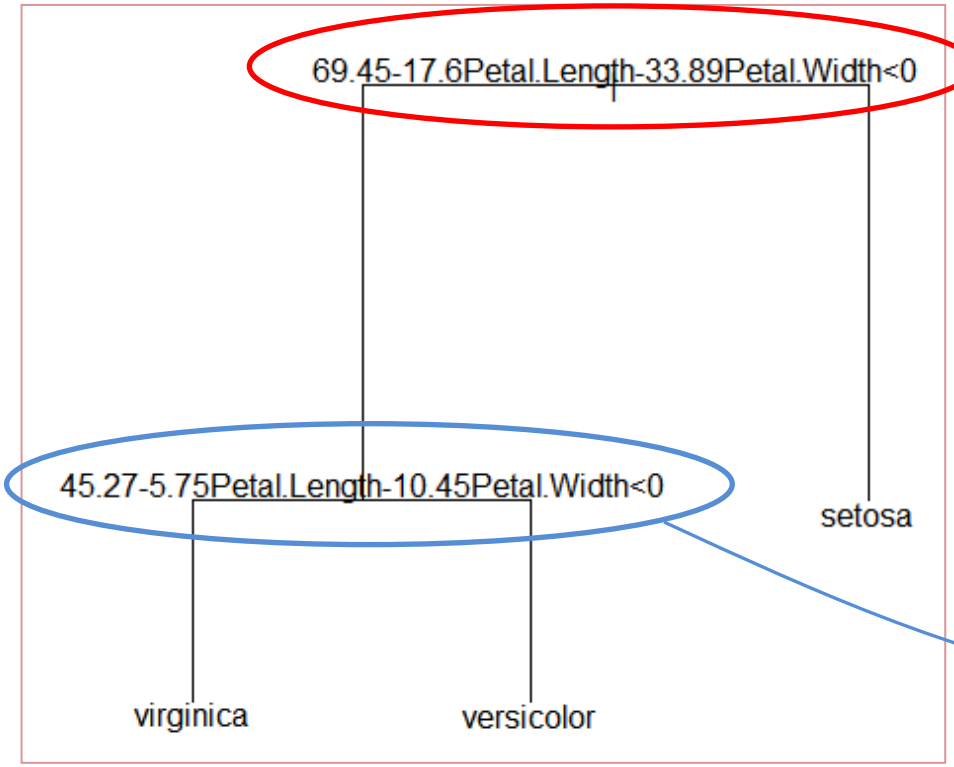
- Interpretation of the rules is harder when the number of nodes increases
- More expensive calculations
- Not really better than standard decision tree methods

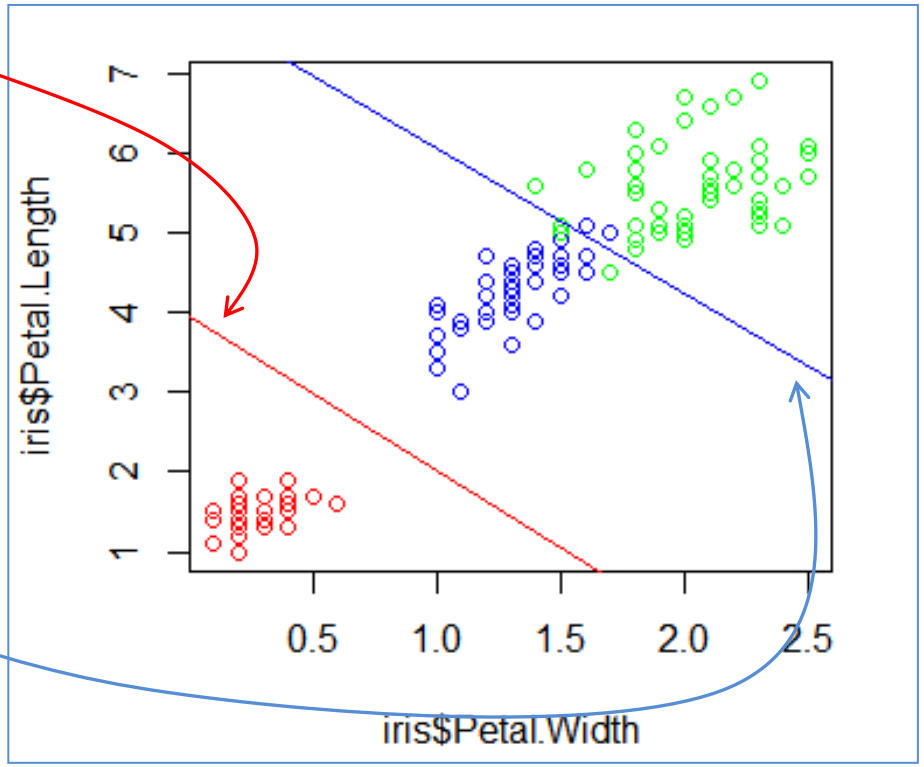
Oblique tree using R ('oblique.tree' package)

IRIS dataset

```
library(oblique.tree)
data(iris)
arbre <- oblique.tree(Species ~ Petal.Length + Petal.Width, data = iris)
plot(arbre)
text(arbre)

plot(iris$Petal.Width,iris$Petal.Length,col=c("red","blue","green")[iris$Species])
abline(a=69.45/17.6,b=33.89/(-17.6),col="red")
abline(a=45.27/5.75,b=10.45/(-5.75),col="blue")
```





Other refinements

- Fuzzy decision trees
- Option trees
- Constructive induction
- Lookahead search

etc… cf. Rakotomalala (2005)

(1) The classification performance improvements on real problems are not significant
(2) The refinements allow above all to obtain shorter trees

References

- L. Breiman, J. Friedman, R. Olshen and C. Stone, "Classification and Regression Trees", Wadsworth Int. Group, 1984.
- G. Kass, "An exploratory technique for Investigating Large Quantities of Categorical Data", Applied Statistics, Vol. 29, N°2, 1980, pp. 119-127.
- R. Quinlan, "C4.5: Programs for machine learning", Morgan Kaufman, 1993.