Decision trees in R – \texttt{rpart()} implementation

- library \texttt{rpart} (but there are also other libraries, e.g. \texttt{tree} and \texttt{party})

\begin{verbatim}
Model=rpart(formula, data=, method=, control=)
\end{verbatim}

- \texttt{?rpart}

- \texttt{formula} in the “traditional” format
  \begin{verbatim}
  \texttt{TargetClass ~ Feature1 + Feature2 + ...}
  \end{verbatim}

- \texttt{data} specifies the input data frame

- \texttt{method} is "class" for classification task

- \texttt{control} other optional parameters
Visualisation using `rpart.plot()`

Visualisation of the model with library `rpart.plot`
Decision Trees – parameters

hypothesis parameters – parameters of the prediction function
- output of the learning algorithm, define the structure of the decision tree

learning parameters – parameters of the learning process
- “configuration” of the learning algorithm
2 phases of decision tree learning:

- growing
- pruning

Learning parameters are used to control these two phases:

- when to stop growing
- how much to prune the tree

... to avoid overfitting and improve performance
Learning parameters in \texttt{rpart}

\texttt{rpart.control}

\textbf{minsplits}

- the minimum number of observations that must exist in a node in order for a split to be attempted

\textbf{cp}

- complexity parameter, influences the depth of the tree

... and others, see \texttt{?rpart.control}

\textbf{Hint:} Try to set different \texttt{cp} and \texttt{minsplits} values in your model learning, then observe and compare different resulting trees
The **cp** parameter is used to control the depth of the growing tree: Any split that does not decrease the **relative training error** by a factor of **cp** is not attempted.

⇒ That means, the learning algorithm measures for each split how it improves the tree relative error and if the improvement is too small, the split will not be performed.

**Relative error** is the error relative to the misclassification error (without any splitting relative error is 100%).
M <- rpart(SENSE ~ A1+A2+A3+A4+A5+A6+A7+A8+A9+A10+A11, data=train, 
    method="class", minsplit=5, cp=0.001)
> M$cptable

<table>
<thead>
<tr>
<th>CP</th>
<th>nsplit</th>
<th>rel error</th>
<th>xerror</th>
<th>xstd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.093053735</td>
<td>0</td>
<td>1.0000000</td>
<td>1.0000000</td>
<td>0.01844043</td>
</tr>
<tr>
<td>0.057667104</td>
<td>1</td>
<td>0.9069463</td>
<td>0.9069463</td>
<td>0.01830335</td>
</tr>
<tr>
<td>0.048492792</td>
<td>2</td>
<td>0.8492792</td>
<td>0.8591088</td>
<td>0.01817412</td>
</tr>
<tr>
<td>0.040629096</td>
<td>3</td>
<td>0.8007864</td>
<td>0.8106160</td>
<td>0.01800131</td>
</tr>
<tr>
<td>0.009174312</td>
<td>4</td>
<td>0.7601573</td>
<td>0.7601573</td>
<td>0.01777550</td>
</tr>
<tr>
<td>0.003931848</td>
<td>9</td>
<td>0.7064220</td>
<td>0.7070773</td>
<td>0.01748535</td>
</tr>
<tr>
<td>0.001000000</td>
<td>10</td>
<td>0.7024902</td>
<td>0.7044561</td>
<td>0.01746957</td>
</tr>
</tbody>
</table>

rel error: relative error on training data

xerror: relative error in x-fold cross-validation

xstd: standard deviation of xerror on x validation folds
Models built with different \textit{cp} value

\begin{itemize}
\item \textit{cp}=0.04
\begin{itemize}
\item yes \quad A4 \geq 0.5 \quad no
\item phone \quad A2 \geq 0.5
\item division \quad A3 \geq 0.5
\item formation \quad A9 \geq 0.5
\item division \quad product
\end{itemize}
\end{itemize}

\begin{itemize}
\item \textit{cp}=0.004
\begin{itemize}
\item yes \quad A4 \geq 0.5 \quad no
\item phone \quad A2 \geq 0.5
\item division \quad A3 \geq 0.5
\item formation \quad A9 \geq 0.5
\item division \quad A5 < 0.5
\item A1 \geq 0.5 \quad product
\item cord \quad A11 \geq 0.5
\item phone \quad A6 < 0.5
\item A10 \geq 0.5 \quad text
\item phone \quad product
\end{itemize}
\end{itemize}
Useful functions

plotcp(model)  visualisation of the cross-validation error depending on cp value

prune(model, cp=)  prune the model based on cp value

> M5$cptable[which.min(M5$cptable[,"xerror"]),"CP"]
[1] 0.001
• visualisation of the cross-validation error depending on cp value
• the horizontal line shows the minimal $x_{error} +$ its standard deviation
How to choose the optimal cp value?

demo code cp-and-pruning.Forbes.R on course page

```r
> m = rpart(profits ~ category + sales + assets + marketvalue, 
  data=F[data.train, 1:8], cp=0.001)
> m$cptable

      CP nsplit rel error  xerror  xstd
 1 0.543259557 0 1.0000000 1.0482897 0.03178559
 2 0.027162978 1 0.4567404 0.4607646 0.02673551
 3 0.007042254 3 0.4024145 0.4446680 0.02640028
 4 0.006036217 6 0.3762575 0.4507042 0.02652763
 5 0.005030181 8 0.3641851 0.4567404 0.02665301
 6 0.004024145 15 0.3279678 0.4768612 0.02705703
 7 0.003018109 19 0.3118712 0.4688129 0.02689795
 8 0.002012072 21 0.3058350 0.4869215 0.02725122
 9 0.001006036 23 0.3018109 0.5171026 0.02780383
10 0.001000000 25 0.2997988 0.5412475 0.02821490
```

NPFL054, 2018 Hladká & Holub Lab 3, page 11/12
How to choose the optimal \( cp \) value?

![Graph showing the relationship between the size of the tree and X-val Relative Error. The graph indicates that as the size of the tree decreases, the X-val Relative Error decreases, reaching a minimum around the Inf mark, and then increases slightly with further decreases in size.]

[NPFL054, 2018 Hladká & Holub Lab 3, page 12/12]