Data-driven methods for predictive modelling

D G Rossiter

Cornell University, Soil & Crop Sciences Section
Nanjing Normal University, Geographic Sciences Department

October 24, 2023
1. Modelling cultures
   - Explanation vs. prediction
   - Data-driven (algorithmic) methods

2. Classification & Regression Trees (CART)
   - Regression trees
   - Sensitivity of Regression Trees
   - Classification trees

3. Random forests
   - Bagging and bootstrapping
   - Building a random forest
   - Variable importance
   - Random forests for categorical variables
   - Predictor selection

4. Cubist

5. Model tuning

6. Spatial random forests

7. Data-driven vs. model-driven methods
1. Modelling cultures
   Explanation vs. prediction
   Data-driven (algorithmic) methods

2. Classification & Regression Trees (CART)
   Regression trees
   Sensitivity of Regression Trees
   Classification trees

3. Random forests
   Bagging and bootstrapping
   Building a random forest
   Variable importance
   Random forests for categorical variables
   Predictor selection

4. Cubist

5. Model tuning

6. Spatial random forests

7. Data-driven vs. model-driven methods
• Statistics starts with **data**: something we have measured

• Data is **generated** by some (unknown) **mechanism**: input (stimulus) \( x \), output (response) \( y \)

• Before analysis this is a **black box** to us, we only have the data itself

• **Two goals** of analysis:
  1. **Prediction** of future responses, given known inputs
  2. **Explanation, Understanding** of what is in the “black box” (i.e., make it “white” or at least “some shade of grey”).
Modelling cultures

Data-driven methods for predictive modelling

Modelling cultures

- Explanation vs. prediction
- Data-driven (algorithmic) methods

Classification & Regression Trees (CART)

- Regression trees
- Sensitivity of Regression Trees
- Classification trees

Random forests

- Bagging and bootstrapping
- Building a random forest
- Variable importance
- Random forests for categorical variables
- Predictor selection

Cubist

Model tuning

Spatial random forests

Data-driven vs. model-driven methods

Data modelling (also called “model-based”)

- assume an empirical-statistical (stochastic) data model for the inside of the black box, e.g., a functional form such as multiple linear, exponential, hierarchical …
- parameterize the model from the data
- evaluate the model using model diagnostics

Algorithmic modelling (also called “data-driven”)

- find an algorithm that produces $y$ given $x$
- evaluate by predictive accuracy (note: not internal accuracy)


https://doi.org/10.1214/ss/1009213726
Data-driven methods for predictive modelling

Modelling cultures

Explanation vs. prediction

Data-driven (algorithmic) methods

Classification & Regression Trees (CART)

Regression trees

Sensitivity of Regression Trees

Classification trees

Random forests

Bagging and bootstrapping

Building a random forest

Variable importance

Random forests for categorical variables

Predictor selection

Cubist

Model tuning

Spatial random forests

Data-driven vs. model-driven methods

Explanation

- Testing a causal theory – why are things the way they are?
- Emphasis is on correct model specification and coefficient estimation
- Uses conceptual variables based on theory, which are represented by measurable variables

Prediction

- Predicting new (space, members of population) or future (time) observations.
- Uses measurable variables only, no need for concepts

The expected prediction error (EPE) for a new observation with value \( x \) is:

\[
\text{EPE} = E\{Y - \hat{f}(x)\}^2 = E\{Y - f(x)\}^2 + \{E(\hat{f}(x)) - f(x)\}^2 + E\{\hat{f}(x) - E(\hat{f}(x))\}^2
\]

Model variance: residual error with perfect model specification (i.e., noise in the relation)

Bias: mis-specification of the statistical model: \( \hat{f}(x) \neq f(x) \)

Estimation variance: the result of using a sample to estimate \( \hat{f}(x) \)
Bias/variance tradeoff: explanation vs. prediction

**Explanation** Bias should be minimized

- correct model specification and correct coefficients → correct conclusions about the theory (e.g., causal relation)

**Prediction** Total EPE should be minimized.

- accept some bias if that reduces the estimation variance
- a simpler model (omitting less important predictors) often has better fit to the data
When does an underspecified model better predict than a full model?

- the data are very **noisy** (large $\sigma$ of the target variable, even with a perfect model);
- the true absolute values of the omitted parameters are small;
- the **predictors are highly correlated**; and
- the sample size is small or the range of omitted variables is narrow.
Mosteller and Tukey (1977): “The whole area of guided regression [an example of, model-based inference] is fraught with intellectual, statistical, computational, and subject matter difficulties.”

It seems we understand nature if we fit a model form, but in fact our conclusions are about the model’s mechanism, and not necessarily about nature’s mechanism.

So, if the model is a poor emulation of nature, the conclusions about nature may be wrong …

…and of course the predictions may be wrong – we are incorrectly **extrapolating**.
The philosophy of data-driven methods

- Also called “statistical learning”, “machine learning”

- Build structures to represent the “black box” without using a statistical model

- Model quality is evaluated by predictive accuracy on test sets covering the target population
  - cross-validation methods can use (part of) the original data set if an independent set is not available
Some data-driven methods

1. Covered in this lecture
   - Classification & Regression Trees (CART)
   - Random Forests (RF)
   - Cubist

2. Others
   - Artificial Neural Networks (ANN)
   - Support Vector Machines
   - Gradient Boosting (xgboost)

3. Relevant R packages: https://cran.r-project.org/web/views/MachineLearning.html
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explaination vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Key references – texts


- Stanford online course, based on James et al. book: https://www.edx.org/course/statistical-learning?index=product&queryID=d36d87b77d62e1e9ba8218e5f169cf38&position=1


1. Modelling cultures
   Explanation vs. prediction
   Data-driven (algorithmic) methods

2. Classification & Regression Trees (CART)
   Regression trees
   Sensitivity of Regression Trees
   Classification trees

3. Random forests
   Bagging and bootstrapping
   Building a random forest
   Variable importance
   Random forests for categorical variables
   Predictor selection

4. Cubist

5. Model tuning

6. Spatial random forests

7. Data-driven vs. model-driven methods
- Typical uses in diagnostics (medical, automotive ...)
- Begin with the full set of possible decisions
- Split into two \textit{binary} subsets based on the values of some \textbf{decision criterion}
- Each branch has a more limited set of decisions, or at least has more information to help make a decision
- Continue \textbf{recursively} on both branches until there is enough information to make a decision

\begin{center}
\textbf{Engineering Flowchart}
\end{center}

\begin{itemize}
\item \textbf{DOES IT MOVE?}
\item \begin{itemize}
\item \textbf{No}
\item \begin{itemize}
\item \textbf{Should it?}
\item \begin{itemize}
\item \textbf{No}
\item \textbf{Problem}
\item \textbf{No Problem}
\end{itemize}
\item \textbf{Yes}
\item \textbf{Problem}
\end{itemize}
\end{itemize}
\item \textbf{Yes}
\item \begin{itemize}
\item \textbf{Should it?}
\item \begin{itemize}
\item \textbf{Yes}
\item \textbf{Problem}
\item \textbf{No Problem}
\end{itemize}
\item \textbf{No}
\end{itemize}
\end{itemize}
A type of decision tree; decision is “what is the predicted response, given values of predictors”?

Aim is to predict the response (target) variable from one or more predictor variables

If response is categorical (class, factor) we build a classification tree

If response is continuous we build a regression tree

Predictors can be any combination of categorical or continuous
Advantages of CART

- A simple model, **no statistical assumptions** other than between/within class variance to decide on splits
  - For example, no assumptions of the distribution of residuals
  - So can deal with non-linear and threshold relations
- No need to transform predictors or response variable
- **Predictive power** is quantified by **cross-validation**; this also controls **complexity** to avoid **over-fitting**
Disadvantages of CART

- No model coefficients to interpret
  - although we can see variable importance overall and influence on each prediction
- Predictive power over a **population** depends on a **sample** that is **representative** of that population
- Quite **sensitive** to the **sample**, even when pruned
- Pruning to a complexity parameter depends on 10-fold cross-validation, which is sensitive to the choice of observations in each fold
- Typically makes only a small number of different predictions ("boxes"), so maps made with it show **discontinuities** ("jumps")
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
- Explanation vs. prediction
- Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
- Regression trees
- Sensitivity of Regression Trees
- Classification trees

Random forests
- Bagging and bootstrapping
- Building a random forest
- Variable importance
- Random forests for categorical variables
- Predictor selection

Cubist

Model tuning

Spatial random forests

Data-driven vs. model-driven methods

R packages

- rpart: “Recursive partitioning for classification, regression and survival trees. An implementation of most of the functionality of the 1984 book by Breiman, Friedman, Olshen and Stone”\(^1\)
- good introduction: vignette("longintro", package="rpart")
- rpart.plot: Plot rpart models

---

• **splitting variable** variable to examine, to decide which branch of the tree to follow

• **root node** 根部节点 variable used for first split; overall mean and total number of observations

• **interior node** 非叶子节点 splitting variable, value on which to split, mean and number to be split

• **leaf** 叶子点 predicted value, number of observations contributing to it

• **cutpoint** of the splitting variable: value used to decide which branch to follow

• **growing** the tree

• **pruning** the tree
• Meuse River soil heavy metals dataset
• **Response** variable: log$_{10}$Zn concentration in topsoil
• **Predictor** variables
  1. distance to Meuse river (continuous)
  2. elevation above sea level (continuous)
  3. flood frequency class (categorical, 3 classes)
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Meuse River study area. Sample points, with Zn concentrations as proportional-size circles, shown in Google Earth
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Example regression tree – first split

Splitting variable: distance to river

Is the point closer or further than 145 m from the river? 101 points yes, 54 points no.
Explanation of first split

- **root**: average $\log_{10}Zn$ of whole dataset $2.56 \log(mg \ kg^{-1})$ fine soil; based on all 155 observations
- **splitting variable at root**: distance to river
- **cutpoint at root**: 145 m
- **leaves**
  - distance $< 145$ m: 54 observations, their mean is $2.87 \log(mg \ kg^{-1})$
  - distance $\geq 145$ m: 101 observations, their mean is $2.39 \log(mg \ kg^{-1})$
- full dataset has been *split* into two *more homogeneous* subsets
For both branches, what is the elevation of the point?

*Note:* this is a coincidence in this case, different splitting variables can be used on different branches.
Explanation of second split

• **interior nodes** were **leaves** after the first split, now ‘roots’ of subtrees
  - *left*: distance \( \geq 145 \) m: 101 observations, their mean is 2.39 log(mg kg\(^{-1}\)) – note smaller mean on left
  - *right*: distance < 145 m: 54 observations, their mean is 2.87 log(mg kg\(^{-1}\))

• **splitting variable at interior node** for < 145 m: elevation
• **cutpoint at interior node** for < 145 m: 8.15 m.a.s.l.
• **splitting variable at interior node** for \( \geq 145 \) m: elevation
• **cutpoint at interior node** for \( \geq 145 \) m: 6.95 m.a.s.l.
• **leaves** 93, 8, 15, 39 observations; means 2.35, 2.84, 2.65, 2.96 log(mg kg\(^{-1}\))

• These leaves are now more homogeneous than the interior nodes.
Example regression tree – third split

Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Example regression tree – third split

\[ \text{dist.m} \geq 145 \]
\[ \text{elev} \geq 6.94 \]
\[ \text{dist.m} \geq 230 \]
\[ \text{dist.m} \geq 75 \]
\[ < 145 \]
\[ < 6.94 \]
\[ < 230 \]
\[ < 75 \]

\[ n=155 \]
\[ n=101 \]
\[ n=93 \]
\[ n=78 \]
\[ n=15 \]
\[ n=8 \]
\[ n=39 \]
\[ n=11 \]
\[ n=28 \]
\[ n=8 \]
\[ n=15 \]
Example regression tree – fourth split

```
dist.m >= 145
  elev >= 6.94
    dist.m >= 230
      elev >= 9.03
        dist.m >= 75
          < 145
            elev >= 8.15
              < 8.15
                2.96
                n=39
            < 6.94
              elev >= 230
                < 230
                  2.31
                  n=78
              < 9.03
                2.22
                n=29
            < 75
              2.85
              n=11
        < 75
          3
          n=28
    < 230
      2.87
      n=54
  < 6.94
    elev >= 9.03
      dist.m >= 75
        < 145
          elev >= 8.15
            < 8.15
              2.96
              n=39
          < 6.94
            elev >= 230
              < 230
                2.31
                n=78
            < 9.03
              2.22
              n=29
          < 75
            2.85
            n=11
      < 75
        3
        n=28
  < 9.03
    2.39
    n=101
```
Example regression tree – fifth split

```
<table>
<thead>
<tr>
<th>dist.m &gt;= 145</th>
<th>2.56</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=155</td>
</tr>
<tr>
<td>dist.m &gt;= 230</td>
<td>2.39</td>
</tr>
<tr>
<td></td>
<td>n=93</td>
</tr>
<tr>
<td>elev &gt;= 6.94</td>
<td>2.35</td>
</tr>
<tr>
<td></td>
<td>n=78</td>
</tr>
<tr>
<td>elev &gt;= 9.03</td>
<td>2.22</td>
</tr>
<tr>
<td></td>
<td>n=29</td>
</tr>
<tr>
<td>dist.m &gt;= 670</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td>n=8</td>
</tr>
<tr>
<td>dist.m &gt;= 365</td>
<td>2.26</td>
</tr>
<tr>
<td></td>
<td>n=21</td>
</tr>
<tr>
<td>dist.m &lt; 670</td>
<td>2.34</td>
</tr>
<tr>
<td></td>
<td>n=31</td>
</tr>
<tr>
<td>dist.m &lt; 365</td>
<td>2.41</td>
</tr>
<tr>
<td></td>
<td>n=18</td>
</tr>
<tr>
<td>dist.m &lt; 600</td>
<td>2.55</td>
</tr>
<tr>
<td></td>
<td>n=15</td>
</tr>
<tr>
<td>dist.m &lt; 550</td>
<td>2.84</td>
</tr>
<tr>
<td></td>
<td>n=8</td>
</tr>
<tr>
<td>dist.m &lt; 500</td>
<td>2.65</td>
</tr>
<tr>
<td></td>
<td>n=15</td>
</tr>
<tr>
<td>dist.m &lt; 450</td>
<td>2.85</td>
</tr>
<tr>
<td></td>
<td>n=11</td>
</tr>
<tr>
<td>dist.m &lt; 400</td>
<td>2.92</td>
</tr>
<tr>
<td></td>
<td>n=14</td>
</tr>
<tr>
<td>dist.m &lt; 350</td>
<td>3.07</td>
</tr>
<tr>
<td></td>
<td>n=7</td>
</tr>
<tr>
<td>dist.m &lt; 300</td>
<td>3.08</td>
</tr>
<tr>
<td></td>
<td>n=7</td>
</tr>
<tr>
<td>elev &gt;= 7.69</td>
<td>2.37</td>
</tr>
<tr>
<td></td>
<td>n=49</td>
</tr>
<tr>
<td>elev &gt;= 9.03</td>
<td>2.35</td>
</tr>
<tr>
<td></td>
<td>n=93</td>
</tr>
<tr>
<td>elev &gt;= 6.99</td>
<td>2.22</td>
</tr>
<tr>
<td></td>
<td>n=29</td>
</tr>
<tr>
<td>elev &gt;= 8.15</td>
<td>2.39</td>
</tr>
<tr>
<td></td>
<td>n=101</td>
</tr>
<tr>
<td>elev &lt; 6.94</td>
<td>2.31</td>
</tr>
<tr>
<td></td>
<td>n=78</td>
</tr>
<tr>
<td>elev &lt; 9.03</td>
<td>2.34</td>
</tr>
<tr>
<td></td>
<td>n=31</td>
</tr>
<tr>
<td>elev &lt; 6.99</td>
<td>2.26</td>
</tr>
<tr>
<td></td>
<td>n=21</td>
</tr>
<tr>
<td>elev &lt; 9.03</td>
<td>2.41</td>
</tr>
<tr>
<td></td>
<td>n=18</td>
</tr>
<tr>
<td>elev &lt; 6.99</td>
<td>2.55</td>
</tr>
<tr>
<td></td>
<td>n=15</td>
</tr>
<tr>
<td>elev &lt; 9.03</td>
<td>2.84</td>
</tr>
<tr>
<td></td>
<td>n=8</td>
</tr>
<tr>
<td>elev &lt; 6.94</td>
<td>2.65</td>
</tr>
<tr>
<td></td>
<td>n=15</td>
</tr>
<tr>
<td>elev &lt; 9.03</td>
<td>2.85</td>
</tr>
<tr>
<td></td>
<td>n=11</td>
</tr>
<tr>
<td>elev &lt; 6.94</td>
<td>2.92</td>
</tr>
<tr>
<td></td>
<td>n=14</td>
</tr>
<tr>
<td>elev &lt; 9.03</td>
<td>3.07</td>
</tr>
<tr>
<td></td>
<td>n=7</td>
</tr>
<tr>
<td>elev &lt; 6.99</td>
<td>3.08</td>
</tr>
<tr>
<td></td>
<td>n=7</td>
</tr>
</tbody>
</table>
```
Example regression tree – maximum possible splits

Example regression tree – maximum possible splits
How are splits decided?

1. Take all possible *predictors* and all possible *cutpoints*
2. Split the data(sub)set at *all combinations*
3. Compute some *measure of discrimination* for all these – i.e., a measure which determine which split is “best”
4. Select the predictor/split that most discriminates
How are splits decided? – Continuous response

Select the predictor/split that most increases *between-class* variance (this decreases *pooled within-class* variance):

$$
\sum_{\ell} \sum_i (y_{\ell,i} - \bar{y}_\ell)^2
$$

- $y_{\ell,i}$ value $i$ of the target in leaf $\ell$
- $\bar{y}_\ell$ is the mean value of the target in leaf $\ell$

So the set of leaves are **more homogeneous**, on average, than the root.
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Example split (1)

```r
> # all the possible cutpoints for distance to river
> (distances <- sort(unique(meuse$dist.m)))

[1]  10  20  30  40  50  60  70  80 100 110 120 130 140 150
[29] 330 340 350 360 370 380 390 400 410 420 430 440 450 460
[43] 470 480 490 500 520 530 540 550 560 570 630 650 660 680
[57] 690 710 720 750 760 860 1000

> # try them all
> for (i in 1:length(distances)) { # try them all
>   branch.less <- meuse$zinc[meuse$dist.m < distances[i]]
>   branch.more <- meuse$zinc[meuse$dist.m >= distances[i]]
>   rss.less <- sum((branch.less-mean(branch.less))^2)
>   rss.more <- sum((branch.more-mean(branch.more))^2)
>   rss <- sum(rss.less + rss.more)
>   results.df[i,2:5] <- c(rss.less, rss.more, rss, 1-rss/tss)
> } # find the best split
> ix.r.squared.max <- which.max(results.df$r.squared)
> print(results.df[ix.r.squared.max,])

   distance rss.less rss.more     rss r.squared
13     140   7127795   3030296 10158091  0.510464

> # plot the results
plot(r.squared ~ distance, data=results.df, type="h",
     col=ifelse(distance==d.threshold,"red","gray"))
```
Example split (2): $R^2$ vs. cutpoint – distance to river

Try to split the **root node** on this predictor:

Best cutpoint is 140 m; this explains 51% of the total variance
Example split (3): $R^2$ vs. cutpoint – elevation

Try to split the **root node** on this predictor:

Best cutpoint is 7.48 m.a.s.l.; this only explains 35% of the total variance; so use the distance to river as the first split
Example split (4a): left first-level leaf

Try to split the **left first-level leaf** (101 observations):

Best cutpoint is 6.99 m.a.s.l.; this explains 93.0% of the variance *in this group*. Splitting at 290 m distance would explain 89.1%.

So split this leaf on *elevation* – it becomes an *interior node*
Try to split the **right first-level leaf** (54 observations):

Best cutpoint is 8.23 m.a.s.l.; this explains 76.6% of the variance *in this group*. Splitting at 60 m distance would explain 72.6%.

So split on *elevation* – it becomes an *interior node*. 
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Internal representation of the tree

```
(node), split, n, deviance, yval
* denotes terminal node
1) root 155 1.513633e+01 2.556160
   2) dist.m>=145 101 4.652110e+00 2.388584
      4) elev>=6.943 93 2.835457e+00 2.349952
         8) dist.m>=230 78 1.735841e+00 2.311717
            16) elev>=9.028 29 3.657173e-01 2.218499
               32) dist.m>=670 8 7.842734e-03 2.107362 *
               33) dist.m<670 21 2.214203e-01 2.260837
                  66) elev>=9.5415 9 7.287443e-02 2.187184 *
                  67) elev<9.5415 12 6.310607e-02 2.316077 *
            17) elev<9.028 49 9.689808e-01 2.366887
               34) dist.m>=250 47 9.067079e-01 2.359533
                  68) dist.m>=525 10 1.400007e-01 2.302179
                     136) dist.m<660 8 5.166170e-02 2.262676 *
                     137) dist.m>=660 2 2.591870e-02 2.460192 *
               69) dist.m<525 37 7.249212e-01 2.375035
                  138) dist.m<510 36 6.193825e-01 2.366133
                     276) elev>=8.4485 13 1.107937e-01 2.319408 *
                     277) elev<8.4485 23 4.641635e-01 2.392544
(etc.)
```
node) the node number, a binary representation of its position in the tree

• e.g., 67 = 1000011 right/left/left ...

split the splitting variable used at this node, the splitting value, and the splitting direction ($\geq$ or $<$)

n number of observations at this node

deviance the sum of squared differences in fit of this mean value and the actual values for the observations in this node

yval the mean value of the observations in this node

The deviance for node $i$ is computed as the total sum of squares:

$$D_i = \sum_{j} (y_j - \mu_i)^2$$

This is progressively smaller at lower levels of the tree.
Controlling tree complexity

- Fitting a full tree, until there is only one observation per leaf, is always **over-fitting** to the sample set, and will not be a good **predictor** of the population.
- A full tree fits some **noise** as well as **structure**.
- Can control by the **analyst** or automatically by **pruning** (see below).
- Analyst can specify:
  - Minimum number of observations in a leaf (fewer: no split is attempted): **minsplit**
  - Maximum depth of tree: **maxdepth**
  - Minimum improvement in pooled within-class vs. between-class variance: **cp** (see below)
Predicting with the fitted tree

- A simple ‘model’ is applied to each leaf:
  - Response variable continuous numeric: mean of observed data in leaf
  - Categorical variable: most frequent category in leaf
- Value at new location is predicted by running the covariate data down the tree
Question: What is the predicted value for a point 100 m from the river and 9 m a.s.l. elevation?
Only one prediction per leaf, applies to all points falling in the leaf.
Pruning – why?

- The splitting can continue until each calibration observation is in its own leaf
- This is almost always over-fitting to the current dataset
- What we want is a tree for the best prediction
- Solution: grow a full tree; then prune it back to a simpler tree with the best predictive power
  - Similar to using the adjusted $R^2$ to avoid over-fitting a multiple linear regression
The cp “complexity parameter”: the increase in fit *below* which the tree is not further expanded

- Default value for the ‘prune’ function is 0.01 (1% increase in $R^2$)
- Can be set by the analyst during **growing**
  - `rpart` optional parameter
- Can also be used as a target for **pruning**
  - `prune` parameter
Pruning – how?

• From the leaves backwards, remove splits that do not decrease the overall lack of fit by a factor of \( cp \)
• Q: How to decide on the value of \( cp \) that gives the best predictive tree?
• A: Use the cross-validation error, also called the out-of-bag error.
  • apply the model to the original data split \( K \)-fold (default 10), each time excluding some observations; compare predictions to actual values
  • Note how this fits the philosophy of data-driven approaches: predictive accuracy is the criterion
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

X-validation error vs. complexity parameter

Horizontal line is 1 standard error above the minimum error. Usually choose the largest \( cp \) below this; here \( cp = 0.01299 \) (about 1.3% improvement in \( R^2 \)).
Full and pruned trees

Full tree built with \( cp=0.003 = 0.3\% \); 27 leaves; pruned to 8 (\( cp=0.013 = 1.3\% \))

**Interpretation:** a noisy dataset if using these two predictors
Variable importance

- Unlike with linear regression we do not get any coefficient or its standard error for each predictor.
- So to evaluate the importance of each predictor we see how much it’s used in the tree:
  - simple:
    - sum of gain in $R^2$ over all splits based on the predictor
  - complicated:
    - permute predictor values;
    - use these to re-build the tree;
    - compute cross-validation error;
    - the larger the difference, the more important
- This also includes **surrogates**, i.e., variables that could have substituted for missing values in another predictor.
Variable importance – example

```
variableImportance

dist.m   52.378203
elev     36.616987
soil     6.128099
ffreq    4.876711
```

Normalized to sum to 100% of the gain in $R^2$

Distance to river is most important.

soil and ffreq were only important as surrogates for dist.m and/or elev, which were the only variables used in the pruned tree.
Surrogate variables

• What is to be done with observations missing a splitting variable?
  • e.g., in this case, a new point without a record of its flooding frequency

• Rather than classify the new observation as NA, maybe we can find **surrogate** variables in the training set: variables that can predict the value of the missing splitting variable at that split.
  • e.g., maybe elev is more useful than random assignment to predict flood frequency.

• These are ranked, and included in the **variable importance**, even if they are not used in the (pruned) tree.

• Observations missing the split variable are classified using the first surrogate; if that is missing also, the second surrogate, etc.

• Observations missing the split variable and all surrogates are randomly assigned to one of the branches.
Map predicted from Regression Tree

This tree: $\log_{10}Zn$ predicted from dist (45% importance); E (17%); soil (15%); N (11%); ffreq. (11%).
Sensitivity of Regression Trees to sample

• **Question:** how sensitive are Regression Trees to the sample?
• **Experiment:** build trees from random samples of 140 of the 155 observations (only 10% not used!)
  • How different are the optimized trees and the predictive maps?
  • What is the distribution of the optimal complexity parameter and the out-of-bag (predictive) error?
Data-driven methods for predictive modelling

Modelling cultures
- Explanation vs. prediction
- Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
- Regression trees
- Sensitivity of Regression Trees
- Classification trees

Random forests
- Bagging and bootstrapping
- Building a random forest
- Variable importance
- Random forests for categorical variables
- Predictor selection

Cubist

Model tuning

Spatial random forests

Data-driven vs. model-driven methods

Sensitivity: complexity and out-of-bag error
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Sensitivity: trees
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Sensitivity: predictive maps
Regression trees are sensitive to the observations

- This is a problem!
- **Solution**: why have one tree when you can have a forest?
Classification trees

- Target variable is a **categorical** (classified) variable
- Tree structure is the same
- Stopping/pruning criterion is the minimum increase in node purity
Select the predictor/split that minimizes the impurity of the set of leaves:

- Misclassification rate: \( \frac{1}{N_m} \sum_{i \in R} I(y_i \neq k(m)) \)
  - \( N_m \): number of observations at node \( m \)
  - \( R_m \): the set of observations
  - \( k(m) \) is the majority class; \( I \) is the logical T/F function

- Impurity is maximal when all classes have same frequency, and minimal when only one class has any observations in the leaf

So the set of leaves are purer (less confusion), on average, than the root.
Predicting with the fitted tree

- A simple ‘model’ is applied to each leaf:
  - Categorical variable: most frequent category in leaf (majority)
  - Value at new location is predicted by running the covariate data down the tree
Data-driven methods for predictive modelling

- (Meuse river) flood frequency **class** (3 levels) predicted from distance to river and elevation
- Result (pruned): number of observations in each class (left); proportion (right) – note class 3 not predicted!
1. **Modelling cultures**
   - Explanation vs. prediction
   - Data-driven (algorithmic) methods

2. **Classification & Regression Trees (CART)**
   - Regression trees
   - Sensitivity of Regression Trees
   - Classification trees

3. **Random forests**
   - Bagging and bootstrapping
   - Building a random forest
   - Variable importance
   - Random forests for categorical variables
   - Predictor selection

4. **Cubist**

5. **Model tuning**

6. **Spatial random forests**

7. **Data-driven vs. model-driven methods**
• Instead of relying on a *single* (hopefully best) tree, maybe it is better to fit *many* trees.

• But...how to obtain *multiple* regression trees if we have only *one* data set?
  • Go into field and collect *new* sample data? too expensive and impractical.
  • *Split* the dataset and fit trees to the separate parts? Too few observations to build a reliable tree.
  • **Solution:** Use the *single* sample to generate an *ensemble* (group) of trees; use these together to predict.
“Bag” = a group of samples “in the bag”; others “out-of-bag”

Suppose we have a large sample that is a good representation of the study area
  - i.e., sample frequency distribution is close to population frequency distribution

Generate a new sample is generated by sampling from the sample!
Bagging (2) – Booststrapping

Standard method for sampling in bagging is called **bootstrapping**

- Select **same number of points** as in sample
- Sample **with replacement** (otherwise you get the same sample)
- So **some observations are used more than once**!
- But, **the sample is supposed to represent the population**, so these could be values that would have been obtained in a new field sample.
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Sampling with replacement

```r
> # sample 20 times from (1, 2, ... 20) with replacement
> (my.sample <- sample(1:20, 20, replace=TRUE))
[1]  7 13  5  2  1  9 19  1  6  2  9  9 12  4 11  9  5 20 20 11
> sort(my.sample)
[1]  1  1  2  2  4  5  5  6  7  9  9  9  9 11 11 12 13 19 20 20
> (1:20) %in% my.sample    # in bag
> !((1:20) %in% my.sample) # Out-of-bag
```
## Example: 10 bootstrap samples from the integers 1 ... 20 – sorted

```
<table>
<thead>
<tr>
<th></th>
<th>b1</th>
<th>b2</th>
<th>b3</th>
<th>b4</th>
<th>b5</th>
<th>b6</th>
<th>b7</th>
<th>b8</th>
<th>b9</th>
<th>b10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>7</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>11</td>
<td>9</td>
<td>7</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>8</td>
<td>13</td>
<td>10</td>
<td>9</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>8</td>
<td>15</td>
<td>10</td>
<td>9</td>
<td>13</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>9</td>
<td>15</td>
<td>10</td>
<td>11</td>
<td>13</td>
<td>13</td>
<td>8</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>12</td>
<td>17</td>
<td>12</td>
<td>16</td>
<td>10</td>
<td>13</td>
<td>14</td>
<td>13</td>
<td>10</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>13</td>
<td>17</td>
<td>14</td>
<td>16</td>
<td>14</td>
<td>13</td>
<td>15</td>
<td>14</td>
<td>14</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>18</td>
<td>14</td>
<td>17</td>
<td>16</td>
<td>14</td>
<td>16</td>
<td>15</td>
<td>17</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>15</td>
<td>17</td>
<td>16</td>
<td>16</td>
<td>18</td>
<td>15</td>
<td>17</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td>16</td>
<td>19</td>
<td>15</td>
<td>18</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td>15</td>
<td>18</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>19</td>
<td>16</td>
<td>19</td>
<td>17</td>
<td>19</td>
<td>18</td>
<td>16</td>
<td>19</td>
<td>14</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>19</td>
<td>17</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>17</td>
<td>20</td>
<td>17</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>19</td>
<td>19</td>
<td>18</td>
<td>20</td>
<td>19</td>
<td>19</td>
<td>20</td>
<td>17</td>
<td>20</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>19</td>
<td>18</td>
<td>20</td>
<td>19</td>
<td>19</td>
<td>20</td>
<td>19</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>
```
Forests with bagging – method

- Fit a **full regression tree** to each bootstrap sample; *do not prune*
- Each bootstrap sample results in a **tree** and in a **predicted value** for any combination values of the predictors
- Prediction is the **average** of the individual predictions from the “forest” of regression trees
- Jumps in predictions are **smoothed**; more precise predictions
Forest with bagging – limitations

- All predictors are tried at each split, so trees tend to be similar.
- Some predictors may never enter into the trees → missing source of diversity.
- Solution: random forest variation of bagging – two sources of randomness:
  - Random 1: sampling by bagging
  - Random 2: choice of predictors at each split (see next)
• Multiple samples obtained by bootstrapping, used to build trees (as in bagging)
• Average predictions over all trees (as in bagging)
• Besides, in each internal node a random subset of splitting variables (predictors) is used
  • Extra source of diversity among trees
  • Predictors that are “outcompeted” in bagging by stronger competitors may now enter the group of trees
randomForest older package, slow, some nice helper functions

ranger “a fast implementation of random forests or recursive partitioning, particularly suited for high dimensional data”.

• Classification, regression, and survival forests are supported.

caret “Classification And REgression Training”; contains functions to streamline the model training process for complex regression and classification problems
Selecting predictors at each split

- **randomForest, ranger parameter** `mtry`: Number of variables randomly sampled as candidates at each split.
  - **ranger** default \( \lfloor \sqrt{p} \rfloor \), where \( p \) is number of possible predictors
  - example: 60 predictors \( \rightarrow \lfloor \sqrt{60} \rfloor = \lfloor 7.74 \rfloor = 7 \) tried at each split
  - **randomForest** default \( \lfloor p/3 \rfloor \)
  - example: 60 predictors \( \rightarrow \lfloor 60/3 \rfloor = \lfloor 20 \rfloor = 20 \) tried at each split

- Can be **tuned** with caret, see below.
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Other control parameters

- number of trees in the forest
  - `ranger` parameter `min.node.size`
  - `randomForest` parameter `ntree`
  - default = 500

- minimal node size
  - `ranger` parameter `min.node.size`
  - `randomForest` parameter `nodesize`
  - default = 5

- *(optional)* names of variables to always try at each split; weights for sampling of training observations (to compensate for unbalanced samples)
Average prediction of many trees, comes close to actual value
Out-of-bag ("OOB") evaluation

- In a bootstrap sample not all samples are present: sampling is with *replacement*.
- Sample data not in bootstrap sample: **out-of-bag** sample: these were *not* used to build the tree.
- These data can be used for **evaluation** ("validation"):  
  - Use the tree fitted on the bootstrap sample to predict at out-of-bag data, i.e., observations *not* used in that bootstrap sample.
  - Compute **squared prediction error** for out-of-bag data.
- This gives a very good estimate of the true prediction error *if* the sample was **representative** of the population.
Out-of-bag RF predictions vs. observed

Average prediction of many trees *not* using an observation. Further from actual value; **better estimate of predictive power**
How many trees are needed to make a forest?

- Plot mean squared out-of-bag error against number of trees
- Check whether this is stable
- If not, increase number of trees

![Plot](image-url)
Variable importance

Importance quantified by permutation accuracy:

- randomize (permute) values of a predictor
  - so the predictor can not have any relation with the target
- build a random forest with this randomized predictors and the other (non-randomized) ones
- compute OOB error; compare with OOB error without randomization
  - the larger the difference, the more important
- Example:

  % Increase in MSE under randomization
  
  ffreq  9.4
  dist.m 67.5
  elev   54.0
Variable importance plot

- dist.m
- elev
- x
- y
- ffreq
- lime
- soil

%IncMSE
The effect of each variable, with the others held constant at their means/most common class.
Data-driven methods for predictive modelling

Explanation vs. prediction

Data-driven (algorithmic) methods

Classification & Regression Trees (CART)

Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest

Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Two-way partial dependence

Prediction of the forest for different values of dist.m and elev
Data-driven methods for predictive modelling

Examining the forest – at what depth in the trees are predictors used?

Earlier in tree → most discriminating
Variable importance for individual predictions

- It is also possible to see how much each variable influences each prediction separately
  - randomForest: option localImp=TRUE; ranger: option local.importance = TRUE
  - package iml “Interpretable Machine Learning”³
- Permutation tests, individual conditional expectation curves, Shapley values (see below)
- This shows which predictions are most influenced by which variables, not the importance of the variables overall

Data-driven methods for predictive modelling

Individual conditional expectation curves

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods
• An idea from game theory, developed by Shapley in 1953\(^4\)
• This divides the contribution of the several variables to a prediction according to their average *marginal* contribution across all possible “coalitions” of variables
• “the only attribution method that satisfies the properties Efficiency, Symmetry, Dummy and Additivity, which together can be considered a definition of a fair payout.”
• Shows the *direction* of the contribution, towards a higher or lower value
• Well-explained by Molnar\(^5\)


Shapley values for the Meuse Zn ‘ranger‘ model – by observation

(closest to river) (furthest from river)
Data-driven methods for predictive modelling

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Shapley values for the Meuse Zn ‘ranger‘ model – by predictor
• Recall: RF is built from many trees, each tree makes a prediction at each location
• These are averaged to get a “best” predictive map
• However, the set of predictions can be considered a probability distribution of the true value
• From this we can make a map of any quantile, e.g., 5% and 95% confidence limits, or prediction interval width
RF uncertainty vs. RK uncertainty

95% prediction interval for topsoil pH prediction from 2024 point observations and 18 covariates Languedoc-Roussillon region (F)
References for quantile random forests


Random forests for categorical variables

- Target variable is **categorical**, i.e., a class
  - Example: Meuse river flooding frequency classes (every year, every 2–5 years, rare or none)
- Final prediction is the class predicted by the **majority** of the regression trees in the forest
- Can also see the **probability** for each class, by predicting with the model with the **type=”prob”** argument to **predict.randomForest**.
Data-driven methods for predictive modelling

Prob. class 1

Prob. class 2

Prob. class 3

Predicted class probability
Data-driven methods for predictive modelling

Modelling cultures
- Explanation vs. prediction
- Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
- Regression trees
- Sensitivity of Regression Trees
- Classification trees

Random forests
- Bagging and bootstrapping
- Building a random forest
- Variable importance
- Random forests for categorical variables
- Predictor selection

Cubist

Model tuning

Spatial random forests

Data-driven vs. model-driven methods

Predicted most probable class
• The vector of predicted probabilities can be used for several uncertainty measures.
• These can be used to assess the certainty of the prediction, and maybe reject predictions that are too unsure.
• **Confusion index**: How much more probable is the majority class $\mu_{\text{max}}$ to its next competitor $\mu_{(\text{max}-1)}$

$$\text{CI} = 1 - \{\mu_{\text{max}} - \mu_{(\text{max}-1)}\}$$

• **Shannon entropy**: Diversity of predictions. For a prediction $z$ with $n$ possible classes:

$$H_z = - \sum_{i=1}^{n} \hat{\pi}(z_i) \cdot \log_n \hat{\pi}(z_i)$$
• naïve agreement: how often a class in the training set is correctly predicted – see with a **confusion matrix** ("cross-classification")

• Out-of-bag (OOB) estimate of error rate

• **Gini impurity**: how often a *randomly chosen* training observation would be *incorrectly* assigned ... ...if it were *randomly labeled* according to the *frequency distribution* of labels in the subset.
A **confusion matrix** (a.k.a. cross-classification matrix) of actual (columns) vs. predicted (rows) classes:

Confusion matrix:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>class.error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>77</td>
<td>7</td>
<td>0</td>
<td>0.08333333</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>40</td>
<td>5</td>
<td>0.16666667</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>9</td>
<td>13</td>
<td>0.43478261</td>
</tr>
</tbody>
</table>
• Recall: aim is **predictive** power

• Intuitively, the class proportions of OOB predictions should match those in the **population**

• If the **sample** proportions match the **population** proportions, this is a good basis
  • although notice, some classes may just be poorly-predicted by the predictor set

• If **not**, can use *a priori* class frequencies to **weight** the selection of observations for in/out-of-bag
  • *ranger* option case.weights

• Another use for case weights: some observations are more reliable than others (similar to weighted linear regression)
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables

Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Predictor selection

- **Problem**: large number of possible predictors, can lead to ...
  - Computational inefficiency
  - Difficult interpretation of variable importance, correlated predictors can substitute for each other
  - Meaningless good fits, even if using cross-validation

- **Solution 1**: expert selection from “known” relations
  - this is then not pure “data mining” for unsuspected relations

- **Solution 2**: (semi-)automatic feature selection, see next.

---

Feature selection methods

Wrapper methods: “evaluate multiple models using procedures that add and/or remove predictors to find the optimal combination that maximizes model performance.”

- risk of over-fitting
- high computational load

Filter methods: “evaluate the relevance of the predictors outside of the predictive models and subsequently model only the predictors that pass some criterion”

- does not account for correlation among predictors
- does not directly assess model performance
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Recursive feature elimination

- A “wrapper” method
- Implemented in caret::rfe “Backwards Feature Selection” function
- Algorithm: “Recursive Feature Elimination (RFE) incorporating resampling”

1. Partition data into training/test sets via resampling
2. Start with full model, compute variable importance
3. for each proposed subset size
   1. Re-compute model with reduced variable sets
   2. Calculate performance profiles using test samples
4. Determine optimum number of predictors
• From the documentation of the caret package (§5).
Data-driven methods for predictive modelling

1. Modelling cultures
   Explanation vs. prediction
   Data-driven (algorithmic) methods

2. Classification & Regression Trees (CART)
   Regression trees
   Sensitivity of Regression Trees
   Classification trees

3. Random forests
   Bagging and bootstrapping
   Building a random forest
   Variable importance
   Random forests for categorical variables
   Predictor selection

4. Cubist

5. Model tuning

6. Spatial random forests

7. Data-driven vs. model-driven methods
• Similar to CART, but instead of single values at leaves it creates a multivariate linear regression for the cases in the leaf

• **Advantage vs. CART**: predictions are continuous, not discrete values equal to the number of leaves in the regression tree.
  • Also can be improved with nearest-neighbours, see below

• **Advantage vs. RF**: model coefficients at leaves can be interpreted

• **Disadvantage**: its algorithm is not easy to understand; however its results are generally quite good.
Refinements to Cubist

- **“Committees”** of models: a sequence of models, where each corrects the errors in the previous one
- **nearest-neighbours adjustment**: modify model result at a prediction point from some number of neighbours in feature (predictor) space.

\[
\hat{y}' = \frac{1}{K} \sum_{i=1}^{K} w_i [t_i + (\hat{y} - \hat{t}_i)]
\]

(1)

where \( t_i \) is the actual value of the neighbour, \( \hat{t}_i \) is its value predicted by the model tree(s), and \( w_i \) is the weight given to this neighbour for the adjustment, based on its distance \( D_i \) from the target point. These are computed as \( w_i = 1/(D_i + 0.5) \) and normalized to sum to one.
Example cubist model

Rule 1/1: [66 cases, mean 2.288309, range 2.053078 to 2.89098, err 0.103603]
   if \( x > 179095 \), dist > 0.211846
   then outcome = 2.406759 - 0.32 dist
Rule 1/2: [9 cases, mean 2.596965, range 2.330414 to 2.832509, err 0.116378]
   if \( x \leq 179095 \), dist > 0.211846
   then outcome = -277.415278 + 0.000847 y + 0.56 dist
Rule 1/3: [80 cases, mean 2.772547, range 2.187521 to 3.264582, err 0.157513]
   if \( \text{dist} \leq 0.211846 \)
   then outcome = 2.632508 - 2.1 dist - 2.4e-05 x + 1.4e-05 y

Rule 2/1: [45 cases, mean 2.418724, range 2.10721 to 2.893762, err 0.182228]
   if \( x \leq 179826 \), ffreq in \{2, 3\}
   then outcome = 128.701732 - 0.000705 x
Rule 2/2: [121 cases, mean 2.443053, range 2.053078 to 3.055378, err 0.181513]
   if \( \text{dist} > 0.0703468 \)
   then outcome = 30.512065 - 0.87 dist - 0.000154 x
Rule 2/3: [55 cases, mean 2.543648, range 2.075547 to 3.055378, err 0.125950]
   if \( \text{dist} > 0.0703468 \), ffreq = 1
   then outcome = 37.730889 - 0.000314 x - 0.35 dist + 6.5e-05 y
Rule 2/4: [34 cases, mean 2.958686, range 2.574031 to 3.264582, err 0.139639]
   if \( \text{dist} \leq 0.0703468 \)
   then outcome = 2.982852 - 0.36 dist
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Map predicted by Cubist

Optimized Cubist prediction
Data-driven methods for predictive modelling

1. Modelling cultures
   Explanation vs. prediction
   Data-driven (algorithmic) methods

2. Classification & Regression Trees (CART)
   Regression trees
   Sensitivity of Regression Trees
   Classification trees

3. Random forests
   Bagging and bootstrapping
   Building a random forest
   Variable importance
   Random forests for categorical variables
   Predictor selection

4. Cubist

5. Model tuning

6. Spatial random forests

7. Data-driven vs. model-driven methods
Data-driven models have parameters that control their behaviour and can significantly affect their predictive power.

- **CART**: complexity parameter
- **randomForest**: number of predictors to try at each split; minimum number of observations in a leaf; number of trees in the forest
  - too many predictors → trees too uniform, loss of diversity; too few → highly-variable trees, poor predictions
  - too few observations per leaf to imprecise prediction; too many → over-fitting
  - too few trees → sub-optimal model; too many trees → wasted computation
- **Cubist**: number of committees; number of nearest neighbours
- The model can be tuned to optimize the selection of these.
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist

Model tuning
Spatial random forests
Data-driven vs. model-driven methods

---

Model tuning – algorithm

1. For each combination of parameters to be optimized:
   1. Split the dataset into some disjunct subsets, for example 10, by random sampling.
   2. For each subset:
      1. Fit the model with the selected parameters on all but one of the subsets (train subset).
      2. Predict at the remaining subset, i.e., the one not used for model building, with the fitted model.
      3. Compute the goodness-of-fit statistics of fitting to the test subset
         e.g., root mean square error (RMSE) of prediction; squared correlation coefficient between the actual and fitted values, i.e., $R^2$ against a 1:1 line.
   3. Average the statistics for the disjunct test subsets.

2. Search the table of results for the best results
   e.g., lowest RMSE, highest $R^2$. 
Model tuning – R implementation

- caret “Classification And REgression Training” package
  - can tune 200+ models; some built-in, some by calling the appropriate package

- method:
  1. set up a vector or matrix with the parameter values to test, e.g.,
     all combinations of 1 …3 splitting variables to try, and 1 …10
     observations per leaf
  2. run the model for all of these and collect the cross-validation
     statistics
  3. select the best one and build a final model
Model tuning example – random forest (1)

```r
> ranger.tune <- train(x = preds, y = response, method = "ranger",
                      tuneGrid = expand.grid(.mtry = 1:3,
                                              .splitrule = "variance",
                                              .min.node.size = 1:10),
                      trControl = trainControl(method = 'cv'))
> print(ranger.tune)

## Resampling: Cross-Validated (10 fold)
## Resampling results across tuning parameters:
##    #    mtry  min.node.size RMSE    Rsquared  MAE
## 1   1     1          199.7651 0.8862826 156.1662
## 1   2     2          200.5215 0.8851154 156.3225
## 1   3     3          200.6421 0.8854146 156.2801
## 2   1     6          201.9809 0.8793349 158.7097
## 2   2     7          202.9065 0.8781754 159.7739
## 2   3     8          202.5687 0.8788200 159.5980
## 3   1     1          199.7651 0.8862826 156.1662
## 3   2     2          200.5215 0.8851154 156.3225
## 3   3     3          200.6421 0.8854146 156.2801
## ...  
## RMSE was used to select the optimal model
## Final values: mtry = 2, min.node.size = 6.
```
Find the minimum RMSE; but favour simpler models (fewer predictors, larger nodes) if not too much difference.
Model tuning example – Cubist (1)

```r
> cubist.tune <- train(x = all.preds, y = all.resp, method = "cubist",
                      tuneGrid = expand.grid(.committees = 1:12,
                                              .neighbors = 0:5),
                      trControl = trainControl(method = 'cv'))

## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 139, 139, 140, 139, 139, 139, ...
## Resampling results across tuning parameters:
##
## committees neighbors  RMSE     Rsquared      MAE
## 1         0       0.1898596 0.6678588 0.1405553
## 1         1       0.1764705 0.6953460 0.1189364
## 1         2       0.1654910 0.7296723 0.1163660
## 1         3       0.1623381 0.7425831 0.1163285
## 1         4       0.1631900 0.7453506 0.1192963
## 2         3       0.1599994 0.7533962 0.1139932
## 2         4       0.1584434 0.7617762 0.1153331
## 2         5       0.1589143 0.7622337 0.1165942

## RMSE was used to select the optimal model using the smallest value.
## The final values: committees = 10, neighbors = 4.
```
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

Model tuning example – Cubist (2)

Criterion: RMSE

Criterion: $R^2$

Adding one neighbour reduces predictive power; adding 2 ...increases it; 3 is close to optimum

Committees improve predictive power; 3 is optimum
Data-driven methods for predictive modelling

1. Modelling cultures
   - Explanation vs. prediction
   - Data-driven (algorithmic) methods

2. Classification & Regression Trees (CART)
   - Regression trees
   - Sensitivity of Regression Trees
   - Classification trees

3. Random forests
   - Bagging and bootstrapping
   - Building a random forest
   - Variable importance
   - Random forests for categorical variables
   - Predictor selection

4. Cubist

5. Model tuning

6. Spatial random forests

7. Data-driven vs. model-driven methods
• Random forests can use *coordinates* and *distances* to geographic features as predictors
  • e.g., E, N, distance to river, distance to a single point …
• Can also use distances to *multiple points* as predictors
  • Distance *buffers*: distance to closest point with some range of values
  • Common approach: compute *quantiles* of the response variable and one buffer for each
  • Each sample point has a distance to the closest point in each quantile
• This uses *separation between point-pairs* of different values, but with *no* model.
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests

Data-driven vs. model-driven methods
Regression tree on 16 distance buffers

Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods
Random forest prediction on 16 distance buffers

Zn, log(mg kg⁻¹)

Random forest fit

Actual value

Random forest prediction on 16 distance buffers

Zn, log(mg kg⁻¹)

Out-of-bag prediction

Actual value
Data-driven methods for predictive modelling

DGR/Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods

OOB error vs. OK cross-validation error

Zn, log(mg kg$^{-1}$)

OK

Note that RF does not use any model of spatial autocorrelation!
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning
Spatial random forests

Data-driven vs. model-driven methods

Random forest map on 16 distance buffers

Resembles OK map, but *no model* was used.
Data-driven methods for predictive modelling

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods

Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees

Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection

Cubist
Model tuning

Spatial random forests
Data-driven vs. model-driven methods

Compare with Ordinary Kriging

Zn, log(mg kg⁻¹)

Ordinary Kriging

Random forest on distance buffers
Data-driven methods for predictive modelling

Modelling cultures
- Explanation vs. prediction

Data-driven (algorithmic) methods
- Classification & Regression Trees (CART)
  - Regression trees
  - Sensitivity of Regression Trees
  - Classification trees
- Random forests
  - Bagging and bootstrapping
  - Building a random forest
  - Variable importance
  - Random forests for categorical variables
  - Predictor selection
- Cubist
- Model tuning

Spatial random forests
- Data-driven vs. model-driven methods

Spatial RF - OK predictions, log(Zn)
1. Modelling cultures
   Explanation vs. prediction
   Data-driven (algorithmic) methods

2. Classification & Regression Trees (CART)
   Regression trees
   Sensitivity of Regression Trees
   Classification trees

3. Random forests
   Bagging and bootstrapping
   Building a random forest
   Variable importance
   Random forests for categorical variables
   Predictor selection

4. Cubist

5. Model tuning

6. Spatial random forests

7. Data-driven vs. model-driven methods
Conclusion: Data-driven vs. model-based methods

- **Data-driven**: main aim is **predictive power**
  - Individual trees can be interpreted (CART), but ensemble methods (random forests, Cubist ...) can not
    - can see variable importance overall and influence on each prediction

- **Model-based**: main aim is **understanding processes**
  - We hope the model is a simplified representation of the process that produced the observations
  - If the model is correct, predictions will be accurate
Conclusion: limitations

- **Data-driven** methods depend on their **training observations**
  - They have no way to extrapolate or even interpolate to unobserved areas in feature space
  - So the observations should cover the entire range of the population

- **Model-based** methods depend on a correct **empirical-statistical model**
  - Model is derived from training observations, but many models are possible
  - Various model-selection techniques
  - Wrong model → poor predictions, incorrect understanding of processes
Data-driven methods for predictive modelling

DGR/罗大维

Modelling cultures
Explanation vs. prediction
Data-driven (algorithmic) methods
Classification & Regression Trees (CART)
Regression trees
Sensitivity of Regression Trees
Classification trees
Random forests
Bagging and bootstrapping
Building a random forest
Variable importance
Random forests for categorical variables
Predictor selection
Cubist
Model tuning
Spatial random forests
Data-driven vs. model-driven methods