Random Forests

M. Gruber

KW 47, Rev.1

In [Bre01b] ist die Rede von zwei grundverschiedenen Ansätzen des statistischen Modellierens, nämlich von *"The Data Modeling Culture"* versus *"The Algorithmic Modeling Culture"*. *Random Forests* geht auf Leo Breiman zurück und fällt in die zweite Kategorie. Der Algorithmus wurde in [Bre01a] erstmals vorgestellt. In [Bre] finden Entwickler und Anwender Hinweise für die praktische Anwendung. Im akademischen Bereich wird die Methode *Random Forests* heute vor allem von Adele Cutler vertreten (siehe [CCS]). Es gibt eine R-Version, die die Entwickler in [LW02] kommentieren.

Unsere Einführung in die Methode *Random Forests* beschränkt sich der Einfachheit halber auf den Klassifizierungsfall. Der Regressionsfall unterscheidet sich aber nicht wesentlich. Wir setzen die R-Implementierung *randomForest* ein. Unsere Beispieldaten sind die Irisdaten [Fis36].

Random Forests ist eine Ensemblemethode. Auf der Lernmenge werden viele — ein Wald *("forest")* — Klassifikations- bzw. Regressionsbäumen von nicht allzugroßer Tiefe entwickelt. Das Kollektiv dieser Bäume stellt den resultierenden Prädiktor dar. Der Ausgabewert des Prädiktors ist der Wert, für den die Mehrheit der Klassifizierungsbäume votiert. Dieses demokratische Konzept bewährt sich erstaunlich gut.

Betrachten wir einen einzelnen Klassifizierungsbaum näher. Die R-Version *randomForest* gestattet es, den Parameter *ntree* auf den Wert 1 zu setzen. Im Anwendungsfall wäre dieser Wert untypisch, denn gewöhnlich arbeitet die Methode mit Hunderten von Bäumen. Mit $ntree = 1$ erhalten wir einen einzigen Klassifizierungsbaum wie etwa den in Abb. 1. Er ist auf einem *Bootstrap sample* (Ziehen mit Zurücklegen) vom Umfang $N = 150$ entstanden. *Bootstrap samples* lassen etwa ein Drittel der Daten ungenutzt (wegen $(1 - 1/N)^N \to 1/e$). Die ungenutzten Daten werden zur Validierung des Baumes herangezogen. Der festgestellte Validierungsfehler des Baumes ist sein *out-of-bag error*. Er berägt in unserem Beispiel 3.4% (vgl. Abb. 2).

Bei der Entwicklung des Baumes wird das *Bootstrap sample* rekursiv binär geteilt. Bei jeder Teilung wird eine bestimmte Zahl (z.B. $\sqrt{d}$, wenn d die die Dimension der Inputmenge ist) zufällig gezogener Attribute auf ihre Eignung zum Teilen der vorliegenden Menge geprüft. Geteilt wird eine Menge $\mathcal{S}$ — wenn k die Nusmmer des zur Teilung herangezogenen Attributs ist — in eine "linke" Teilmenge $\mathcal{S}_L = \{x \in \mathcal{S} \mid x_k < \eta\}$ und eine "rechte" Teilmenge $\mathcal{S}_R = \{x \in \mathcal{S} \mid x_k \geq \eta\}$.

Welches der zufällig gezogenen Attribute zur Teilung verwendet wird, hängt von der Güte der

Teilung ab, zu der es fähig ist. Die Güte einer Teilung wird mit der Reinheit der entstehenden Teilmengen gemessen. Im Irisbeispiel gibt es die Klassen $i = 1, 2, 3$ (für Iris-setosa, Iris-versicolor und Iris-virginica). Eine Teilung ist umso besser, je kleiner der Giniwert

$$n_L \sum_{1 \leq i \leq 3} p_{iL}(1 - p_{iL}) + n_R \sum_{1 \leq i \leq 3} p_{iR}(1 - p_{iR}) \tag{1}$$

ist. Dabei ist p_{iL} die relative Häufigkeit der Klasse i in der linken bzw. p_{iR} deren relative Häufigkeit in der rechten Teilmenge. Die Zahlen n_L und n_R sind die Mächtigkeiten der linken bzw. rechten Teilmenge. Für jedes Attribut wird zur Berechnung des Gini-Wertes der beste Schwellenwert η genommen. Von den als Kandidaten ausgelosten Attributen gewinnt das mit dem kleinsten Giniwert. Siegerattribute und Schwellenwerte sind in Abb.1 zu sehen.

Die Gesamtheit der Bäume stellt den *random forest* dar.

Ein interessantes Nebenprodukt des Erzeugungsprozesses ist die *variable importance*, angegeben in der Spalte `MeanDecreaseGini` in Abb.3. Sei V die betrachtete Variable. Zur Berechnung der Wichtigkeit von V werden alle Teilungen ausgewertet, zu denen V herangezogen wurde. Die jeweils realisierten Giniwertdifferenzen werden aufaddiert und die Summe durch die Anzahl der Bäume dividiert.

Ein anderes Wichtigkeitsmaß ist die `MeanDecreaseAccuracy`. Man hat bereits den OOB-Klassifizierungsfehler (OOB steht für *out-of-bag*). Permutiert man in der OOB-Menge die Wertebelegung für das Attribut V, erhält man einen OOB-Klassifizierungsfehler einer fiktiven Lernmenge. In der Regel ist er größer als der der echten Lernmenge. Die Zunahme ist die `MeanDecreaseAccuracy`. Sie wird klassenweise oder klassenübergreifend angegeben.

Literatur

[Bre] Leo Breiman. Manual on setting up, using, and understanding Random Forests V4.0.

[Bre01a] Leo Breiman. Random Forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[Bre01b] Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 2001.

[CCS] Adele Cutler, David Richard Cutler, and John R Stevens. Random Forests. In *Ensemble machine learning*, pages 157–175. Springer US.

[Fis36] Ronald Aylmer Fisher. Iris Data Set, 1936.

[LW02] Andy Liaw and Matthew Wiener. Classification and Regression by randomForest. *R News*, 2(3):18–22, 2002.

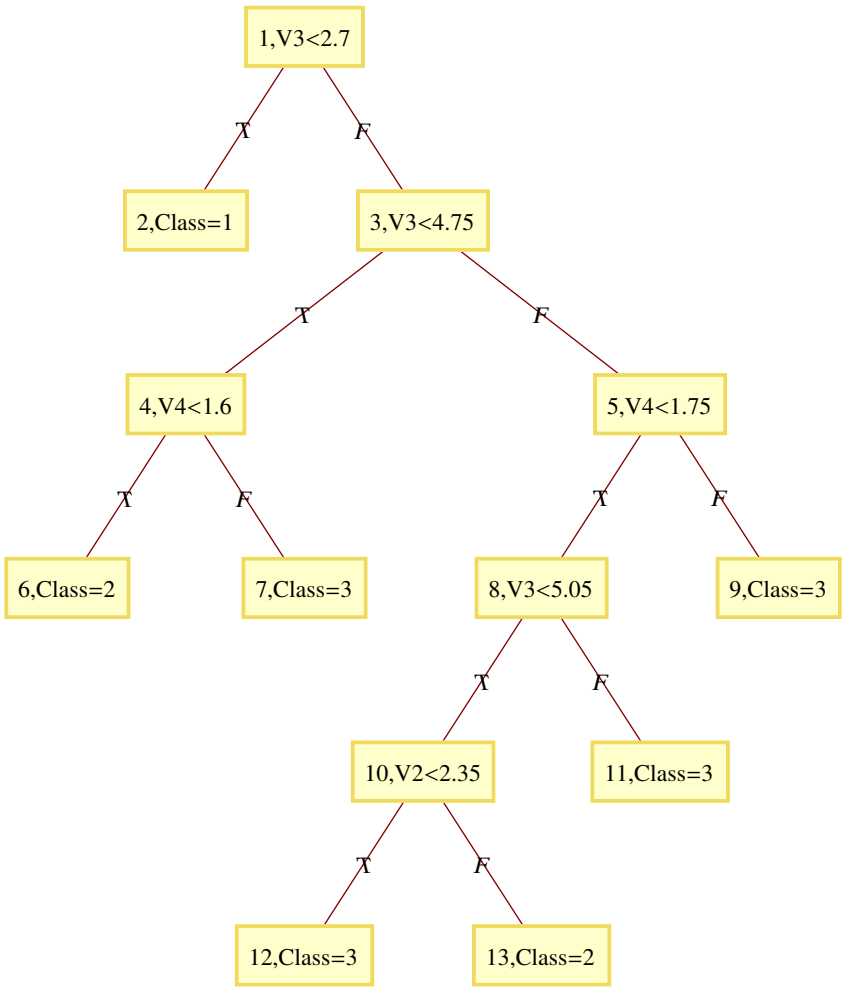

Abbildung 1: Ein typischer Klassifizierungsbaum in *Random Forests*.

```
$call
randomForest(formula = formula.rf, data = iris.local, importance = TRUE,
    ntree = 1)

$type
[1] "classification"

$predicted
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20
<NA> <NA>    1    1 <NA> <NA>    1    1    1 <NA> <NA>    1    1    1 <NA> <NA> <NA> <NA> <NA> <NA>
  21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37   38   39   40
   1 <NA>    1 <NA>    1 <NA>    1    1 <NA>    1 <NA> <NA>    1 <NA>    1    1 <NA> <NA> <NA>    1
  41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56   57   58   59   60
<NA> <NA> <NA>    1    1 <NA> <NA>    1    1    1    2 <NA> <NA> <NA>    2 <NA>    2 <NA>    2 <NA>
  61   62   63   64   65   66   67   68   69   70   71   72   73   74   75   76   77   78   79   80
   2    2 <NA> <NA> <NA>    2 <NA>    2 <NA>    2 <NA>    2 <NA> <NA>    2 <NA> <NA>    2    2 <NA>
  81   82   83   84   85   86   87   88   89   90   91   92   93   94   95   96   97   98   99  100
<NA>    2 <NA> <NA> <NA> <NA> <NA>    2 <NA>    2    2 <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
 101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120
<NA>    3 <NA> <NA> <NA> <NA>    2 <NA>    3 <NA> <NA>    3    3 <NA> <NA> <NA>    3 <NA>    3    2
 121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140
   3 <NA> <NA> <NA>    3 <NA> <NA>    3 <NA> <NA> <NA> <NA>    3 <NA> <NA> <NA> <NA>    3 <NA> <NA>
 141  142  143  144  145  146  147  148  149  150
   3 <NA> <NA>    3    3 <NA> <NA> <NA>    3    3
Levels: 1 2 3

$err.rate
            OOB 1 2         3
[1,] 0.03448276 0 0 0.1111111

$confusion
   1  2  3 class.error
1 23  0  0   0.0000000
2  0 17  0   0.0000000
3  0  2 16   0.1111111
```

Abbildung 2: R-Ausgabe nach Erzeugung eines Klassifizierungsbaumes.

```
           1         2         3 MeanDecreaseAccuracy MeanDecreaseGini
V1 0.0000000 0.0000000 0.0000000           0.00000000        1.3333333
V2 0.0000000 0.0000000 0.0000000           0.00000000        0.4166667
V3 0.0000000 0.1176471 0.0000000           0.03448276       12.0628205
V4 0.6521739 0.7058824 0.2777778           0.55172414       82.3338462
```

Abbildung 3: R-Ausgabe zur *Variable importance*.