Abstract

The successful tree-based methodology has one serious disadvantage: lack of stability. That is, regression tree model depends on the training set and even small change in a predictor value could lead to a quite different model. In order to solve this problem single trees are combined into one model. There are three aggregation methods used in classification: bootstrap aggregation (bagging), adaptive resample and combine (boosting) and adaptive bagging (hybrid bagging-boosting procedure).

In the field of regression a variant of boosting, i.e. gradient boosting, can be used. Friedman (1999) proved that boosting is equivalent to a stepwise function approximation in which in each step a regression tree models residuals from last step model.

Key words: tree-based models, regression, boosting.

1. INTRODUCTION

The goal of a regression is to find a function $F^*(x)$ that maps $x$ to $y$:

$$ F^*(x) : x \rightarrow y, $$

and minimises the expected value of a specified loss function $L(y, F(x))$ over the joint distribution of all $(x, y)$ values:

$$ F^*(x) = \arg \min_{F(x)} E_y \sum_x L(y, F(x)), $$

given a sample (called “training set”):

* Professor, Institute of Statistics, University of Economics in Katowice.
The most frequently used loss function for measuring errors between \( y \) and \( F(x) \) is the squared error:

\[
L(y, F(x)) = (y - F(x))^2.
\]

In this paper we consider \( F(x) \) having an additive form:

\[
F(x) = \sum_{m=0}^{M} \beta_m f_m(x, a_m).
\]

where \( f_m(x, a) \) is a simple function of \( x \) with parameters \( a \) (called "base learner"), for example the linear function:

\[
f_m(x, a_m) = \sum_{l=1}^{L} a_m x_l.
\]

When the base learner (6) is a tree, the parameters \( a \) are the splitting variables, split locations and mean values of \( y \) in regions \( R_k \).

II. REGRESSION TREES

The tree corresponds to an additive model in the form of:

\[
f(x, a) = \sum_{k=1}^{K} a_k I(x \in R_k),
\]

where \( R_k \) are hyper-rectangular disjoint regions in the \( M \)-dimensional feature space, \( a_k \) denotes real parameters and \( I \) is an indicator function (Gatnar, 2001).

Each real-valued dimension of the region \( R_k \) is characterised by its upper and lower boundary: \( v_{km}^{(d)} \) i \( v_{km}^{(g)} \) respectively. Therefore the region induces a product of \( M \) indicator functions:

\[
I(x \in R_k) = \prod_{m=1}^{M} I(v_{km}^{(d)} \leq x_m \leq v_{km}^{(g)}),
\]

If \( x_m \) is a categorical variable, the region \( R_k \) is defined as:
\[ I(x \in R_k) = \prod_{m=1}^{M} I(x_m \in B_{km}), \] (9)

where \( B_m \) is a subset of the set of the variable values.

The parameter estimation formula depends on the way how the homogeneity of the region \( R_k \) is measured. In the simplest case, when variance is used, the best estimate is the mean of all \( y \) values in \( R_k \):

\[ a_k = \frac{1}{N(k)} \sum_{i=1}^{N(k)} y_i, \] (10)

where \( N(k) \) is the number of objects from training set belong to region \( R_k \).

Tree-based regression models are represented by step functions (Fig. 1).

\[ \begin{align*}
\text{Figure 1. Example of a step function} \\
\end{align*} \]

Because their lack of smoothness could be sometimes a disadvantage, Friedman (1991) proposed to use splines (in the MARS procedure) to solve this problem.

III. BOOSTING

The successful tree-based methodology has one undesirable feature: lack of stability. That is a regression tree model depends on the training set and even small change in a predictor value could lead to a quite different model.

To solve this problem in the field of classification single trees are combined into one model and then averaged. There are three aggregation methods developed so far:
1) bootstrap aggregation (bagging), developed by Breiman (1996),
2) adaptive resample and combine (boosting), proposed by Freund and
Shapire (1996),
3) adaptive bagging, proposed by Breiman (1999).

Boosting is seen as the most successful and powerful idea in statistical
learning (Hastie et al., 2001). It was developed by Freund and Shapire
(1996) originally for classification problems, to produce the most accurate
model as a committee of many "weak" classifiers.

Given a set of training data \( \mathcal{D} \) and classifier \( f_m(x, a) \) producing values
from the set \( \{-1, +1\} \), the algorithm AdaBoost trains the classifier on
modified training sample, giving higher weights to cases that are currently
misclassified. This repeats for a sequence of weighted samples and the result
is a linear combination of the classifiers from each stage.

The algorithm works as follows:
1. Start with equal weights for each case:
   \[ w_i = \frac{1}{N}, \quad \forall i = 1, \ldots, N \]  
   \[ (11) \]

2. Repeat for \( m = 1 \) to \( M \):
   a) fit the classifier: \( f_m(x, a) \) to the training data using weights \( w_i \),
   b) compute the classification error:
   \[ e_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq f_m(x_i, a))}{\sum_{i=1}^{N} w_i}, \]
   \[ (12) \]
   c) compute the classifier weight:
   \[ \beta_m = \log \left( \frac{1 - e_m}{e_m} \right), \]
   \[ (13) \]
   d) set weights for cases:
   \[ w_i \leftarrow w_i \cdot e^\beta_m I(y_i \neq f_m(x_i, a)), \]
   \[ (14) \]

3. Final classifier is:
   \[ F(x) = \text{sgn} \left( \sum_{m=0}^{M} \beta_m f_m(x, a) \right), \]
   \[ (15) \]

In the step 2d) cases misclassified by \( f_m(x, a) \) have their weights increased
and then they form the classifier \( f_{m+1}(x, a) \).
IV. GRADIENT BOOSTING

Friedman (1999) developed a variant of boosting, i.e. "gradient boosting" of trees which produces highly robust models, especially appropriate for imperfect data. He proved that boosting is equivalent to forward stepwise modelling, that is sequentially adding new functions to the expansion:

\[ F(x) = f_0(x_i) + \beta_1 f_1(x_i) + \beta_2 f_2(x_i) + \ldots \]  \hspace{1cm} (16)

Using steepest-descent method from numerical minimisation, the negative gradient:

\[ g_m(x) = \left[ \frac{\partial E_y(L(y, F(x))|x)}{\partial F(x)} \right]_{F(x) = F_{m-1}(x)} \]  \hspace{1cm} (17)

define the "steepest-descent" direction:

\[ f_m(x) = -\lambda_m g_m(x) \]  \hspace{1cm} (18)

and:

\[ F_{m-1}(x) = \sum_{i=0}^{m-1} f_i(x). \]  \hspace{1cm} (19)

The weights \( \lambda_m \) in (18) are estimated as:

\[ \lambda_m = \arg\min_{\lambda} E_{y, x} L(y, F_{m-1}(x) + \lambda \cdot f_m(x)), \]  \hspace{1cm} (20)

and the approximation updated:

\[ F_m(x) = F_{m-1}(x) + \lambda_m \cdot f_m(x). \]  \hspace{1cm} (21)

For squared error loss function (4), or its minor modification:

\[ L(y, F(x)) = \frac{1}{2} (y - F(x))^2, \]  \hspace{1cm} (22)

the negative gradient is just the residual:

\[ -g_m = y_i - F_{m-1}(x). \]  \hspace{1cm} (23)
The gradient boosting is a stepwise function approximation in which each step models residuals from last step model. If the base learner \( f_m(x, a) \) is a regression tree \((7)\), then the boosted tree is induced according to the procedure:

1. Initialise:

\[
F_0(x) = \bar{y}.
\]  

(24)

2. For \( m = 1 \) to \( M \):
   a) repeat for each \( i = 1, ..., N \):

\[
u_i = y_i - F_{m-1}(x).
\]  

(25)

b) grow regression tree for the residuals \( u_i \) finding homogeneous regions \( R_{jm} \),

c) for \( j = 1, ..., J_m \) compute:

\[
a_{jm} = \arg \min_a \sum_{x_j \in R_{jm}} (y_i - (F_{m-1}(x_i) + a))^2.
\]  

(26)

d) modify:

\[
F_m(x) = F_{m-1}(x) + \sum_{j=1}^{J_m} a_{jm} I(x \in R_{jm}).
\]  

(27)

3. The final model:

\[
F^*(x) = F_M(x).
\]  

(28)

The tree is grown to group observations into homogeneous subsets. Once we have the subsets our update quantities for each subset are computed in a separate step.

V. EXAMPLE

Consider Boston Housing data set (Harrison and Rubinfeld, 1978). The data consisted of 14 variables measured for each of 506 census tracts in the Boston area. The dependent variable is \( MV \) – median of neighborhood home value and independent variables are: \( CRIM \) – crime rate,
RM – average number of rooms, LSTAT – percent lower-status population, etc.

Average value of MV is $22.533. We start model $F_0(x)$ with the mean (24) and construct residuals. The residuals are computed with two-node tree and the tree separates positive from negative residuals.

Then we update the model, obtain new residuals and repeat the process (e.g. twice). Estimated function consists of three parts and is shown in Figure 2.

\[
\begin{align*}
    MV &= 22.5 + RM < 6.8 \\
    &\quad \begin{cases}
        \text{yes} & \text{LSTAT < 14.3} \\
        \text{no} & \text{LSTAT < 14.3}
    \end{cases} \\
    &\quad \begin{cases}
        \text{yes} & +0.4 \\
        \text{no} & -8.4
    \end{cases} \\
    &\quad \begin{cases}
        \text{yes} & +13.7 \\
        \text{no} & -5.2
    \end{cases} \\
    &\quad \begin{cases}
        \text{yes} & +8.4 \\
        \text{no} & -0.3
    \end{cases} \\
    &\quad \begin{cases}
        \text{yes} & CRIM < 8.2 \\
        \text{no} & CRIM < 8.2
    \end{cases} \\
    &\quad \begin{cases}
        \text{yes} & +3.2 \\
        \text{no} & -4.4
    \end{cases}
\end{align*}
\]

Figure 2. Boosting tree for Boston data

As we can see (Fig. 2) only three independent variables were selected to the model$^1$: RM, LSTAT and CRIM.

The resulted boosting tree is better model, in terms of goodness-of-fit, than other regression models. In Table 1 we present a comparison of the value of $R^2$ for two data sets: Boston Housing and California Housing$^2$.

---

$^1$ We used the MART system implemented in the S-Plus environment.

$^2$ The California Housing data set is available from Statlib repository. It was analysed by Pace and Barry (1997) and consists of data from 20460 neighborhoods (1990 census block groups) in California.
Table 1. Comparison of R2 for two data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>Single regression tree</th>
<th>Gradient boosting tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston Housing</td>
<td>0.67</td>
<td>0.84</td>
</tr>
<tr>
<td>California Housing</td>
<td>0.70</td>
<td>0.86</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

There are several advantages of using the method of gradient boosting in nonparametric regression. Boosting regression trees can cope with outliers, are invariant to monotone transformations of variables, and can handle missing values. They also automatically select variables to the model and perform regression very fast.

The regression model obtained in the form of a boosting tree is also extremely easy to interpret and to use for prediction.

REFERENCES

Eugeniusz Gatnar

GRADIENTOWA ODMIANA METODY \textit{BOOSTING} W ANALIZIE REGRESJI

Streszczenie

Szeroko stosowane w praktyce metody nieparametryczne wykorzystujące tzw. drzewa regresyjne mają jedną istotną wadę. Otoż wykazują one niestabilność, która oznacza, że niewielka zmiana wartości cech obiektów w zbiorze uczącym może prowadzić do powstania zupełnie innego modelu. Oczywiście wpływa to negatywnie na ich trafność prognoistyczną. Tę wadę można jednak wyeliminować, dokonując agregacji kilku indywidualnych modeli w jeden.

Znane są trzy metody agregacji modeli i wszystkie opierają się na losowaniu ze zwracaniem obiektów ze zbioru uczącego do kolejnych prób uczących: agregacja bootstrapowa (\textit{boosting}), losowanie adaptacyjne (\textit{bagging}) oraz metoda hybrydowa, łącząca elementy obu poprzednich.

W analizie regresji szczególnie warto zastosować gradientową, sekwencyjną, odmianę metody boosting. W istocie polega ona wykorzystaniu drzew regresyjnych w kolejnych krokach do modelowania reszt dla modelu uzyskanego w poprzednim kroku.