Boosting, Bagging and Random Forests

Alan Lee

Department of Statistics
Course STATS 760 Lecture 5

April 29, 2013
Outline

Boosting

Bagging

Random Forests
Boosting

The basic idea here is

1. Fit a model to the data. Then repeat
   1.1 Modify the data in some way
   1.2 Fit a model to the modified data

2. Average the results
Application to regression

Forward stagewise modelling: (HTF p 341)

Many regression models fit an additive combination of “basis functions” i.e. express $f$ as

$$f(x) = \sum_{m=1}^{M} \beta_m b(x, \gamma_m)$$

where the $\beta_m$ are regression coefficients and $b(x, \gamma)$ is a “basis function” which depends on a parameter $\gamma$. Examples:

- Linear: $b(x, \gamma) = x_j$
- Trees: $b(x, \gamma) = I(x \in R_m)$
- Neural nets: $b(x, \gamma) = \sigma(\gamma_0 + \gamma^T x)$
- Mars: $b(x, \gamma) = (x - \gamma)_+$ or $(\gamma - x)_+$
FSM Algorithm

Let $L(y, \eta)$ be a loss function, e.g. $L(y, \eta) = (y - \eta)^2$.

1. Set $f_0(x) = 0$.

2. For $m=1,2\ldots,M$
   2.1 Compute
   \[
   (\hat{\beta}, \hat{\gamma}) = \arg\min_{\beta, \gamma} \sum_{i=1}^{n} L(y_i, f_{m-1}(x_i) + \beta b(x, \gamma))
   \]
   2.2 Set $f_m(x) = f_{m-1}(x) + \hat{\beta} b(x, \hat{\gamma})$. 
Squared error loss, trees

If \( L(y, \eta) = (y - \eta)^2 \), then at each stage we are fitting a simple model involving one \( \beta \) and \( \gamma \) to the residuals from the previous fit. Note that to predict at \( x \), we add up the predictions for each model.

Modification for trees: We replace 2.1 by

\[
\arg\min_{\beta, \theta} \sum_{i=1}^{n} L(y_i, f_{m-1}(x_i) + T(x_i, \theta))
\]

where \( T \) is a tree which can be represented by a sum of basis functions:

\[
T(x, \theta) = \sum_{j=1}^{J} \beta_j I(x \in R_j)
\]
Points to note

- Trees are usually small (2-5 terminal nodes)
- M is large, say 00's. We are averaging many small noisy trees.
- Implemented in R by the mboost package - see documentation on the web page
- We need to choose M (number of boosting steps) and J (tree size, number of terminal nodes)
Example

We use the bodyfat data set in the mboost package. This consists of data on 71 healthy female subjects, containing body fat measurements (variable DEXfat) and nine anthropometric measurements. The idea is to develop a predictive model for body fat.

We will standardise the data and fit a boosted tree model, using 200 boosting iterations and trees with a maximum of 4 terminal nodes. The code is on the next slide.
Example

data("bodyfat", package = "mboost")
bodyfat = standardise(bodyfat)  # see model answer, Ass2
# we have to specify trees in a model formula
fm = DEXfat ~ btree(hipcirc, age, elbowbreadth, kneebreadth,, anthro3a, anthro3b, anthro3c, anthro4, tree_controls = ctree_control(maxdepth = 5))
# Now fit model
myfit = mboost(fm, data = bodyfat,
control = boost_control(mstop = 200, nu = 0.1))
Error plots

```r
par(mfrow=c(1,1))
my.risk = cvrisk(myfit, grid = 1:20*10)
plot(my.risk)
```
Partial dependence plots

plot(myfit)
Bagging

*Bagging* stands for *Bootstrap Aggregation*. The idea is: to make a prediction at \( x \), we

- Draw \( B \) bootstrap samples
- Fit a model \( f \) to each sample
- Average the predictions from each model

Often done with trees, with a small tweak in the above. This results in Random Forests. Random Forests were invented by Leo Breiman, a Berkeley professor, and further developed by Adele Cutler (an Auckland graduate)
Random Forests

We apply the algorithm above to trees, modifying it as follows. To predict the response at $x$:

- Draw $B$ bootstrap samples.
- For each sample, fit a tree. At each split, select the splits using only a randomly chosen subset of $m$ variables, rather than choosing from all of them.
- Average the predictions $f(x)$ using only the OOB samples (OOB = “out-of-bag” samples, those not containing $x$.) Overfitting does not arise since we are using OOB samples only.
Assessing variable importance

We would like to know which variables are important in the predictions. There are two methods for calculating an “importance measure” for each variable \( j \). For a single tree

- **Method 1:** For each non-terminal node, record (a) which variable we are splitting on, and (b) the decrease in RSS associated with the split. Then for each variable, sum up the RSS decreases for the nodes that were split on that variable. (see next slide for method 2)
Assessing variable importance (cont)

- Method 2: In every tree grown in the forest, calculate the predictions for the OOB cases (those not included in the bootstrap sample.) Now randomly permute the values of variable $j$ in the OOB cases and work out predictions for these. Subtract the prediction error of the unpermuted variable from that of the permuted variable. This should be large for important variables.

In both cases, we average over all trees in the forest to get an importance score for variable $j$. 
Partial dependence plots

How can we explore the relationship between individual variables (or subsets of variables) and the predictor? One way is to consider the average value of $f$ for a fixed value $x_j$ of a particular variable (say $X_j$), averaged over the other variables. For example, suppose we want to understand the relationship between $X_1$ and $f$. For a fixed value $x_1$ of $X_1$, we can calculate

$$\frac{1}{n} \sum_{i=1}^{n} \hat{f}(x_1, x_i, 2, \ldots, x_{ik}).$$

We can repeat this for different values of $x$ and plot the result. This is called a Partial Dependence Plot.
## Example: bodyfat

### Code for fitting random forest:

```r
library(randomForest)
myfit2=randomForest(DEXfat ~ ., data=bodyfat, ntree=200, mtry=5, maxnodes=5, importance=TRUE)
importance(myfit2)
```

### Table: Variable Importance

<table>
<thead>
<tr>
<th>Variable</th>
<th>%IncMSE</th>
<th>IncNodePurity</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>-0.4079780</td>
<td>12.16666</td>
</tr>
<tr>
<td>waistcirc</td>
<td>11.8841105</td>
<td>2561.47017</td>
</tr>
<tr>
<td>hipcirc</td>
<td>11.0321544</td>
<td>2173.09379</td>
</tr>
<tr>
<td>elbowbreadth</td>
<td>-0.0303593</td>
<td>10.27547</td>
</tr>
<tr>
<td>kneebreadth</td>
<td>3.8233991</td>
<td>321.13977</td>
</tr>
<tr>
<td>anthro3a</td>
<td>4.7300259</td>
<td>581.31088</td>
</tr>
<tr>
<td>anthro3b</td>
<td>4.4563992</td>
<td>449.90686</td>
</tr>
<tr>
<td>anthro3c</td>
<td>7.4780828</td>
<td>916.24997</td>
</tr>
<tr>
<td>anthro4</td>
<td>4.3821385</td>
<td>439.23765</td>
</tr>
</tbody>
</table>
Error Plots

plot(myfit2)
Importance Plots

```r
varImpPlot(myfit4)
```

![Importance Plots](image)
Dependence Plots

partialPlot(myfit2, x.var = "waistcirc", pred.data=bodyfat)
## California Housing data

<table>
<thead>
<tr>
<th>Method</th>
<th>Out-of-sample error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>0.356</td>
</tr>
<tr>
<td>MARS $nk=40$</td>
<td>0.263</td>
</tr>
<tr>
<td>NNet4</td>
<td>0.249</td>
</tr>
<tr>
<td>PPR6</td>
<td>0.235</td>
</tr>
<tr>
<td>Boosting</td>
<td>0.222</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.177</td>
</tr>
</tbody>
</table>
References

2. Bagging: HTF Chapter 8 (section 8.7).
3. Random Forests HTF Chapter 15, see also http://www.stat.berkeley.edu/~breiman/RandomForests/