Boosting

Myra O'Regan

November 22, 2017

Boosting - Adaboost

- For weak classifiers slightly better than random
- Two very different ways of looking at this
- N observations in the training sets
- Sample with replacement N observations
- Look to see which obs are misclassified
- Weight the cases which are misclassified more
- Sample again
- Repeat above steps
- Weighted Combination at the end

Boosting - Adaboost Original form

- y is coded as -1 and $+1$
- Initial compute initial weights $= w_i^{(0)} = 1/N$
- Let M $=$ no. of trees
- For m$=$1 to M $\{$
- Fit a classifier $T_m(x)$ to training data with $w_i^{(m)}$
- Compute $err_m = \frac{\sum_{i=1}^{N} w_i^{(m)} I(y_i \neq T_m(x))}{\sum w_i^{(m)}}$
- Compute $\alpha_m = ln((1 - err_m)/err_m)$
- Update weight $w_i^{m+1} = w_i^m exp[\alpha_m . I(y_i \neq T_m(x)]$
- In some versions weights are normalised to sum to 1
- Output sign$(\sum_{m=1}^{M} \alpha_m . T_m(x))$

Some points about Boosting

- Compute $\alpha_m = ln((1 - err_m)/err_m)$
- A classifier with 50% error is given zero weight
- A classifier with error rate $< 50\%$ given a negative weight

Illustration of boosting

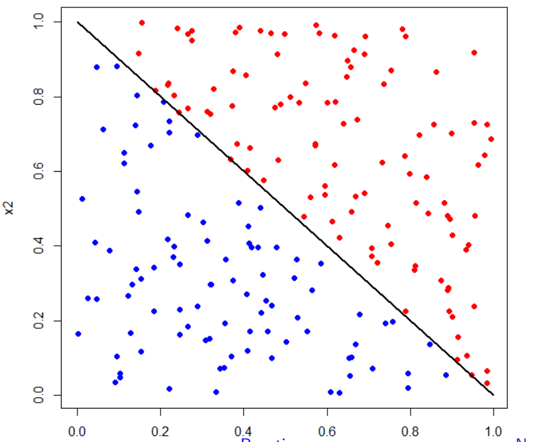
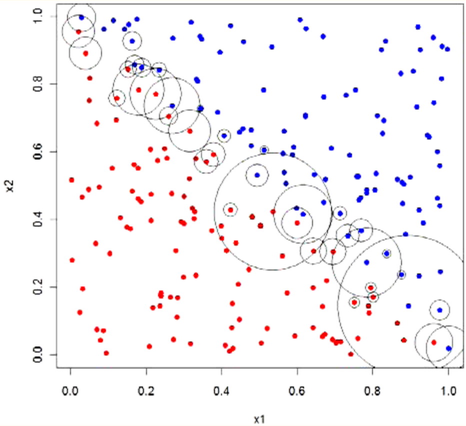


Illustration of boosting



Weights after a few iterations

Generic algorithm for Ensembles

Step 1- Choose $\{\mathbf{p_m}\}$
$F_0(\mathbf{x}) = \mathbf{0}$
For m=1 to M {

$$\mathbf{p_m} = \mathbf{argmin_p} \sum_{\mathbf{i} \in \mathbf{S_{n(\eta)}}} \mathbf{L}(\mathbf{y_i}, \mathbf{F_{m-1}}(\mathbf{x_i}) + \mathbf{T}(\mathbf{x_i}; \mathbf{p}))$$

$$T_m(\mathbf{x}) = \mathbf{T}(\mathbf{x_i}; \mathbf{p})$$

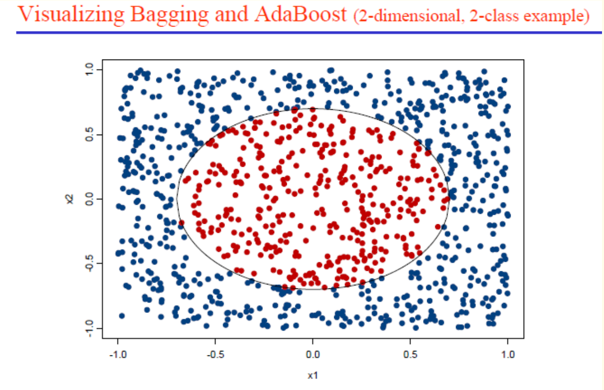
$$F_m(\mathbf{x}) = \mathbf{F_{m-1}}(\mathbf{x}) + \nu \mathbf{T_m}(\mathbf{x})$$

}
write $\{T_m(x)\}_1^M$

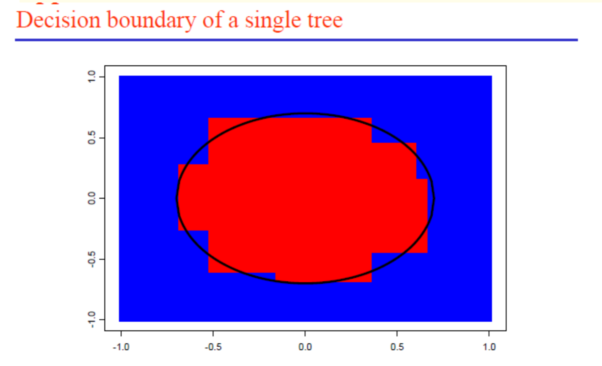
Boosting - Adaboost

- Second way
- y is coded as +1 and -1
- $L(y, \hat{y}) = exp(y.\hat{y})^2$
- $\upsilon = 1$
- $\eta = N$
- $T_m(x)$: any weak learner
- $c_0 = 0, \{c_m\}_1^M$; sequential partial regression coefficients
- Use **gbm** package in R

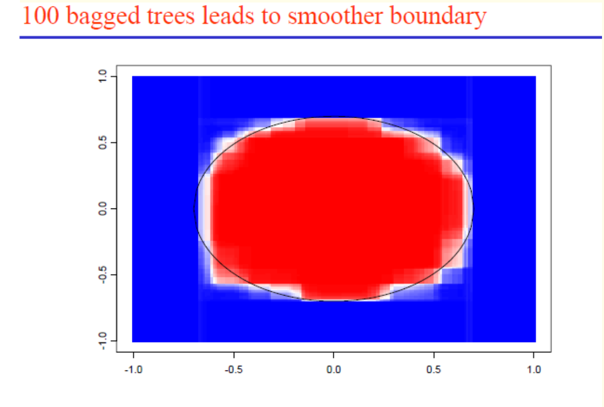
Illustration of boosting, bagging and single trees



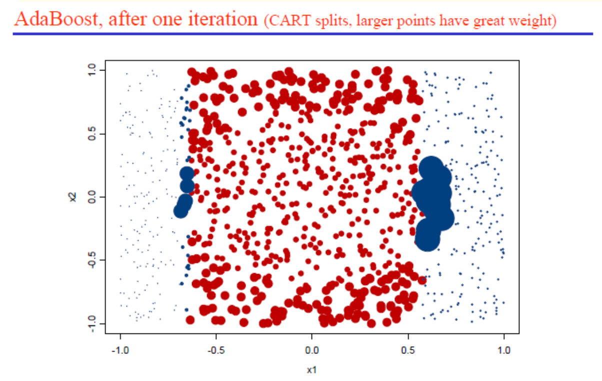
Single tree



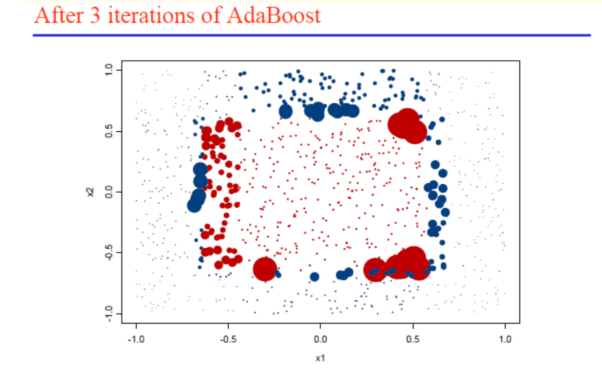
100 bagged trees



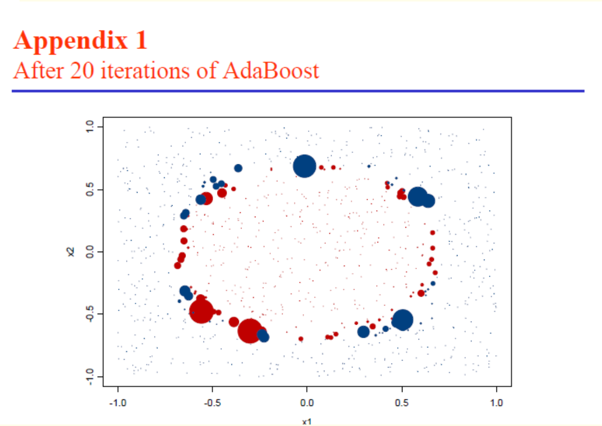
Boosting after 1 iteration



Boosting after 3 iterations



Boosting after 20 iterations



Boosting after 100 iterations