

STAT406- Methods of Statistical Learning Lecture 17

Matias Salibian-Barrera

UBC - Sep / Dec 2018

Random forests

- Feature ranking - relative importance of each variable
- Given a single tree T , at each node t split we can compute the sum of reductions in sum of squares (or gini or deviance measures) m_t^2
- We assign this squared measure m_t^2 to the variable (feature) used in the split

Random forests

- To each feature, we assign the sum of “squared gains” attributed to it
- For the i -th variable X_i we have

$$\mathcal{J}_i^2(T) = \begin{cases} m_t^2 & \text{if split involved } X_i \\ 0 & \text{otherwise} \end{cases}$$

Random forests

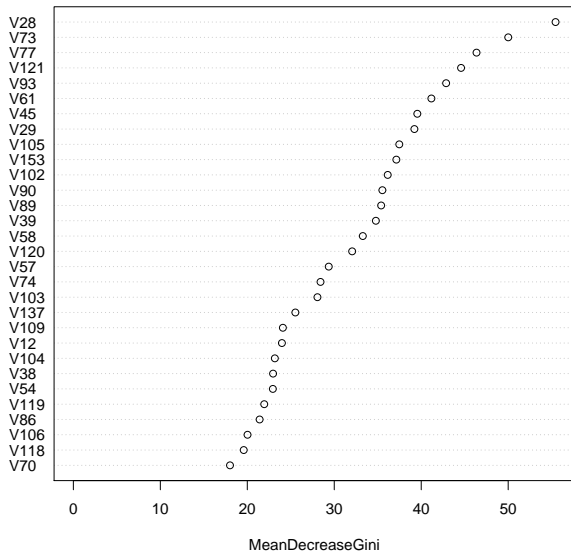
- For a random forest we use

$$\mathcal{J}_i^2 = \sum_T \mathcal{J}_i^2(T)$$

- In other words, we sum (or average) the importance of the variable across the trees in the forest

Random Forest - plot

Zip codes



Ensembles

- Ensembles of classifiers
- Combine classifiers trained on the same (or similar [e.g. bootstrapped]) data
- Consensus is reached by (equally weighted) voting or averaged estimated probabilities.
- Bagging and Random Forests are examples of ensembles.

Boosting

- Originally proposed for classification
- Main idea: sequentially re-train a simple classifier assigning more importance to points that were previously misclassified

Boosting

- The end result is a weighted average of all the classifiers
- Interesting ideas:
 - Not all components of the ensemble are treated equally
 - Members of the ensemble use information about other members
 - The underlying loss function has a “margin” (unlike 0-1 losses)

Boosting - AdaBoost.M1

Algorithm. Data (y_i, \mathbf{x}_i) , with $y_i \in \{-1, 1\}$

- Set initial weights $w_i = 1/n$, $1 \leq i \leq n$
- For $j = 1, \dots, K$
- Build a classifier $T_j(\mathbf{x})$ to the data using weights w_i , $1 \leq i \leq n$

Boosting - AdaBoost.M1

- Let

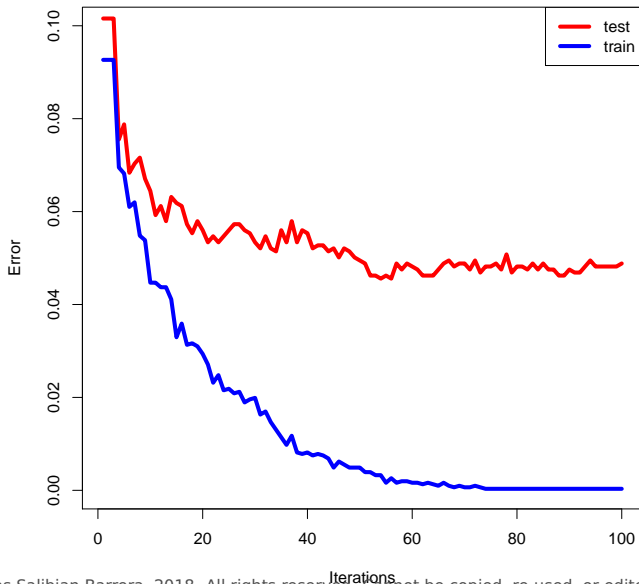
$$e_{\mathbf{j}} = \frac{\sum_{\mathbf{i}=1}^n w_{\mathbf{i}} I(y_{\mathbf{i}} \neq T_{\mathbf{j}}(\mathbf{x}_{\mathbf{i}}))}{\sum_{\ell=1}^n w_{\ell}}$$

- Let $\alpha_{\mathbf{j}} = \log((1 - e_{\mathbf{j}})/e_{\mathbf{j}})$ and
 $w_{\mathbf{i}} = w_{\mathbf{i}} \exp(\alpha_{\mathbf{j}} I(y_{\mathbf{i}} \neq T_{\mathbf{j}}(\mathbf{x}_{\mathbf{i}})))$, $\mathbf{i} = 1, \dots, n$
- Final classifier:

$$T(\mathbf{x}) = \text{sign} \left(\sum_{\mathbf{j}=1}^K \alpha_{\mathbf{j}} T_{\mathbf{j}}(\mathbf{x}) \right)$$

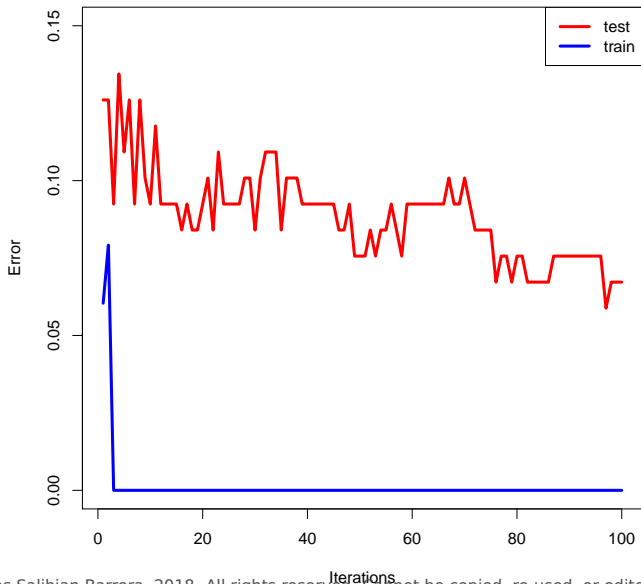
Error evolution - Spam data

AdaBoost error Vs number of trees



Error evolution - Isolet

AdaBoost error Vs number of trees



Boosting

- Boosting is fitting an **additive model**
- ... using a **forward search algorithm**
- ... and a **specific loss function**

Boosting

- Think of classifiers of the form

$$G(\mathbf{x}) = \sum_{j=1}^K \beta_j f(\mathbf{x}, \gamma_j)$$

where $f(\mathbf{x}, \gamma_j)$ are simple base classifiers (e.g. trees)

Boosting

- Given a data set $(y_i, \mathbf{x}_i), i = 1, \dots, n$

$$\begin{aligned} \min_G \sum_{i=1}^n L(y_i, G(\mathbf{x}_i)) &= \\ &= \min_{\beta, \gamma} \sum_{i=1}^n L(y_i, \sum_{j=1}^K \beta_j f(\mathbf{x}_i, \gamma_j)) \end{aligned}$$

where $\beta = (\beta_1, \dots, \beta_K)'$ and
 $\gamma = (\gamma_1, \dots, \gamma_K)'$

Boosting

- Find approximate solutions sequentially
- Start with $f_0(\mathbf{x}) = 0$
- `for(j in 1:K)`
- Find

$$(\beta_j, \gamma_j) = \arg \min_{\beta, \gamma} \sum_{i=1}^n L(y_i, f_{j-1}(\mathbf{x}_i) + \beta f(\mathbf{x}_i, \gamma))$$

- Let $f_j(\mathbf{x}) = f_{j-1}(\mathbf{x}) + \beta_j f(\mathbf{x}, \gamma_j)$

Counterexample

```
> set.seed(123)
> n <- 100
> x1 <- rnorm(n)
> x2 <- rnorm(n)
> y <- 2 - x1 + 6*x2 + rnorm(n, sd=.7)
> m1 <- lm(y~x1+x2)
> ( obj1 <- sum( resid(m1)^2 ) )
[1] 43.01311
>
> r1 <- y - mean(y)
> m2 <- lm(r1~x1-1)
> r2 <- resid(m2)
> m3 <- lm(r2~x2-1)
> ( obj2 <- sum( resid(m3)^2 ) )
[1] 109.3264
```