2.5
Random Forrests

SS 2019

Prof. Dr. Rainer Lienhart

www.multimedia-computing.{de,org}

Reference

- Friedman, J., Hastie, T. and Tibshirani, R. *Additive Logistic Regression: a Statistical View of Boosting* Annals of Statistics 28(2), 337-407. 2000.

  ➔ Figures and notation are taken from this reference

What is Boosting

- Meta-Learning algorithm
- Boosting is a way of combining the performance of many “weak” classifiers to produce a powerful “committee”.
- Fits an additive model $\sum_m f_m$ (x) in a forward stagewise manner
- Several variants:
  - Discrete Adaboost
  - Real AdaBoost
  - LogitBoost
  - Gentle AdaBoost

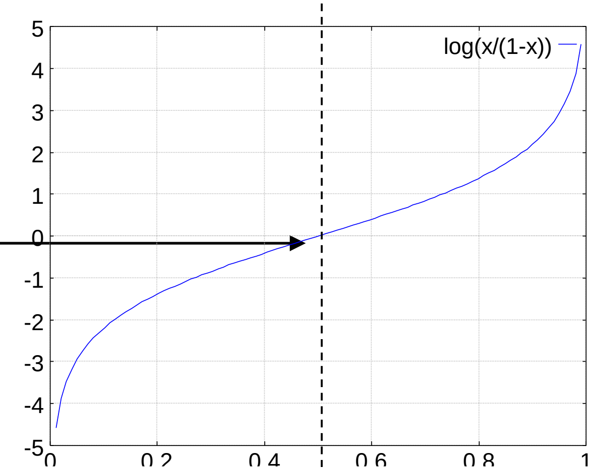
Discrete AdaBoost

Discrete AdaBoost [Freund and Schapire (1996b)]

1. Start with weights $w_i = 1/N$, $i = 1, \dots, N$.
2. Repeat for $m = 1, 2, \dots, M$:
   (a) Fit the classifier $f_m(x) \in \{-1, 1\}$ using weights w_i on the training data.
   (b) Compute $\mathrm{err}_m = E_w[1_{(y \neq f_m(x))}]$, $c_m = \log((1 - \mathrm{err}_m)/\mathrm{err}_m)$.
   (c) Set $w_i \leftarrow w_i \exp[c_m 1_{(y_i \neq f_m(x_i))}]$, $i = 1, 2, \dots, N$, and renormalize so that $\sum_i w_i = 1$.
3. Output the classifier $\mathrm{sign}[\sum_{m=1}^{M} c_m f_m(x)]$.

ALGORITHM 1. *E_w represents expectation over the training data with weights $w = (w_1, w_2, \dots, w_N)$, and $1_{(S)}$ is the indicator of the set S. At each iteration, AdaBoost increases the weights of the observations misclassified by $f_m(x)$ by a factor that depends on the weighted training error.*

Logit-transform:]0,1[→]-∞,+∞[



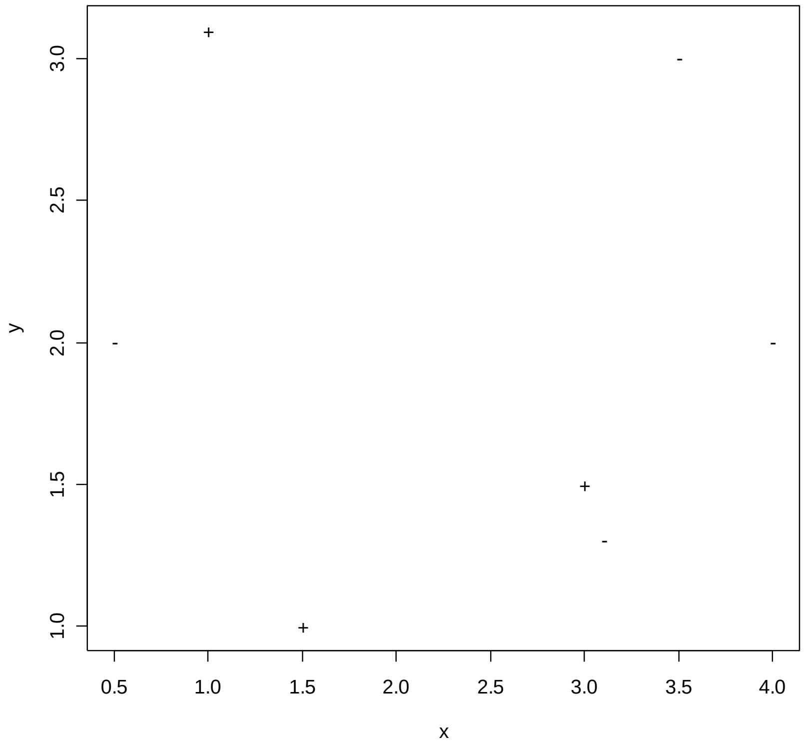
Example (1)

- Training Set

```
+: (1   , 3.1)
   (1.5 , 1)
   (3   , 1.5)
-: (0.5 , 2)
   (4   ,  2)
   (3.5 , 3)
   (3.1 , 1.3)
```

- R

```
x <- c(1, 1.5, 3, 0.5, 4, 3.5, 3.1)
y <- c(3.1, 1, 1.5, 2, 2, 3,1.3)
w <- rep(1/7, 7)
c <- c('+', '+','+', '-','-','-','-')
plot(x,y,pch = c)
```



Example (2)

Sort with respect to ‘x’

V	C	# correct class. >+	+<
		3	4
0.5	-		
		4	3
1	+		
		3	4
1.5	+		
		2	5
3	+		
		1	**6**
3.1	-		
		2	5
3.5	-		
		3	4
4	-		
		4	3

Sort with respect to ‘y’

V	C	# correct class. >+	+<
		3	4
1	+		
		2	**5**
1.3	-		
		3	4
1.5	+		
		2	**5**
2.0	-		
2.0	-		
		4	3
3	-		
		5	2
3.1	+		
		4	3

Example (3)

- Choose x

$$f_1(x,y) = \begin{cases} +1 & if\ (x < 3.05) \\ -1 & otherwise \end{cases}$$

$$E_w(1_{f_1(x,y) \neq c(x,y)}) = 1/7$$

$$c_1 = \log \frac{6/7}{1/7} = \log \frac{6}{1} = \log 6$$

$$\exp(\log(6)) = 6$$

- Round 2

w <- 1/12*(1,1,1,6,1,1,1)

Example (4)

Sort with respect to 'x'

V	C	>+	Σ	+<	Σ
		3	3	4	**9**
0.5	–				
		4	**9**	3	3
1	+				
		3	8	4	4
1.5	+				
		2	7	5	5
3	+				
		1	6	6	6
3.1	–				
		2	7	5	5
3.5	–				
		3	8	4	4
4	–				
		4	**9**	3	3

Sort with respect to 'y'

V	C	>+	Σ	+<	Σ
		3	3	4	9
1	+				
		2	2	**5**	**10**
1.3	–				
		3	3	4	5.5
1.5	+				
		2	2	**5**	**10**
2.0	–				
2.0	–				
		4	9	3	3
3	–				
		5	**10**	2	2
3.1	+				
		4	9	3	3

Example (5)

- Choose x

$$f_2(x,y)=\begin{cases}+1 & if\,(y<1.7)\\ -1 & otherwise\end{cases}$$

$$E_w(1_{f_2(x,y)\neq c(x,y)}) = 2/12$$

$$c_2 = \log\frac{10/12}{2/12} = \log 5$$

$$\exp(\log(5)) = 5$$

- Round 3

  w <- 1/20*

  (5,1 ,1,6,1,1, 5)

- …

Real AdaBoost

Real AdaBoost

1. Start with weights $w_i = 1/N$, $i = 1, 2, \dots, N$.
2. Repeat for $m = 1, 2, \dots, M$:
   (a) Fit the classifier to obtain a class probability estimate $p_m(x) = \hat{P}_w(y = 1|x) \in [0, 1]$, using weights w_i on the training data.
   (b) Set $f_m(x) \leftarrow \frac{1}{2} \log p_m(x)/(1 - p_m(x)) \in R$.
   (c) Set $w_i \leftarrow w_i \exp[-y_i f_m(x_i)]$, $i = 1, 2, \dots, N$, and renormalize so that $\sum_i w_i = 1$.
3. Output the classifier $\operatorname{sign}[\sum_{m=1}^{M} f_m(x)]$.

ALGORITHM 2. *The Real AdaBoost algorithm uses class probability estimates $p_m(x)$ to construct real-valued contributions $f_m(x)$.*

Example1

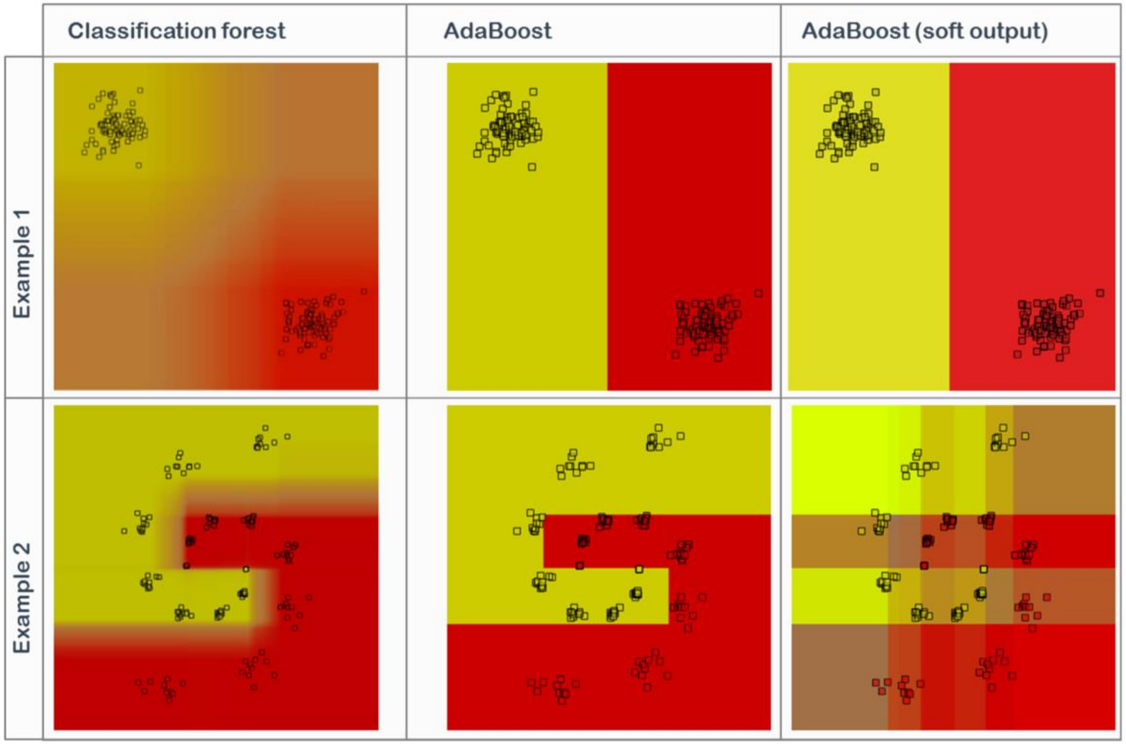


Fig. 3.14 Comparison between classification forests and boosting on two examples. Forests produce a smooth, probabilistic output. High uncertainty is associated with regions between different classes or away from training data. Capitalized produces a hard output. Interpreting the output of a boosted strong classifier as real valued does not seem to produce meaningful confidence. The forest parameters are: $D = 2$, $T = 200$, and we use axis-aligned weak learners. Boosting was also run with 200 axis-aligned stumps and the remaining parameters optimized to achieve best results.

Figure taken from: Antonio Criminisi, Jamie Shotton, Ender Konukoglu . *Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning*. In Foundations and Trends® in Computer Graphics and Vision, Vol. 7: No 2-3, pp 81-227, 2011.

Example 2

- Later when we talk about fast object detection.