Boosting

Bagging: Bootstrap Aggregating

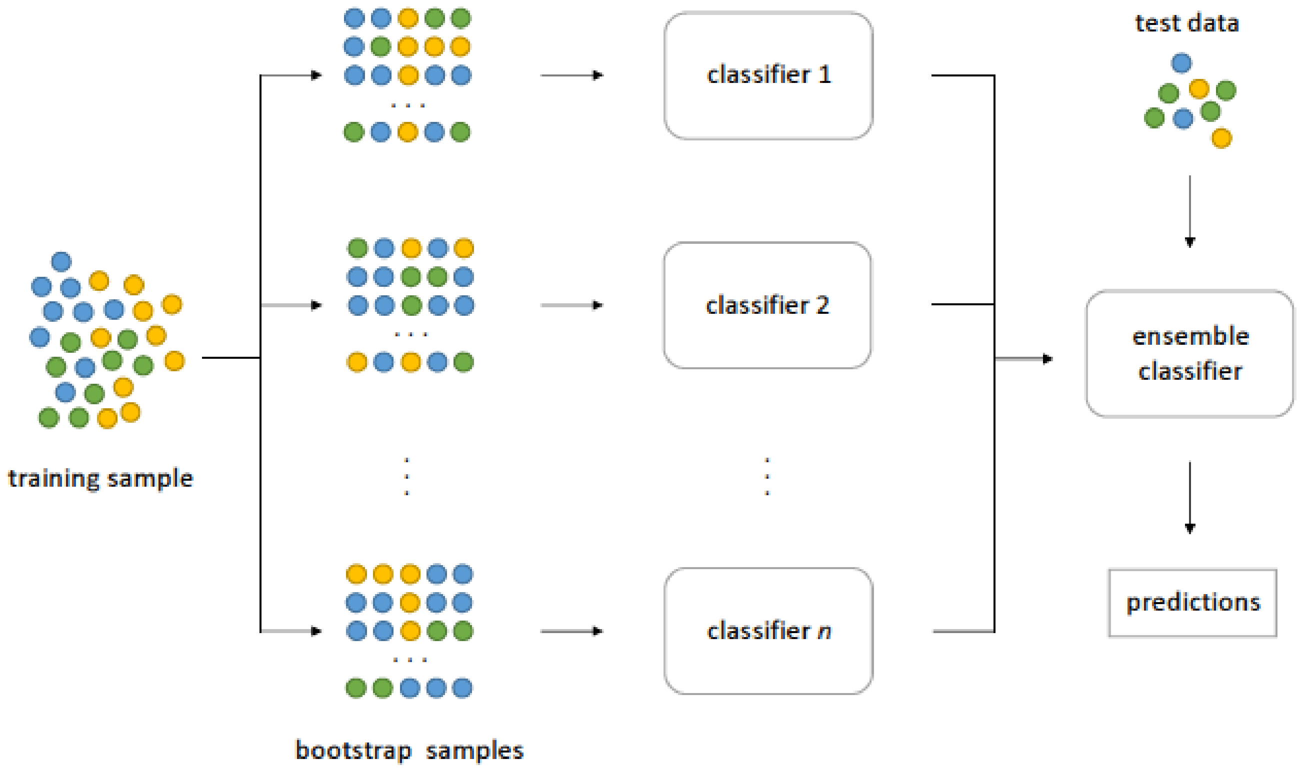

Figure 5: The bagging approach. Several classifier are trained on bootstrap samples of the training data. Predictions on test data are obtained combining the predictions of the trained classifiers with a majority voting scheme.

- Classifiers are built in parallel
- Classifiers are trained on subsamples of the data
- Every classifier has a similar weight towards the final prediction

Image taken from Data Mining: Accuracy and Error Measures for Classification and Prediction. Galdi and Tagliaferri, 2018

Boosting

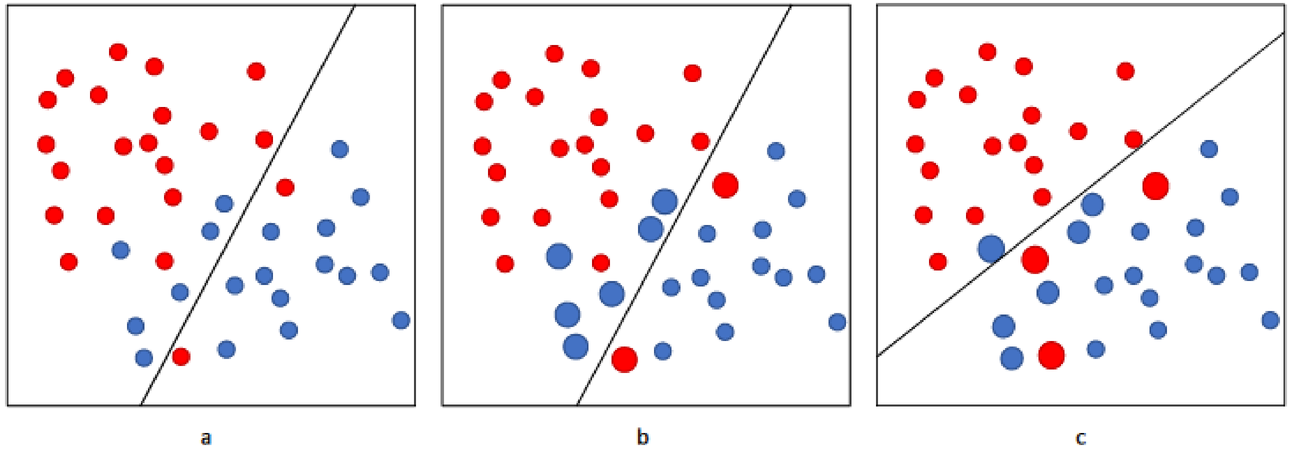

Figure 6: The boosting approach. A classifier is trained on the original data (a). The weights of misclassified instances (dot size in the figure) are increased (b). A new classifier is trained on the new data set and weights are updated accordingly (c).

Image taken from Data Mining: Accuracy and Error Measures for Classification and Prediction. Galdi and Tagliaferri, 2018

- Classifiers are built sequentially
- Classifiers are trained on all data
- Observations are given **weights** which reflect how difficult they are to classify.
- Each classifier's prediction has a different **weight** towards final decision, based on their accuracy

Boosting

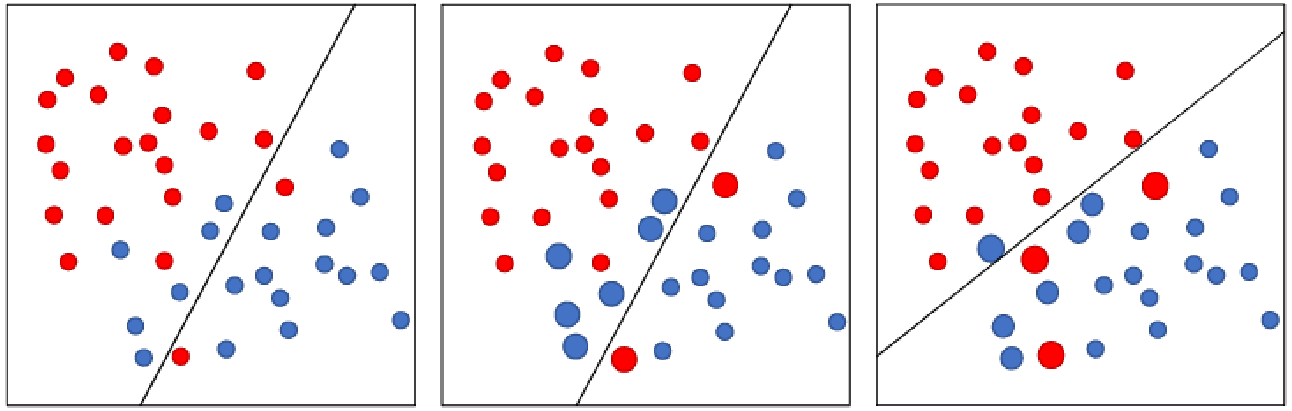

Figure 6: The boosting approach. A classifier is trained on the original data (a). The weights of misclassified instances (dot size in the figure) are increased (b). A new classifier is trained on the new data set and weights are updated accordingly (c).

Image taken from Data Mining: Accuracy and Error Measures for Classification and Prediction. Galdi and Tagliaferri, 2018

- First classifier: all observations are given the same weight
- Second classifier: observations miss-classified by previous classifier have a higher weight.
- The weights are adjusted in each iteration, that is for each new classifier
- The classifiers also get a weight based on their accuracy, that weights their overall contribution to the final prediction

AdaBoost

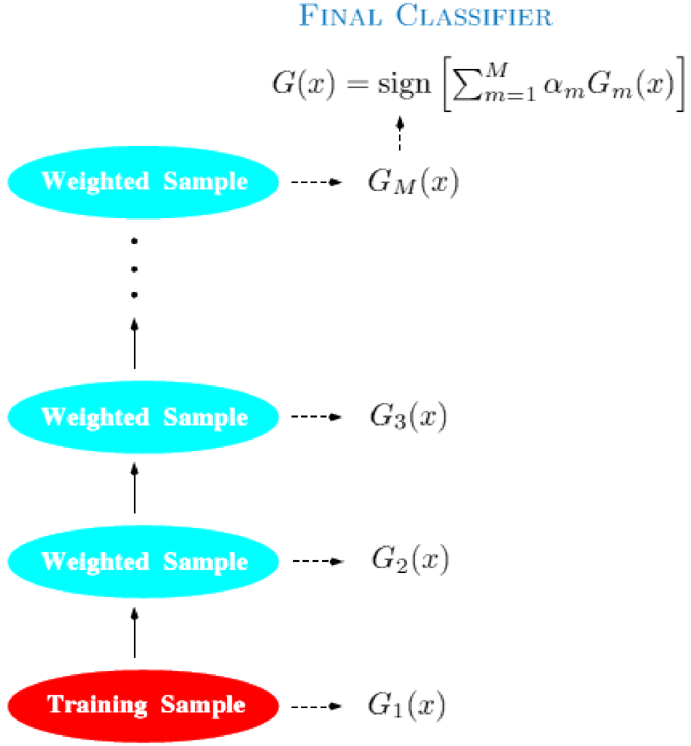

FIGURE 10.1. *Schematic of AdaBoost. Classifiers are trained on weighted versions of the dataset, and then combined to produce a final prediction.*

Algorithm 10.1 *AdaBoost.M1.*

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

Images taken from Elements of Statistical Learning, Hastie et al

Gradient Boosting Machines

AdaBoost

$$G(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right)$$

- Gm(x) is each classifier
- αm is its weight towards the final outcome

Generalized additive model

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

- b(x,γm) is each classifier
- For trees γm refers to the tree splits
- βm is its weight towards the final outcome

Gradient Boosting Machines

Generalized additive model

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

Algorithm 10.2 *Forward Stagewise Additive Modeling.*

1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M:

   (a) Compute

   $$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)).$$

   (b) Set $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.

Gradient Boosting Machines

Generalized additive model

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

- At each iteration, that is, each new classifier, minimizes the difference between its predictions and the residuals of the previous classifier.

Algorithm 10.2 *Forward Stagewise Additive Modeling.*

1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M:
   (a) Compute

$$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)).$$

   (b) Set $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.

Residuals of previous classifier

$$= (\underbrace{y_i - f_{m-1}}_{\text{Residuals of previous classifier}}(x_i) - \beta b(x_i; \gamma))^2$$

$$= (r_{im} - \beta b(x_i; \gamma))^2,$$

THANK YOU

www.trainindata.com