An Introduction to Gradient Boosting Machine

Chengliang Tang

10/18/2016

What is Boosting?

- One of the most powerful learning ideas introduced in the last twenty years.
- Originally designed for classification problems, also extended to regression problems.
- **Motivation**: combine the outputs of many “weak” classifiers to produce a powerful “committee”.
- “Weak” classifier: classifiers with error rates slightly better than a random guessing.
- Attention: weak learners should not be highly correlated .

How to combine?

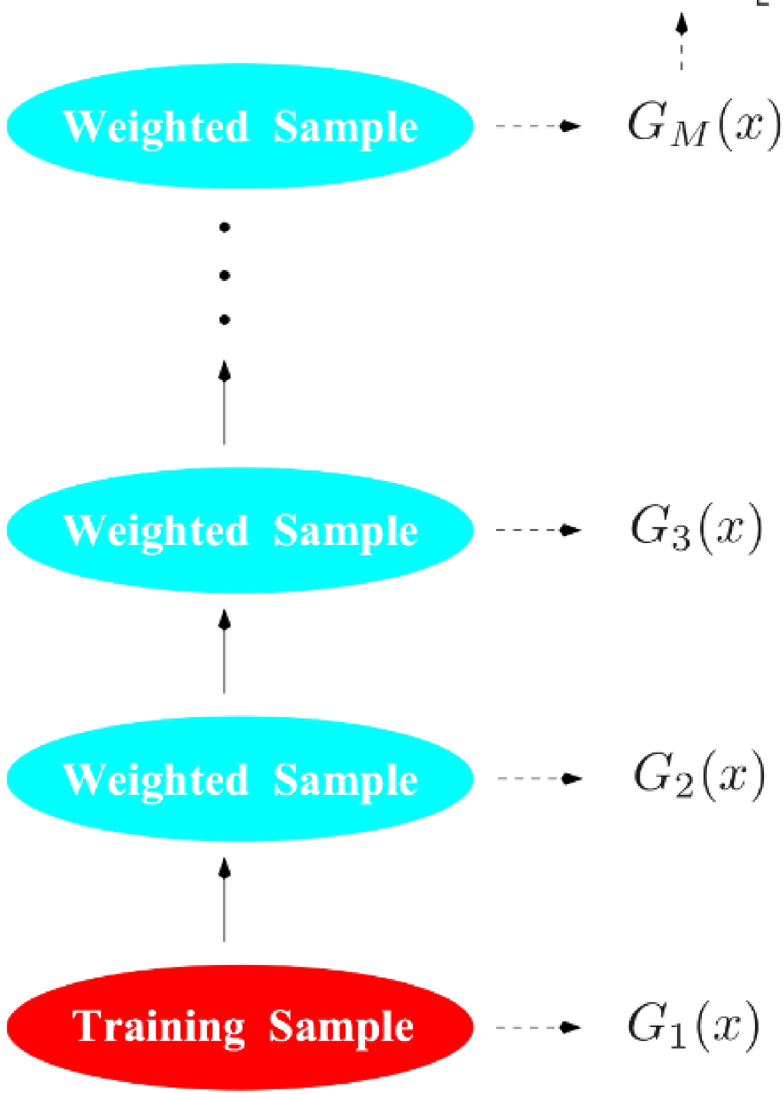
- Repeated modified versions of data
- A sequence of weak classifiers $G_m(x)$
- A weighted majority vote

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

How to combine?

FINAL CLASSIFIER

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

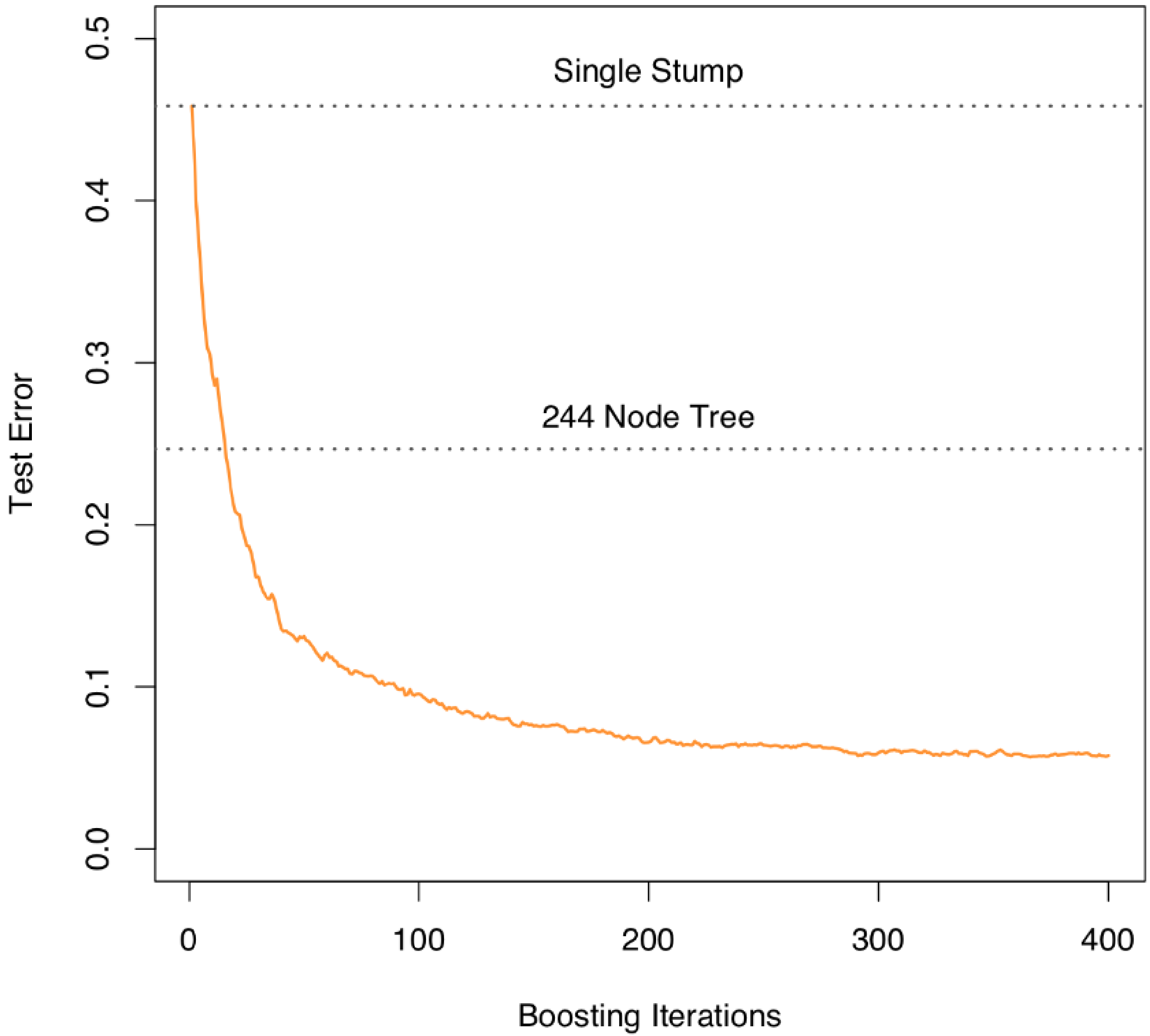


Example: AdaBoost.M1

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

Performance



Why is Boosting successful?

- Basis function expansions

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

- More general,

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

Why is Boosting successful?

- Minimizing loss function

$$\min_{\{\beta_m,\gamma_m\}_1^M} \sum_{i=1}^{N} L\left(y_i, \sum_{m=1}^{M} \beta_m b(x;\gamma_m) \right)$$

- Greedy approximation:

Sequentially adding new basis functions to the expansion

Forward stagewise additive learning

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

Why is Boosting successful?

- AdaBoost.M1 is equivalent to forward stagewise additive modeling using **the exponential loss function**

$$L\big(y, f(x)\big) = \exp(-yf(x))$$

i.e.,

$$(\beta_m, G_m) = arg \min_{\beta,G} \sum_{i=1}^{N} \exp[-y_i(f_{m-1}(x_i) + \beta G(x_i))]$$

Exponential loss

- Binary classification: Output $Y \in \{-1,1\}$
- Regression function for classification $f(x)$:
$$G(x) = sign(f(x))$$
- Margin: $yf(x)$ Loss: $\exp(-yf(x))$
- Large positive margin means the prediction is correct and well within the classification boundary.
- Large negative margin means the prediction is wrong and very much across the classification boundary.

Boosting trees

- Decision trees:

Partition the space of all joint predictor variable values into disjoint regions $R_j, j = 1,2,\ldots,J$ (terminal nodes of the tree)

- Prediction rule

$$x \in R_j \Longrightarrow f(x) = \gamma_j$$

- Formal expression

$$T(x;\Theta) = \sum_{j=1}^{J} \gamma_j I(x \in R_j)$$

Boosting trees

- Boosted tree model

$$f_M(x) = \sum_{m=1}^{M} T(x; \Theta_m)$$

- Each step m,

$$\widehat{\Theta}_m = arg \min_{\Theta_m} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + T(x_i; \Theta_m))$$

Algorithm 10.3 *Gradient Tree Boosting Algorithm.*

1. Initialize $f_0(x) = \arg\min_\gamma \sum_{i=1}^N L(y_i, \gamma)$.
2. For $m = 1$ to M:

   (a) For $i = 1, 2, \ldots, N$ compute

   $$r_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{m-1}}.$$

   (b) Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm},\ j = 1, 2, \ldots, J_m$.

   (c) For $j = 1, 2, \ldots, J_m$ compute

   $$\gamma_{jm} = \arg\min_\gamma \sum_{x_i \in R_{jm}} L\left(y_i, f_{m-1}(x_i) + \gamma\right).$$

   (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

3. Output $\hat{f}(x) = f_M(x)$.

Parameter Setting

- Tree size

$$J_m = J, m = 1,2, \dots, M$$

- Boosting iterations

$$L(f_M) \searrow, as\ M \nearrow$$

M^*: monitor prediction risk as a function of M on a validation sample

- Shrinkage

$$f_m(x) = f_{m-1}(x) + \nu \cdot \sum_{j=1}^{J} \gamma_{jm} I(x \in R_{jm})$$

Thanks!