# Some investigations on Adaptive Boost Decision Tree method

Ligang Xia
Warwick University

September 19, 2018

# Introduction

- Adaptive BDT and Gradient BDT used in HEP for signal-background separation, object identification, ...
- They are to minimize some type of lose function.

| AdaBDT | values |
|---|---|
| input variables | $\vec{x} = (x_1, x_2, \cdots)$ |
| true value $Y$ | -1, 1 |
| guess value $k_m$ | -1, 1 |
| guess weight $\alpha_m$ | $\frac{1}{2} \ln \frac{1-\epsilon_m}{\epsilon_m}$ |
| punishment for wrong guesses | apply a weight of $e^{\alpha_m}$ for next tree |
| final BDT score $y_m$ | $y_m = y_{m-1} + \alpha_m k_m = \sum_{i=1}^{m} \alpha_i k_i$ |
| lose function | $L_m(\vec{x}, y_m) = \sum_{\vec{x}} e^{-Y(\vec{x}) y_m(\vec{x})}$ |

Here $\epsilon_m$ is the miscalssification rate for the $m-$th tree.
Here are two questions.

1. Is the minimization of the lose function equivant to the maximization of the statistic significance ?
2. How to introduce systematic uncertainties in the training (further, how to optimize this process)?

# Is the minimization of the lose function equivant to the maximization of the statistic significance ?

Here are the steps.

1. Obtain the ditribution of the BDT score
2. Calculate the significance based on the distributions of BDT score
3. Compare the lose function and the significance

# Probability distribution function of the BDT score

Let $g_m(y_m)$ be the probability distribution function of the BDT score after $m$ trees.

$$\int_y^{y+\delta} g_m(y_m)dy_m = \int_{y<y_m(\vec{x})<y+\delta} f(\vec{x})d\vec{x} \tag{1}$$

This is difficult as $y(\vec{x})$ is unknown. We use the iteration formula.

$$y_m(\vec{x}) = y_{m-1}(\vec{x}) + \alpha_m k_m(\vec{x}) \tag{2}$$

$$y_m = y_{m-1} + \alpha_m k_m \tag{3}$$

Taking $y_m$, $y_{m-1}$ and $k_m$ as random variables and assuming $y_{m-1}$ and $k_m$ are independent, we have

$$\int_y^{y+\delta} g_m(y_m)dy_m = \int_{y<y_m<y+\delta} g_{m-1}(y_{m-1})f(k_m)dy_{m-1}dk_m \tag{4}$$

$$= \int_{y<y_m<y+\delta} g_{m-1}(y_m - \alpha_m k_m)f(k_m)dy_m dk_m \tag{5}$$

$$\text{for signal} = \int_y^{y+\delta} g_{m-1}(y_m - \alpha_m)(1-\epsilon_m) + g_{m-1}(y_m + \alpha_m)\epsilon_m dy_m \tag{6}$$

# Probability distribution function of the BDT score

After $m$ trees, the BDT score PDF for signal is

$$g_m(y) = g_{m-1}(y-\alpha_m)(1-\epsilon_m) + g_{m-1}(y+\alpha_m)\epsilon_m \tag{7}$$

and for background is

$$g_m(y) = g_{m-1}(y+\alpha_m)(1-\epsilon_m) + g_{m-1}(y-\alpha_m)\epsilon_m \,. \tag{8}$$

Here are two cases.

$$g_1(y) = \quad g_0(y-\alpha_1)(1-\epsilon_1) + g_0(y+\alpha_1)\epsilon_1 \tag{9}$$

$$\begin{aligned} g_2(y) = & \quad g_1(y-\alpha_2)(1-\epsilon_2) + g_1(y+\alpha_2)\epsilon_2 \qquad (10)\\ = & \quad g_0(y-\alpha_1-\alpha_2)(1-\epsilon_1)(1-\epsilon_2) \\ & + g_0(y+\alpha_1-\alpha_2)\epsilon_1(1-\epsilon_2) \\ & + g_0(y-\alpha_1+\alpha_2)(1-\epsilon_1)\epsilon_2 \\ & + g_0(y+\alpha_1+\alpha_2)\epsilon_1\epsilon_2 \qquad (11) \end{aligned}$$

The two cases above illustrate how we obtain the final PDF starting from $g_0(y)$ which will be shown to be $\delta(y)$. For signal, it is

$$g_m(y) = \sum_{\sigma_1=\pm 1} \cdots \sum_{\sigma_m=\pm 1} g_0(y + \sum_{i=1}^{m} \sigma_i\alpha_i)\Pi_{i=1}^{m}(\frac{1-\sigma_i}{2} + \sigma_i\epsilon_i) \tag{12}$$

# Probability distribution function of the BDT score

Looking at the case of $m = 1$, after 1 tree, the BDT score $y_1$ has only two values, namely, $+\alpha_1$ with a probability of $1 - \epsilon_1$ and $-\alpha_1$ with a probability of $\epsilon_1$ for signal. Thus $g_0(y) = \delta(y)$. Before any training, the PDF ($g_0(y)$) does not have any preference.

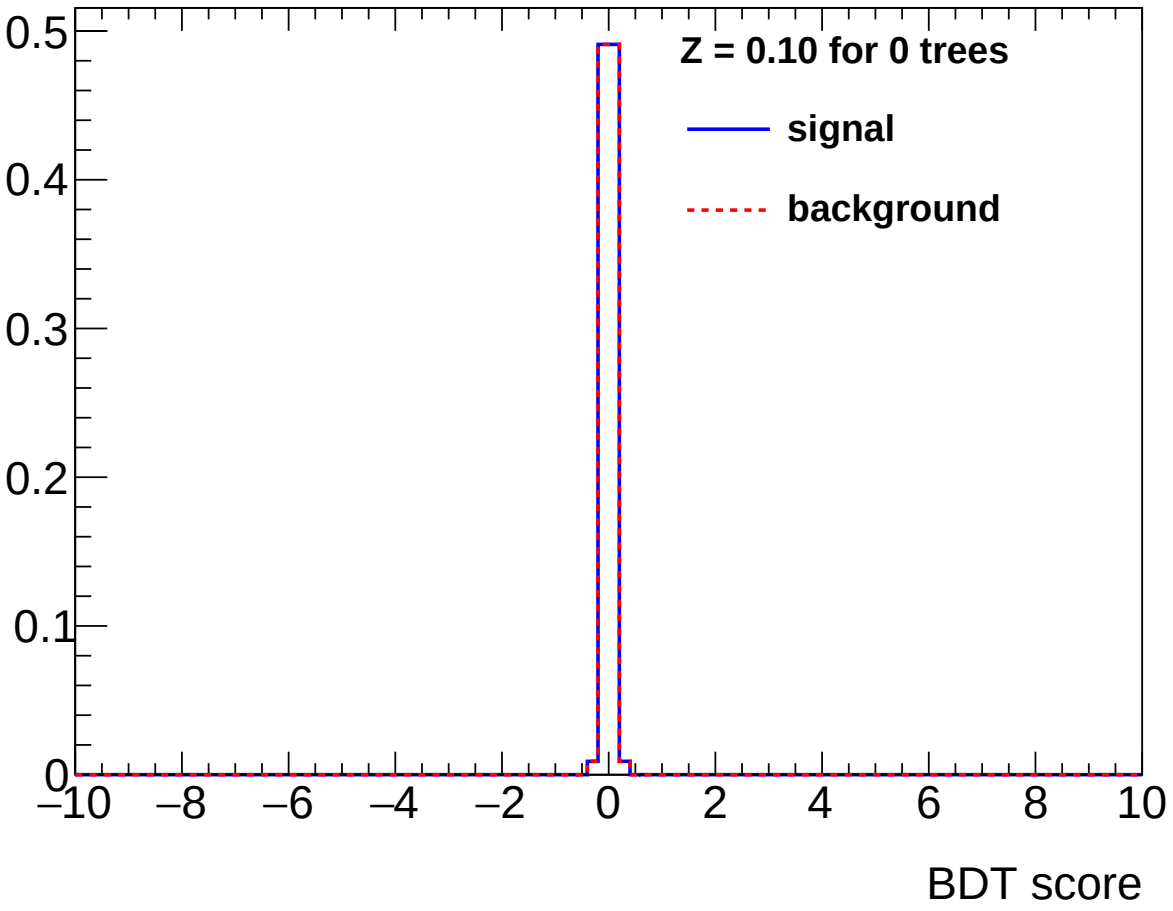
$$\begin{aligned} g_1(y) = & \quad g_0(y - \alpha_1)(1 - \epsilon_1) + g_0(y + \alpha_1)\epsilon_1 \\ = & \quad \delta(y - \alpha_1)(1 - \epsilon_1) + \delta(y + \alpha_1)\epsilon_1 \end{aligned} \tag{13}$$

Before calculating the final PDF, $g_m(y)$, let us look at how the BDT score distribution evolve with the number of trees. In the following example, we are using

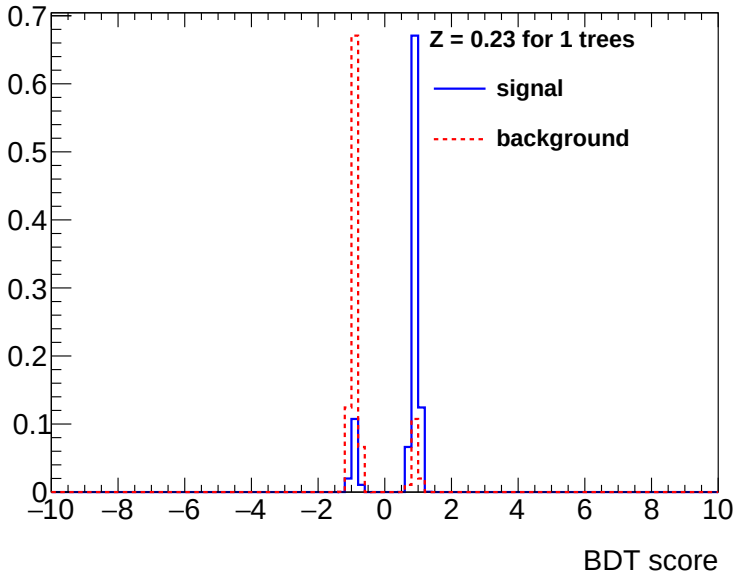
- They are from the iteration formula $y_m = y_{m-1} + \alpha_m k_m$, not from real AdaBDT training.
- $\epsilon_m = 0.5 - 0.4 \times e^{-0.1m}$ so that $\epsilon_0 = 0.1$ (this is a strong classifier to speed up the convergence)
- Number of signal events: $N_S = 1$ and number of background events: $N_B = 100$
- Use a narrow gaussian distribution to imitate $\delta(y)$

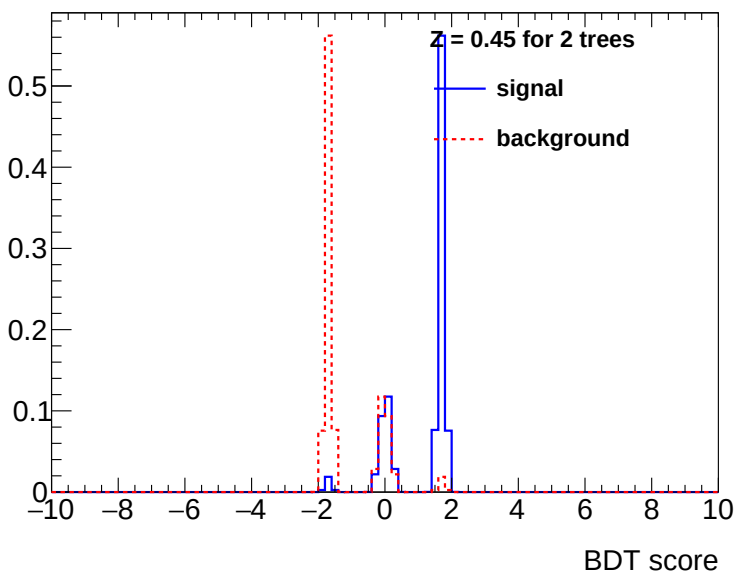
# Probability distribution function of the BDT score

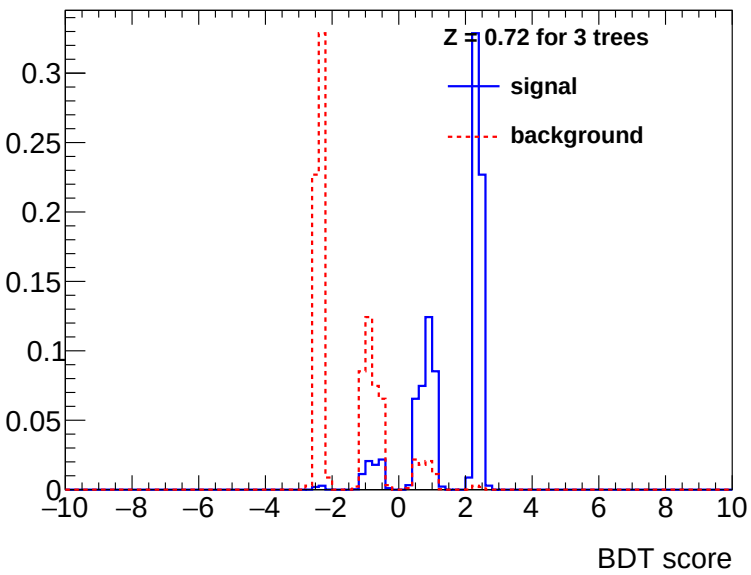
- this is the case of $m = 0$ before training
- $Z$ is the binned significance calculated from $N_S = 1$, $N_B = 100$ and the BDT score distribution.

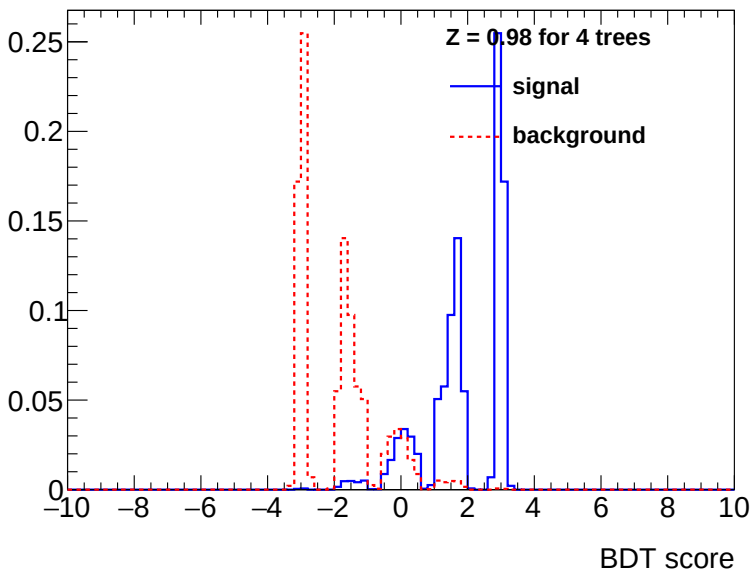


# Probability distribution function of the BDT score

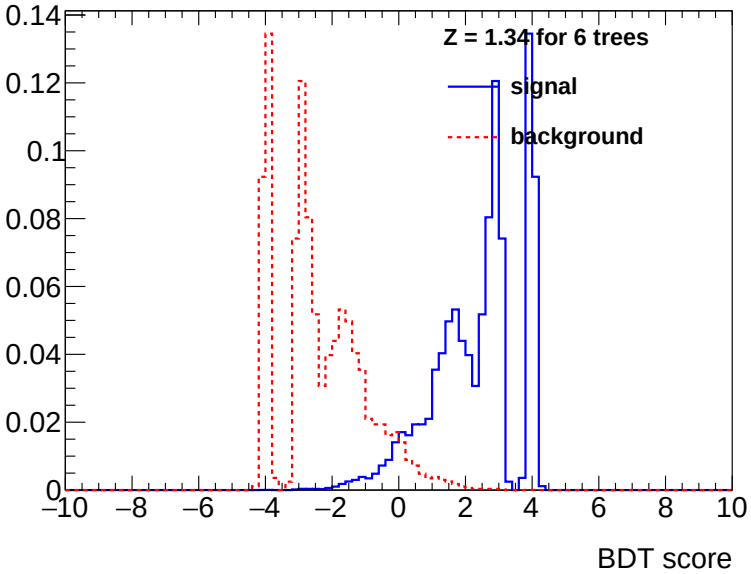


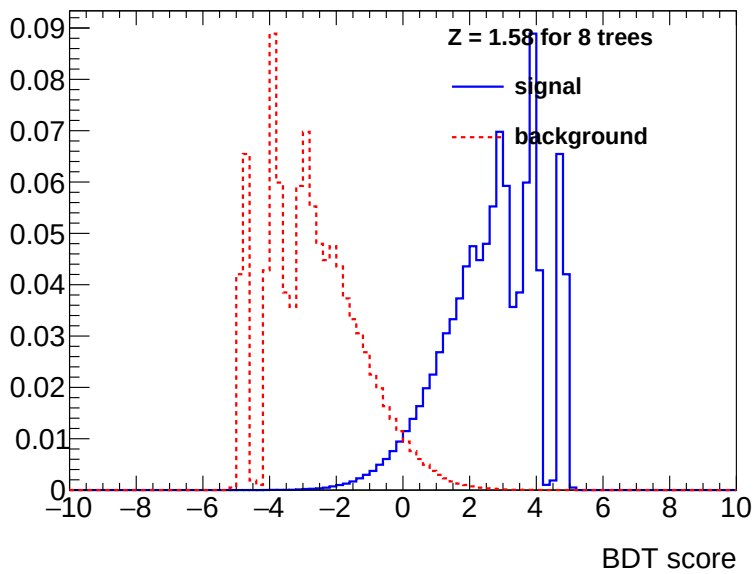



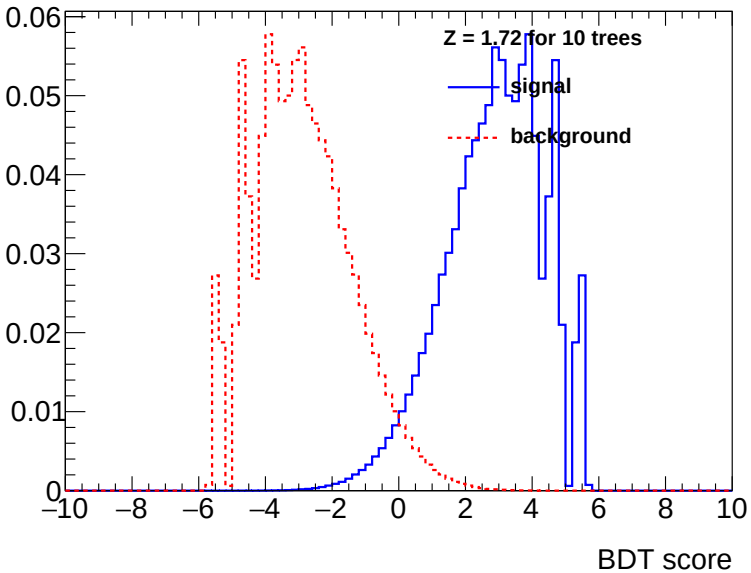



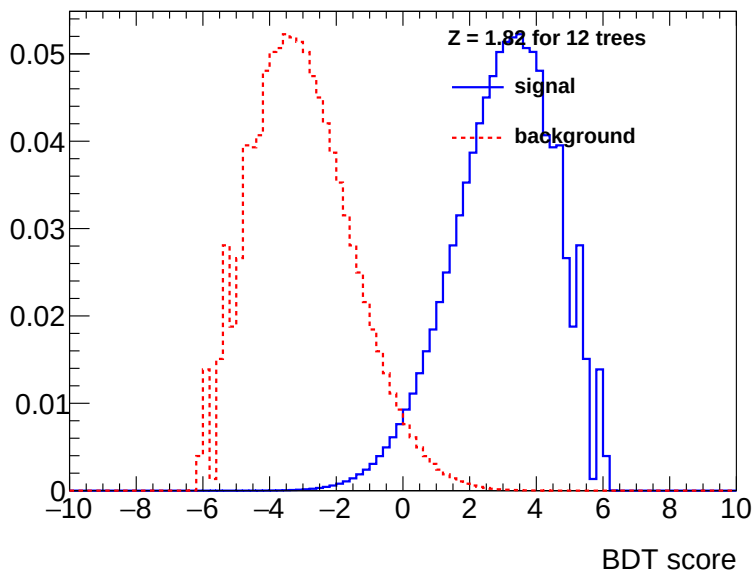



# Probability distribution function of the BDT score



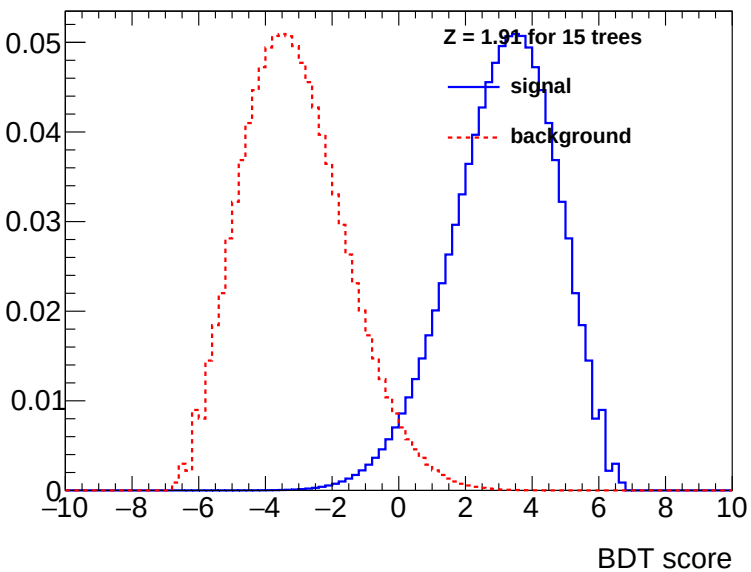



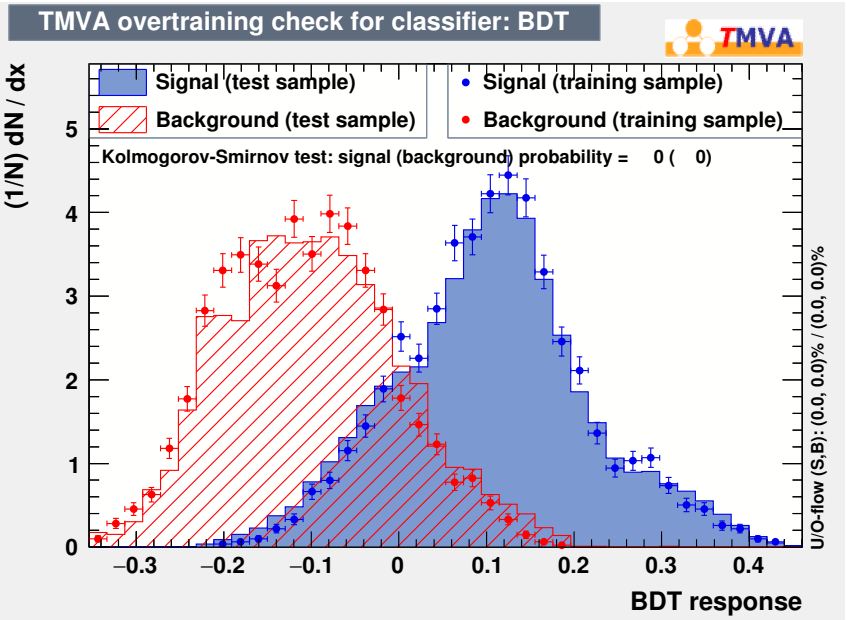



# Probability distribution function of the BDT score

- this is the case of $m = 15$ ( we would need 1000 trees in reality)
- right plot is from a real AdaBDT training
- the final distribution is close to gaussian



# Probability distribution function of the BDT score

For signal, the PDF of BDT score is

$$g_m(y) = \sum_{\sigma_1=\pm 1} \cdots \sum_{\sigma_m=\pm 1} \delta(y + \sum_{i=1}^{m} \sigma_i \alpha_i) \Pi_{i=1}^{m} (\frac{1-\sigma_i}{2} + \sigma_i \epsilon_i) \tag{14}$$

Let us try to simplify it assuming all trees are weak classifiers, namely, $\epsilon_i \to \frac{1}{2}^-$ ( 1/2 is the mis-classification rate of random guess) and $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i} \to 0^+$. We use the characteristic function.

$$\begin{aligned}
\phi_y(t) &= \int_{-\infty}^{+\infty} g_m(y) e^{iyt} dy & (15)\\
&= \sum_{\sigma_1=\pm 1} \cdots \sum_{\sigma_m=\pm 1} e^{-it \sum_{i=1}^{m} \sigma_i \alpha_i} \Pi_{i=1}^{m} (\frac{1-\sigma_i}{2} + \sigma_i \epsilon_i) & (16)\\
&= \sum_{\sigma_1=\pm 1} \cdots \sum_{\sigma_m=\pm 1} \Pi_{i=1}^{m} e^{-it\sigma_i \alpha_i} (\frac{1-\sigma_i}{2} + \sigma_i \epsilon_i) & (17)\\
&= \Pi_{i=1}^{m} \sum_{\sigma_i=\pm 1} e^{-it\sigma_i \alpha_i} (\frac{1-\sigma_i}{2} + \sigma_i \epsilon_i) & (18)\\
&= \Pi_{i=1}^{m} (\epsilon_i e^{-it\alpha_i} + (1-\epsilon_i) e^{it\alpha_i}) & (19)\\
&= \Pi_{i=1}^{m} (\cos(-t\alpha_i) + i(1-2\epsilon_i) \sin(t\alpha_i)) & (20)
\end{aligned}$$

# Probability distribution function of the BDT score

Finally, we can apply the limit $\epsilon_i \to \frac{1}{2}^-$ ($\alpha_i = \frac{1}{2}\ln\frac{1-\epsilon_i}{\epsilon_i} \to 1-2\epsilon_i \to 0^+$).

$$\begin{aligned} \phi_y(t) &= \Pi_{i=1}^m(\cos(-t\alpha_i) + i(1-2\epsilon_i)\sin(t\alpha_i)) && (21)\\ \ln\phi_y(t) &= \sum_{i=1}^m \ln(\cos(-t\alpha_i) + i(1-2\epsilon_i)\sin(t\alpha_i)) && (22)\\ &\approx -\frac{1}{2}\sigma^2 t^2 + i\mu t && (23) \end{aligned}$$

with

$$\sigma = \sqrt{\sum_{i=1}^m \alpha_i^2}\,, \quad \mu = \sum_{i=1}^m (1-2\epsilon_i)\alpha_i \approx \sigma^2\,. \tag{24}$$

Therefore, we obtain the PDF of the BDT score under the weak-classifier limit.

$$g_m^S(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \text{ for signal} \tag{25}$$

$$g_m^B(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y+\mu)^2}{2\sigma^2}} \text{ for background} \tag{26}$$

# Significance and the lose function

The significance using the BDT score as observable is

$$Q \equiv Z^2 = \sum_{i\in\text{bins}} 2((s_i+b_i)\ln(1+\frac{s_i}{b_i})-s_i) \tag{27}$$

$$\approx \sum_{i\in\text{bins}} \frac{s_i^2}{b_i} = \frac{N_S^2}{N_B}\sum_{i\in\text{bins}} \frac{(g_m^S(y_i))^2}{g_m^B(y_i)}\Delta y \tag{28}$$

$$= \frac{N_S^2}{N_B}\int \frac{(g_m^S(y))^2}{g_m^B(y)}dy = \frac{N_S^2}{N_B}e^{2\frac{\mu^2}{\sigma^2}} \approx \frac{N_S^2}{N_B}e^{2\mu} \quad \text{using } \mu\approx\sigma^2 \tag{29}$$

For lose function, we have

$$L_m(\vec{x},y_m) = \sum_{\vec{x}} e^{-Y(\vec{x})y_m(\vec{x})} = \sum_{\vec{x}} e^{-Y(\vec{x})(y_{m-1}(\vec{x})+\alpha_m k_m(\vec{x}))} \tag{30}$$

$$= \sum_{\vec{x}} e^{-Y(\vec{x})(y_{m-1}(\vec{x}))}(\epsilon_m e^{-\alpha_m}+(1-\epsilon_m)e^{\alpha_m}) \tag{31}$$

$$= L_{m-1}(\vec{x},y_{m-1})(\epsilon_m e^{-\alpha_m}+(1-\epsilon_m)e^{\alpha_m}) \tag{32}$$

where we use $\epsilon_m = \sum_{Y(\vec{x})\neq k_m(\vec{x})} e^{-Y(\vec{x})y_{m-1}(\vec{x})} / \sum_{\vec{x}} e^{-Y(\vec{x})y_{m-1}(\vec{x})}$.

# Significance and the lose function

The significance using the BDT score as observable is

$$Q \equiv Z^2 \approx \frac{N_S^2}{N_B} e^{2\frac{\mu^2}{\sigma^2}} \approx \frac{N_S^2}{N_B} e^{2\mu} \quad \text{using } \mu \approx \sigma^2 \tag{33}$$

For lose function, we have

$$\begin{aligned} L_m(\vec{x}, y_m) &= L_{m-1}(\vec{x}, y_{m-1})(\epsilon_m e^{-\alpha_m} + (1-\epsilon_m)e^{\alpha_m}) && (34)\\ &= \Pi_{i=1}^{m} \left(\epsilon_i e^{-\alpha_i} + (1-\epsilon_i)e^{\alpha_i}\right) && (35)\\ &= \phi_y(i) && (36)\\ &\approx e^{\frac{1}{2}\sigma^2 - \mu} \approx e^{-\frac{1}{2}\mu}. && (37) \end{aligned}$$

Eventually, we have

$$Z^2 \approx \frac{N_S^2}{N_B} L^{-4} \tag{38}$$

under the weak-classifier limit.

# How to introduce systematic uncertainties in the training (further, how to optimize this process)?

I do not have a conclusive strategy. Here are some ideas.

1. Training on both nominal and systematic samples for both signal and background. Choose the most conservative one (time and CPU consuming, cannot consider correlation)
2. Trainsfer the uncertainty of input variables, $\Delta\vec{x}$ to that of the BDT weight, $\Delta\alpha_i$. And enhance the punish weight $e^{\alpha_i} \rightarrow e^{\alpha_i+|\Delta\alpha_i|}$. (applicable, how to consider correlation)
3. During the growth of a tree, the tree branch split is determined by the Gini index. Replace Gini index by the significance. Maximize the signifiacnce to determine split position. We can use the significance as the BDT score. We can easily introduce systematic uncertainties to the significance. (I actually tried this idea, but the performance is not as good as the normal BDT).

# Summary

- Is the minimization of the lose function equivant to the maximization of the statistic significance ? Yes, under the week classifier limit.
- How to introduce systematic uncertainties in the training (further, how to optimize this process)? There are ideas. Will try

# BACK UP

BACK UP