g14 analysis using new statistical data analysis methods

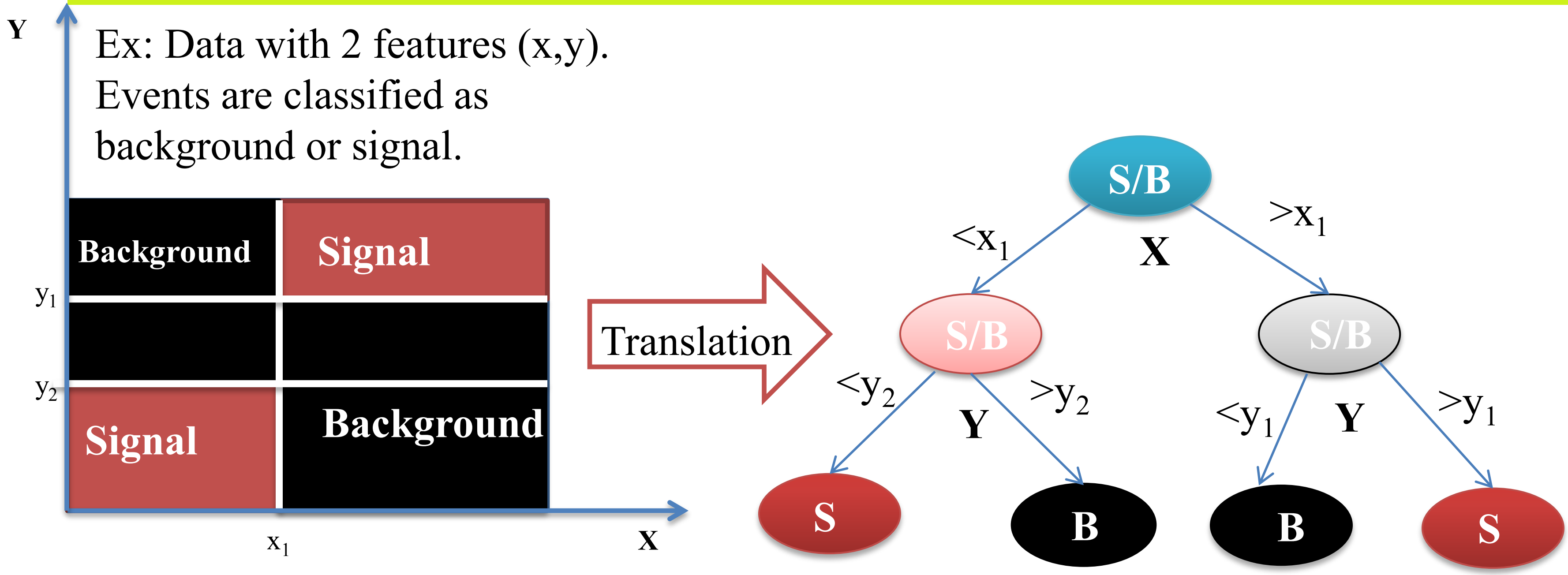
Dao Ho

03/07/2014

Outline

- **Boosted Decision Trees (BDT)**: a supervised-learning classifier, i.e, the BDT is **trained** before can be applied to **classification** tasks
- **Kernel Density Estimator (KDE)**: a smooth and continuous method to estimate density distributions
- **Bootstrap**: a data-resampling method to estimate standard deviation, or confidence interval

Decision Tree introduction:



- Decision tree is similar to "cut" method
- More advanced because it can classify high dimensional (>3) data simultaneously
- BOOSTED decision trees (better than a single tree) are very efficient (much better than the cut method)

Boosted Decision Tree (BDT):

- BDT: well-known, well-tested machine learning algorithm for classification tasks
- Use widely at CERN, incorporated into ROOT[1]
- Standard analysis tool for Hall D GlueX collaboration
- Introduced by Mike Williams (MIT) in several talks at GlueX meetings

---

- **NOT this talk**: explain the underlying mechanism of BDT (theory and implementation)
- **This talk: illustration the steps of using BDT for event selection in g14 analysis**
- **This talk: showing an improved performance over standard cut method in g14 analysis.**

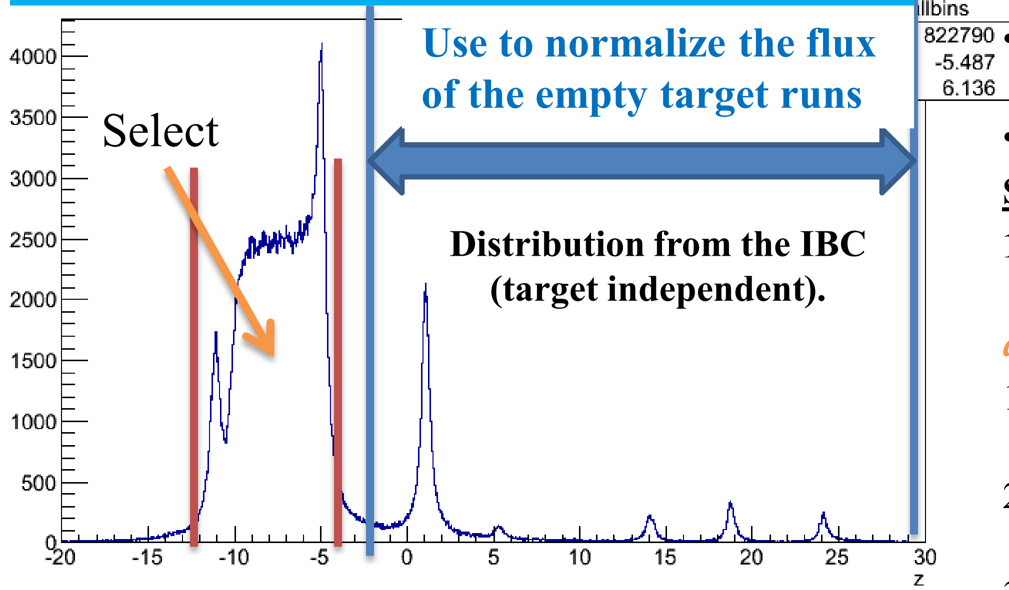
[1] TMVA: toolkit for multivariate data analysis with ROOT

BDT in action:

- **Background Subtraction: E asymmetry for $\gamma n(p_s) \rightarrow p\pi^-(p_s)$ reaction**
  - g14 targets have Al wires inside, and cell wall (background to remove)
  - DON'T select events with LARGE spectator proton momentum → **USE CUTS**
  - g14 experiment had empty target run, used for background subtraction
- **BDT: E asymmetry for $\gamma n(p_s) \rightarrow p\pi^-(p_s)$ reaction**
  - Using empty run as **background training data**
  - Used simulated $\gamma n(p_s) \rightarrow p\pi^-(p_s)$ reaction as **signal training data**
  - Train the trees to develop algorithm for classification
  - Check for overfitting
  - Employed the trained trees for classifying signal, and background in g14 gold2 target run period
- **Compare the two methods**

Background subtraction:

z-component of interaction vertex



- Background (BG) comes mainly from **Al wires inside the target and KelF target cell.**
- Empty target runs to obtain BG distribution.

Steps:

1. **Apply cuts to clean up gold2 target data.**

→Missing mass, missing momentum, coplanary angle, and target dimension cuts.

1. Run the same analysis on empty target data.
2. Normalize the IBC flux with full target data and obtain the scaling factor.
3. Subtract scaled BG (from empty runs) to align yield ($Y^{3/2}$) and anti align yield ($Y^{1/2}$) of full target runs.

Verified for empty run $Y^{1/2} \approx Y^{3/2}$

Y_{BG}=1/2*($Y^{1/2}+Y^{3/2}$)* scaling factor

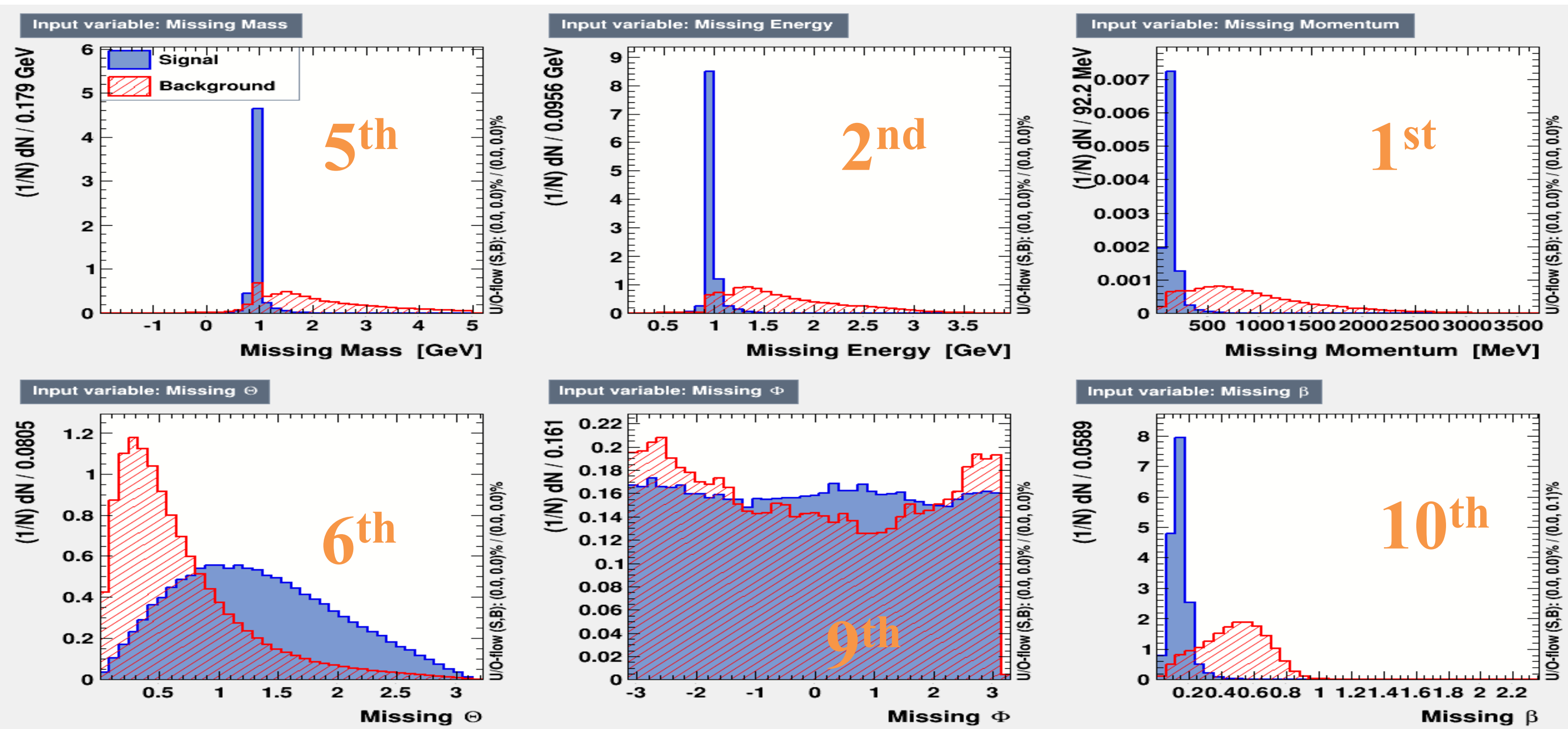
$Y_{HD}^{1/2} = Y_{full}^{1/2} - Y_{BG}$ $\quad$ $Y_{HD}^{3/2} = Y_{full}^{3/2} - Y_{BG}$

$$E = (P_{\gamma} \times P_{target})^{-1} \times ({Y_{HD}}^{1/2} - {Y_{HD}}^{3/2})/({Y_{HD}}^{1/2} + {Y_{HD}}^{3/2})$$

Event selection using BDT:

Procedure: **TRAINING**, CHECKING ,and APPLYING

- Empty target run as background and MC as signal training data sets
- Provide 11 variables for the BDT to use to develop classification algorithm



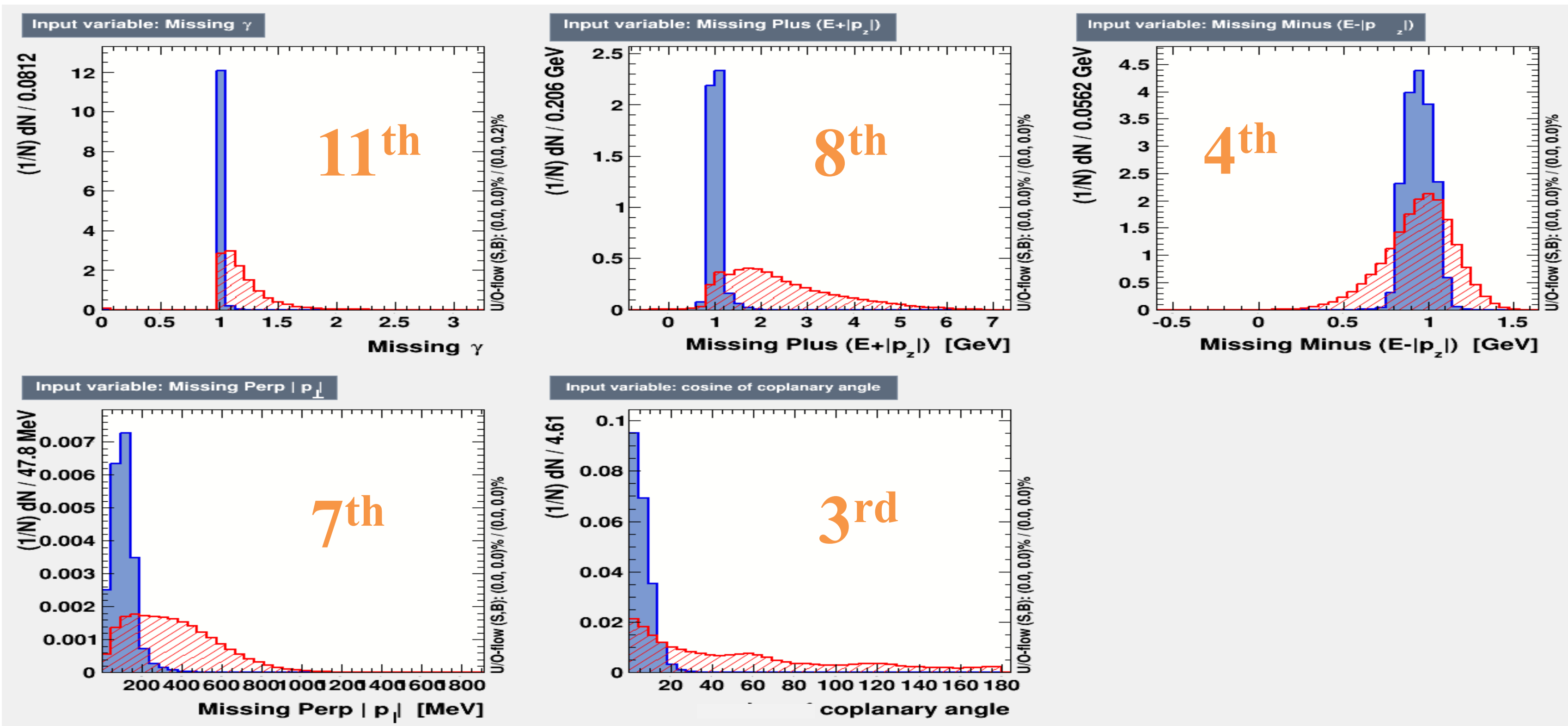
Signal and **background** training data

Ranking (how important) of variables for this classification task

Event selection using BDT:

Procedure: **TRAINING**, CHECKING ,and APPLYING

- Empty target run as background and MC as signal training data sets
- Provide 11 variables for the BDT to use to develop classification algorithm



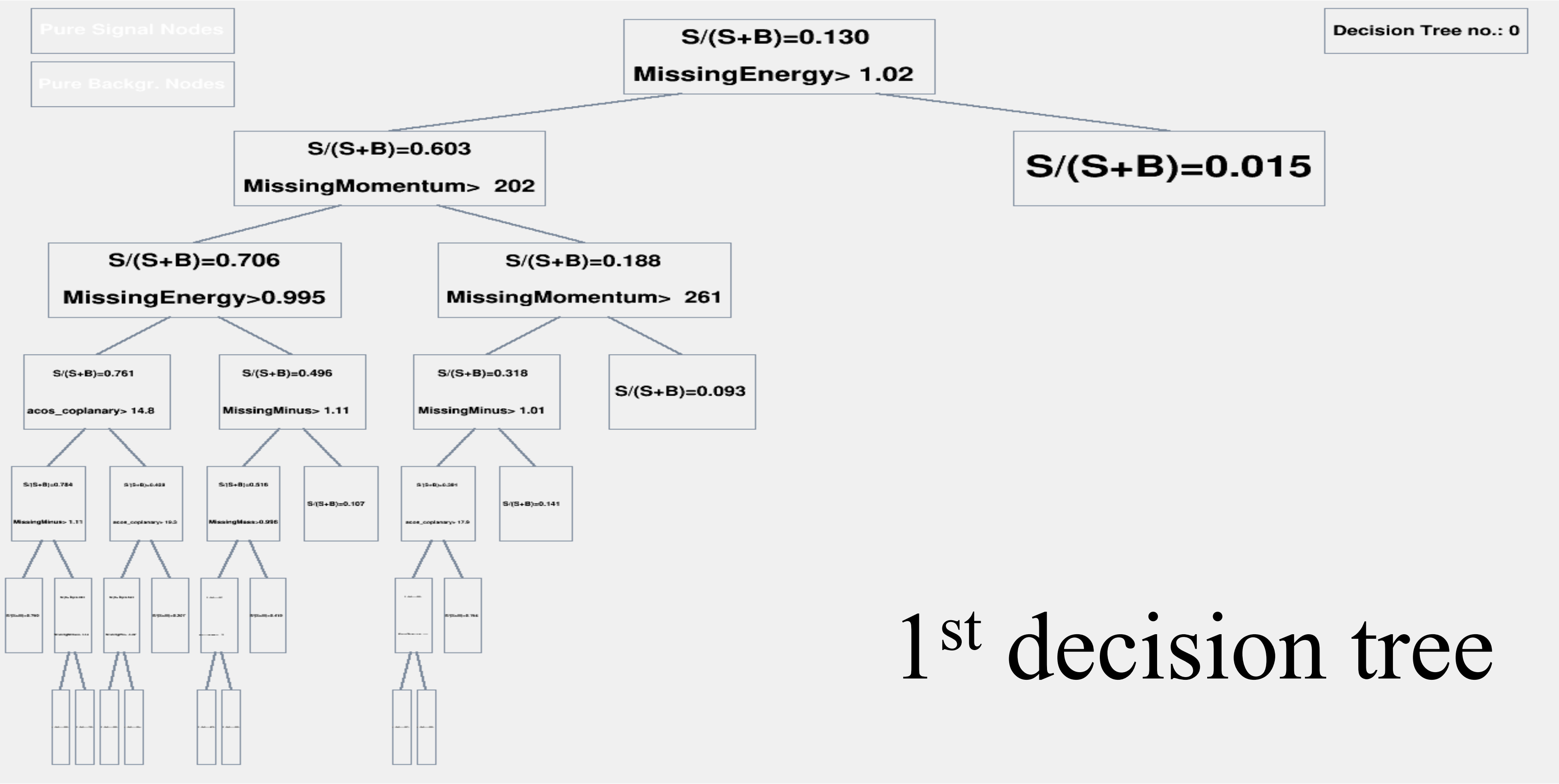
Signal and **background** training data

Ranking (how important) of variables for this classification task

Event selection using BDT:

Procedure: **TRAINING**, CHECKING ,and APPLYING

- Forest of trees (500 trees) developed algorithm to recognize differences between two classes of data (signal/background)

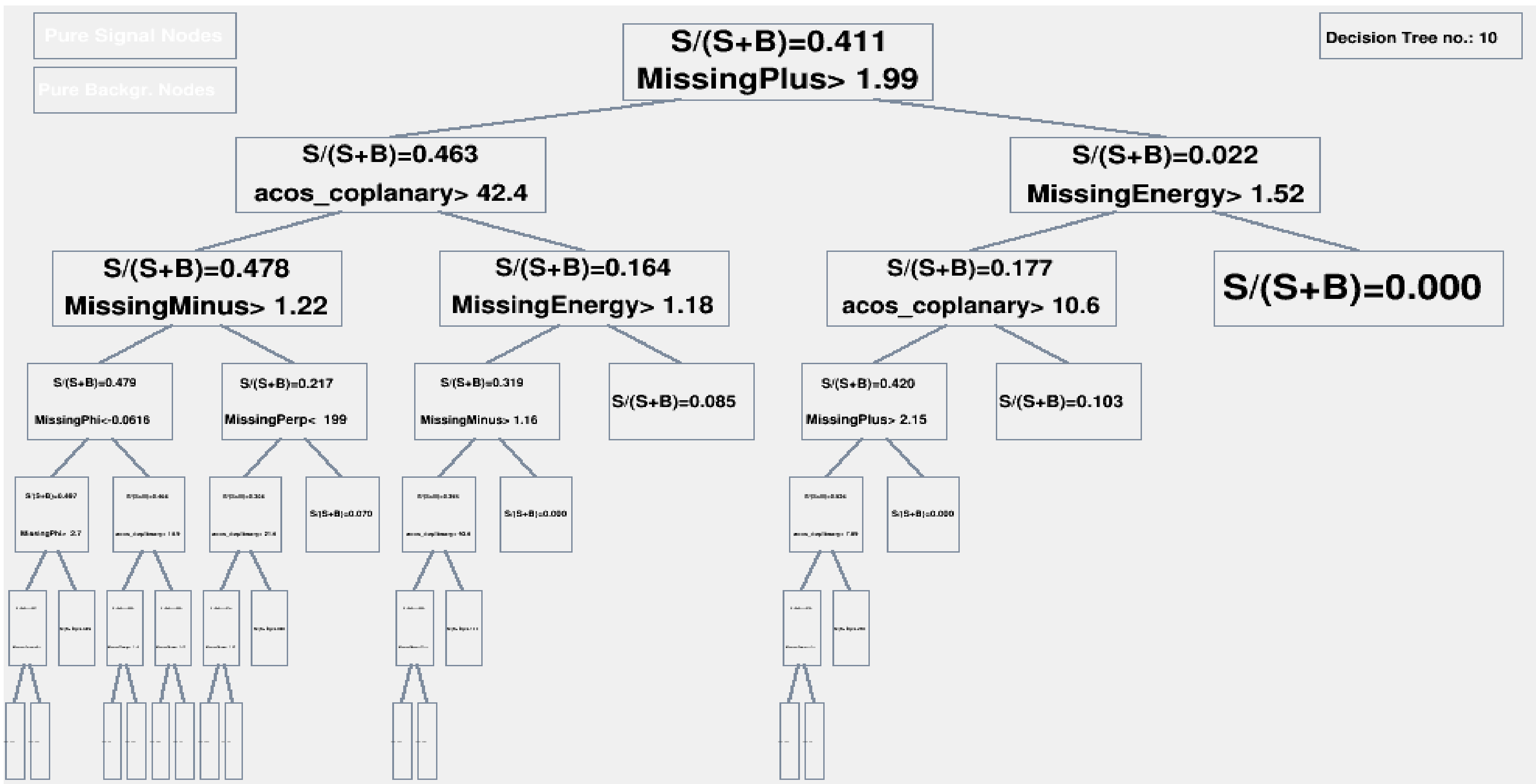


1st decision tree

Event selection using BDT:

Procedure: **TRAINING**, CHECKING ,and APPLYING

- Forest of trees (500 trees) developed algorithm to recognize differences between two classes of data (signal/background)

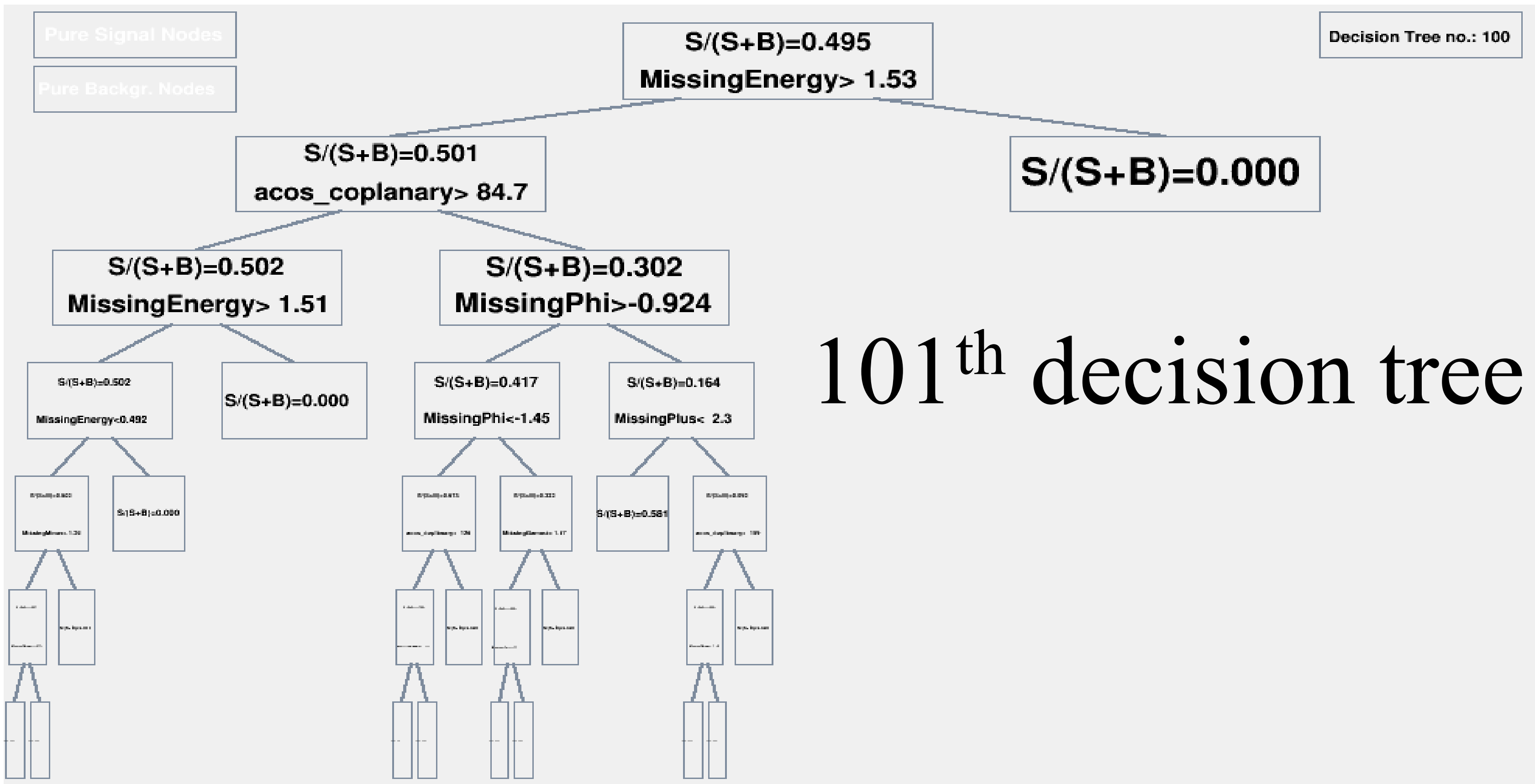


11th decision tree

Event selection using BDT:

Procedure: **TRAINING**, CHECKING ,and APPLYING

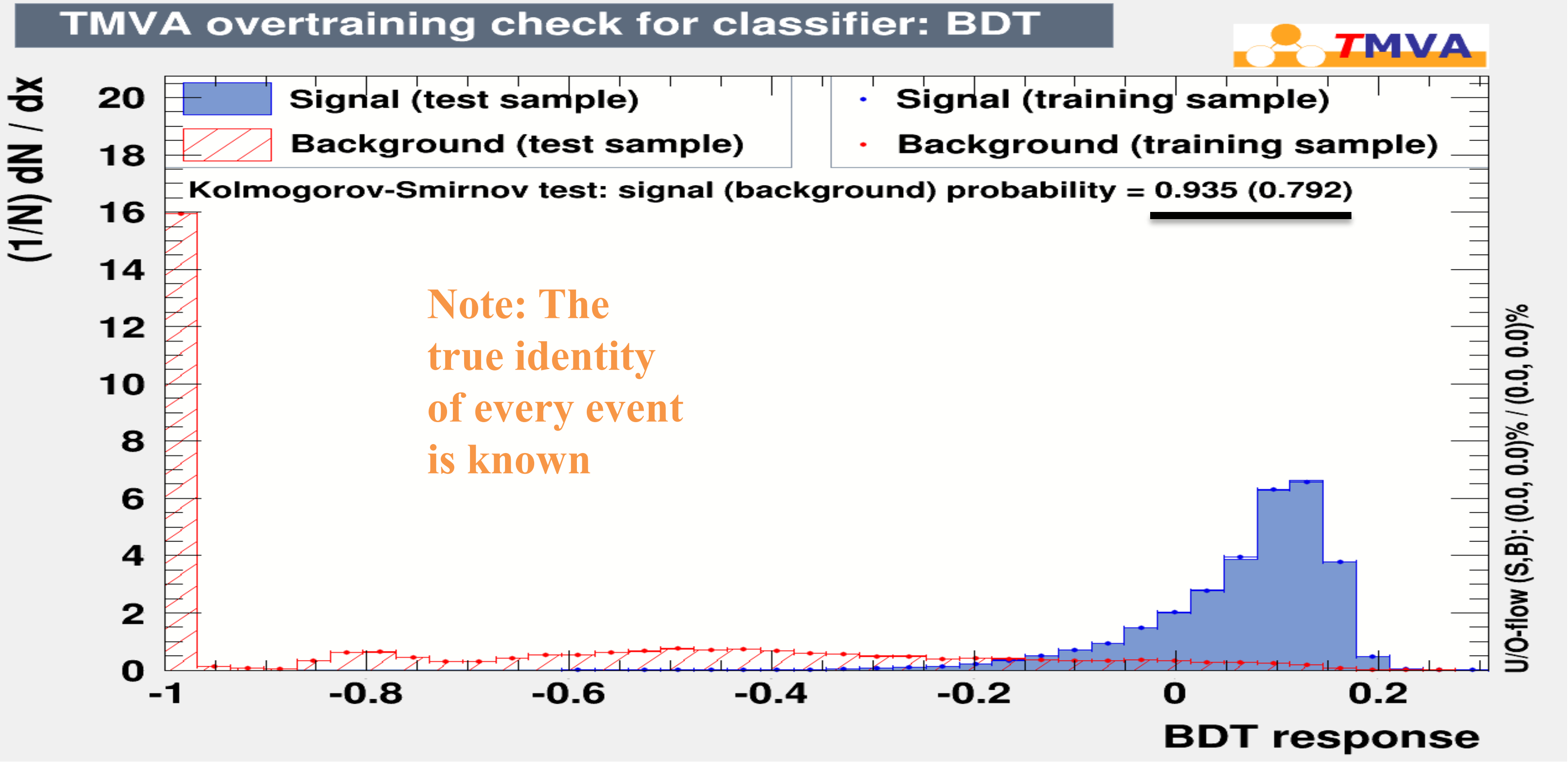
- Forest of trees (500 trees) developed algorithm to recognize differences between two classes of data (signal/background)



Event selection using BDT:

Procedure: TRAINING, **CHECKING** ,and APPLYING

- Each event is given an output between -1 to 1.
- Closer to -1: more likely background, closer to 1: more likely signal
- Performance on test sample is similar on training sample →GOOD



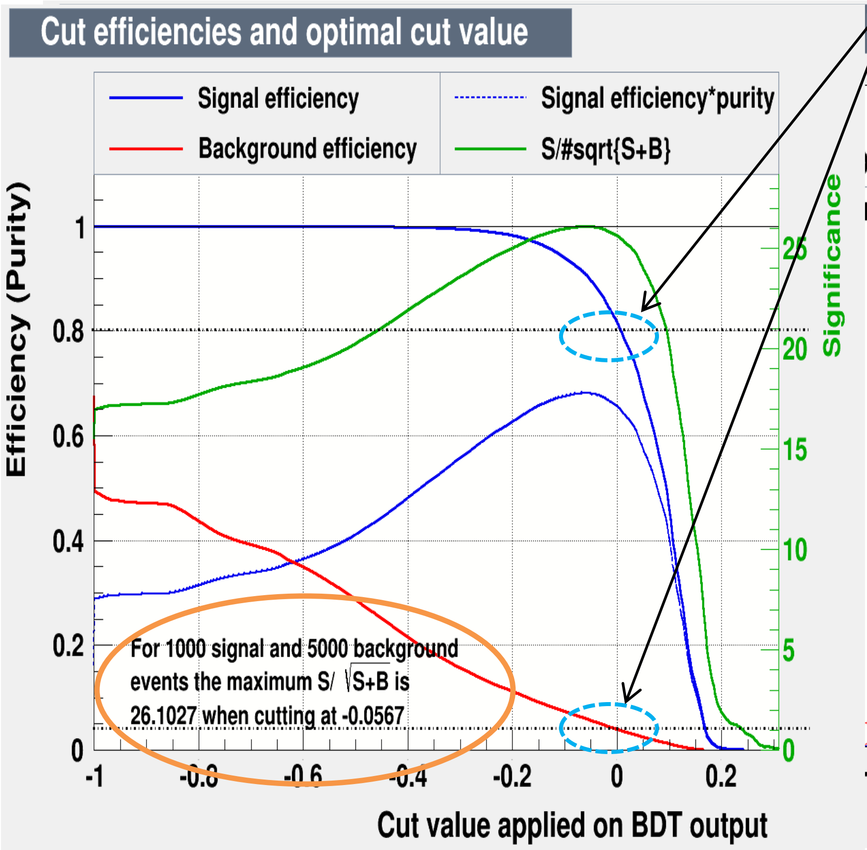
Event selection using BDT:

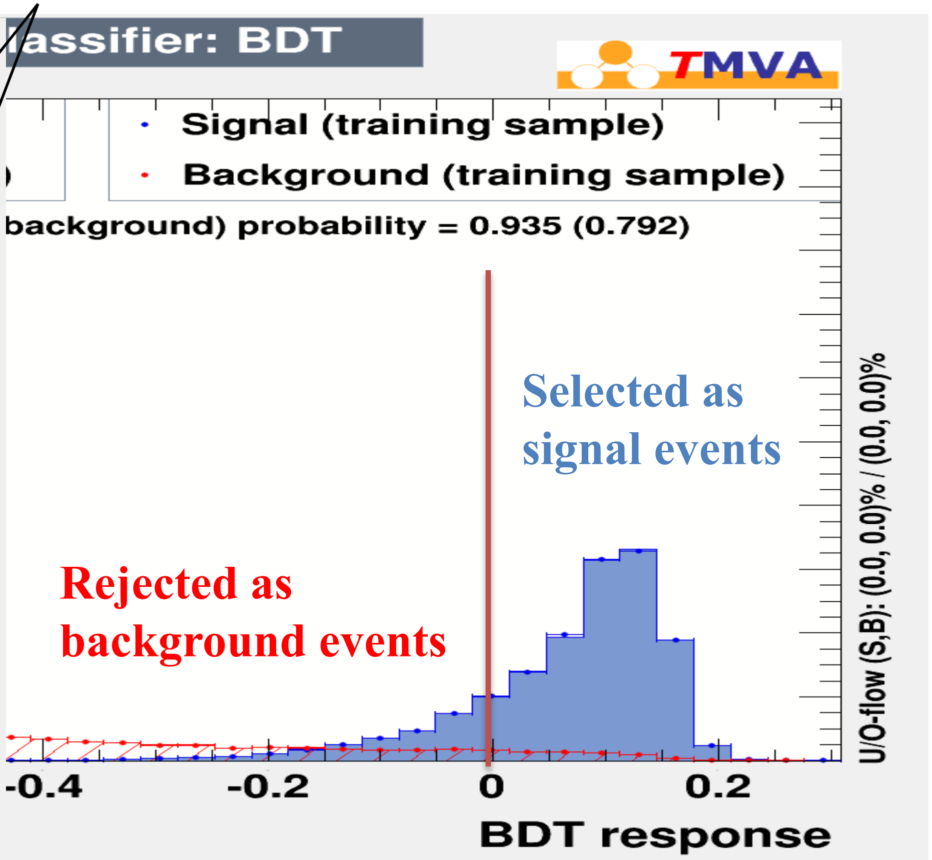
Procedure: TRAINING, CHECKING ,and APPLYING

- Each event is given an output between -1 to 1; **A cut at zero is chosen**

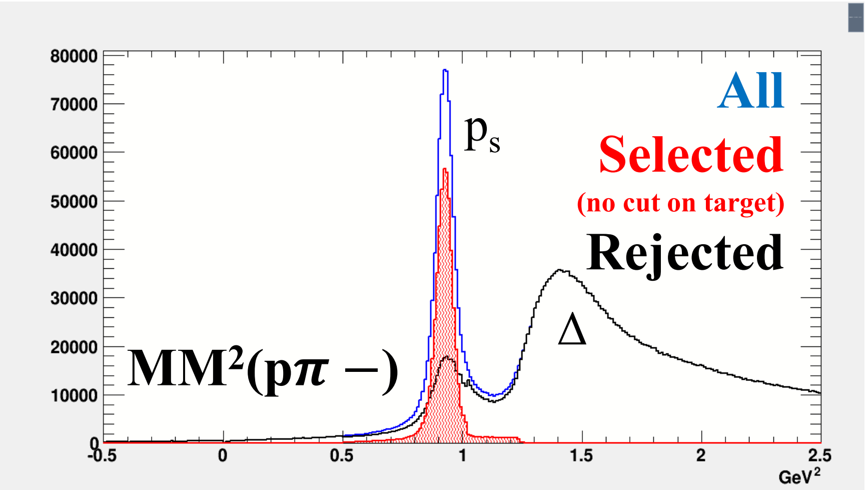
→**4% true background** from BG data **classified as signal (being selected)**

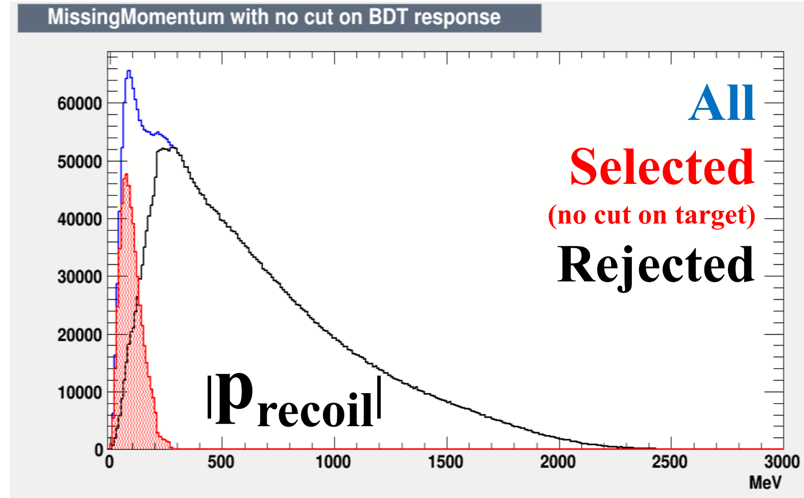
→**80% true signal** from signal data **survived the cut at zero**

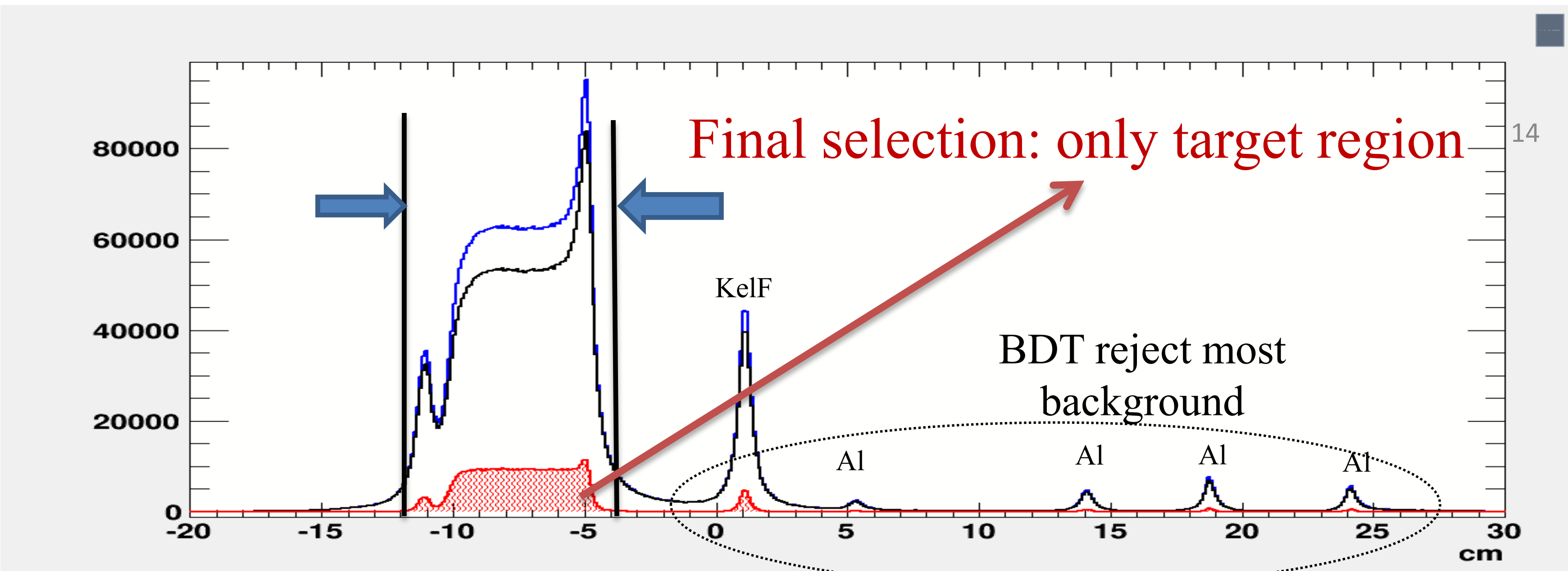




Event selection using BDT: APPLYING







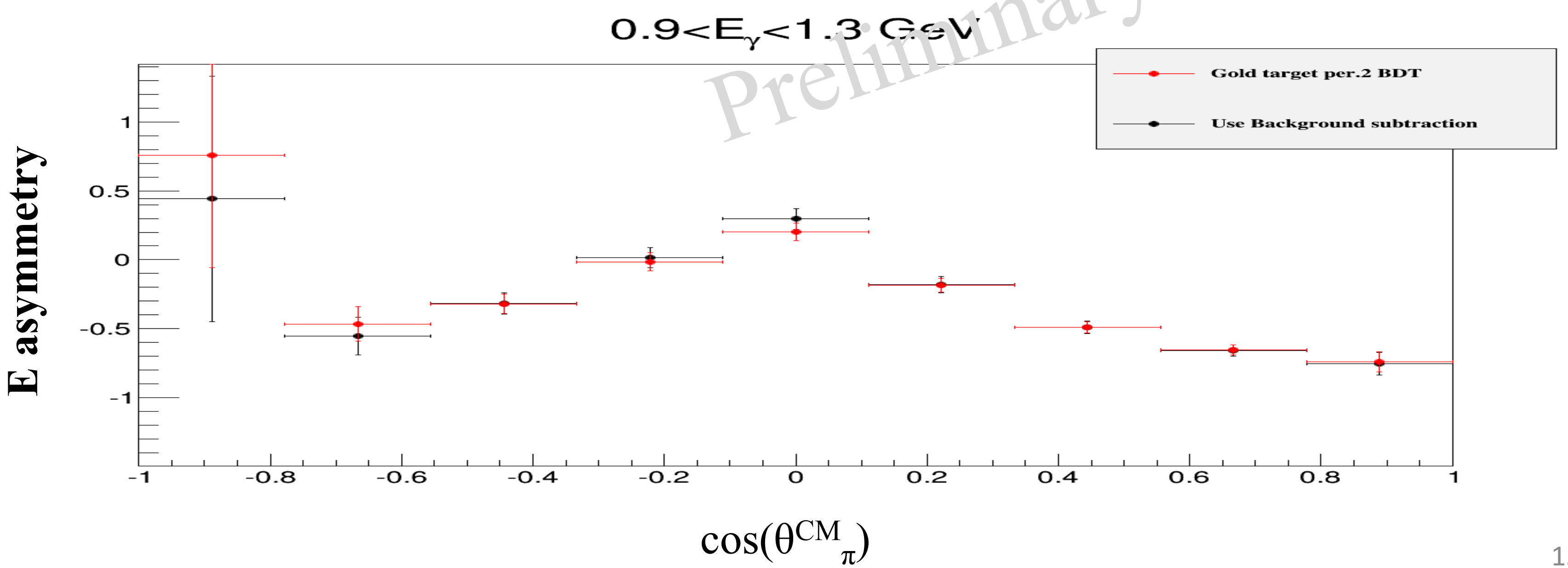
z-component of interaction vertex; this variable NOT used in training the BDT

Background subtraction vs. BDT:

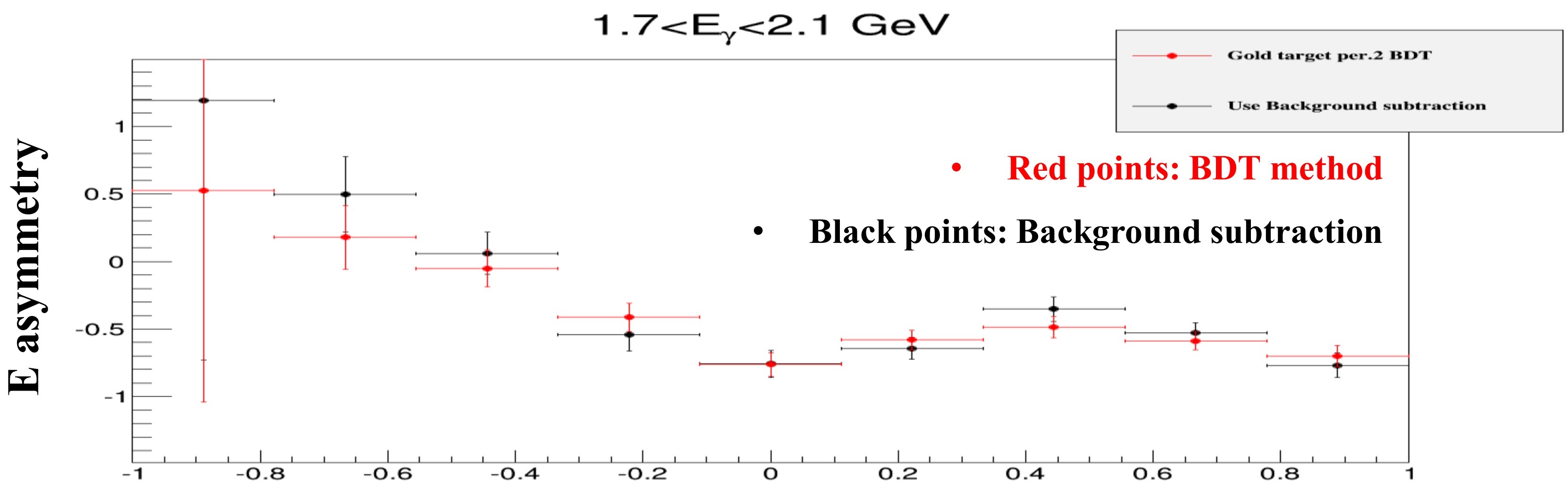
Eγ (GeV)	Total Events	Estimated BG
0.9-1.3	194,528	27,856
0.9-1.3	347,293	32,993
1.3-1.7	73,959	10,126
1.3-1.7	134,476	12,775
1.7-2.1	25,149	3,472
1.7-2.1	44,960	4,271

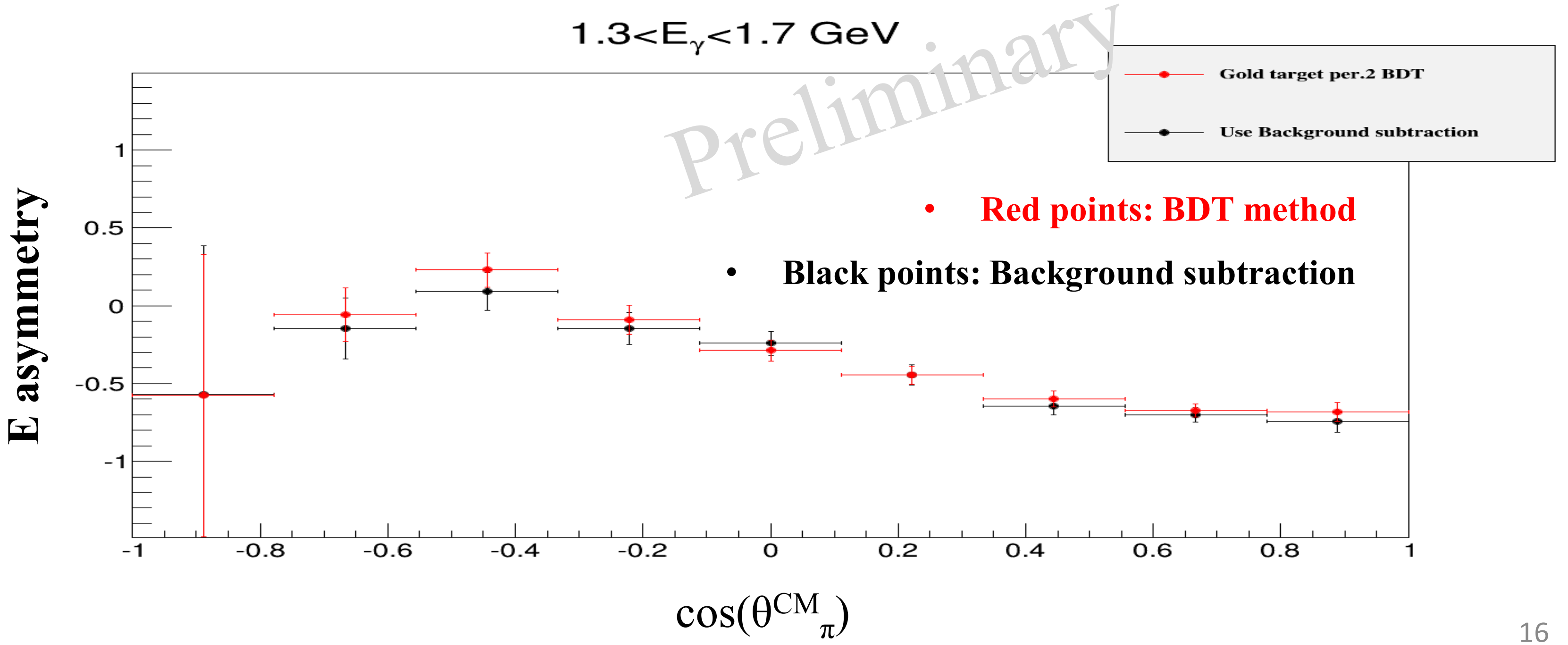
- Using gold2 period data for this check
- **Red points: BDT**
- **Black points: Background subtraction**
- **BDT total events≈1.80 cut method total**
- **BDT remained BG ≈1.18 cut method BG**

→A big gain in data with a small increase in remaining background

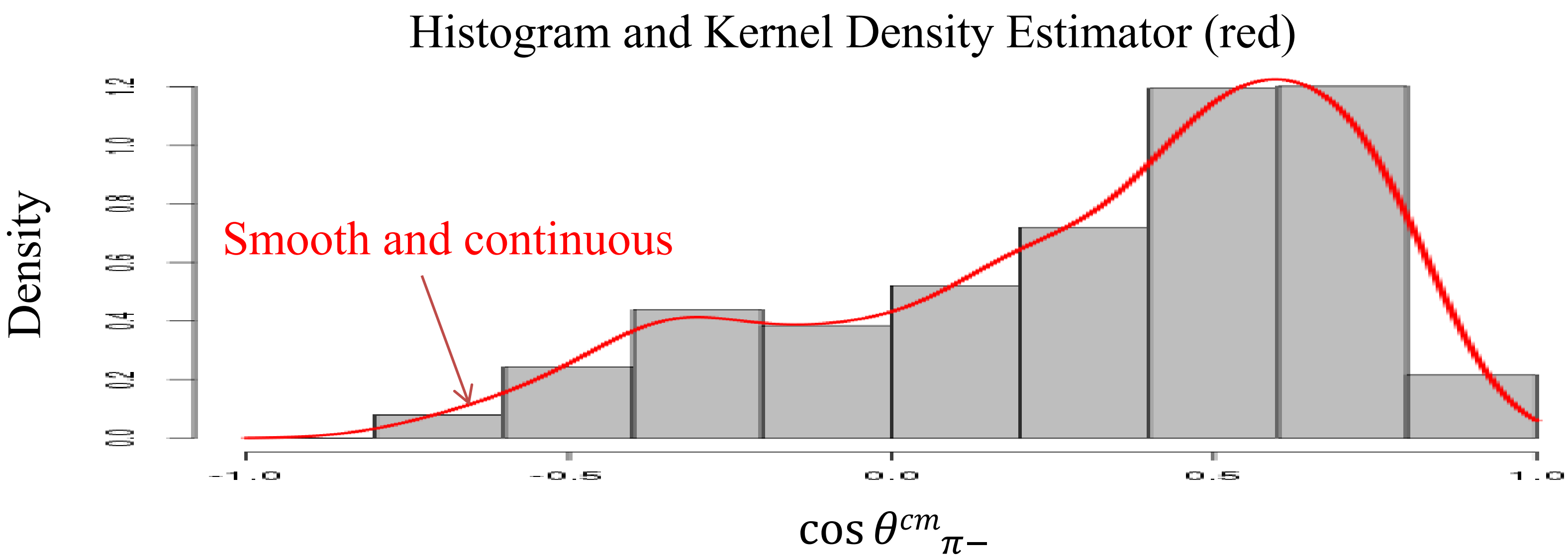


Background subtraction vs. BDT:





Kernel Density Estimator (KDE) introduction:



- KDE is a **non-parametric method** to estimate a probability density distribution (others is the histogram estimator)[2]
- Every data point X_i is "**smeared**" by a density function with mean at X_i (for example, Gaussian density function)
- KDE then "**sums up**" these distributions (one at each data point X_i) to estimate the underlying distribution that the X_is were sampled from

[2] Density Estimation for statistics and data analysis by B.W. Silverman

Mathematical Formula for Kernel Density Estimator (KDE):

Example: Given 15 sample points below, divided into 3 bins (see figure); $X_i \in (0.0, 6.0), i \in \{1,2,3 \dots, 14,15\}$

Sample Data Points (X_i)	
0.73	0.20
0.51	1.24
1.90	1.05
1.48	2.50
3.70	3.30
4.80	5.00
4.50	4.20
5.80	

$\boldsymbol{\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)}$ where

- X_i is the sample data points,
- h is the bandwidth (smoothing parameter),
- *K(x)* can be any symmetric density function.
- *Often, K(x)*$=\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x)^2}$, the normal distribution

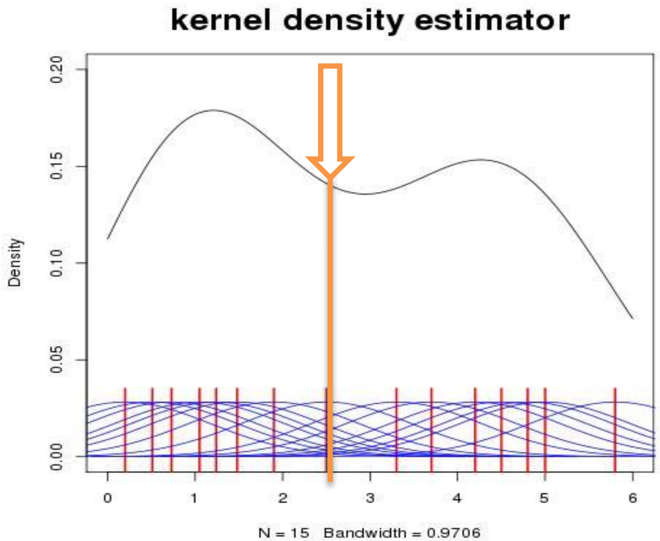
Ex:

$$\hat{f}_h(2.5) = \frac{1}{15h}\sum_{i=1}^{15}\frac{1}{\sqrt{2\pi}} exp\left(-\frac{(2.5-X_i)^2}{2h^2}\right)$$

- $\hat{f}_d(x)$ is estimated using all data points (all X_is).
- *K(x)* is smooth, so $\hat{f}_d(x)$ is smooth (continuous).

How to pick a good value for h→→ →back up slide

Mathematical Formula for Kernel Density Estimator (KDE):



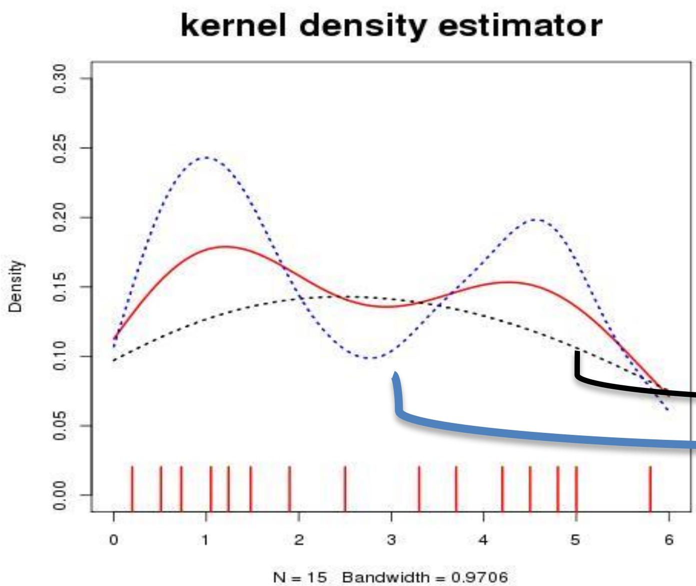
$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right)$ where

- X_i is the sample data points,
- h is the bandwidth (smoothing parameter),
- *K(x)* can be any symmetric density function.
- *Often, K(x)*$=\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(x)^2}$, the normal distribution

Ex:

$$\hat{f}_h(2.5) = \frac{1}{15h}\sum_{i=1}^{15}\frac{1}{\sqrt{2\pi}}exp\left(-\frac{(2.5-X_i)^2}{2h^2}\right)$$

- $\hat{f}_d(x)$ is estimated using all data points (all X_is).
- *K(x)* is smooth, so $\hat{f}_d(x)$ is smooth (continuous).
- h is strongly influenced on the resulting estimate,

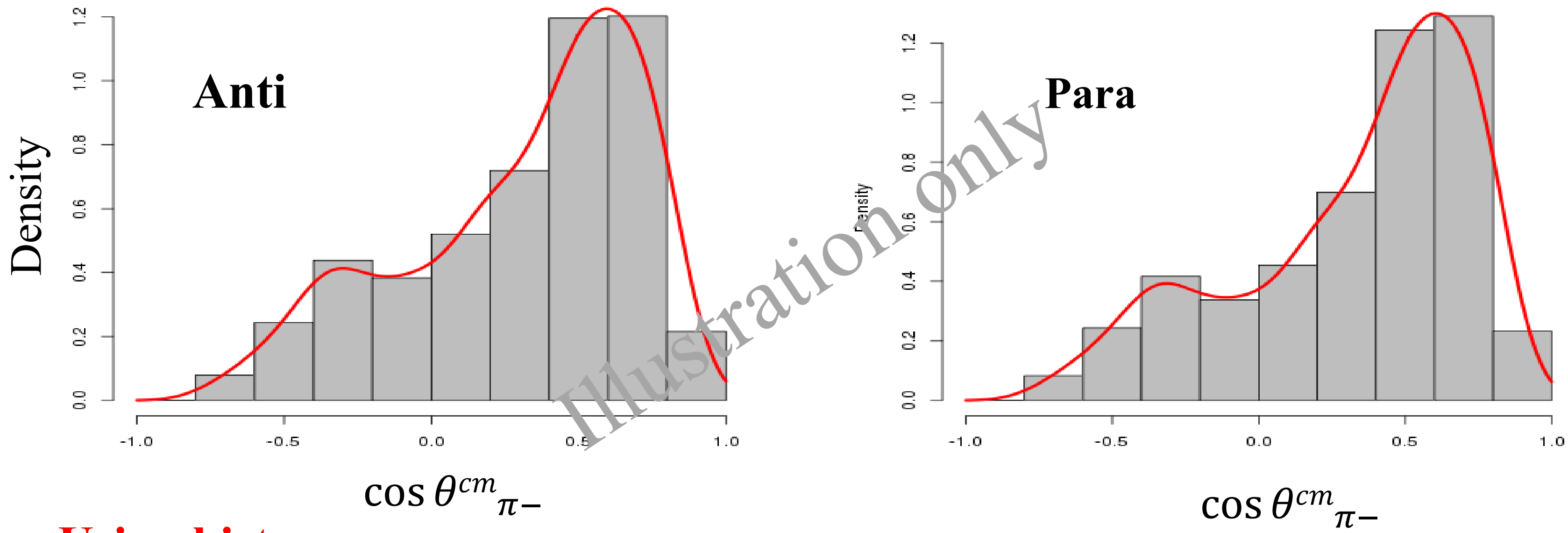


too large h obscures underlying structure

too small h results in many fluctuations

Using KDE to “plot” E asymmetry:

Histogram and KDE for gold2, left is anti ($Y^{1/2}$), right is para ($Y^{3/2}$)



Using histogram

$E(x_{B_j}) = \dfrac{1}{pol}\dfrac{n_{B_j}{}^{\downarrow\uparrow} - n_{B_j}{}^{\uparrow\uparrow}}{n_{B_j}{}^{\downarrow\uparrow} + n_{B_j}{}^{\uparrow\uparrow}}$ where $n_{B_j}{}^{\uparrow\uparrow}$ = # events in bin B_j for para data.

Using KDE

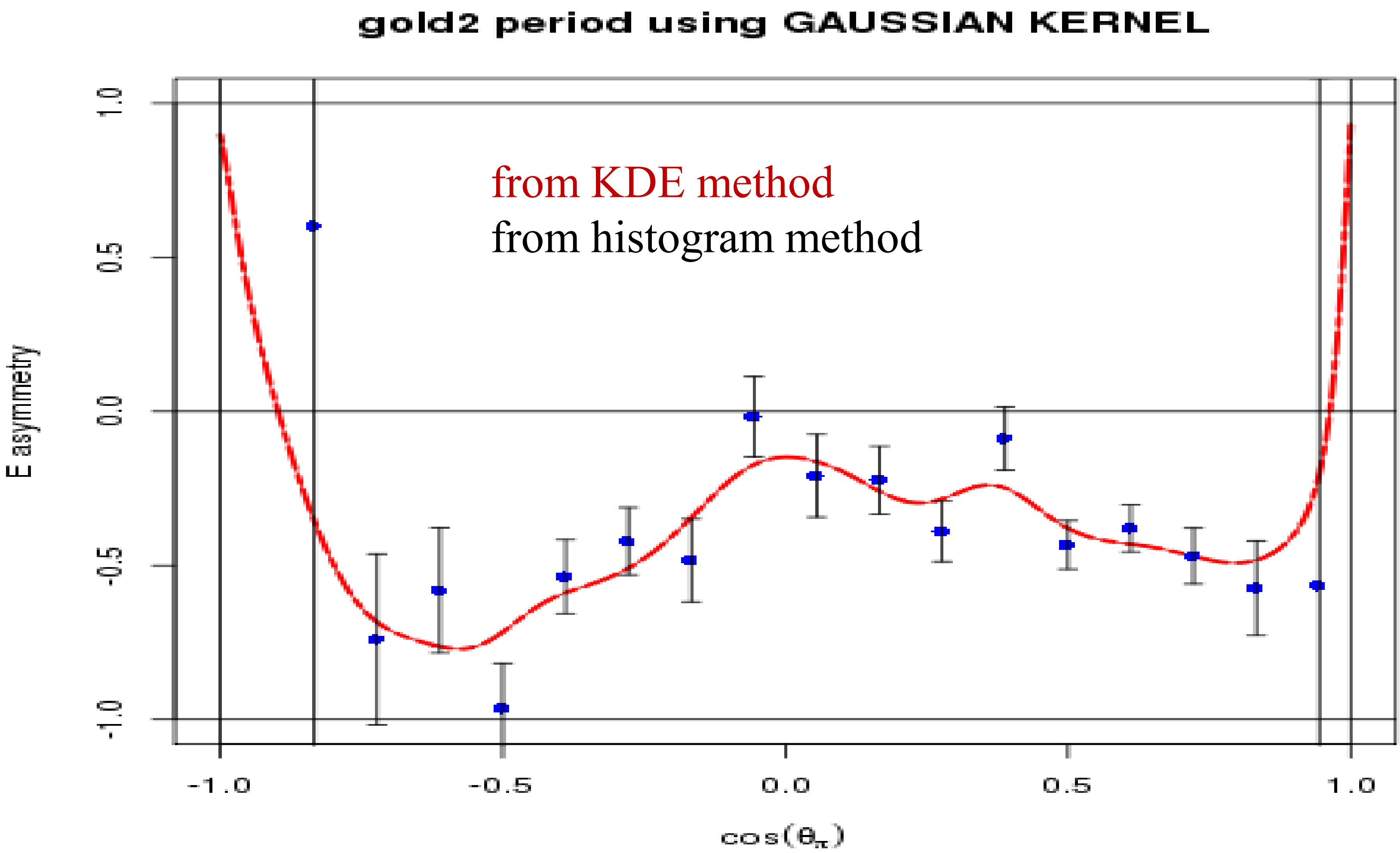
$$E(x) = \frac{1}{pol}\frac{n^{\downarrow\uparrow}\hat{f}(x)_{\downarrow\uparrow} - n^{\uparrow\uparrow}\hat{f}(x)_{\uparrow\uparrow}}{n^{\downarrow\uparrow}\hat{f}(x)_{\downarrow\uparrow} + n^{\uparrow\uparrow}\hat{f}(x)_{\uparrow\uparrow}}$$

where $n^{\uparrow\uparrow}$ = total # events for para data, and

$\int_{-1}^{1}\widehat{f}(x)_{\downarrow\uparrow}dx = \int_{-1}^{1}\widehat{f}(x)_{\uparrow\uparrow}dx = 1$

Using KDE to "plot" E asymmetry:

0.9 GeV<E_γ<1.0 GeV; gold2 run period



Bootstrap short introduction [3]

Measure value of X n times

Estimate f(X)=mean f(X_i)

Need to estimate uncertainty

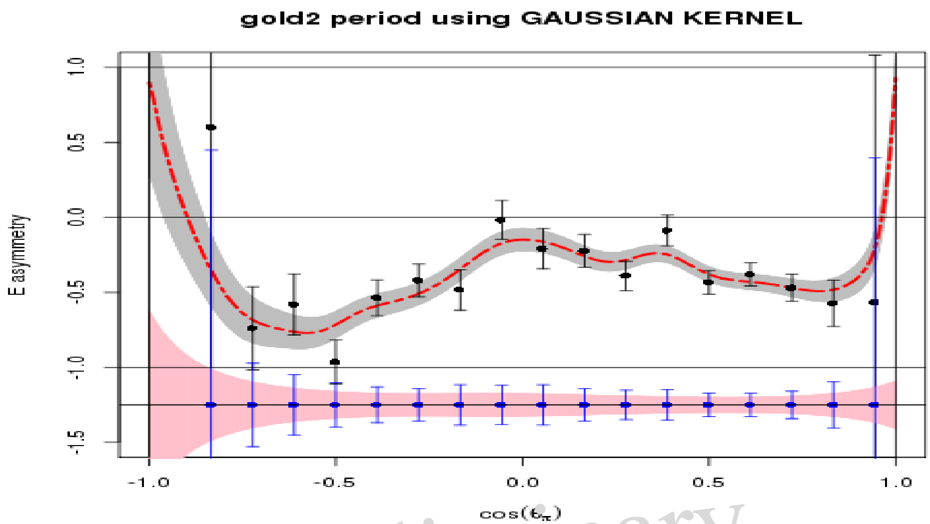
- ***Draw new samples from the true distribution, i.e, repeat the experiment m times.***
- Estimate uncertainty from the m samples

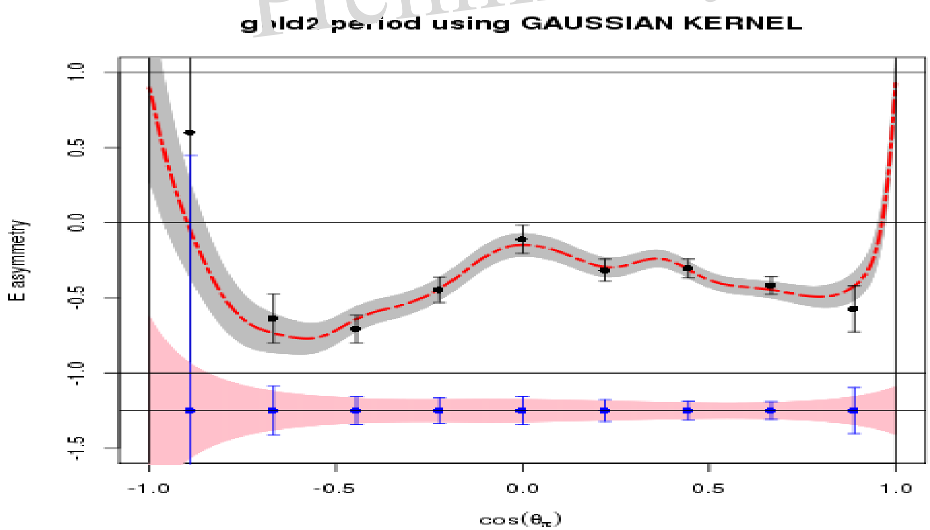
- **CAN'T measure again, then sampling the n X_i with replacement m′ times.**
- ***Draw new samples from an approximate distribution (the data obtained)***
- Estimate uncertainty by from the m′ samples

→this is the bootstrap method

[3] Probability and Statistics by DeGroot- Schervish

Plotting E asymmetry





0.9 GeV<E_γ<1.0 GeV; gold2 run period

- E curve from KDE method **agrees** well with histogram method for two different sets of binning.
- Histogram: compromise between a more precise measurement (small uncertainty) or a better representation of data (more bins).
- KDE does NOT have that issue, **better choice for low statistics data**, still has **bias near boundaries**.
- KDE makes plotting a **continuous error**, or **confidence level band** possible.

Future Works:

- Study and improve MC simulation for g14 run
- Use BDT for $K^0\Lambda$ and $K^0\Sigma^0$ selections
- Learn and implement bias reduction techniques for bounded data (cosine near ± 1)
- Learn more advanced KDE methods to estimate low statistic data better
- Apply these new tools to low statistics $K^0\Lambda$ and $K^0\Sigma^0$ data

Back Up

Decision Tree introduction: **training**

Goal: *To improve data "purity" after each node splitting using information entropy*

$$Entropy(n_{sig}, n_{bg}) = -\frac{n_{sig}}{n_{total}} log \frac{n_{sig}}{n_{total}} - \frac{n_{bg}}{n_{total}} log \frac{n_{bg}}{n_{total}}$$

Pick a **variable** V_i and its cut **value** ***v*** from a **set of variables** to **maximize** the following formula:

$$Gain(n_{sig}, n_{bg}, V_i) = Entropy(n_{sig}, n_{bg}) - \sum_{A_1, A_2} \frac{n_{A_j}}{n_{total}} Entropy\,(n^{A_j}{}_{sig}, n^{A_j}{}_{bg})$$

where A_1 and A_2 are two sets of events separated by v (one set with V_i<v, the other with V_i >v),

→→Pick the cut **value** ***v*** of variable V_i such that the percentage of signal or background events after splitting is **larger** than the original percentages.

→→**Repeat** for **all** *Vs* and select one *V* with **maximum gain**

Decision Tree introduction: **training**

- Good: easy to understand, straight forward implementation, and no event removed.

- **Bad: the tree can perfectly classify training data if given enough splittings; overfitting always occurs → highly sensitive to statistical fluctuation.**

- **Solution: limit amount of splittings, and BOOSTing**

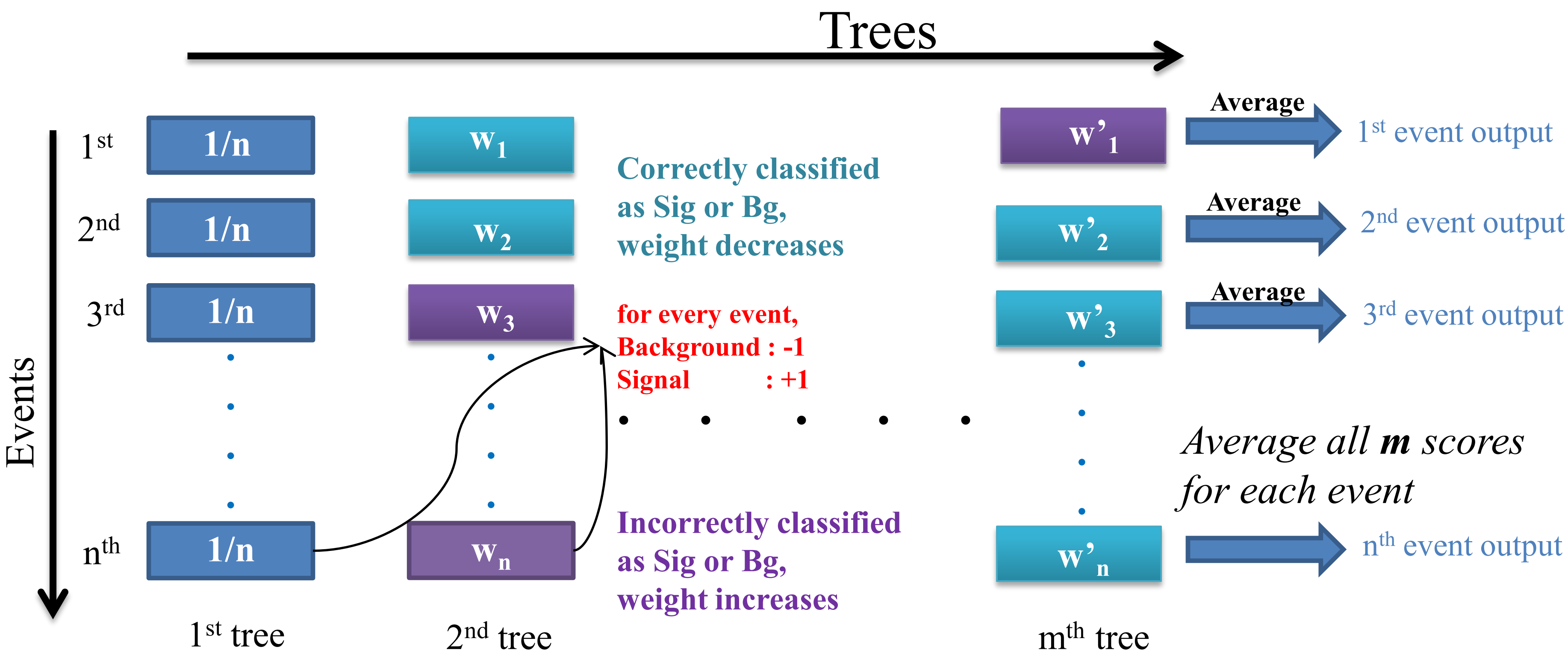
Boosting means using decision tree **multiple times** on **reweighted training data** to improve performance

→ Build a forest of many ***different*** decision trees

BOOSTED Decision Tree introduction:

Boosting: run decision tree **multiple times** on **reweighted training data**

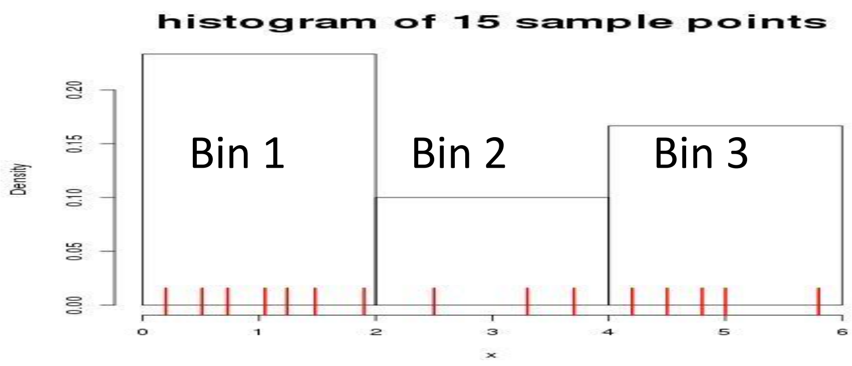
→→ Build a forest of many ***different*** decision trees



- Each new tree is built differently due to having a new set of weights for all events
- Each new tree focuses more on wrongly classified events by previous trees

Mathematical Formula for Histogram Density Estimator:

Example: Given 15 sample points below, divided into 3 bins (see figure); $X_i \in (0.0, 6.0), i \in \{1,2,3 \dots, 14,15\}$



Sample Data Points (X_i)	
0.73	0.20
0.51	1.24
1.90	1.05
1.48	2.50
3.70	3.30
4.80	5.00
4.50	4.20
5.80	

Draw back of histogram method (more severe with low-statistics data):

- **Sensitive to bin width (bias toward data points closed to bin edges).**
- **Use only partial data for estimation. i.e., to estimate f(x) with $x \in B_j$ use only data points $X_i \in B_j$.**
- **$\hat{f}_d(x)$ is a discontinuous function (step function).**

Define: $B_j = (x_0 + (j-1)d,\ x_0 + jd)$

where x_0 is the origin, and d is bin width.

$$\hat{f}_d(x) = \frac{1}{nd} \sum_{i=1}^{n} \sum_{j=1}^{\#bins} 1_{(X_i \in Bj)} 1_{(x \in Bj)}$$

where X_i is the sample data points. $\hat{f}_d(x)$ is constant for all x in a same bin.

X_i : recorded sample data points; x: estimating points

Using Least Square Cross Validation to pick h:

Consider:

$\hat{f}_h(x) = \sum_i^n \frac{1}{nh} K\left(\frac{x-X_i}{h}\right), f(x) \equiv$ true density function

Define:

$$L_h = \int dx(\hat{f}_h(x) - f(x))^2 = \int dx(\hat{f}_h(x))^2 - 2\int dx(\hat{f}_h(x) f(x)) + \int dx(f(x))^2 \geq 0$$

$$\rightarrow \tilde{L}_h = \int dx(\hat{f}_h(x))^2 - 2\int dx(\hat{f}_h(x) f(x)) \geq -\int dx(f(x))^2 = constant$$

exact approximate

$$\rightarrow \tilde{L}_h \approx \sum_i^n \sum_j^n \frac{1}{(nh)^2} \int K\left(\frac{x-X_i}{h}\right) K\left(\frac{x-X_j}{h}\right) dx - \frac{2}{n-1} \sum_{k\neq l}^n \sum_l^n \frac{1}{nh} K\left(\frac{X_k - X_l}{h}\right) \geq -const\ (*)$$

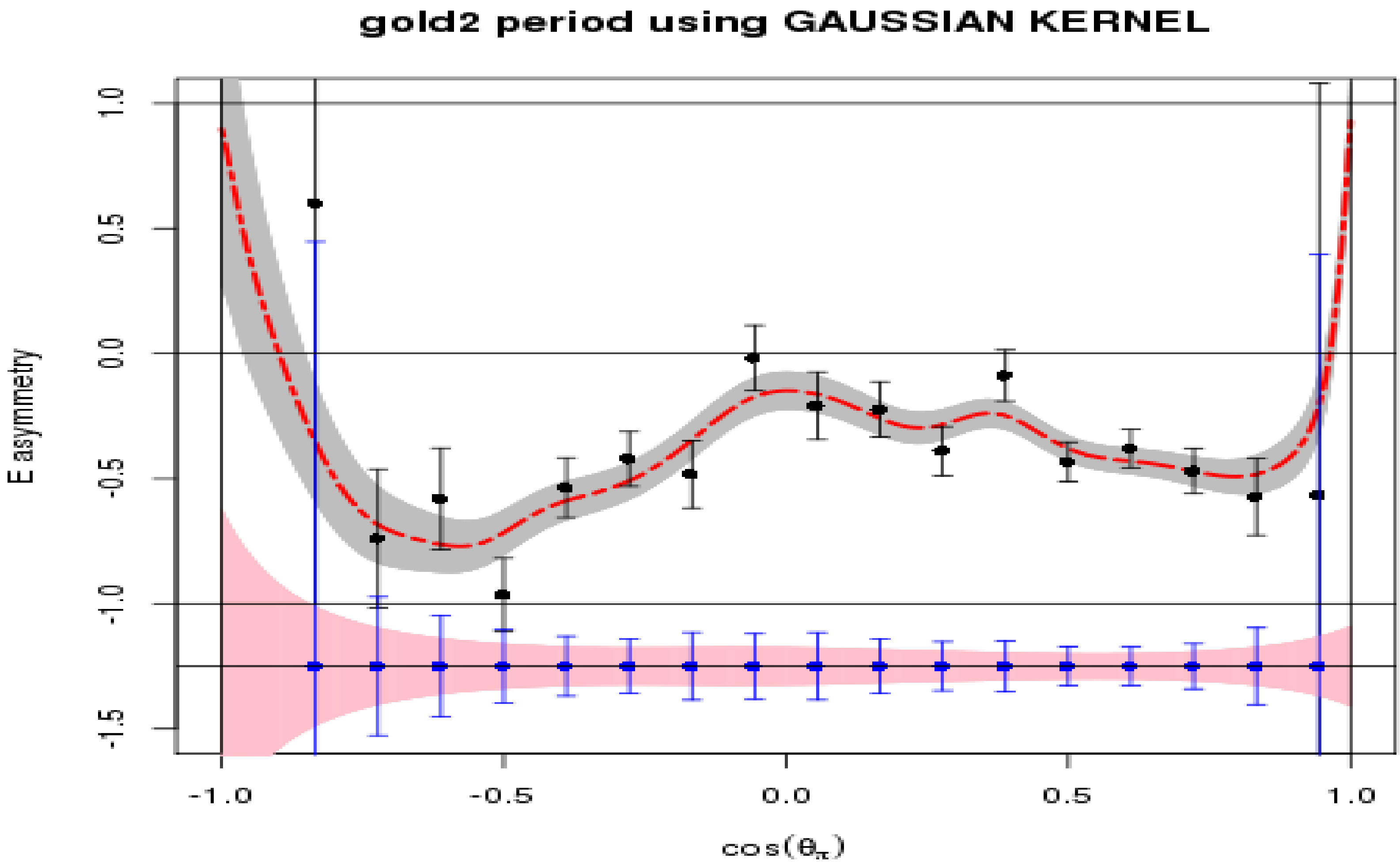
→Find h which minimizes (*)

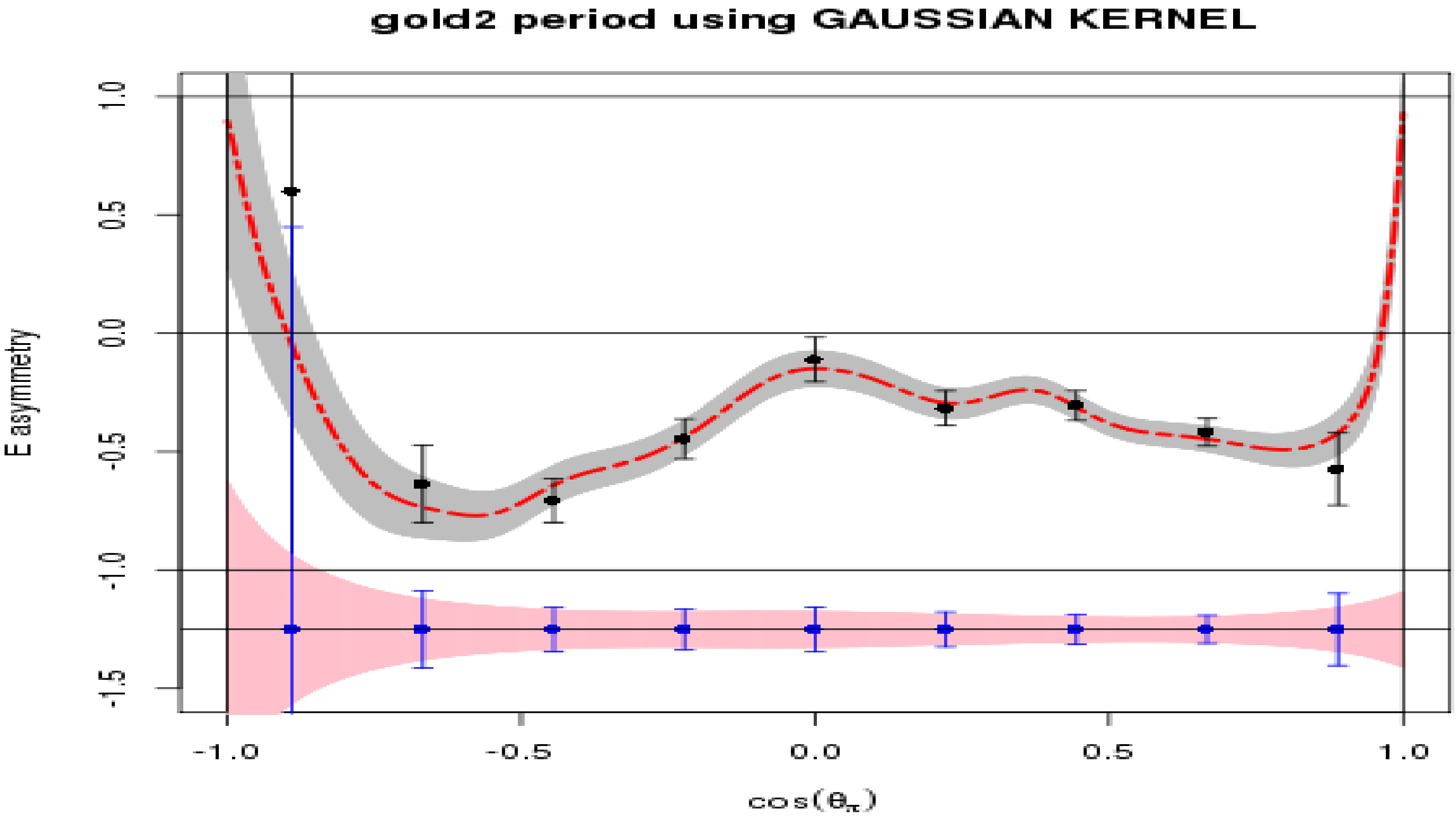
Advantage : non-parametric method to estimate the smoothing parameter h

Disadvantage: computing intensive n^2 operations

if there are many pairs of X_i, X_j with $X_i \approx X_j$, then $\tilde{L}_h \rightarrow -\infty, as\ h \rightarrow 0$;

tends to pick small h (undersmooth)


gold2 period using GAUSSIAN KERNEL
E asymmetry
1.0
0.5
0.0
-0.5
-1.0
-1.5
-1.0
-0.5
0.0
0.5
1.0
cos(θπ)


gold2 period using GAUSSIAN KERNEL
E asymmetry
1.0
0.5
0.0
-0.5
-1.0
-1.5
-1.0
-0.5
0.0
0.5
1.0
cos(θπ)