Machine learning approach to analyze cosmics data

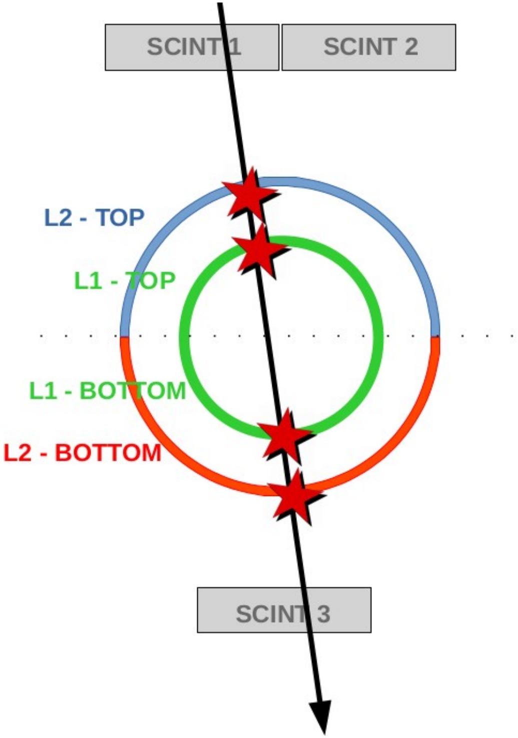
Lia Lavezzi

Università degli Studi di Torino / INFN–TO

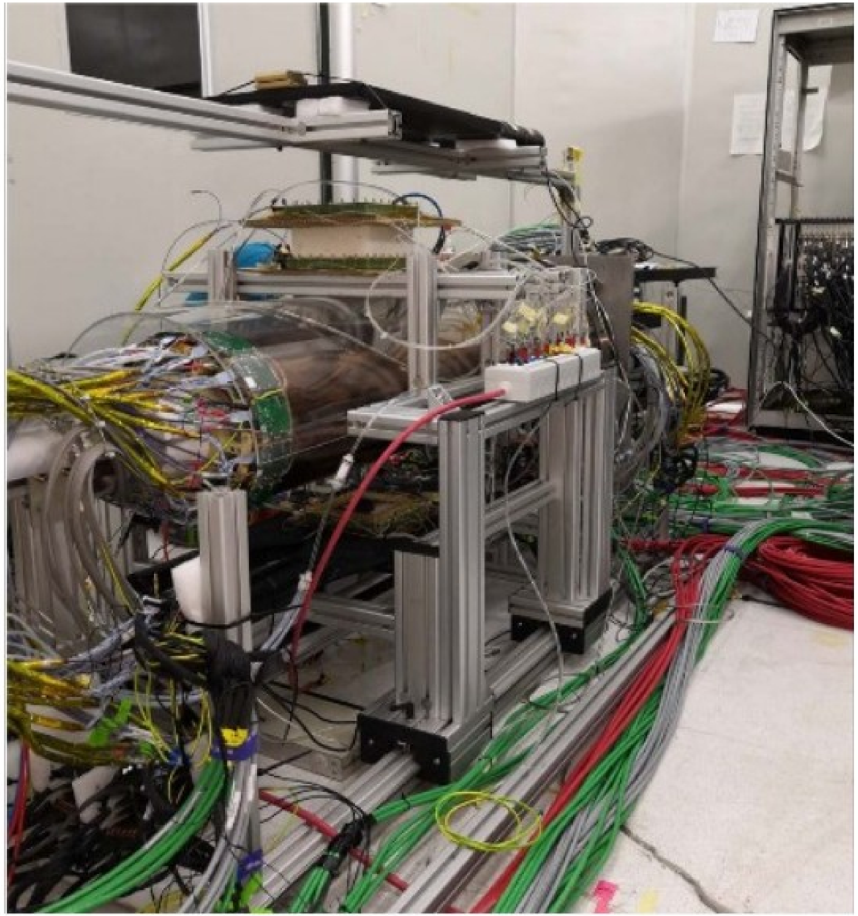
BESIII Italy Collaboration Meeting – Torino, 28/29 September 2022

Good old cosmic ray data

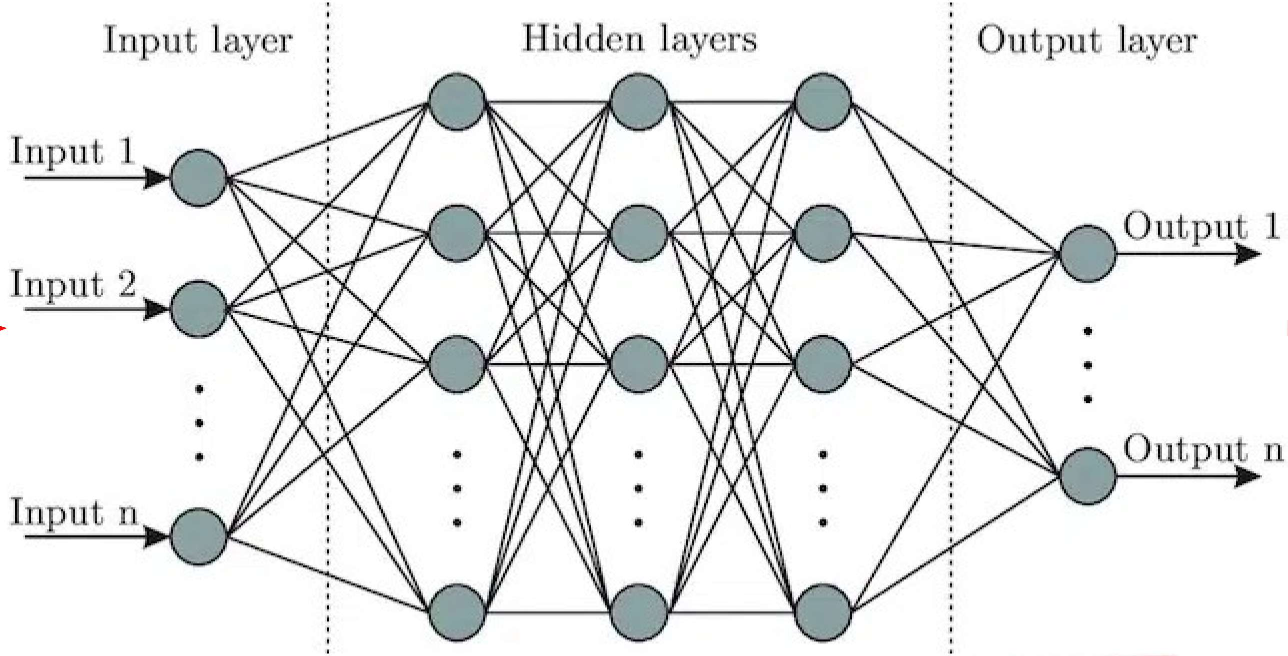
- Machine Learning to separate signal from noise at hit/cluster level
- Ongoing thesis @ Ferrara



SETUP - LAYER1 + LAYER2



Machine learning



Measured quantities:

- hit/cluster charge
- signal start time, length …
- charge sharing between x/v
- …

Classification:

- Signal
- Noise

Signal and Noise

The network must be **trained** to **learn** from **true** and **known** events

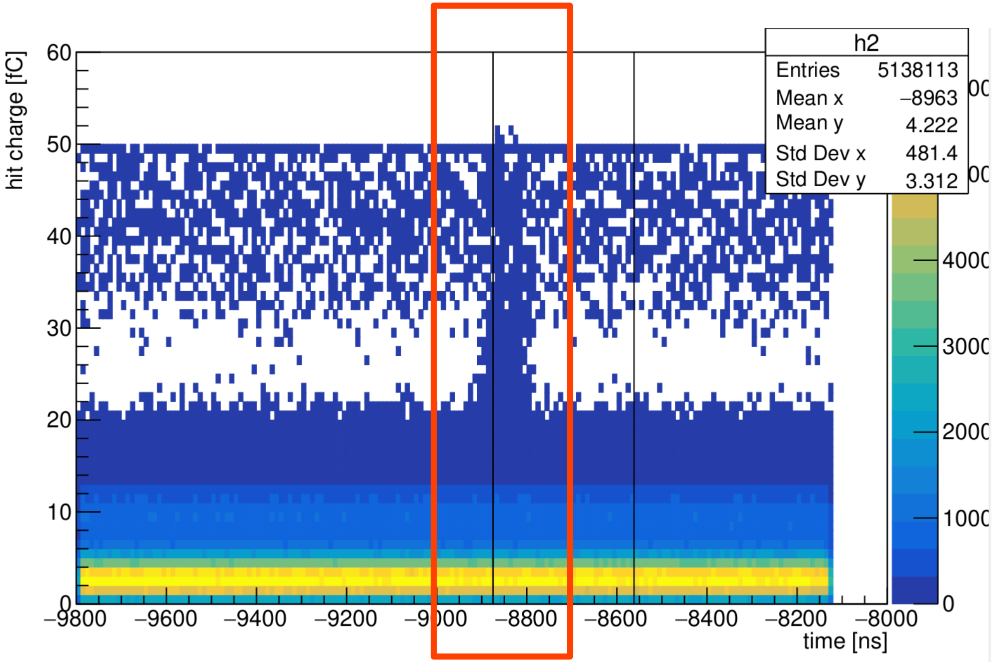
- **Signal:** golden run, no. 17
  Event selection with:
  - cgemboss.6.6.5.g
  - alignment: fixed
  - 4 plane fit → all 4 planes firing
  - no more than 100 cluster–2D
  - no chi2 cut
  - 129357 events

- **Noise:** run 44–49
  - hit in trigger time window
  - 98363 events

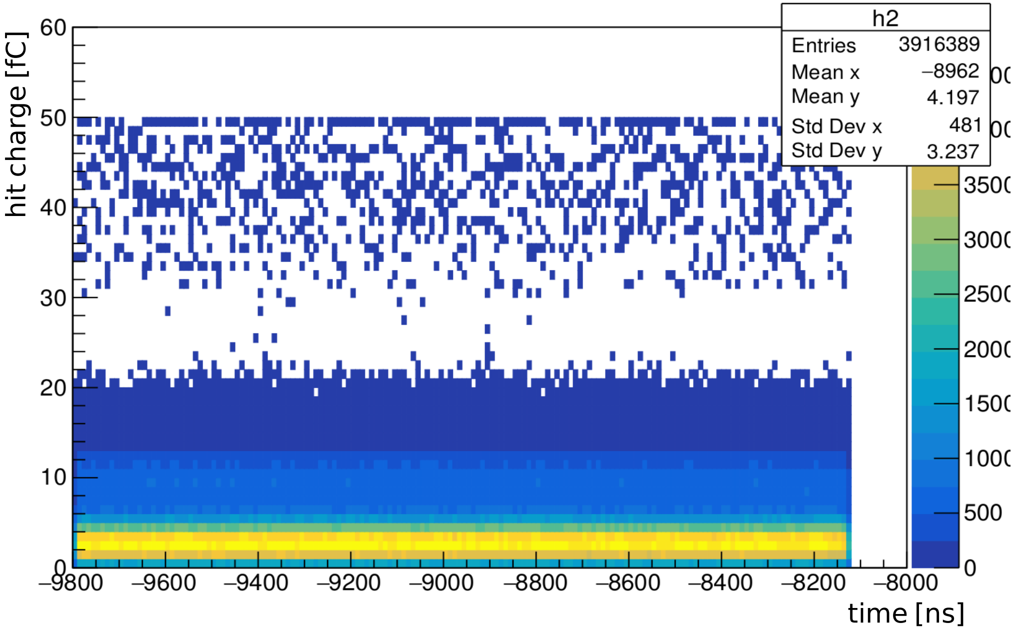
RUN ID				
AQUIRED	SHARED	GAIN	THRSH	# EVENTS
351	17	std	std	234154
421	44	off (HV=0)	std	18853
422	45	off (HV=0)	std	3021
423	46	off (HV=0)	std	21168
424	47	off (HV=0)	std	24288
426	48	off (HV=0)	std	14469
428	49	off (HV=0)	std	16564

Noise cleaning

Run **44** – **49**: hit charge *vs* hit time shows a **structure** close to the event window



Without run **45** – **46** the structure disappears → exclude them



Noise cleaning

The network must be **trained** to **learn** from **true** and **known** events

- **Signal:** golden run, no. 17

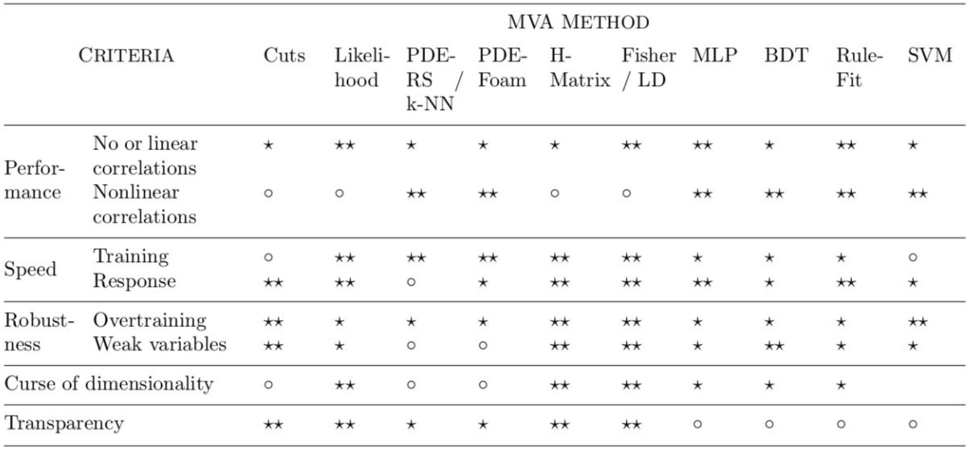
  Event selection with:
  - cgemboss.6.6.5.g
  - alignment: fixed
  - 4 plane fit → all 4 planes firing
  - no more than 100 cluster–2D
  - no chi2 cut
  - 129357 events

- **Noise:** run 44, 47, 48, 49
  - hit in trigger time window
  - 74174 events

RUN ID				
AQUIRED	**SHARED**	**GAIN**	**THRSH**	**# EVENTS**
351	17	std	std	234154
421	44	off (HV=0)	std	18853
~~422~~	~~45~~	~~off (HV=0)~~	~~std~~	~~3021~~
~~423~~	~~46~~	~~off (HV=0)~~	~~std~~	~~21168~~
424	47	off (HV=0)	std	24288
426	48	off (HV=0)	std	14469
428	49	off (HV=0)	std	16564

TMVA package

- Toolkit for Multi Variate Analysis
- C++ Object Oriented / ROOT
- Many algorithms available:

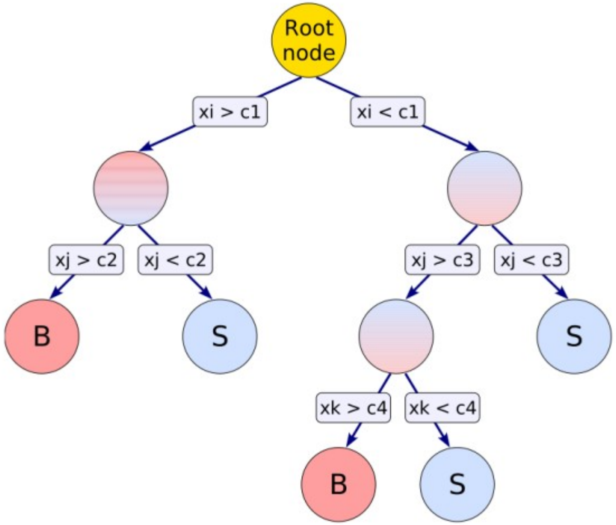


CRITERIA		MVA METHOD Cuts	Likelihood	PDE-RS / k-NN	PDE-Foam	H-Matrix	Fisher / LD	MLP	BDT	Rule-Fit	SVM
Performance	No or linear correlations	⋆	⋆⋆	⋆	⋆	⋆	⋆⋆	⋆⋆	⋆	⋆⋆	⋆
	Nonlinear correlations	◦	◦	⋆⋆	⋆⋆	◦	◦	⋆⋆	⋆⋆	⋆⋆	⋆⋆
Speed	Training	◦	⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆⋆	⋆	⋆	⋆	◦
	Response	⋆⋆	⋆⋆	◦	⋆	⋆⋆	⋆⋆	⋆⋆	⋆	⋆⋆	⋆
Robustness	Overtraining	⋆⋆	⋆	⋆	⋆	⋆⋆	⋆⋆	⋆	⋆	⋆	⋆⋆
	Weak variables	⋆⋆	⋆	◦	◦	⋆⋆	⋆⋆	⋆	⋆⋆	⋆	⋆
Curse of dimensionality		◦	⋆⋆	◦	◦	⋆⋆	⋆⋆	⋆	⋆	⋆	
Transparency		⋆⋆	⋆⋆	⋆	⋆	⋆⋆	⋆⋆	◦	◦	◦	◦

Table 6: Assessment of MVA method properties. The symbols stand for the attributes "good" (⋆⋆), "fair" (⋆) and "bad" (◦). "Curse of dimensionality" refers to the "burden" of required increase in training statistics and processing time when adding more input variables. See also comments in the text. The FDA method is not listed here since its properties depend on the chosen function.

Most straightforward one for starters

Boosted Decision Tree



- Series on nodes: @ each node a cut decides *left/right*
- End nodes (*leafs*) are signal/background
- Several trees with different *weights* to form a *forest*
- BDT outcome is the response of the forest

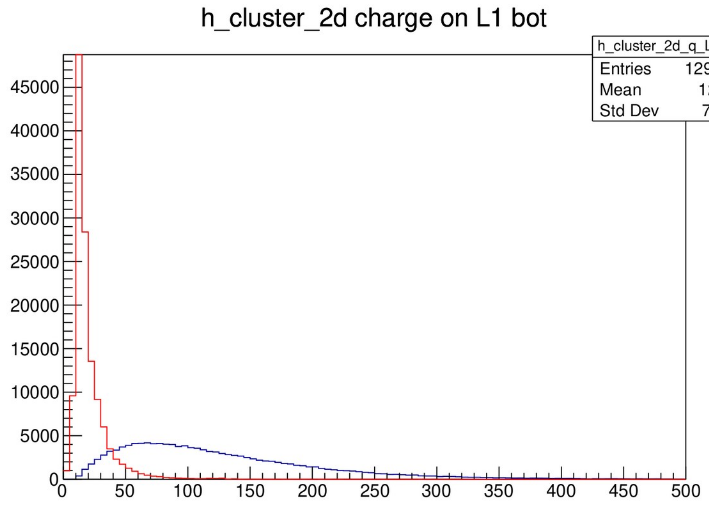
$$\hat{y}(x) = \sum_t w_t h_t(x)$$

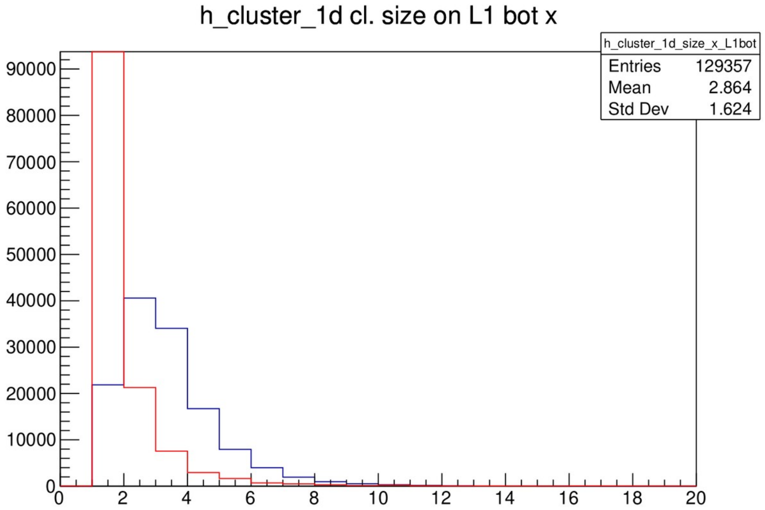
Variable selection

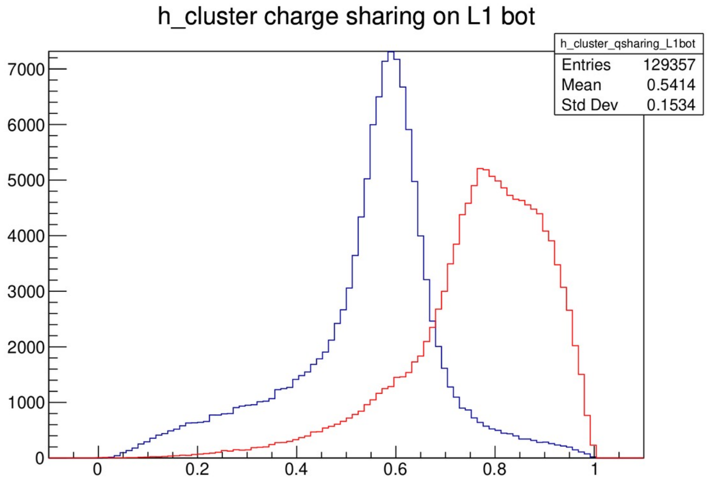
L1 bottom

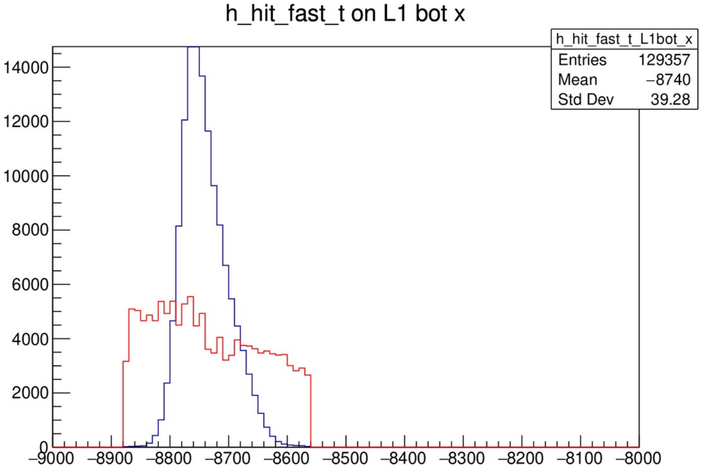
Signal

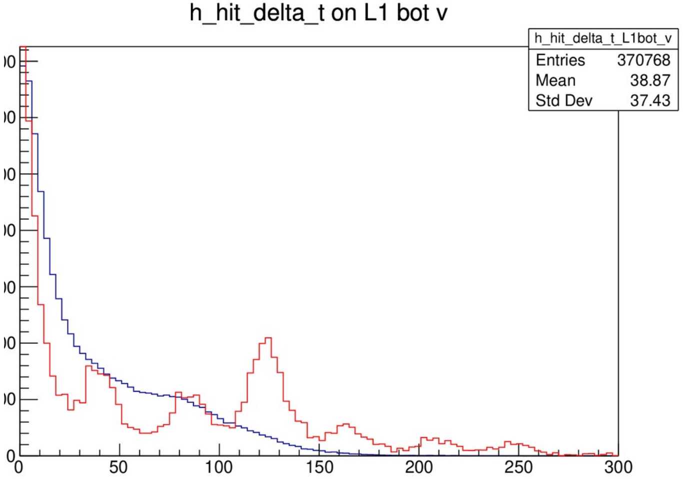
Noise


h_cluster_2d charge on L1 bot


h_cluster_1d cl. size on L1 bot x

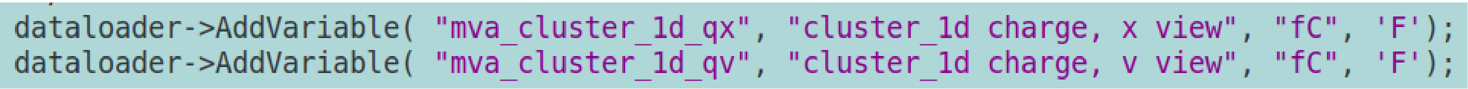

h_cluster charge sharing on L1 bot


h_hit_fast_t on L1 bot x


h_hit_delta_t on L1 bot v

TMVA variables

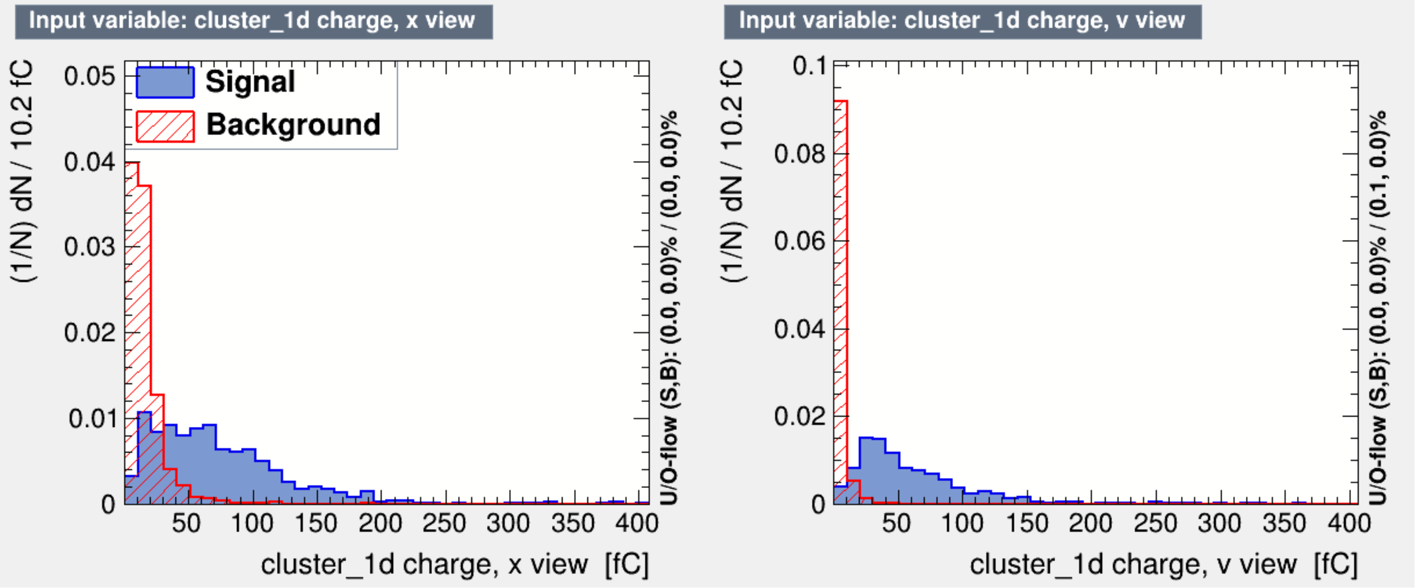
Very first attempt:



```
dataloader->AddVariable( "mva_cluster_1d_qx", "cluster_1d charge, x view", "fC", 'F');
dataloader->AddVariable( "mva_cluster_1d_qv", "cluster_1d charge, v view", "fC", 'F');
```

- #signal events for training = 1000
- #background events for training = 1000
- split mode = random
- method: BDT
- NTrees=850
- MinNodeSize=2.5%
- MaxDepth=3
- BoostType=AdaBoost
- AdaBoostBeta=0.5
- UseBaggedBoost
- BaggedSampleFraction=0.5
- SeparationType=GiniIndex
- nCuts=20"

Variable values



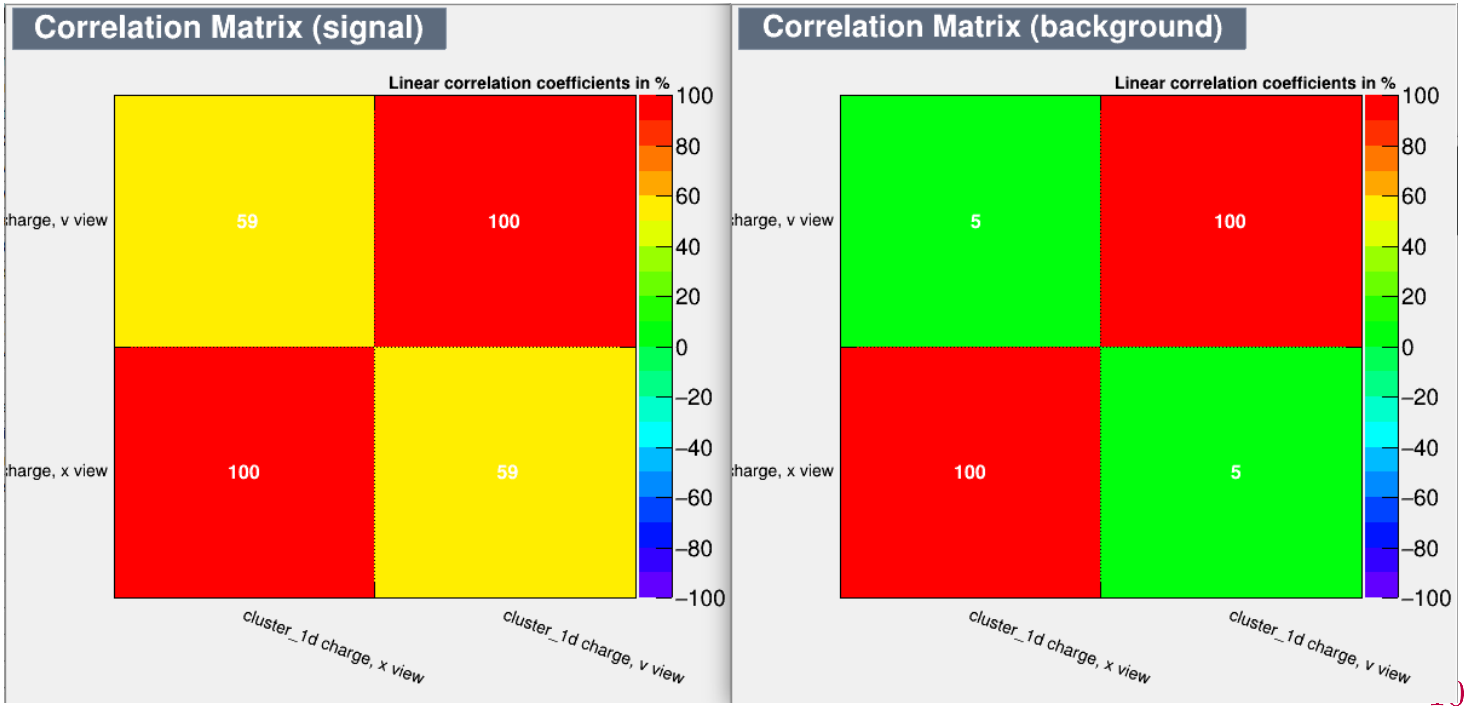
TMVA variables

Very first attempt:

```
dataloader->AddVariable( "mva_cluster_1d_qx", "cluster_1d charge, x view", "fC", 'F');
dataloader->AddVariable( "mva_cluster_1d_qv", "cluster_1d charge, v view", "fC", 'F');
```

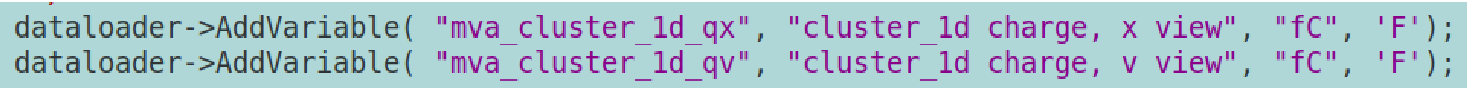
- #signal events for training = 1000
- #background events for training = 1000
- split mode = random
- method: BDT
- NTrees=850
- MinNodeSize=2.5%
- MaxDepth=3
- BoostType=AdaBoost
- AdaBoostBeta=0.5
- UseBaggedBoost
- BaggedSampleFraction=0.5
- SeparationType=GiniIndex
- nCuts=20"

Correlations



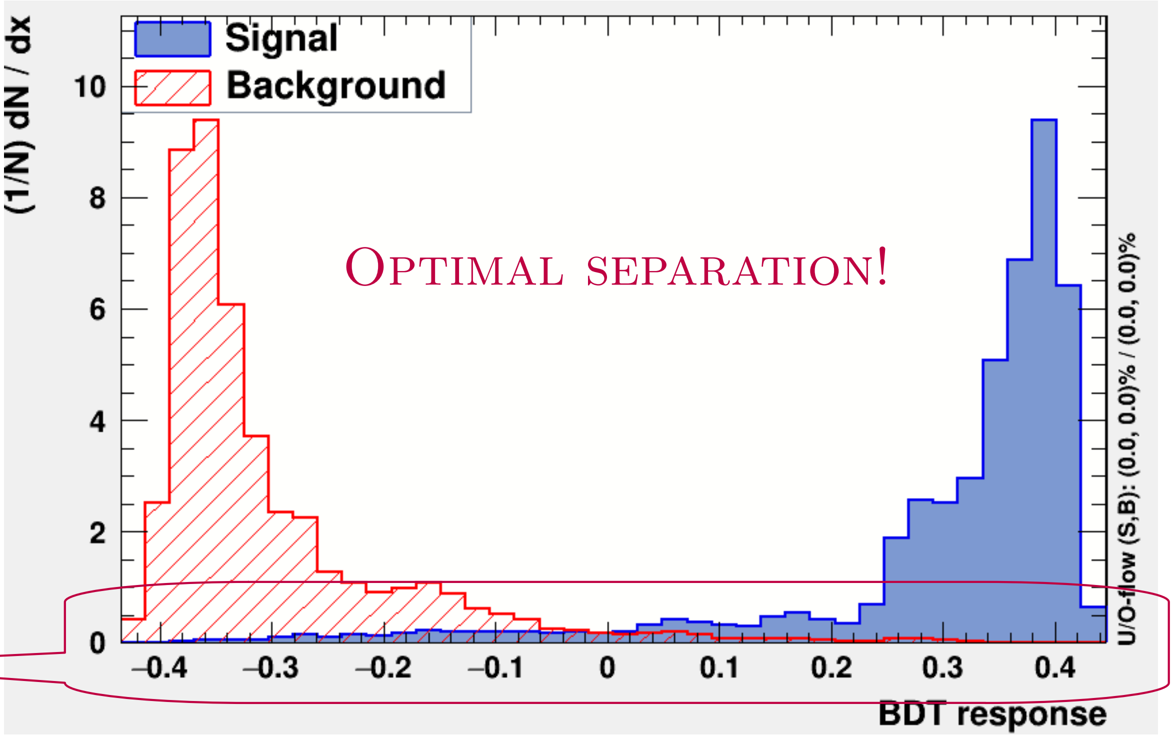
TMVA response

Very first attempt:



```
dataloader->AddVariable( "mva_cluster_1d_qx", "cluster_1d charge, x view", "fC", 'F');
dataloader->AddVariable( "mva_cluster_1d_qv", "cluster_1d charge, v view", "fC", 'F');
```

- #signal events for training = 1000
- #background events for training = 1000
- split mode = random
- method: BDT
- NTrees=850
- MinNodeSize=2.5%
- MaxDepth=3
- BoostType=AdaBoost
- AdaBoostBeta=0.5
- UseBaggedBoost
- BaggedSampleFraction=0.5
- SeparationType=GiniIndex
- nCuts=20"

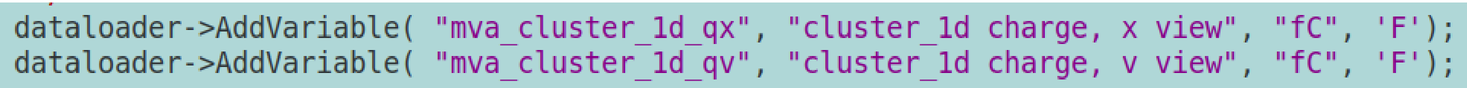


Optimal separation!

Select cut

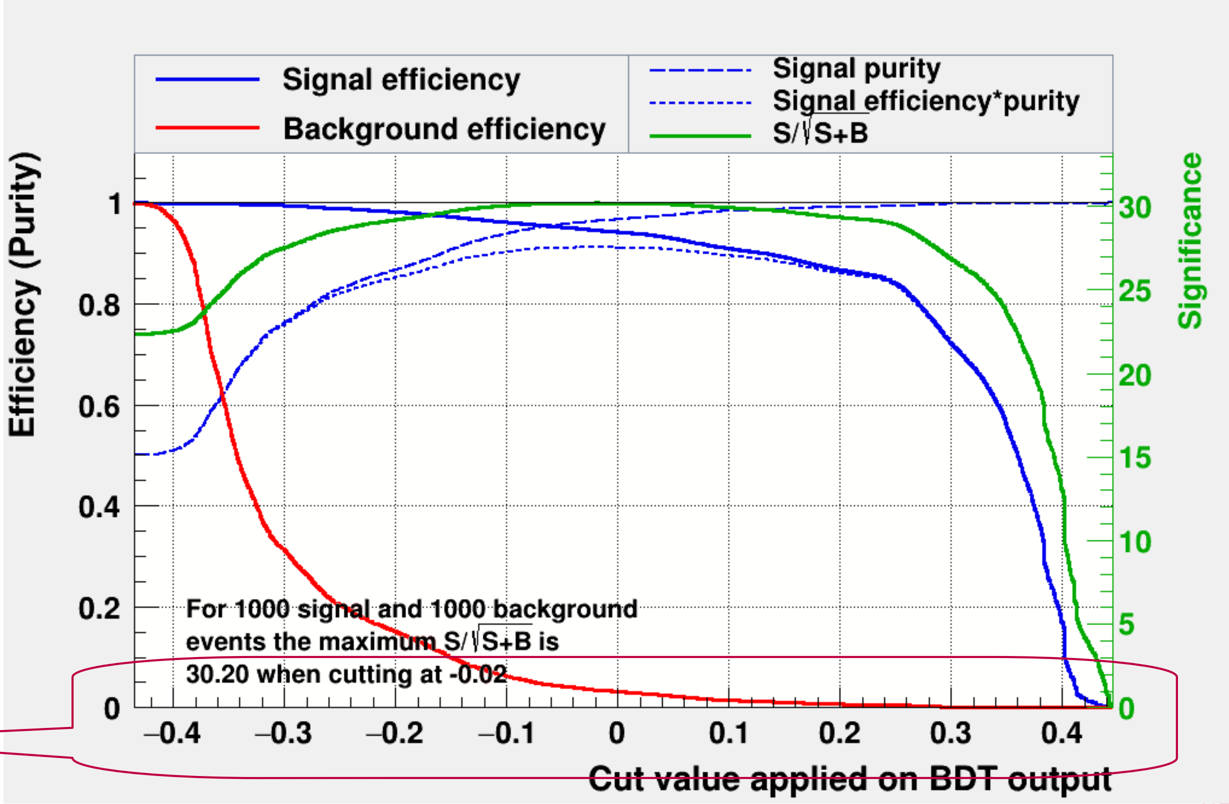
TMVA: efficiency

Very first attempt:



```
dataloader->AddVariable( "mva_cluster_1d_qx", "cluster_1d charge, x view", "fC", 'F');
dataloader->AddVariable( "mva_cluster_1d_qv", "cluster_1d charge, v view", "fC", 'F');
```

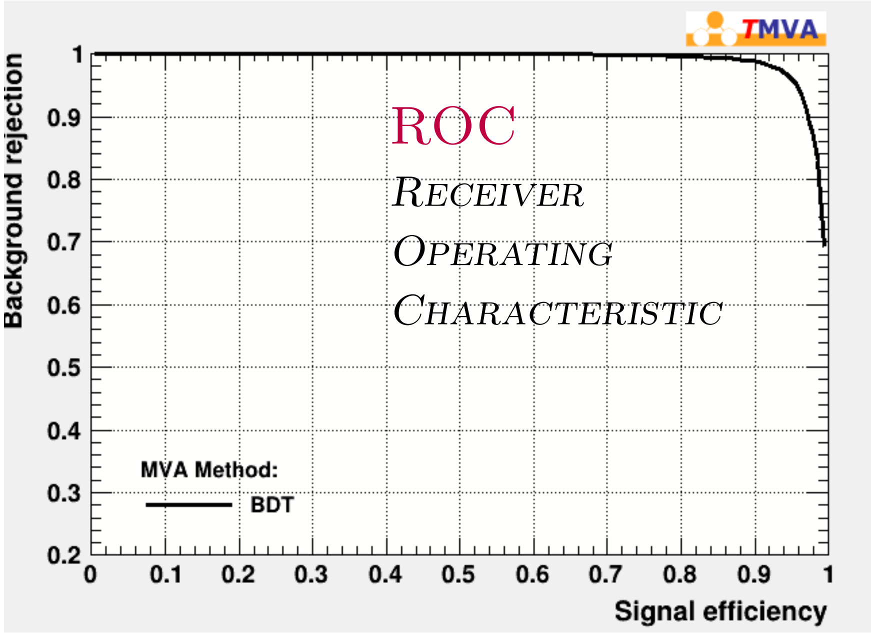
- #signal events for training = 1000
- #background events for training = 1000
- split mode = random
- method: BDT
- NTrees=850
- MinNodeSize=2.5%
- MaxDepth=3
- BoostType=AdaBoost
- AdaBoostBeta=0.5
- UseBaggedBoost
- BaggedSampleFraction=0.5
- SeparationType=GiniIndex
- nCuts=20"



For 1000 signal and 1000 background events the maximum $S/\sqrt{S+B}$ is 30.20 when cutting at -0.02

Select cut

TMVA: ROC & TEST



ROC

RECEIVER

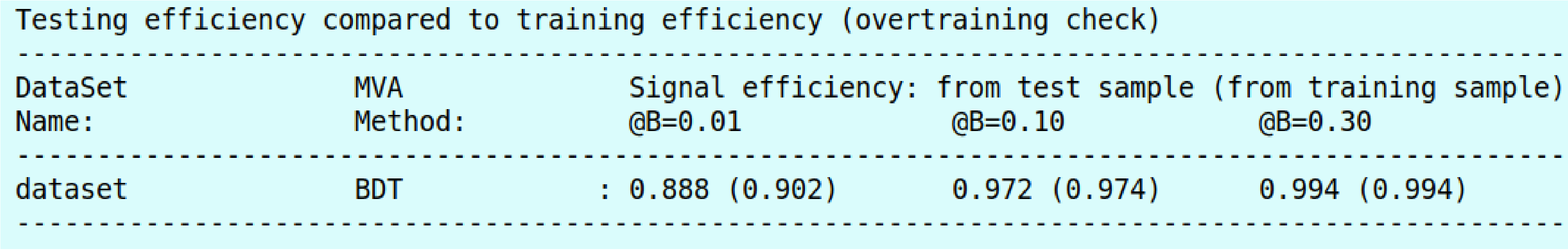
OPERATING

CHARACTERISTIC

We want **good noise rejection** and **good signal efficiency** → give–and–take

- efficiency = 1 – rejection
- given @ three background eff. Levels

example:
@B=0.10 → background rejection = 0.90 → signal efficiency = 0.972



```
Testing efficiency compared to training efficiency (overtraining check)
--------------------------------------------------------------------------------------------
DataSet                 MVA                 Signal efficiency: from test sample (from training sample)
Name:                   Method:             @B=0.01               @B=0.10              @B=0.30
--------------------------------------------------------------------------------------------
dataset                 BDT               : 0.888 (0.902)         0.972 (0.974)        0.994 (0.994)
--------------------------------------------------------------------------------------------
```

Summary

- The whole procedure works
- Different functions have been understood
- Signal and noise training/testing samples have been updated

To do list

- Optimize the BDT parameters
- Test other variables
- Test other ML methods

Summary

- The whole procedure works
- Different functions have been understood
- Signal and noise training/testing samples have been updated

To do list

- Optimize the BDT parameters
- Test other variables
- Test other ML methods