Progress Update

Trevor Towstego
UofT Neutrino/DM Meeting
April 18, 2018

TMVA Weights

- NormMode=NumEvents
  - Normalizes events such the average weight is 1 (i.e. total number of events is normalized to equal total number of MC events)
    - This is done separately for signal and background
    - Normalizing weight is applied after considering user-specified event-by-event weight
  - This changes the relative number of signal/background events, but *only affects training*
- NormMode=None
  - Use event-by-event weights as given by user
  - If weights are very small or very large, some algorithms may become unstable
- User-specified overall weights can also be applied to the signal and background separately

Approach:

- With NormMode=None, see if making event weights larger, but keeping signal/background ratio the same, improves performance
  - Do this using overall weights on signal and background
- With NormMode=NumEvents, results are not incorrect, but the algorithm might not be optimal with a modified signal/background ratio
  - Advantage here is that weights tend to be around order of 1, and so MLP/BDT performance may be more predictable/reliable

NormMode=NumEvents

```
Number of events in input trees (after possible flattening of arrays):
    Signal          -- number of events       : 7432   / sum of weights: 1.44486
    Background      -- number of events       : 9016   / sum of weights: 3.46606
    Signal     tree -- total number of entries: 7432
    Background tree -- total number of entries: 9016
Preselection:
    No preselection cuts applied on event classes
Weight renormalisation mode: "NumEvents": renormalise independently the ...
... class weights so that Sum[i=1..N_j]{w_i} = N_j, j=0,1,2...
... (note that N_j is the sum of training and test events)
--> Rescale Signal     event weights by factor: 5143.75
--> Rescale Background event weights by factor: 2601.22
Number of training and testing events after rescaling:
---------------------------------------------------
Signal     -- training entries            : 3716 (sum of weights: 3681.65)
Signal     -- testing entries             : 3716 (sum of weights: 3750.35)
Signal     -- training and testing entries: 7432 (sum of weights: 7432)
Background -- training entries            : 4508 (sum of weights: 4625.46)
Background -- testing entries             : 4508 (sum of weights: 4390.54)
Background -- training and testing entries: 9016 (sum of weights: 9016)
```

global scaling factors

2Reπ Results

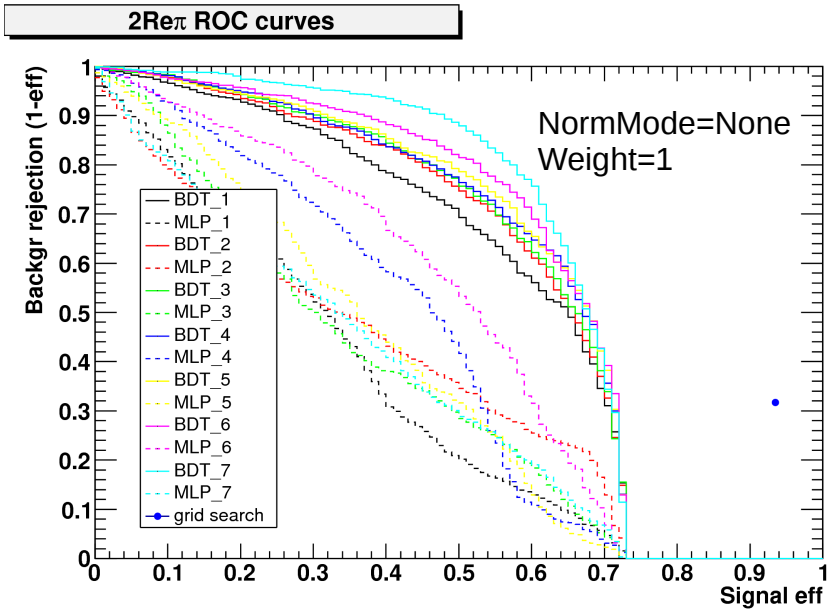
	2Repi	MLP		BDT	
	Cuts	Notes	FOM	Notes	FOM
1	p_low p_e-p_pi 2Repi-like vs 2Ree nll m_epi	HiddenLayers=N+5	0.495 0.682 0.650 0.684	MaxDepth=3	0.573 0.739 0.739 0.742
2	p_low p_e-p_pi 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll m_epi	HiddenLayers=N+5	0.511 0.651 0.650 0.679	MaxDepth=3	0.588 0.753 0.753 0.755
3	p_low p_e-p_pi 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll m_epi cos(theta)	HiddenLayers=N+5	0.495 0.688 0.650 0.691	MaxDepth=3	0.593 0.759 0.759 0.764
4	p_low p_e-p_pi 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll m_epi cos(theta)	HiddenLayers=N+5,N	0.494 0.692 0.650 0.722	MaxDepth=4	0.610 0.780 0.780 0.775

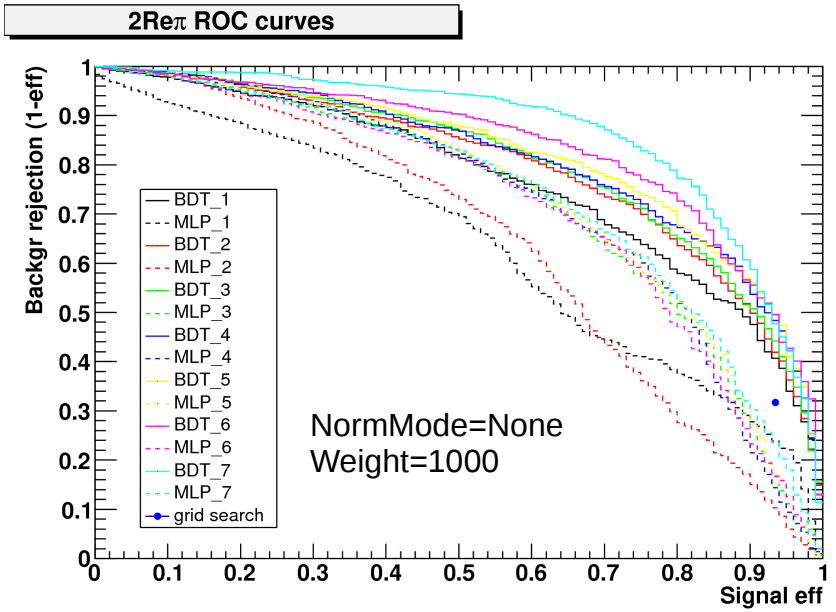
	2Repi	MLP		BDT	
	Cuts	Notes	FOM	Notes	FOM
5	p_low p_e p_pi 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll m_epi cos(theta)	HiddenLayers=N+5,N	0.494 0.694 0.650 0.697	MaxDepth=4	0.616 0.780 0.780 0.776
6	p_low p_e p_pi 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll m_epi cos(theta)	HiddenLayers=N+5,N,N	0.495 0.682 0.650 0.707	MaxDepth=5	0.637 0.804 0.804 0.811
7	p_low p_e p_pi 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll m_epi cos(theta) towall e towall pi	HiddenLayers=N+5,N,N	0.494 0.700 0.650 0.722	MaxDepth=5	0.683 0.842 0.842 **0.843**

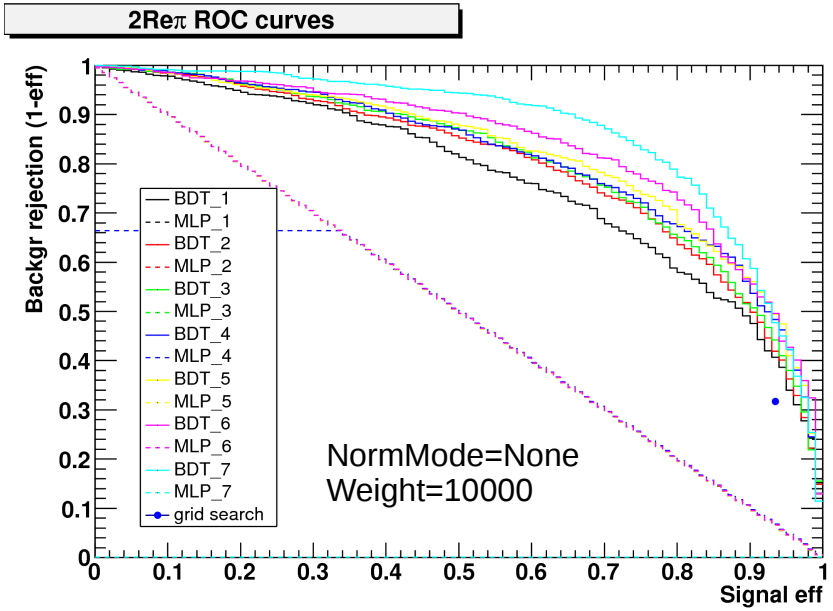
baseline: 0.652
grid: 0.710

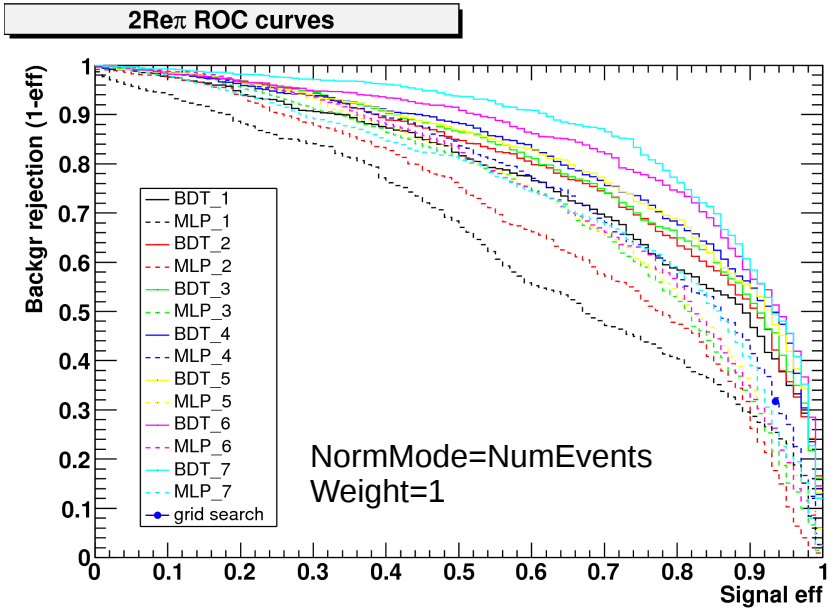
FOM
NormMode=None , Weight=1
NormMode=None , Weight=1000
NormMode=None , Weight=10000
NormMode=NumEvents, Weight=1

2Reπ Results









2Reπ1de Results

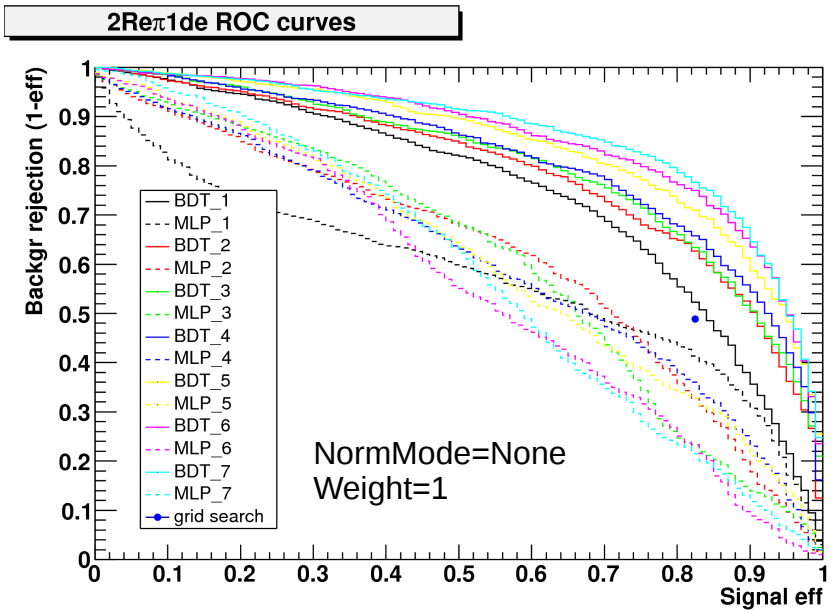
	2Repi1de	MLP		BDT	
	Cuts	Notes	FOM	Notes	FOM
1	p_e-p_pi d2se	HiddenLayers= N+5	1.169 1.171 1.172 1.171	MaxDepth=3	1.186 1.186 1.186 1.186
2	p_e-p_pi d2se 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll	HiddenLayers= N+5	1.146 1.211 1.211 1.205	MaxDepth=3	1.247 1.247 1.247 1.245
3	p_e-p_pi d2se 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll cos(theta)	HiddenLayers= N+5	1.149 1.218 1.224 1.221	MaxDepth=3	1.248 1.248 1.248 1.247
4	p_e-p_pi d2se 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll cos(theta)	HiddenLayers= N+5,N	1.147 1.224 1.232 1.228	MaxDepth=4	1.268 1.268 1.268 1.266

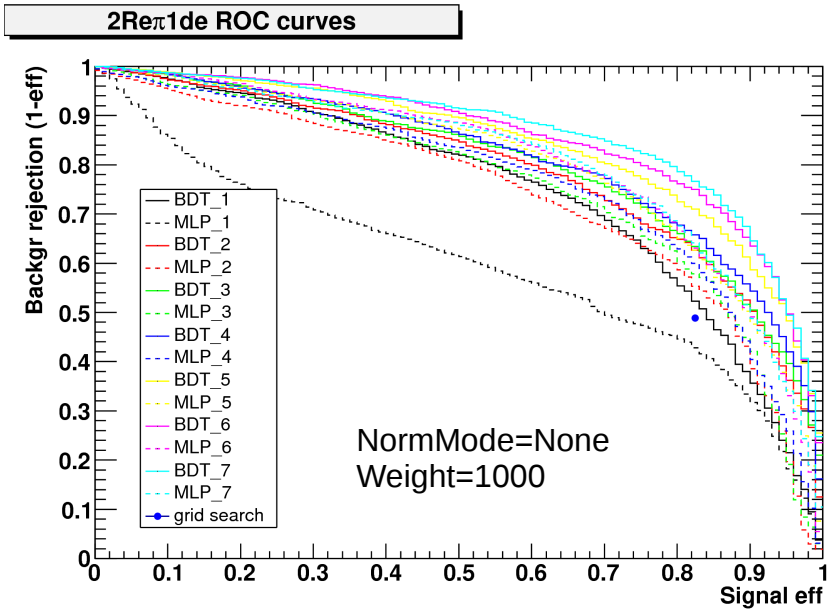
	2Repi1de	MLP		BDT	
	Cuts	Notes	FOM	Notes	FOM
5	p_e p_pi d2se 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll cos(theta)	HiddenLayers= N+5,N	1.154 1.240 1.242 1.241	MaxDepth=4	1.296 1.296 1.296 1.294
6	p_e p_pi d2se 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll cos(theta)	HiddenLayers= N+5,N,N	1.145 1.243 1.243 1.238	MaxDepth=5	1.315 1.315 1.315 1.314
7	p_e p_pi d2se 2Repi vs 2Rpie nll 2Rpie vs 2Ree nll cos(theta) towall e towall pi	HiddenLayers= N+5,N,N	1.148 1.244 1.243 1.242	MaxDepth=5	1.329 1.329 1.329 **1.333**

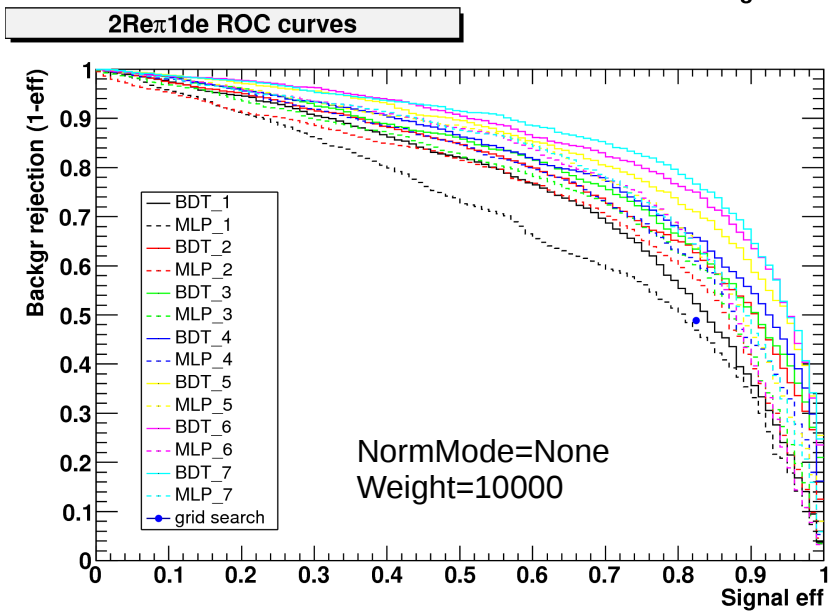
baseline: 1.145
grid: 1.197

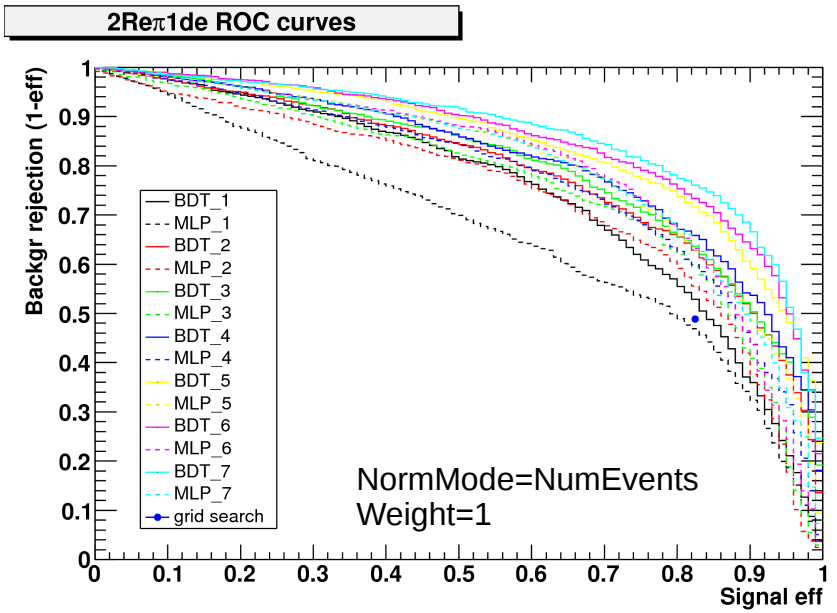
FOM
NormMode=None , Weight=1
NormMode=None , Weight=1000
NormMode=None , Weight=10000
NormMode=NumEvents, Weight=1

2Reπ1de Results









Other Observations

From TMVA User Manual:

In some cases event weights are given by Monte Carlo generators, and may turn out to be overall very small or large numbers. To avoid artifacts due to this, TMVA can internally renormalise the signal and background training(!) weights such that their respective sums of effective (weighted) events is equal.

Note: All this renormalisation only affects the training events as the training of some classifiers is sensitive to the relative amount of signal and background in the training data. On the other hand, the background or signal efficiency of the trained classifier as determined from the test sample is independent of the relative abundance of signal and background events.

- Other users have found issues with MLP when using large event weights
  - Neuron activation value calculations become unstable

```
--- MLP                                : Begin training
--- MLP  NormMode=None Weight=1        : Training Network
--- MLP                                : [>>>>>>>>>>>>>>>>>>>>>>>>] (99%, time left: 0 sec) [<D^2> (train/test): 6.636e-05/6.496e-05]

--- MLP                                : Begin training
--- MLP  NormMode=None Weight=1000     : Training Network
--- MLP                                : [>>>>>>.................] (27%, time left: 15 sec) [<D^2> (train/test): 0.06007/0.06044]

--- MLP                                : Begin training
--- MLP  NormMode=None Weight=10000    : Training Network
--- MLP                                : [>>>>>>>>>>>>>>>>>......] (71%, time left: 5 sec) [<D^2> (train/test): 0.6304/6.505e+05]

--- MLP                                : Begin training
--- MLP  NormMode=NumEvents Weight=1   : Training Network
--- MLP                                : [>>>>>>>>>>>.............] (48%, time left: 10 sec) [<D^2> (train/test): 0.2372/0.2373]
```

Thoughts

- Might be best just to stick with `NormMode=NumEvents`
  - Signal to background ratio will be off when training, but it seems to be more stable and reliable