

Use of BDT

BDT tutorial : `$ROOTSYS/tmva/test/`

Four basic scripts:

Script name	function
TMVAClassification.C	Training and test
TMVAClassificationApplication.C	Application
TMVAGui.C	Plot
tmvaglob.C	Plot style

TMVAClassification.C Training and test

1. How to run

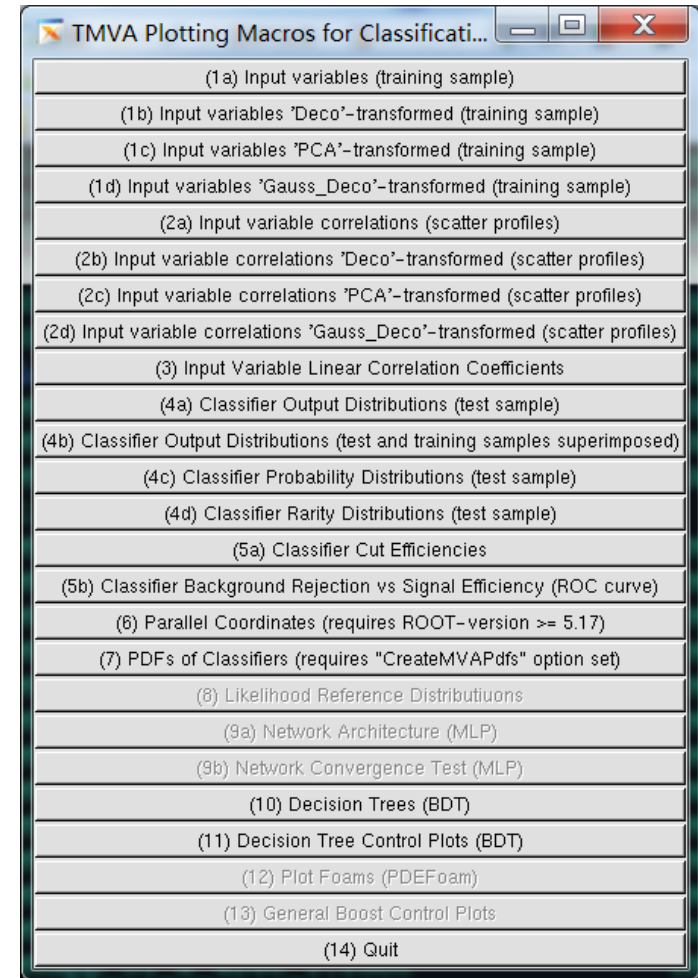
- root -l
TMVAClassification.C\(\"Fisher,Likelihood\")
Replace 'Fisher,Likelihood' with 'BDT'
结果是产生一个root文件和weight目录(xml文件), 并画图
- If the MC events have been trained and tested, use
root -l TMVAGui.C

调用之前生成的root文件, 并且画图

2. Detail

- TMVA::Factory *factory = new
TMVA::Factory("TMVAClassification",
outputFile,"!V:!Silent:Color:DrawProgressBar:Transformations=l;D;P;G,D:AnalysisType=Classification");
- factory->AddVariable("var1", "Variable 1", "units", 'F');
- factory->AddVariable("var2", "Variable 2", "units", 'F');

Only 'l' and 'F'



TMVAClassification.C Training and test

- `factory->AddSignalTree (TreeName_s, signalWeight);`
- `factory->AddBackgroundTree(TreeName_b, backgroundWeight);`
- The events are weighted with the measured cross sections

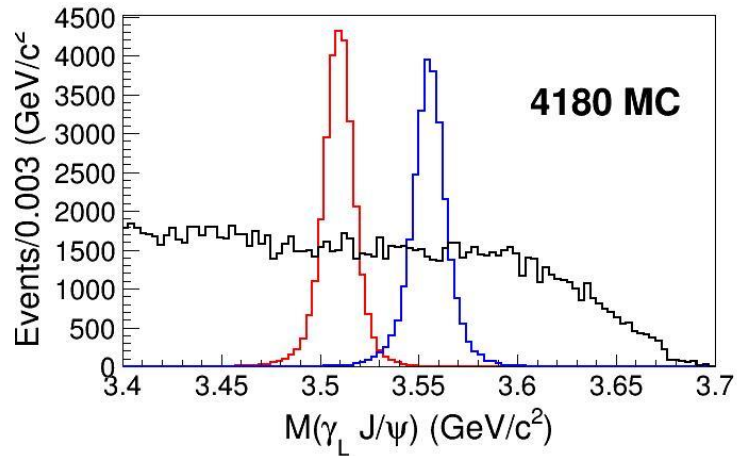
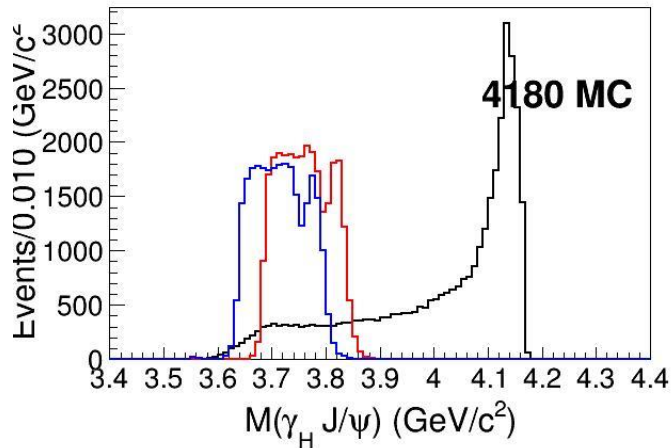
- $$N_s w_s = \mathcal{L} \sigma_1 \epsilon_1 \mathcal{B}_1 + \mathcal{L} \sigma_2 \epsilon_2 \mathcal{B}_2 = N_s^{data}$$

- $$N_b w_b = \mathcal{L} \sigma_b \epsilon_b = N_b^{data}$$

N_s is the number of input signal MC events and N_b is the number of background MC

QUESTION 1: $\epsilon_b = ?$

TMVAClassification.C Training and test



Training region:

$$3.6 < M(\gamma_H J/\psi) < 3.9 \quad 3.45 < M(\gamma_H J/\psi) < 3.6$$

fit region:

$$3.5 < M(\gamma_H J/\psi) < 4.0 \quad 3.35 < M(\gamma_H J/\psi) < 3.7$$

在归一的时候，对于本底的估计，一种是通过截面，另一种是根据拟合结果，但两者在误差范围内不一致（450+3， 527+-24）

QUESTION 2: 在训练的时候，我们是否要归一？

TMVAClassification.C Training and test

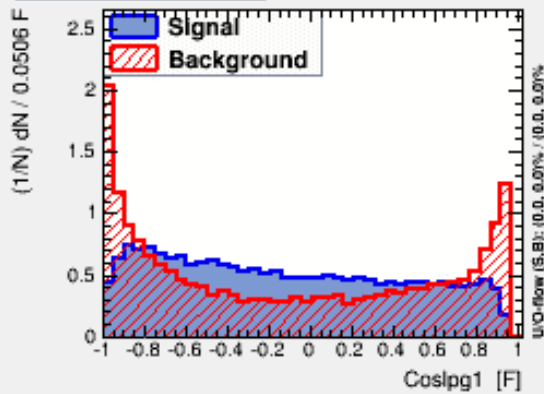
- `factory->PrepareTrainingAndTestTree(mycuts, mycutb, "nTrain_Signal=0:nTrain_Background=0:SplitMode=Random:NormMode=None:!V");`
- `factory->BookMethod(TMVA::Types::kBDT, "BDT", "!H:!V:NTrees=850:nEventsMin=150:MaxDepth=3:BoostType=AdaBoost:AdaBoostBeta=0.5:SeparationType=GiniIndex:nCuts=20:PruneMethod=NoPruning");`
- `factory->TrainAllMethods();`
- `factory->TestAllMethods();`
- `factory->EvaluateAllMethods();`



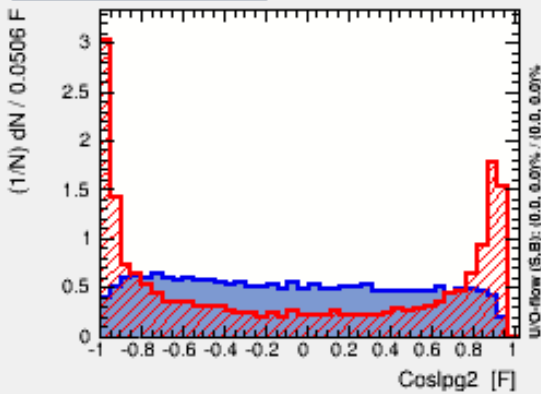
Preselection

Input variables (training sample)

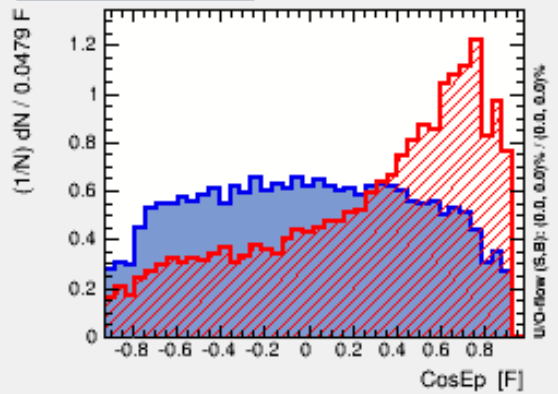
Input variable: Coslpg1



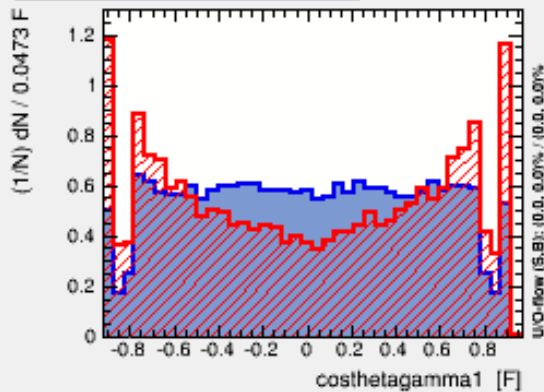
Input variable: Coslpg2



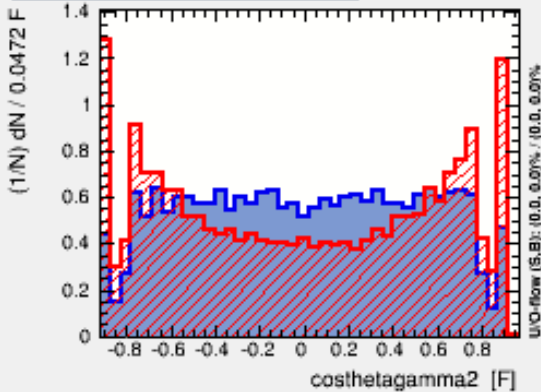
Input variable: CosEp



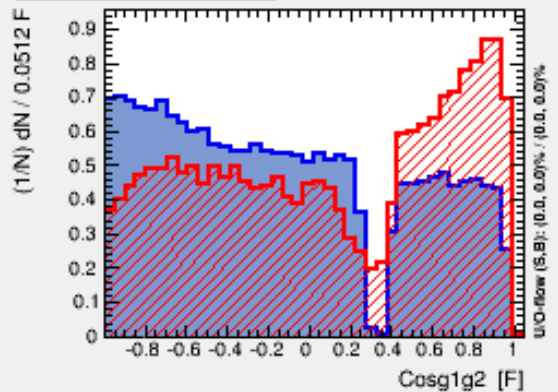
Input variable: costhetagamma1

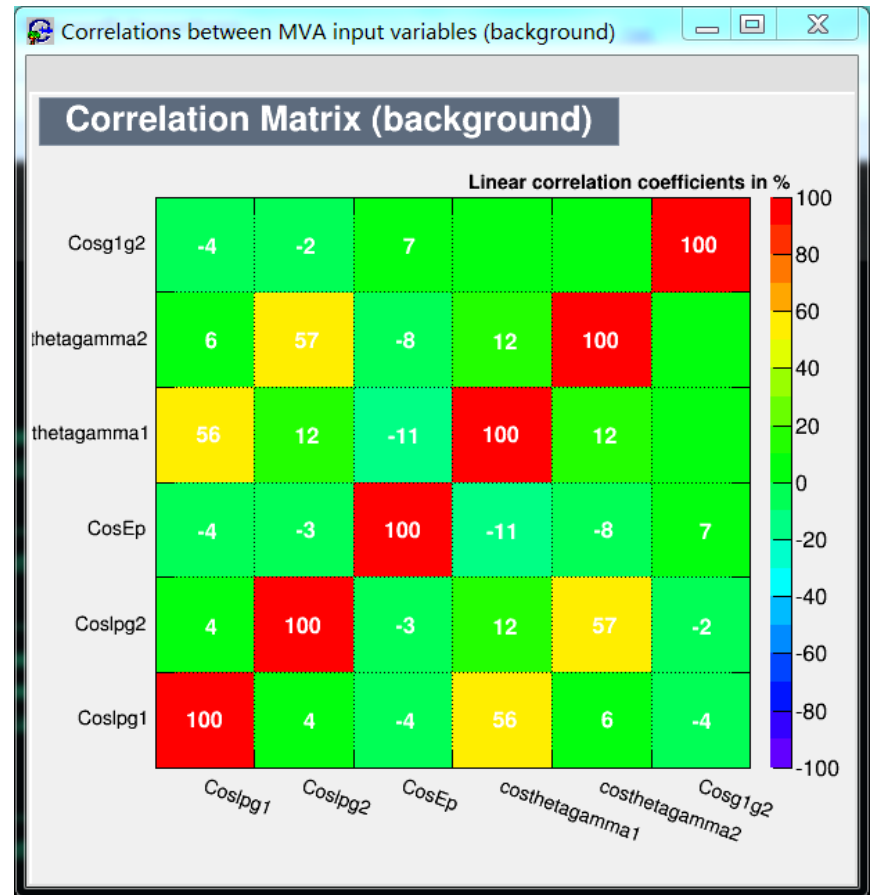
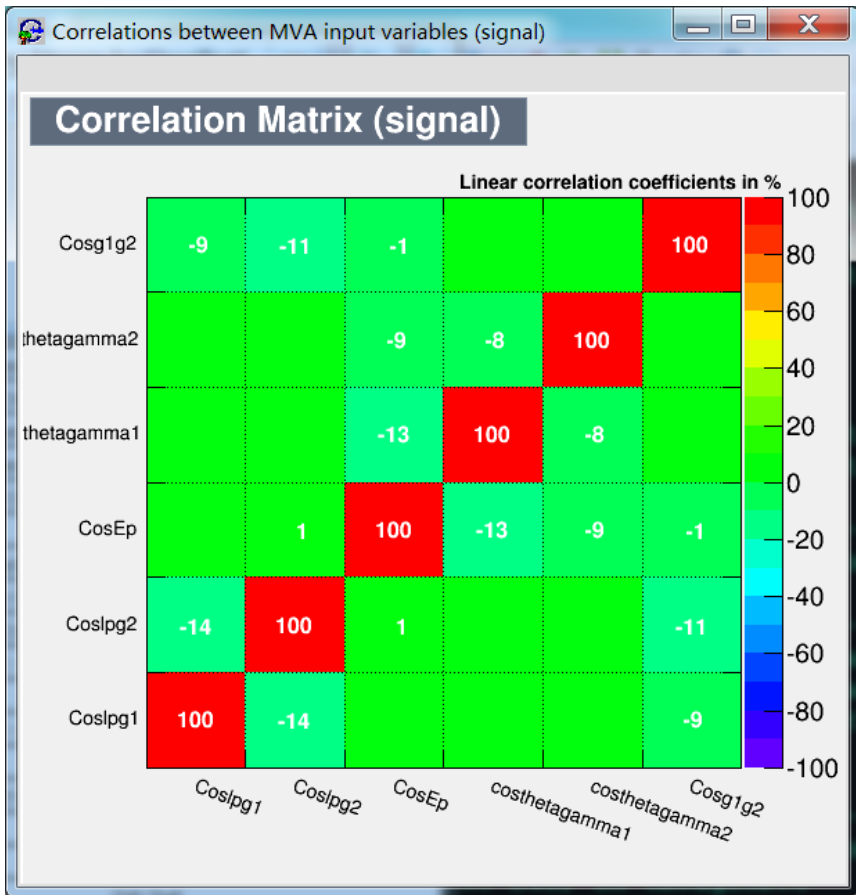


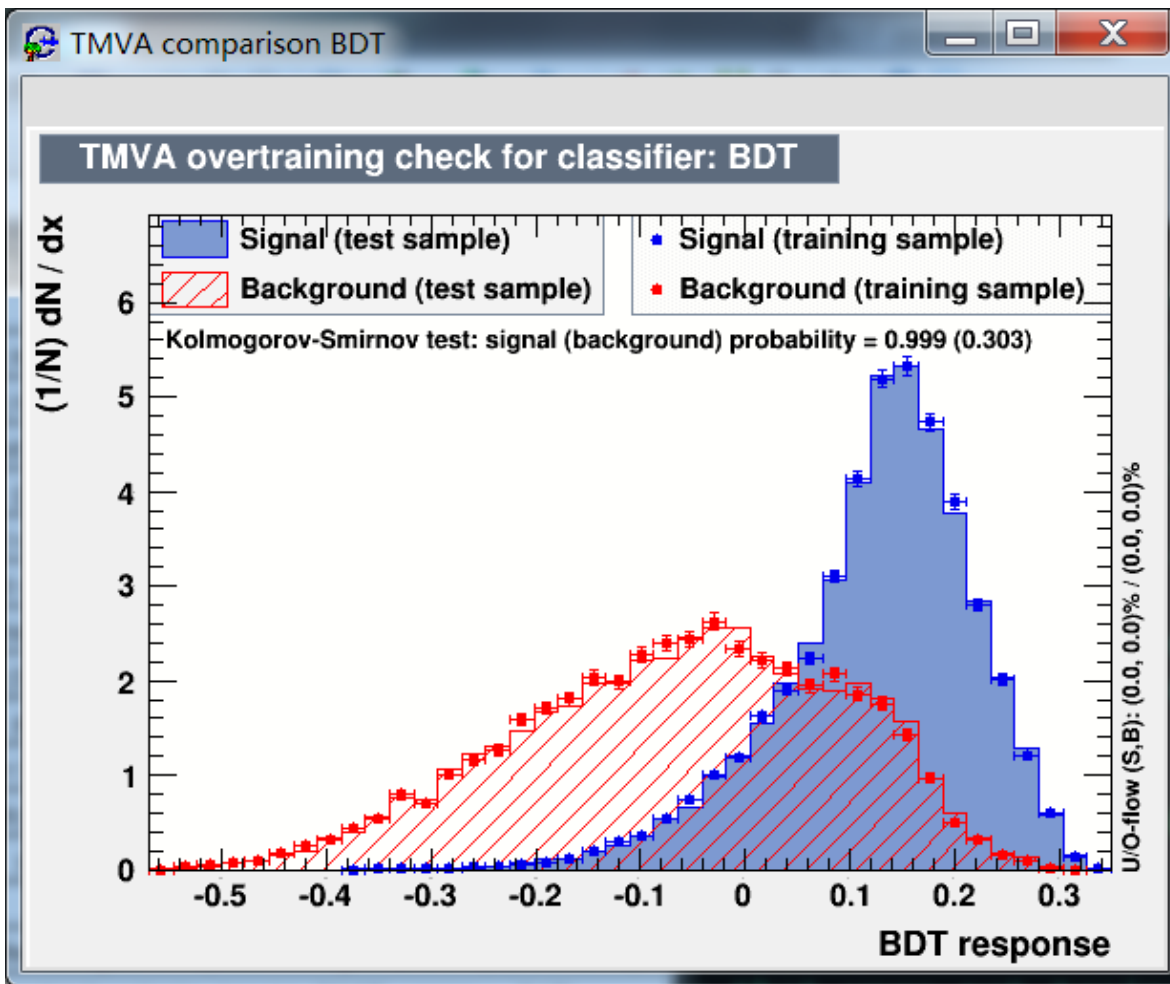
Input variable: costhetagamma2



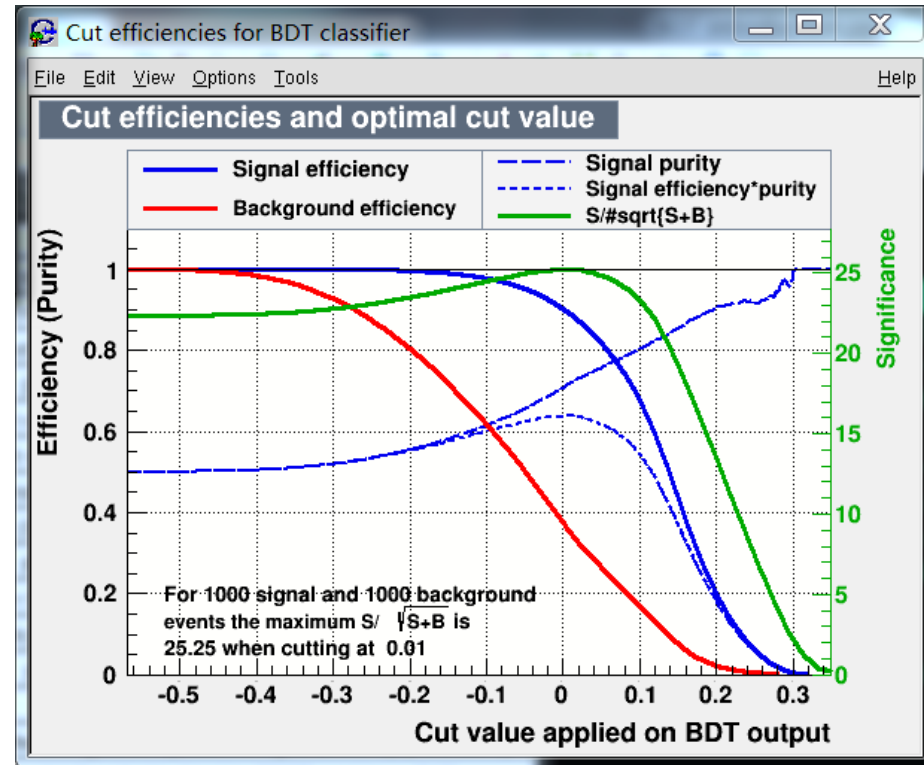
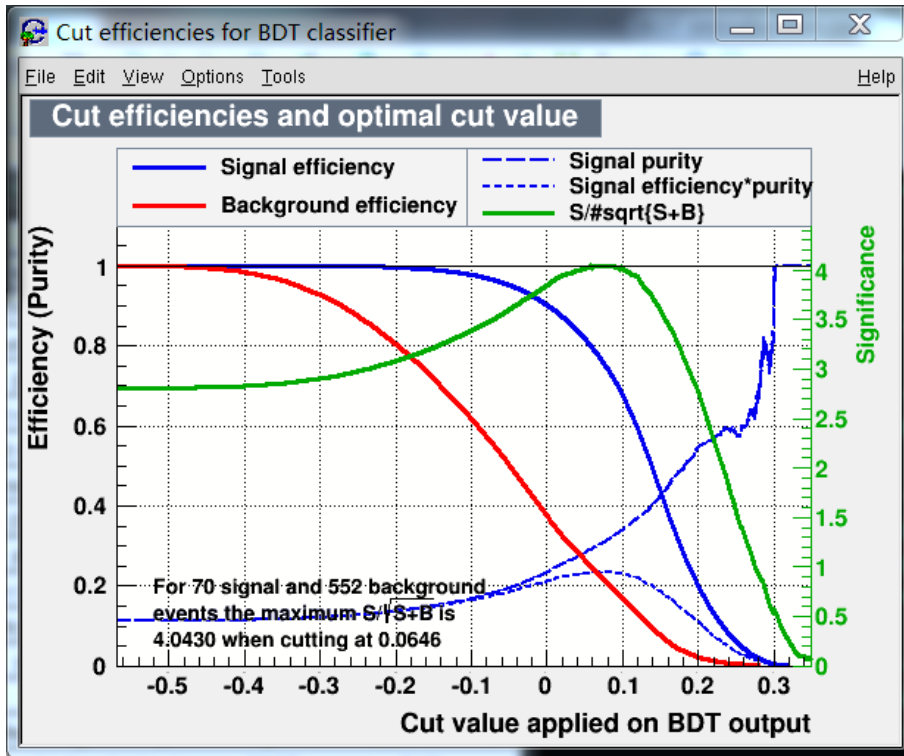
Input variable: Cosg1g2

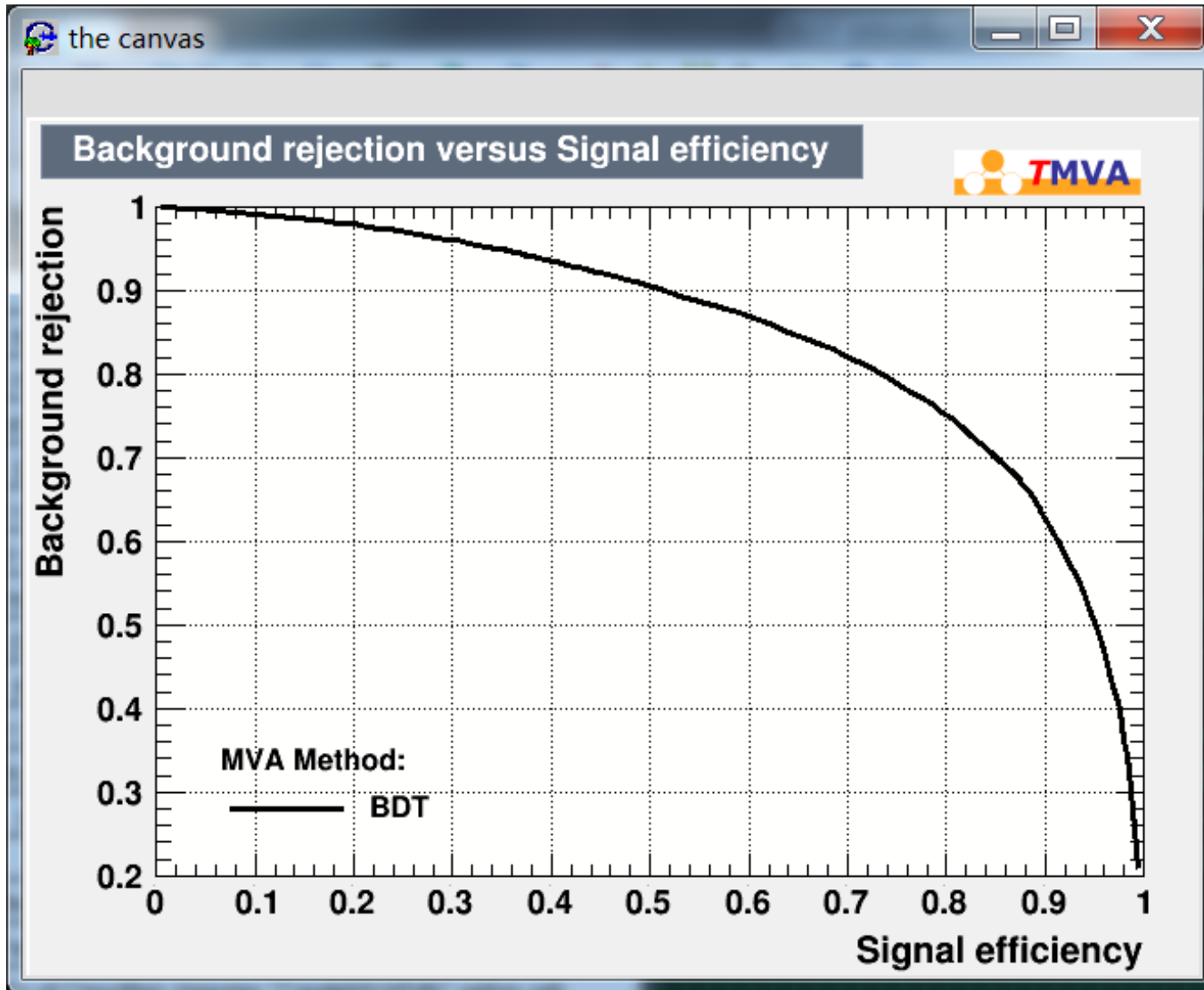






QUESTION 3: 这个一致性检验值 probability 是否是越大越好





QUESTION 4: 改变maxdepth, Ntrees, Minevens,
较好的曲线分布对应的一致性检验值反而小

TMVAClassificationApplication.C

1. How to run

- `root -l TMVAClassificationApplication.C\(\“BDT\”\)`

2. Detail

```
TMVA::Reader *reader = new TMVA::Reader( "!Color:!Silent" );  
Float_t var1, var2;  
reader->AddVariable( "var1",      &var1 );  
reader->AddVariable( "var1",      &var2 );
```

跟训练的顺序保持一致

```
//Book method  
reader->BookMVA( methodName, weightfile );
```

TMVAClassificationApplication.C

```
//Book output trees to save the BDT respond
```

```
TFile *target = new TFile( "TMVApp.root" , "RECREATE" );  
TTree* tree = new TTree("BDT","BDT");
```

```
double MVA_BDT;  
tree->Branch("MVA_BDT",&MVA_BDT,"MVA_BDT/D");
```

```
for (Long64_t ievt=0; ievt<Data_tree->GetEntries();ievt++)  
{  
    Data_tree->GetEntry(ievt);
```

```
    var1 = var1_data;  
    var2 = var2_data;
```

数据变量

```
    MVA_BDT = reader->EvaluateMVA("BDT method" );  
    tree->Fill();  
}
```

TMVA reader变量

TMVAClassificationApplication.C

