TMVA cut optimization tutorials

Jing Wang

Strategy - two steps

Goal: Determine cut values for some variables, to remove most background and remain most signals
Macros: https://github.com/boundino/tutorialTMVA

myTMVA

- For some specific variables, there are infinite groups of cut values that lead to same signal efficiency
- Among those cut values, myTMVA (based on ROOT TMVA package) gives a group of cut values for each signal efficiency, which removes most background

readxml

- readxml looks for the signal efficiency whose corresponding cut values result in maximum S/sqrt(S+B)

Strategy - S and B

*“cuts” below refers to the optimized cuts we are looking for

S: signal candidates number in signal region **after** cuts
B: background candidates number in signal region **after** cuts

therefore

S = S’ * signal cuts efficiency
B = B’ * background cuts efficiency

where

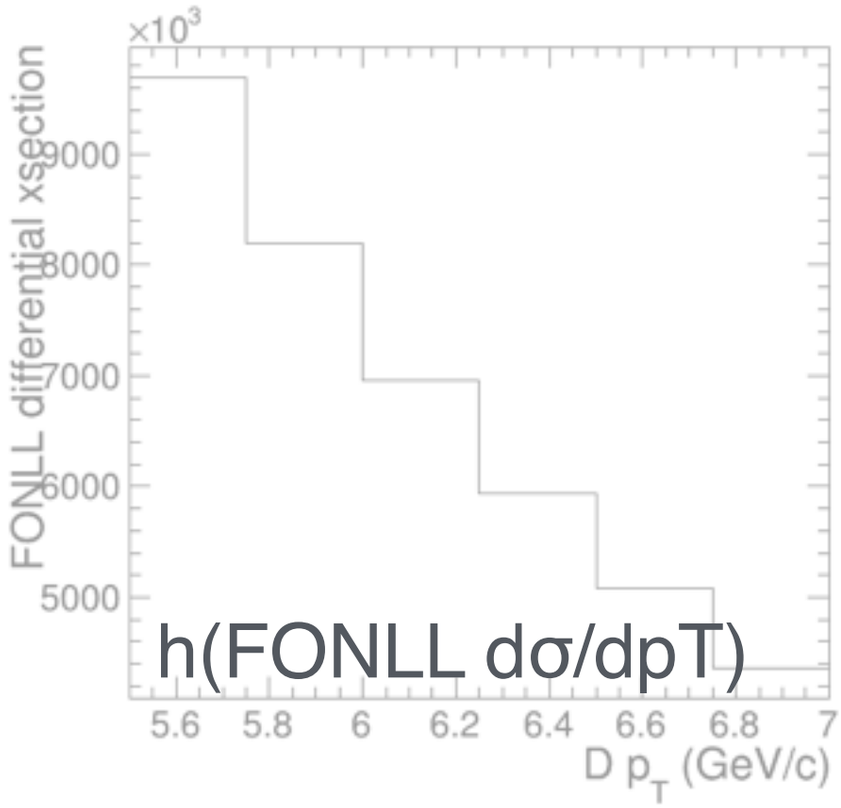
S’: signal candidates number in signal region **before** cuts
B’: background candidates number in signal region **before** cuts

so

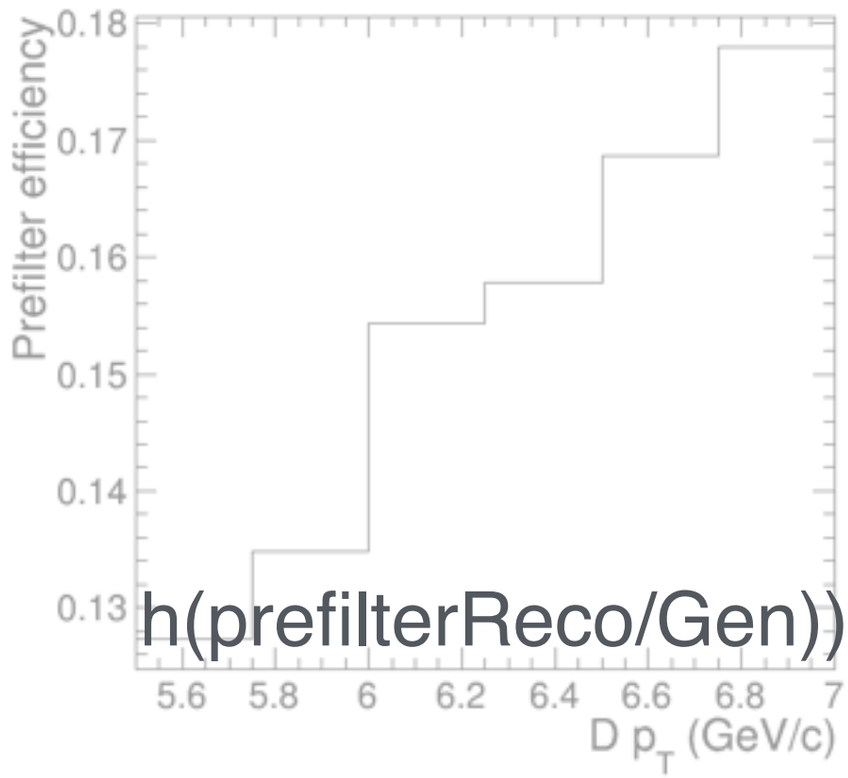
We can calculate S/sqrt(S+B) as long as we know S’ and B’

Strategy - Calculation of S’

S’: signal candidates number in signal region **before** cuts



S’ = Integral(h(FONLL dσ/dpT) *



h(prefilterReco/Gen))

* Taa * Event number * Branching fraction * Raa

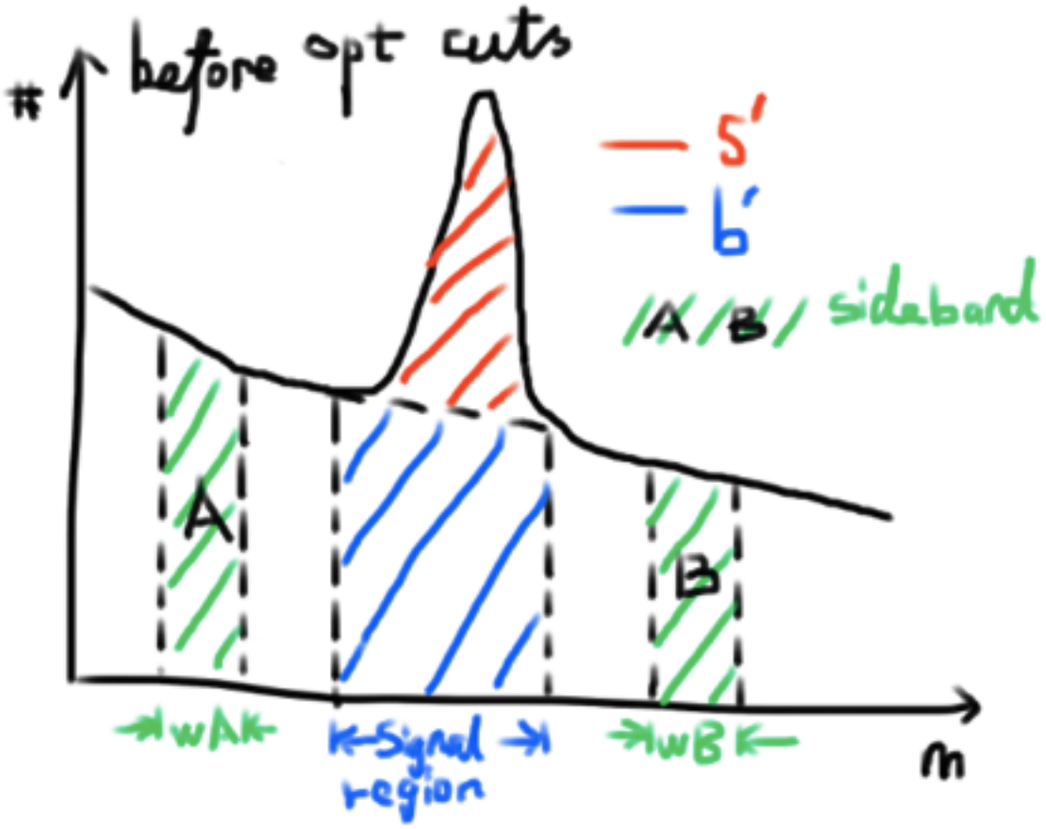
Strategy - Calculation of B’

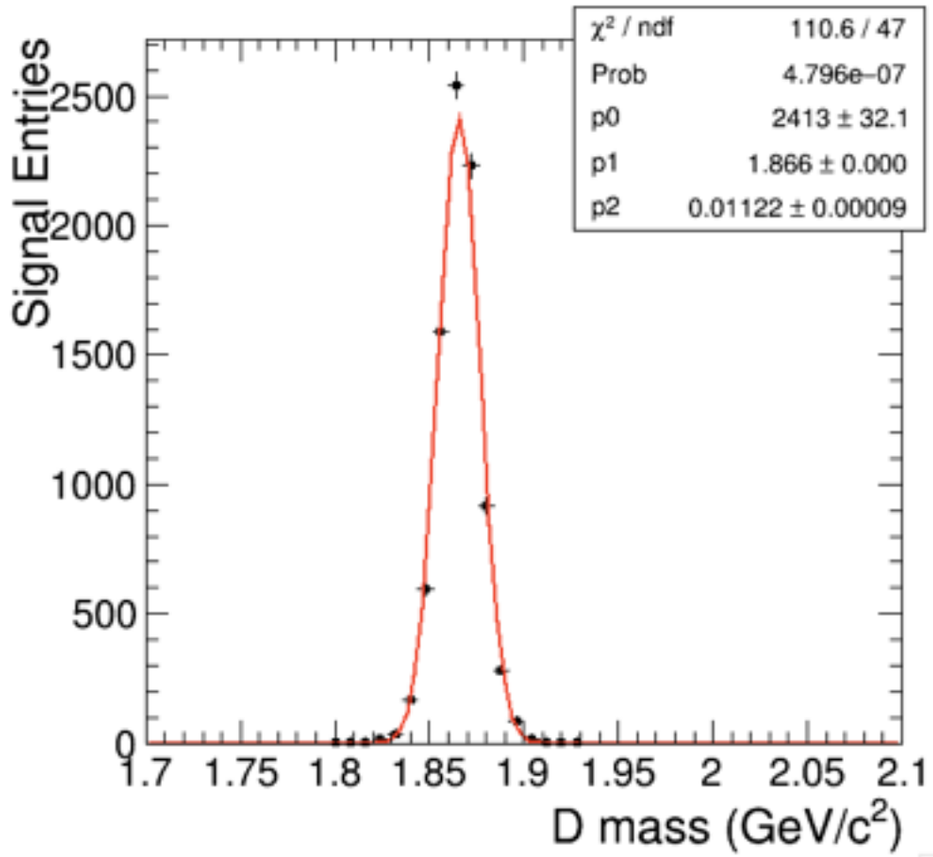
B’: background candidates number in signal region **before** cuts

B’ = (Candidates number in sideband) * (wSignalregion/wSideband)

wSideband = wA+wB
Sideband: 0.1<|m-mPDG|<0.15 GeV/c

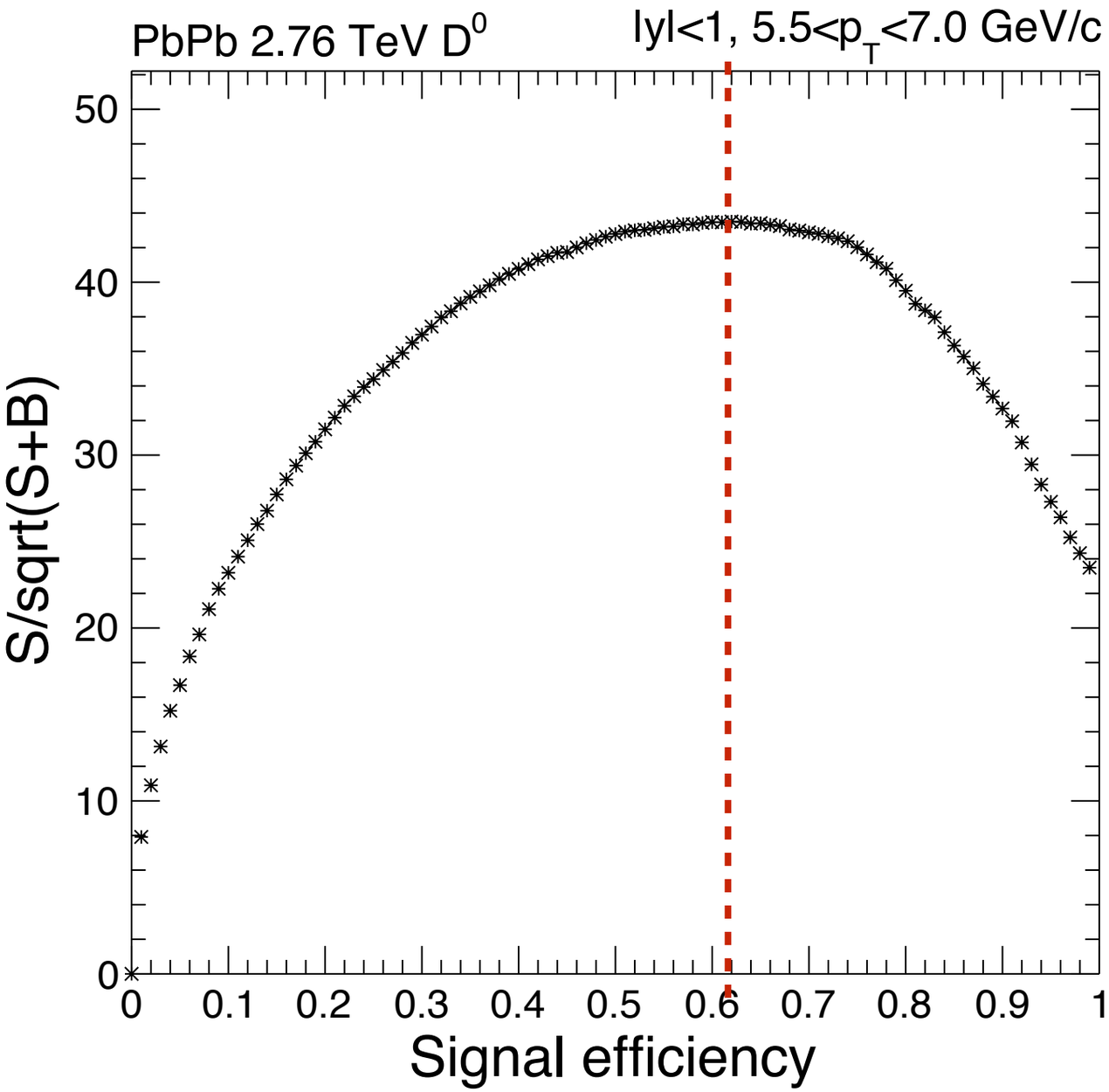
wSignalregion = 4σ





MC signal mass spectrum

Strategy - Example of readxml output



After myTMVA training, we have a single group of cut values for each signal efficiency.

The signal efficiency at which S/sqrt(S+B) reach maximum

Thus, the cuts value in response to this signal efficiency is what we want.

Macros - TMVA

https://github.com/boundino/tutorialTMVA/blob/master/myTMVA/TMVAClassification.C

L83-137: Choose the MVA method

```
82     // --- Cut optimisation
83     Use["Cuts"]            = 0;
84     Use["CutsD"]           = 0;
85     Use["CutsPCA"]         = 0;
86     Use["CutsGA"]          = 0;
87     Use["CutsSA"]          = 1;
88     //
89     // --- 1-dimensional likelihood ("naive Bayes estimator")
90     Use["Likelihood"]      = 0;
91     Use["LikelihoodD"]     = 0; // the "D" extension indicates decorrelated input variables (see option strings)
92     Use["LikelihoodPCA"]   = 0; // the "PCA" extension indicates PCA-transformed input variables (see option strings)
93     Use["LikelihoodKDE"]   = 0;
94     Use["LikelihoodMIX"]   = 0;
95     //
96     // --- Mutidimensional likelihood and Nearest-Neighbour methods
97     Use["PDERS"]           = 0;
98     Use["PDERSD"]          = 0;
99     Use["PDERSPCA"]        = 0;
100    Use["PDEFoam"]         = 0;
101    Use["PDEFoamBoost"]    = 0; // uses generalised MVA method boosting
102    Use["KNN"]             = 0; // k-nearest neighbour method
103    //
104    // --- Linear Discriminant Analysis
105    Use["LD"]              = 0; // Linear Discriminant identical to Fisher
106    Use["Fisher"]          = 0;
107    Use["FisherG"]         = 0;
108    Use["BoostedFisher"]   = 0; // uses generalised MVA method boosting
109    Use["HMatrix"]         = 0;
110    //
```

Macros - TMVA

https://github.com/boundino/tutorialTMVA/blob/master/myTMVA/TMVAClassification.C

L192-193: Add the variables you want to study

```
192     factory->AddVariable("dcandffls3d");//>
193     factory->AddVariable("dcandfprob");//>
```

L209-210: Add input files of signal and background. Here inputS is the signal MC, and inputB is the data sample.

```
209     TFile *inputS = TFile::Open("/data/wangj/TutorialsSamples/Dmesonana_hiforest_official_PbPbD0tokaonpion_Pt0153050_tkpt1p0eta1
210     TFile *inputB = TFile::Open("/data/wangj/TutorialsSamples/Dmesonana_Rereco_MBtrig_d0pt0_y1p2_tk1p0_eta1p1_d2p0_alpha0p2_tigh
```

L216-217: Register the trees of signal and background.

```
216     TTree *signal     = (TTree*)inputS->Get("ntDzero");
217     TTree *background = (TTree*)inputB->Get("ntDzero");
218
219     //global event weights per tree (see below for setting event-wise weights)
220     Double_t signalWeight     = 1.0;
221     Double_t backgroundWeight = 1.0;
222
223     // You can add an arbitrary number of signal or background trees
224     factory->AddSignalTree    ( signal,     signalWeight     );
225     factory->AddBackgroundTree( background, backgroundWeight );
```

Macros - TMVA

https://github.com/boundino/tutorialTMVA/blob/master/myTMVA/TMVAClassification.C

L274-275: Set the pre-filters before training.

```
274     TCut mycuts = "dcandy>-1.&&dcandy<1.&&dcanddau1pt>1.0&&dcanddau2pt>1.0&&(matchedtogen&&nongendoublecounted)&&dcandffls3d>2.0
275     TCut mycutb = "MinBias&&dcandy>-1.&&dcandy<1.&&dcanddau1pt>1.0&&dcanddau2pt>1.0&&(TMath::Abs(dcandmass-1.865)>0.10&&TMath::A
```

L314-316: Set the details of training.

```
314     if (Use["CutsSA"])
315        factory->BookMethod( TMVA::Types::kCuts, "CutsSA",
316                             "!H:!V:FitMethod=SA:EffSel:MaxCalls=150000:KernelTemp=IncAdaptive:InitialTemp=1e+6:MinTemp=1e-6:Eps=
```

especially set the direction of the variable cuts
FMax means var > var_cut, while FMin means var < var_cut

```
:VarProp[0]=FMax:VarProp[1]=FMax"
```

The output (also input of the next step) of TMVA:
https://github.com/boundino/tutorialTMVA/blob/master/myTMVA/weights/TMVAClassification_CutsSA.weights.xml

Macros - readxml

https://github.com/boundino/tutorialTMVA/blob/master/readxml/readxml.cc
https://github.com/boundino/tutorialTMVA/blob/master/readxml/readxml.h

L23-105: Read the info from .xml file

```
23  //read weight file
24  const char* filename = "../myTMVA/weights/TMVAClassification_CutsSA.weights.xml";
25  void *doc = TMVA::gTools().xmlengine().ParseFile(filename,TMVA::gTools().xmlenginebuffersize());
26  void* rootnode = TMVA::gTools().xmlengine().DocGetRootElement(doc); // node "MethodSetup"
27  TString fullMethodName("");
28  TMVA::gTools().ReadAttr(rootnode, "Method", fullMethodName);
29
30  cout<<endl;
31  cout<<" ╔═══════════════════════════════════════════════════════════╗"<<endl;
32  cout<<" |                Cut Opt Configuration                      |"<<endl;
33  cout<<" ├──────────────┬────────────────────────────────────────────┤"<<endl;
34  cout<<" | "<<setiosflags(ios::left)<<setw(10)<<"Method"<<" | "<<setiosflags(ios::left)<<setw(26)<<fullMethodName<<" | "<<seti
35
36  void *opts = TMVA::gTools().GetChild(rootnode,"Options");
37  void* opt = TMVA::gTools().GetChild(opts,"Option");
38
39  TString varProp("");
40  while (opt)
41    {
42      TString optname("");
43      TMVA::gTools().ReadAttr(opt, "name", optname);
44      if (optname=="VarProp") varProp = TMVA::gTools().GetContent(opt);
45      opt = TMVA::gTools().GetNextChild(opt);
46    }
47
48  TObjArray *marginclass = varProp.Tokenize(" ");
49  std::vector<TString> margins;//avoid objarrays
50  for(int i=0;i<marginclass->GetEntries();i++)
51    {
52      margins.push_back(((TObjString *)(marginclass->At(i)))->String());
53    }
54  void* variables = TMVA::gTools().GetChild(rootnode,"Variables");
```

Macros - readxml

https://github.com/boundino/tutorialTMVA/blob/master/readxml/readxml.cc
https://github.com/boundino/tutorialTMVA/blob/master/readxml/readxml.h

L111: Calculate S' and B'

```
110    //
111    calRatio(weights);//weight signal and background
112    //
```

L120-179: Calculate S/sqrt(S+B) and plot [S/sqrt(S+B) - signalEff]

```
120    Double_t max = wSignal*effS[1]/sqrt(wSignal*effS[1]+wBackground*effB[1]);
121    int maxindex = 1;
122    effS[0]=0;
123    for(int i=1;i<100;i++)
124      {
125        effSig[i] = wSignal*effS[i]/sqrt(wSignal*effS[i]+wBackground*effB[i]);
126        if(effSig[i]>max)
127          {
128            max=effSig[i];
129            maxindex=i;
130          }
131      }
132    cout<<endl;
133    cout<<" ╔═══════════════════════════════════════════════════════╗"<<endl;
134    cout<<" |                      Opt Result                       |"<<endl;
135    cout<<" ├──────────┬──────────┬──────────┬──────────┬──────────┤"<<endl;
136    cout<<" | "<<setiosflags(ios::left)<<setw(10)<<"Sig eff"<<" | "<<setiosflags(ios::left)<<setw(10)<<effS[maxindex]<<" | "<<set
```