实验数据多元统计分析

李 晶
北京大学

2016 年 9 月 13 日

粒子物理数据分析基础和前沿研讨会
2016 年 09 月 10 日到 14 日
中国科学院高能物理研究所

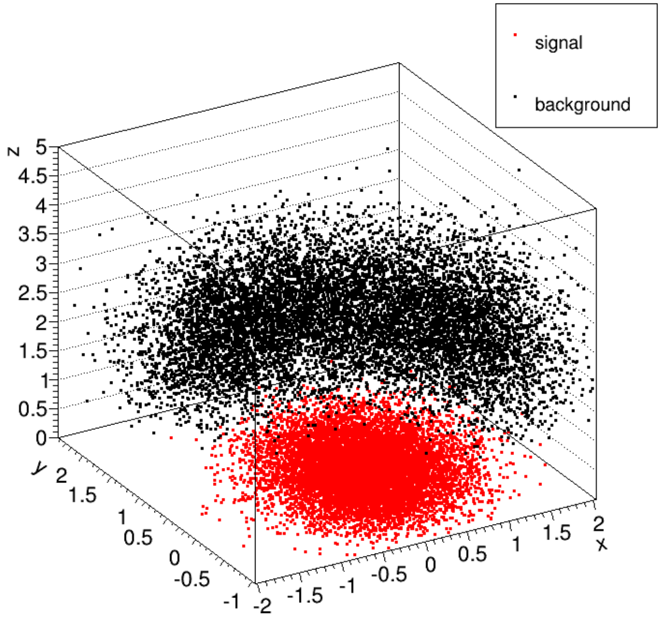
目录

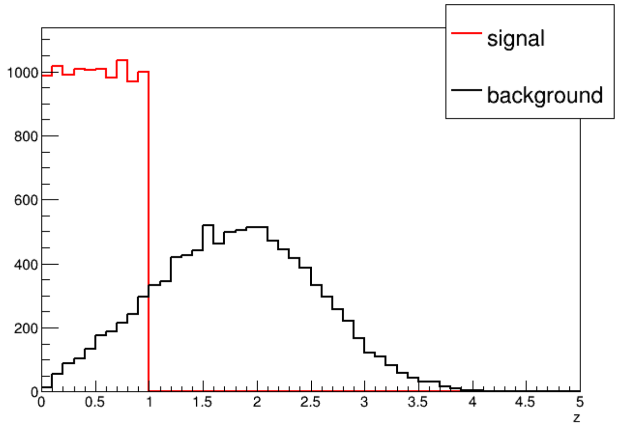
- 背景介绍
- 线性判别方法，Fisher 线性判别
- 人工神经网络
- 决策树判别
- 其他判别方法
- 小结

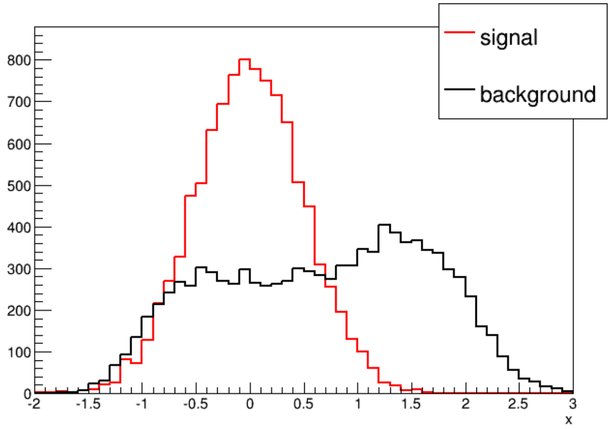
背景介绍

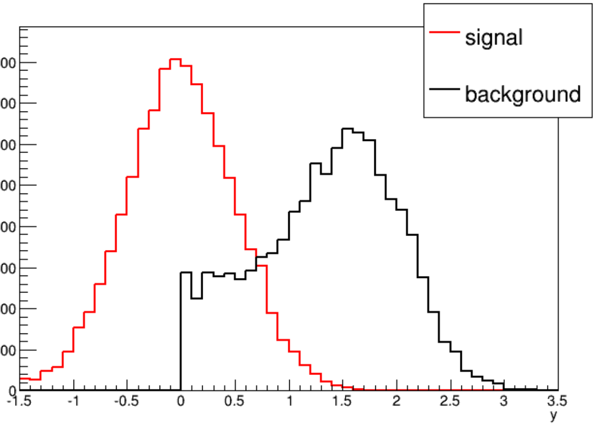
- 实验数据多元统计分析（ Multivariate methods ， MVA ）近年来广泛应用于高能物理实验数据分析中
- 实验中每一个事例采集的数据形成 n 维特征空间中的数据向量的一个样本点 **x**
- 定义一个判别函数（ Test statics ） t(**x**)，由 t(**x**)=t_{cut}确定决策面决策面（ decision boundary ）
  - 把 n 维特征空间分成信号和本底决策域（ critical region ）
  - 根据待识别样本的特征向量的观测值将样本归之为某个类别

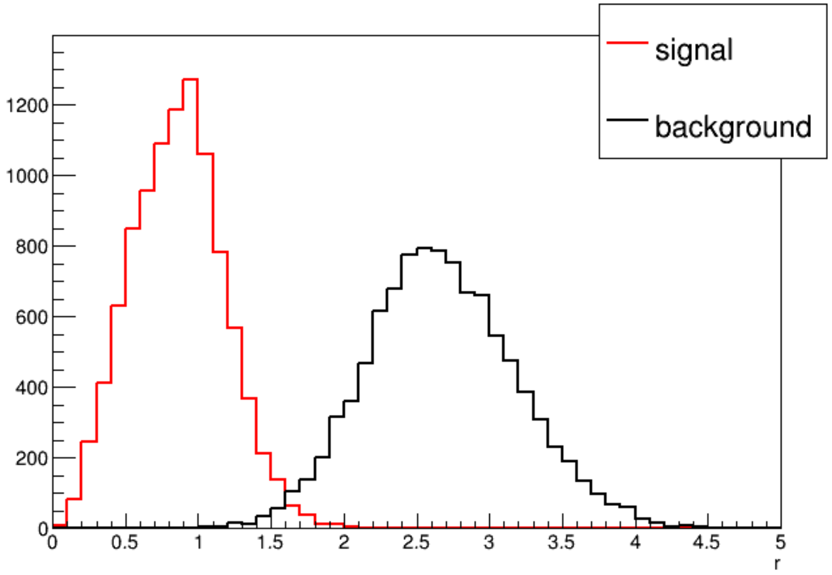
观测量： x, y, z


signal
background
z
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
y
2
1.5
1
0.5
0
-0.5
-1
x
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

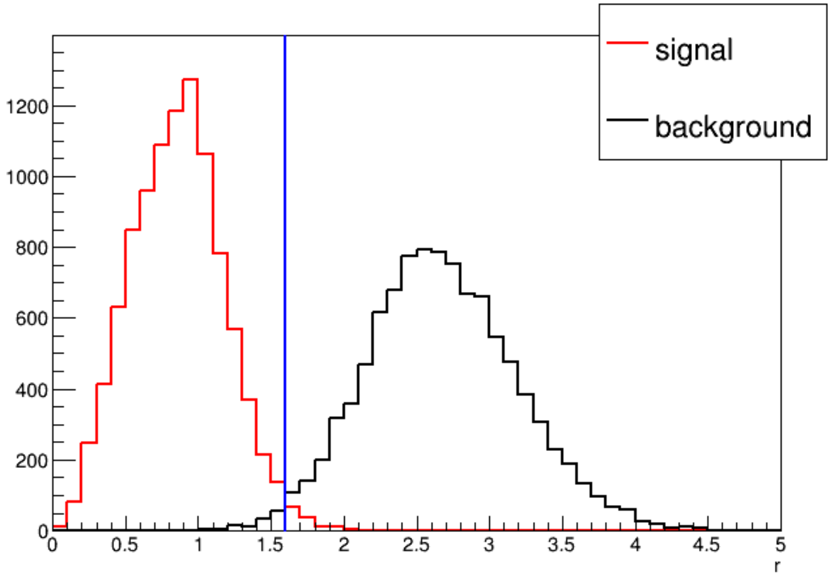

signal
background
1000
800
600
400
200
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
z


signal
background
800
700
600
500
400
300
200
100
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
x


signal
background
800
700
600
500
400
300
200
100
0
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
y

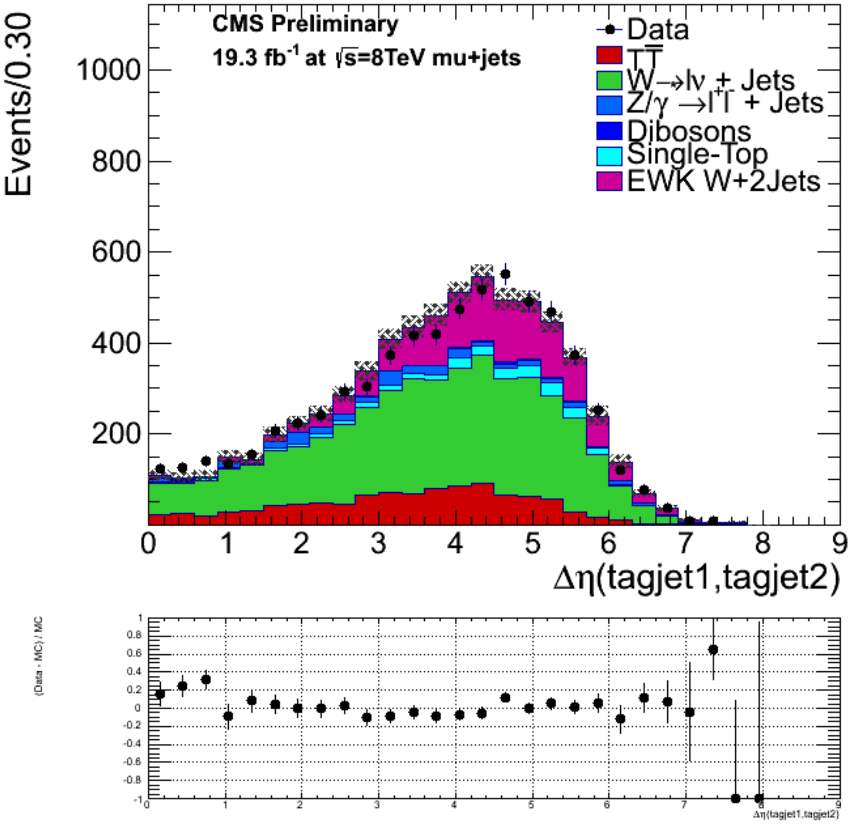


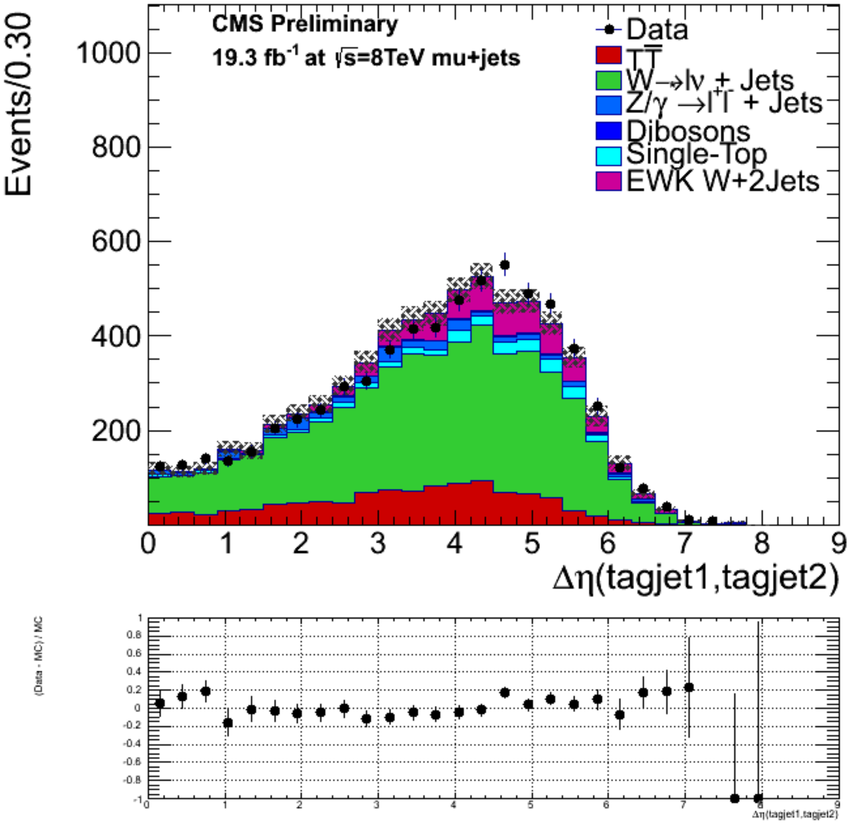
- 判别函数 $r=\mathrm{sqrt}(x^2+y^2+z^2)$



- 判别函数 $r=\mathrm{sqrt}(x^2+y^2+z^2)$
- 决策面 r=1.6
- 决策域
  - 信号： r<1.6
  - 本底： r>1.6

怎样确定决策面和决策域？

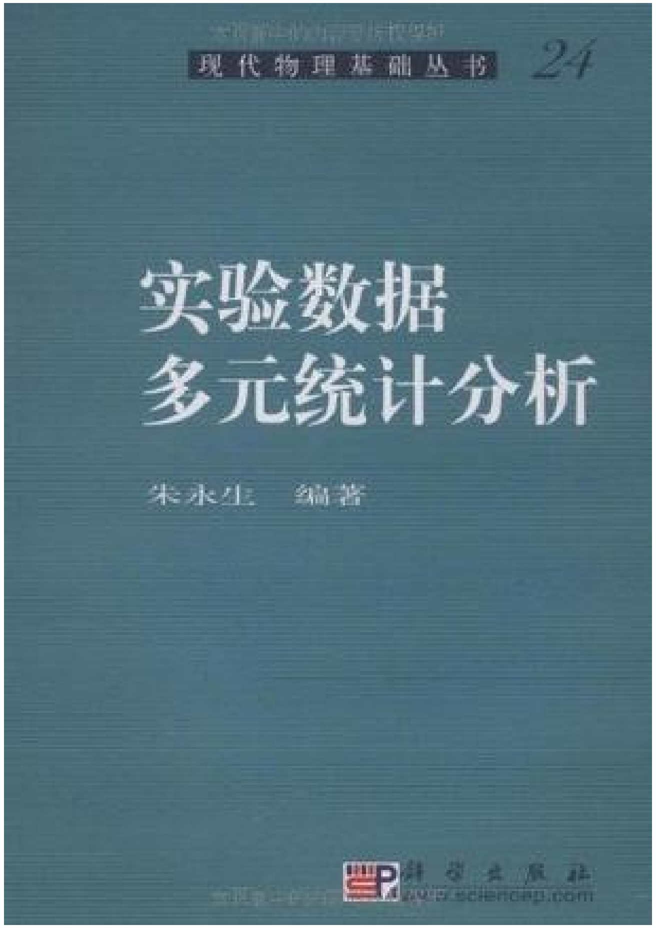




arXiv:1607.06975

- 有时具有区分度的判别函数很难寻找

- 多元统计分析方法
  - 通过统计的方法确定最优化的决策面和决策域
- 线性判别方法： Fisher 线性判别
- 非线性判别方法：人工神经网络，决策树判别



现代物理基础丛书 24

实验数据
多元统计分析

朱永生 编著

科学出版社
www.sciencep.com

- 朱永生（编著）, 实验数据多元统计分析，科学出版社，北京，2009

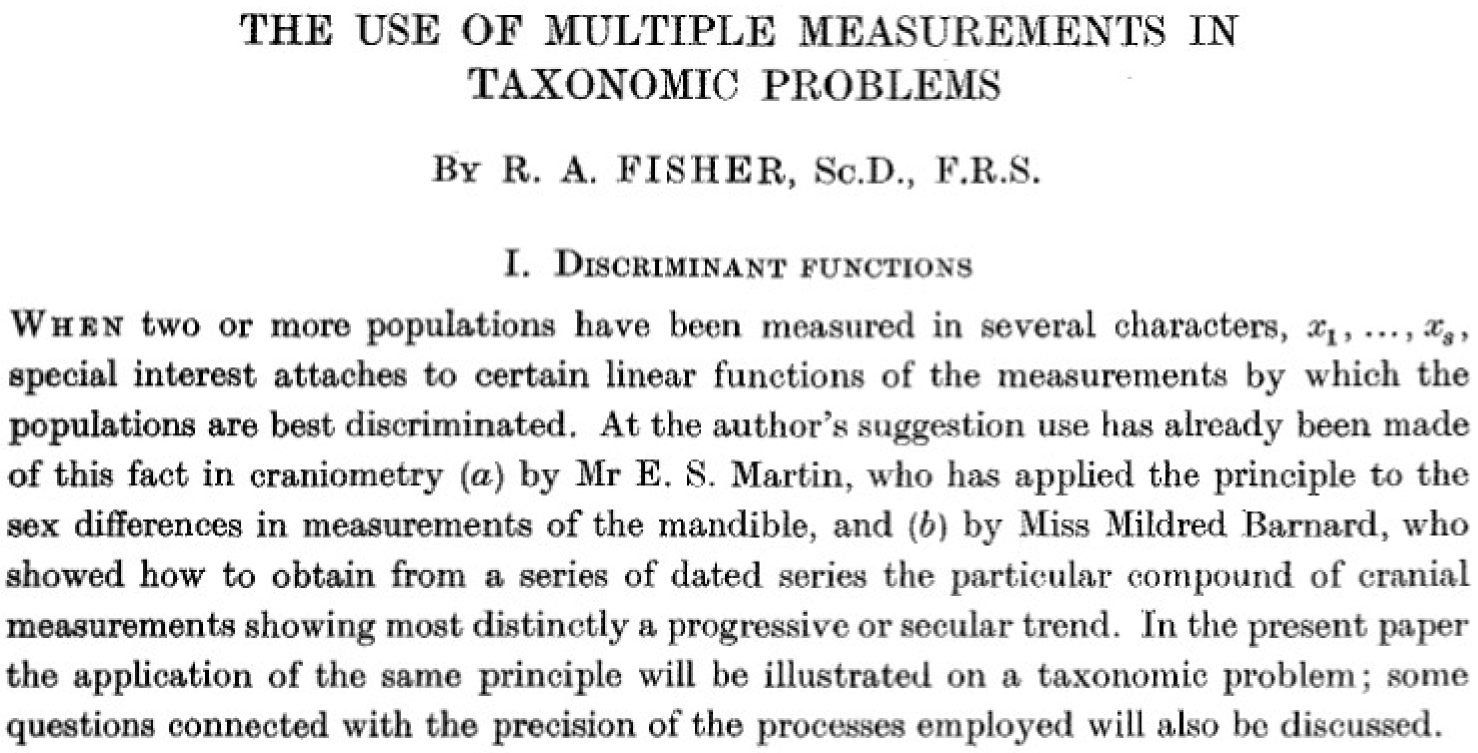
- C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006
- T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, 2nd ed., Springer, 2009
- R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd ed., Wiley, 2001
- A. Webb, Statistical Pattern Recognition, 2nd ed., Wiley, 2002.
- Ilya Narsky and Frank C. Porter, Statistical Analysis Techniques in Particle Physics, Wiley, 2014.

线性判别方法

- 线性判别函数
  - $t(\mathbf{x})=\Sigma w_i x_i + w_0$
- $t(\mathbf{x})=0$ 构成 n 维空间中的（超）平面 → 降低维数
  - 低维空间会给问题的分析和计算带来很多方便，而高维空间往往会使某些解析和计算方法难以实现
- $t(\mathbf{x})>0$，则 $\mathbf{x}\in$ 信号；$t(\mathbf{x})<0$，则 $\mathbf{x}\in$ 本底
- 权向量 **w**: 决策面的法向量
  - 找到某一个方向，使得不同模式的样本在 **w** 上面的投影是最容易区分开的

Fisher 线性判别

- R.A. Fisher, Annals Eugenics 7, 179 (1936).



THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS

By R. A. FISHER, Sc.D., F.R.S.

I. DISCRIMINANT FUNCTIONS

WHEN two or more populations have been measured in several characters, $x_1, \ldots, x_s$, special interest attaches to certain linear functions of the measurements by which the populations are best discriminated. At the author's suggestion use has already been made of this fact in craniometry (*a*) by Mr E. S. Martin, who has applied the principle to the sex differences in measurements of the mandible, and (*b*) by Miss Mildred Barnard, who showed how to obtain from a series of dated series the particular compound of cranial measurements showing most distinctly a progressive or secular trend. In the present paper the application of the same principle will be illustrated on a taxonomic problem; some questions connected with the precision of the processes employed will also be discussed.

Fisher 线性判别

- 我们希望经过投影后
  - 不同类样本尽可能分离的开一些，即两类均值之差越大越好
  - 各类样本内部尽量密集，即类内离散度越小越好
- Fisher 准测函数

$$J(\mathbf{w}) = \frac{(E[y|s] - E[y|b])^2}{V[y|s] + V[y|b]}$$

- 利用∂ J / ∂w$_i$ =0 可以求得当 J 取极大值时的 **w**

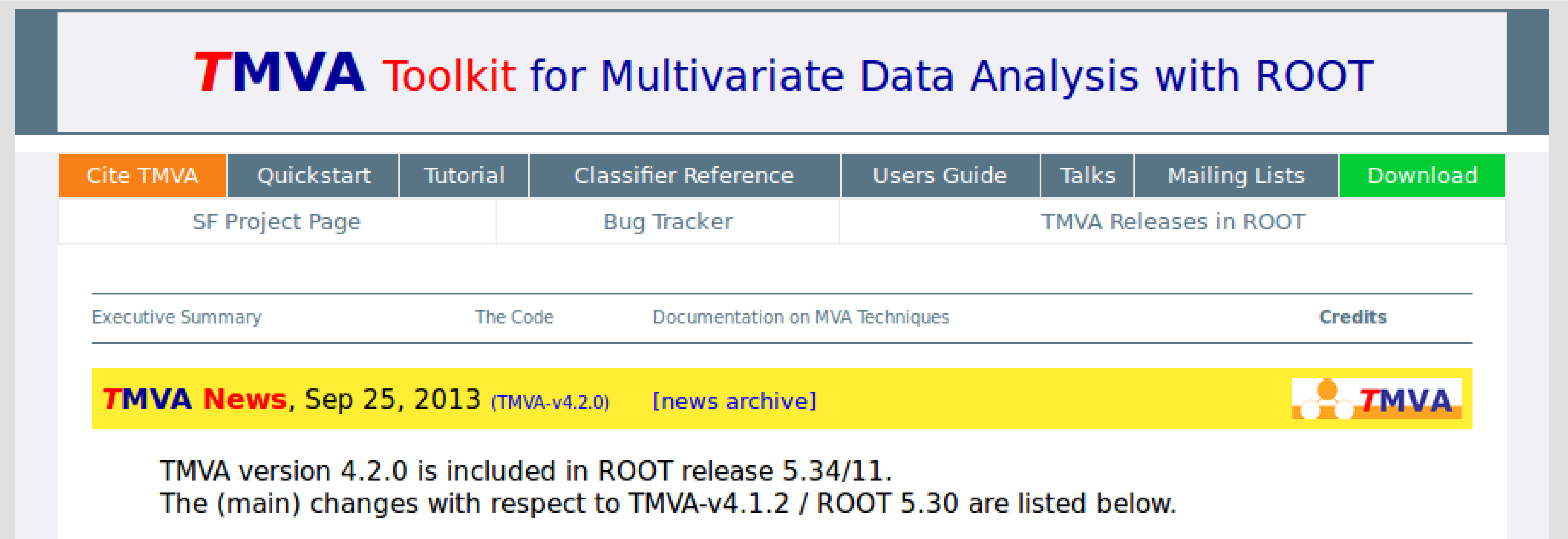
$$\mathbf{w} \propto W^{-1}(\boldsymbol{\mu}_{\mathrm{b}} - \boldsymbol{\mu}_{\mathrm{s}})$$

$$W_{ij} = \mathrm{cov}[x_i, x_j|\mathrm{s}] + \mathrm{cov}[x_i, x_j|\mathrm{b}]$$

$$\mu_{i,\mathrm{s}} = E[x_i|s], \qquad \mu_{i,\mathrm{b}} = E[x_i|b]$$

TMVA package

- Toolkit for multivariate data analysis (TMVA)
- A. Hoecker et al., “TMVA — Toolkit for Multivariate Data Analysis”, (2007). arXiv:physics/0703039.
- http://tmva.sourceforge.net/

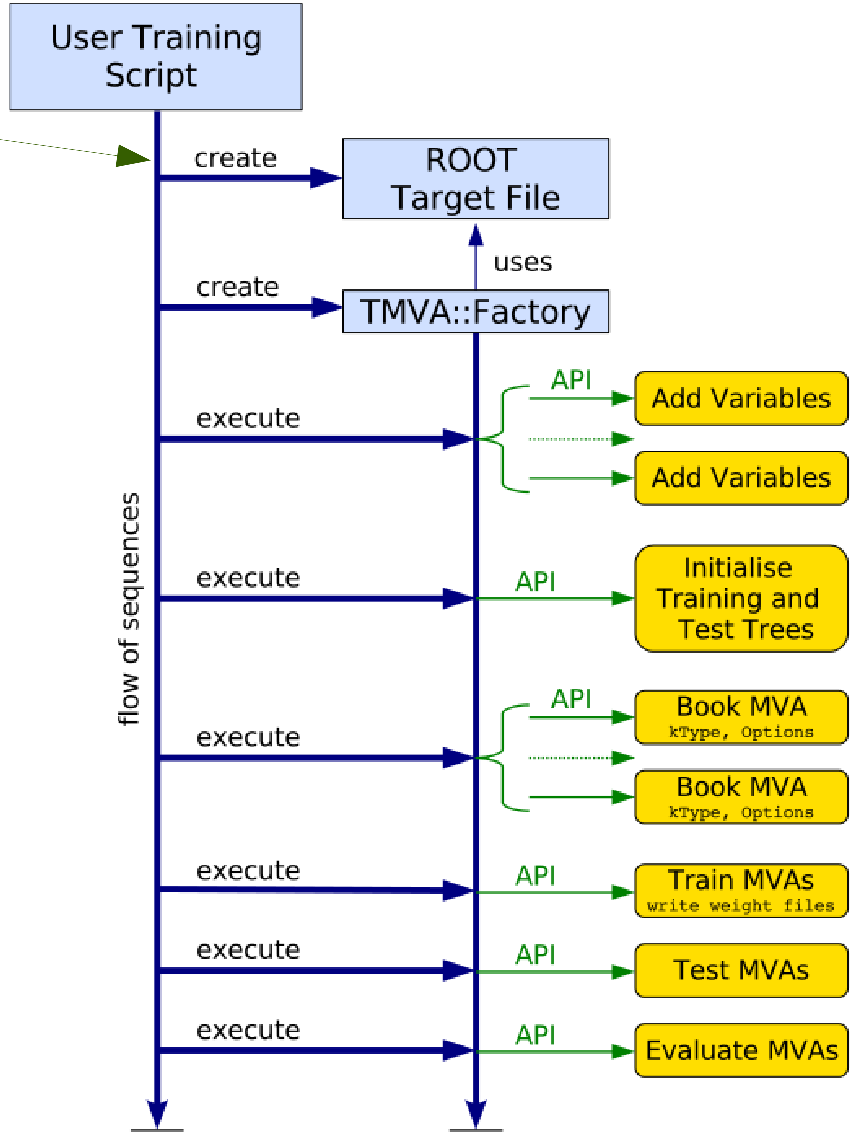


TMVA Toolkit for Multivariate Data Analysis with ROOT

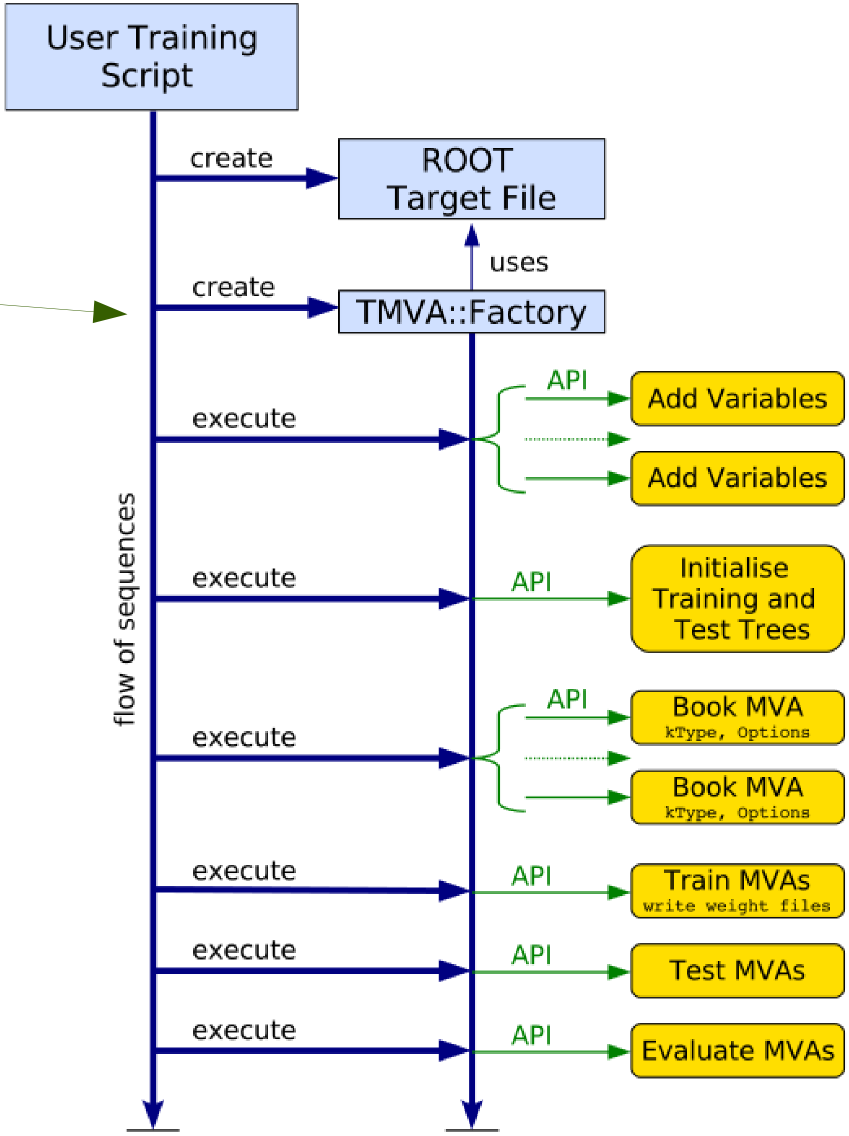
TMVA News, Sep 25, 2013 (TMVA-v4.2.0) [news archive]

TMVA version 4.2.0 is included in ROOT release 5.34/11.
The (main) changes with respect to TMVA-v4.1.2 / ROOT 5.30 are listed below.

```
TString outfileName = Form("TMVAoutput_
%s.root",Label.c_str());
TFile * outputFile = new
Tfile(outfileName,"RECREATE");
std::string factoryOptions( "!V:!
Silent:Transformations=I;P;G" );
TMVA::Factory *factory = new
TMVA::Factory( "TMVAClassificationCategory",
outputFile, factoryOptions );
factory->AddVariable("x",'F');
factory->AddVariable("y",'F');
factory->AddVariable("z",'F');
factory->AddSignalTree    (SigTree,1.);
factory->AddBackgroundTree(BkgTree,1.);
factory->PrepareTrainingAndTestTree
(mycuts,mycutb,"nTrain_Signal=0:nTrain_Background
=0:nTest_Signal=0:nTest_Background=0:SplitMode=R
andom:NormMode=NumEvents:!V");
factory->BookMethod(TMVA::Types::kFisher, "Fisher",
"H:!V:Fisher"); // factory->BookMethod( Types::kFisher,
"Fisher", "<options>" );
factory->TrainAllMethods();
factory->TestAllMethods();
factory->EvaluateAllMethods();
```



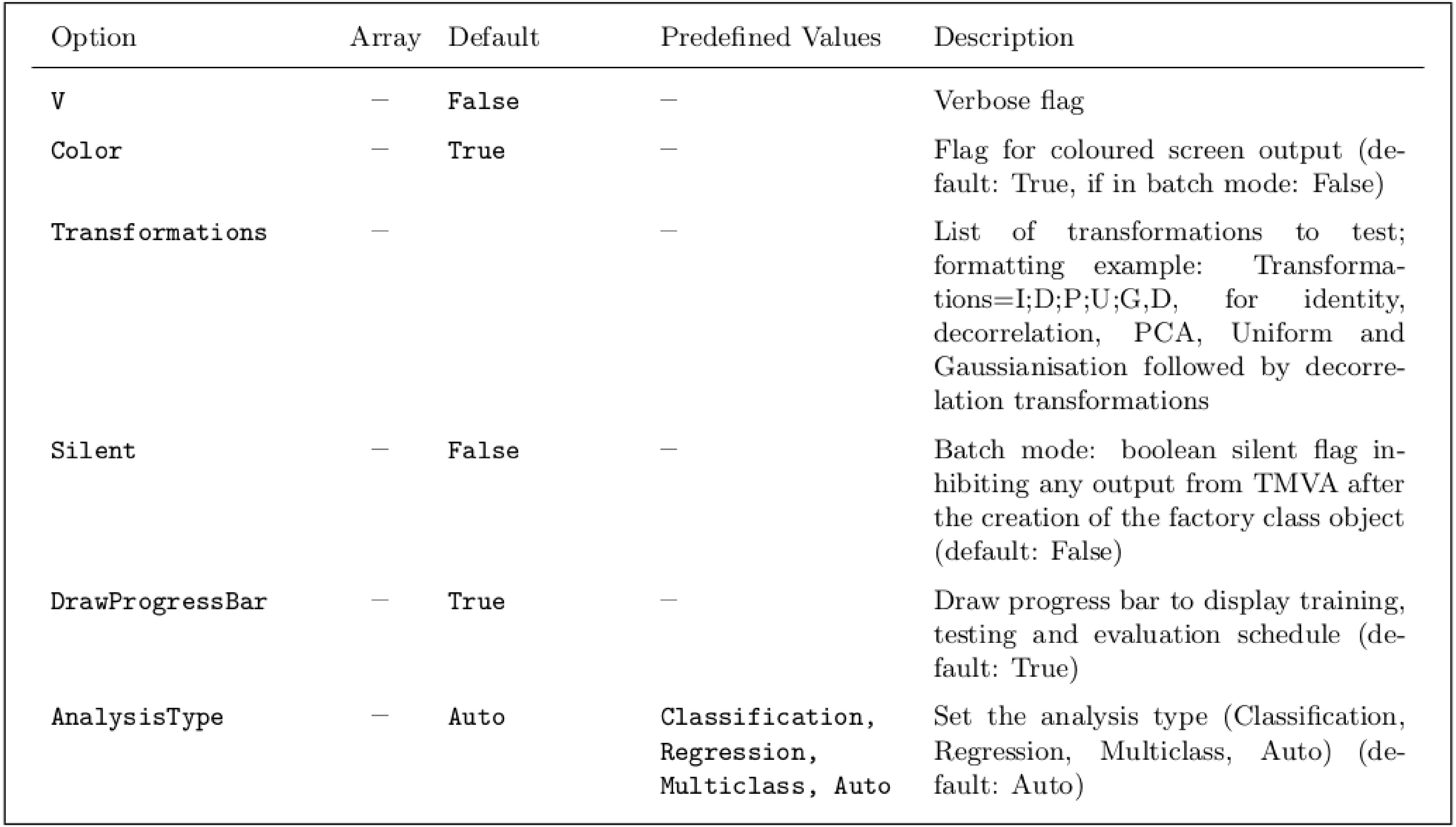
```
TString outfileName = Form("TMVAoutput_
%s.root",Label.c_str());

TFile * outputFile = new
Tfile(outfileName,"RECREATE");

std::string factoryOptions( "!V:!
Silent:Transformations=I;P;G" );

TMVA::Factory *factory = new
TMVA::Factory( "TMVAClassificationCategory",
outputFile, factoryOptions );

factory->AddVariable("x",'F');

factory->AddVariable("y",'F');

factory->AddVariable("z",'F');

factory->AddSignalTree    (SigTree,1.);

factory->AddBackgroundTree(BkgTree,1.);

factory->PrepareTrainingAndTestTree
(mycuts,mycutb,"nTrain_Signal=0:nTrain_Background
=0:nTest_Signal=0:nTest_Background=0:SplitMode=R
andom:NormMode=NumEvents:!V");

factory->BookMethod(TMVA::Types::kFisher, "Fisher",
"H:!V:Fisher"); // factory->BookMethod( Types::kFisher,
"Fisher", "<options>" );

factory->TrainAllMethods();

factory->TestAllMethods();

factory->EvaluateAllMethods();
```



TMVA Users Guide

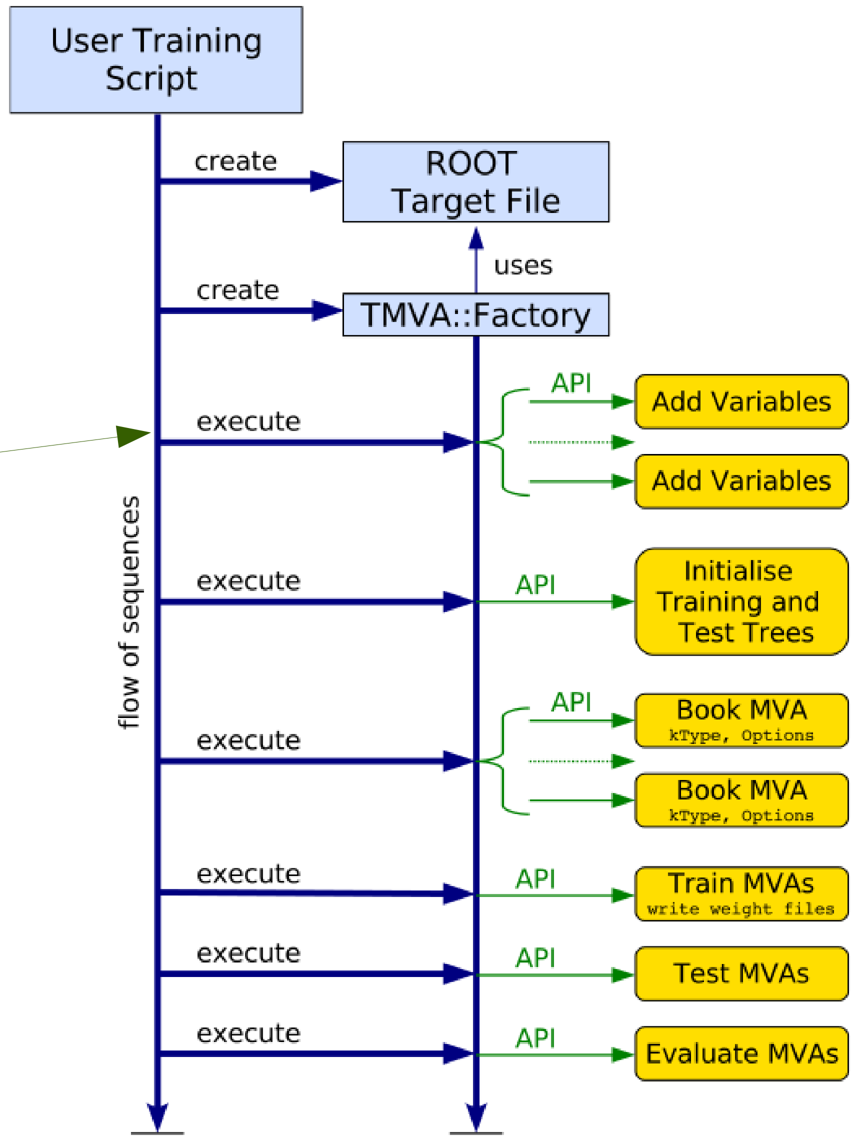
TMVA::Factory* factory = new TMVA::Factory("<JobName>", outputFile, "<options>");

TMVA Users Guide



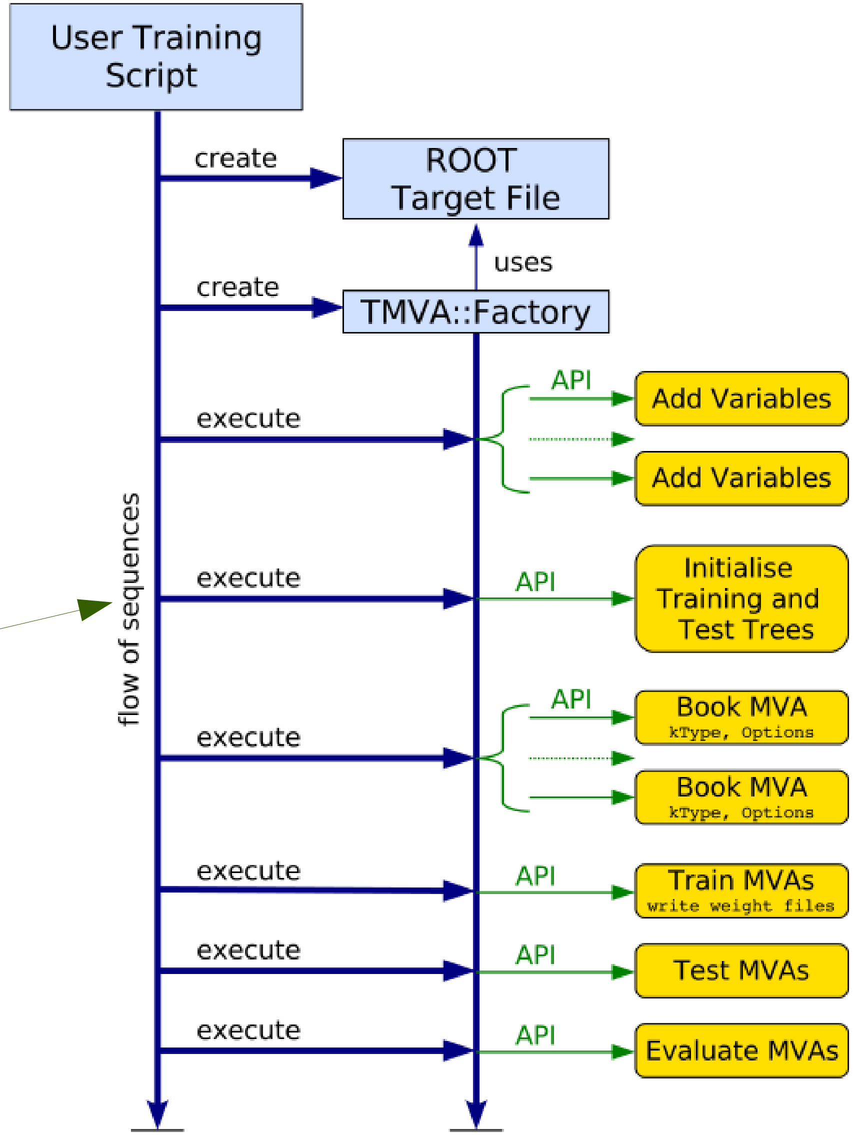
Option	Array	Default	Predefined Values	Description
V	—	False	—	Verbose flag
Color	—	True	—	Flag for coloured screen output (default: True, if in batch mode: False)
Transformations	—		—	List of transformations to test; formatting example: Transformations=I;D;P;U;G,D, for identity, decorrelation, PCA, Uniform and Gaussianisation followed by decorrelation transformations
Silent	—	False	—	Batch mode: boolean silent flag inhibiting any output from TMVA after the creation of the factory class object (default: False)
DrawProgressBar	—	True	—	Draw progress bar to display training, testing and evaluation schedule (default: True)
AnalysisType	—	Auto	Classification, Regression, Multiclass, Auto	Set the analysis type (Classification, Regression, Multiclass, Auto) (default: Auto)

```
TString outfileName = Form("TMVAoutput_
%s.root",Label.c_str());
TFile * outputFile = new
Tfile(outfileName,"RECREATE");
std::string factoryOptions( "!V:!
Silent:Transformations=I;P;G" );
TMVA::Factory *factory = new
TMVA::Factory( "TMVAClassificationCategory",
outputFile, factoryOptions );
factory->AddVariable("x",'F');
factory->AddVariable("y",'F');
factory->AddVariable("z",'F');
factory->AddSignalTree    (SigTree,1.);
factory->AddBackgroundTree(BkgTree,1.);
factory->PrepareTrainingAndTestTree
(mycuts,mycutb,"nTrain_Signal=0:nTrain_Background
=0:nTest_Signal=0:nTest_Background=0:SplitMode=R
andom:NormMode=NumEvents:!V");
factory->BookMethod(TMVA::Types::kFisher, "Fisher",
"H:!V:Fisher"); // factory->BookMethod( Types::kFisher,
"Fisher", "<options>" );
factory->TrainAllMethods();
factory->TestAllMethods();
factory->EvaluateAllMethods();
```

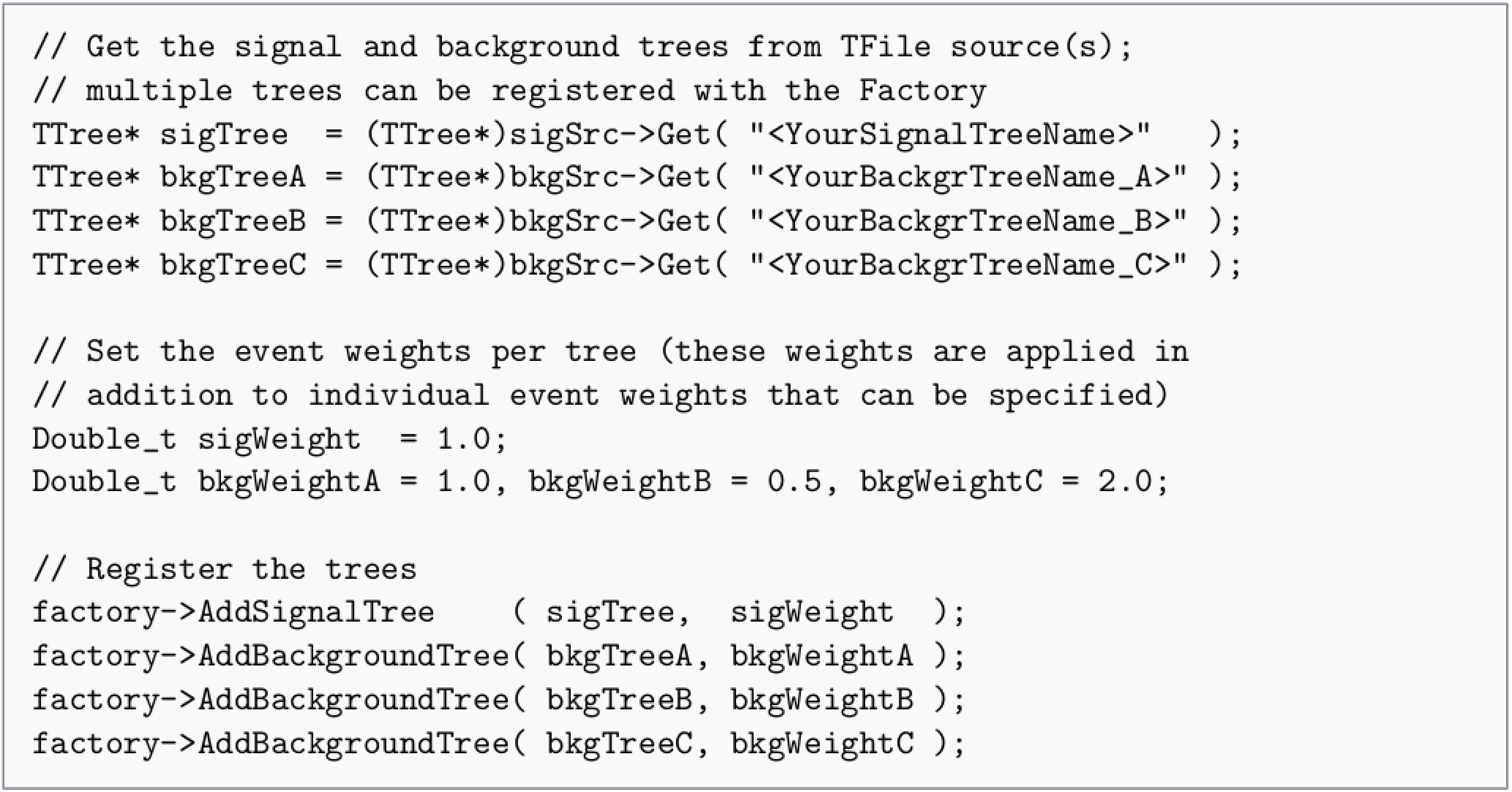


TMVA Users Guide

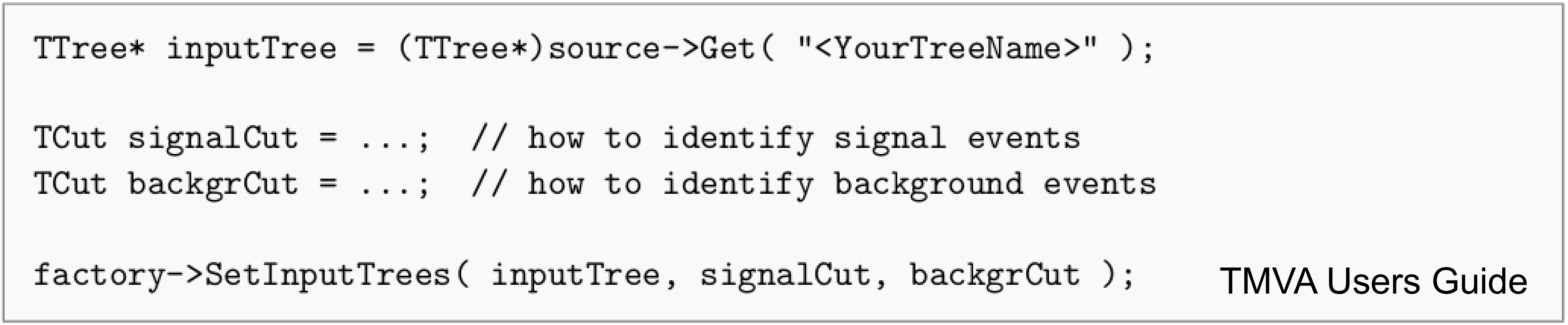
```
TString outfileName = Form("TMVAoutput_
%s.root",Label.c_str());
TFile * outputFile = new
Tfile(outfileName,"RECREATE");
std::string factoryOptions( "!V:!
Silent:Transformations=I;P;G" );
TMVA::Factory *factory = new
TMVA::Factory( "TMVAClassificationCategory",
outputFile, factoryOptions );
factory->AddVariable("x",'F');
factory->AddVariable("y",'F');
factory->AddVariable("z",'F');
factory->AddSignalTree    (SigTree,1.);
factory->AddBackgroundTree(BkgTree,1.);
factory->PrepareTrainingAndTestTree
(mycuts,mycutb,"nTrain_Signal=0:nTrain_Background
=0:nTest_Signal=0:nTest_Background=0:SplitMode=R
andom:NormMode=NumEvents:!V");
factory->BookMethod(TMVA::Types::kFisher, "Fisher",
"H:!V:Fisher"); // factory->BookMethod( Types::kFisher,
"Fisher", "<options>" );
factory->TrainAllMethods();
factory->TestAllMethods();
factory->EvaluateAllMethods();
```



TMVA Users Guide

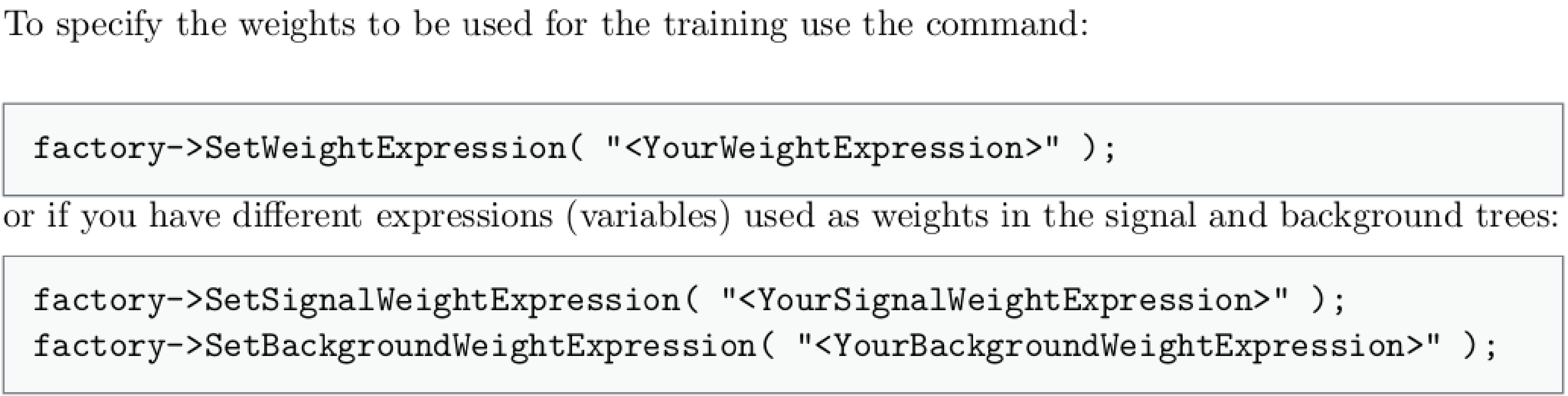


```
// Get the signal and background trees from TFile source(s);
// multiple trees can be registered with the Factory
TTree* sigTree  = (TTree*)sigSrc->Get( "<YourSignalTreeName>"   );
TTree* bkgTreeA = (TTree*)bkgSrc->Get( "<YourBackgrTreeName_A>" );
TTree* bkgTreeB = (TTree*)bkgSrc->Get( "<YourBackgrTreeName_B>" );
TTree* bkgTreeC = (TTree*)bkgSrc->Get( "<YourBackgrTreeName_C>" );

// Set the event weights per tree (these weights are applied in
// addition to individual event weights that can be specified)
Double_t sigWeight  = 1.0;
Double_t bkgWeightA = 1.0, bkgWeightB = 0.5, bkgWeightC = 2.0;

// Register the trees
factory->AddSignalTree    ( sigTree,  sigWeight  );
factory->AddBackgroundTree( bkgTreeA, bkgWeightA );
factory->AddBackgroundTree( bkgTreeB, bkgWeightB );
factory->AddBackgroundTree( bkgTreeC, bkgWeightC );
```



```
TTree* inputTree = (TTree*)source->Get( "<YourTreeName>" );

TCut signalCut = ...;  // how to identify signal events
TCut backgrCut = ...;  // how to identify background events

factory->SetInputTrees( inputTree, signalCut, backgrCut );
```

TMVA Users Guide

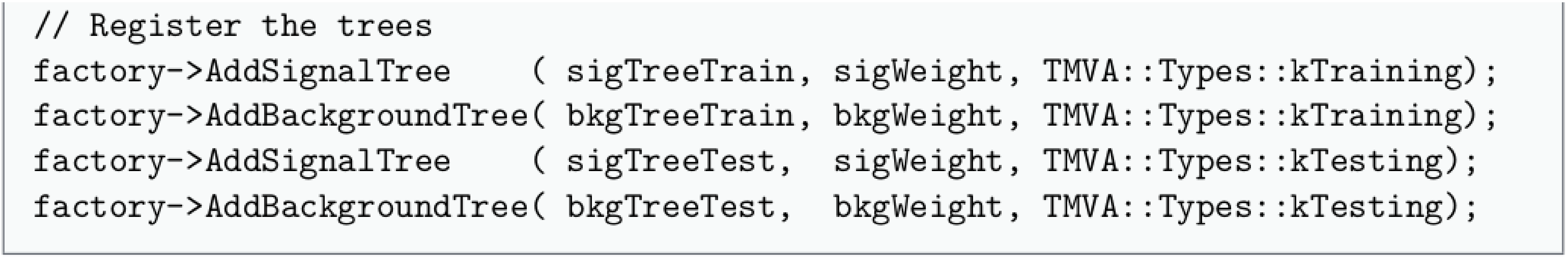


To specify the weights to be used for the training use the command:

```
factory->SetWeightExpression( "<YourWeightExpression>" );
```

or if you have different expressions (variables) used as weights in the signal and background trees:

```
factory->SetSignalWeightExpression( "<YourSignalWeightExpression>" );
factory->SetBackgroundWeightExpression( "<YourBackgroundWeightExpression>" );
```

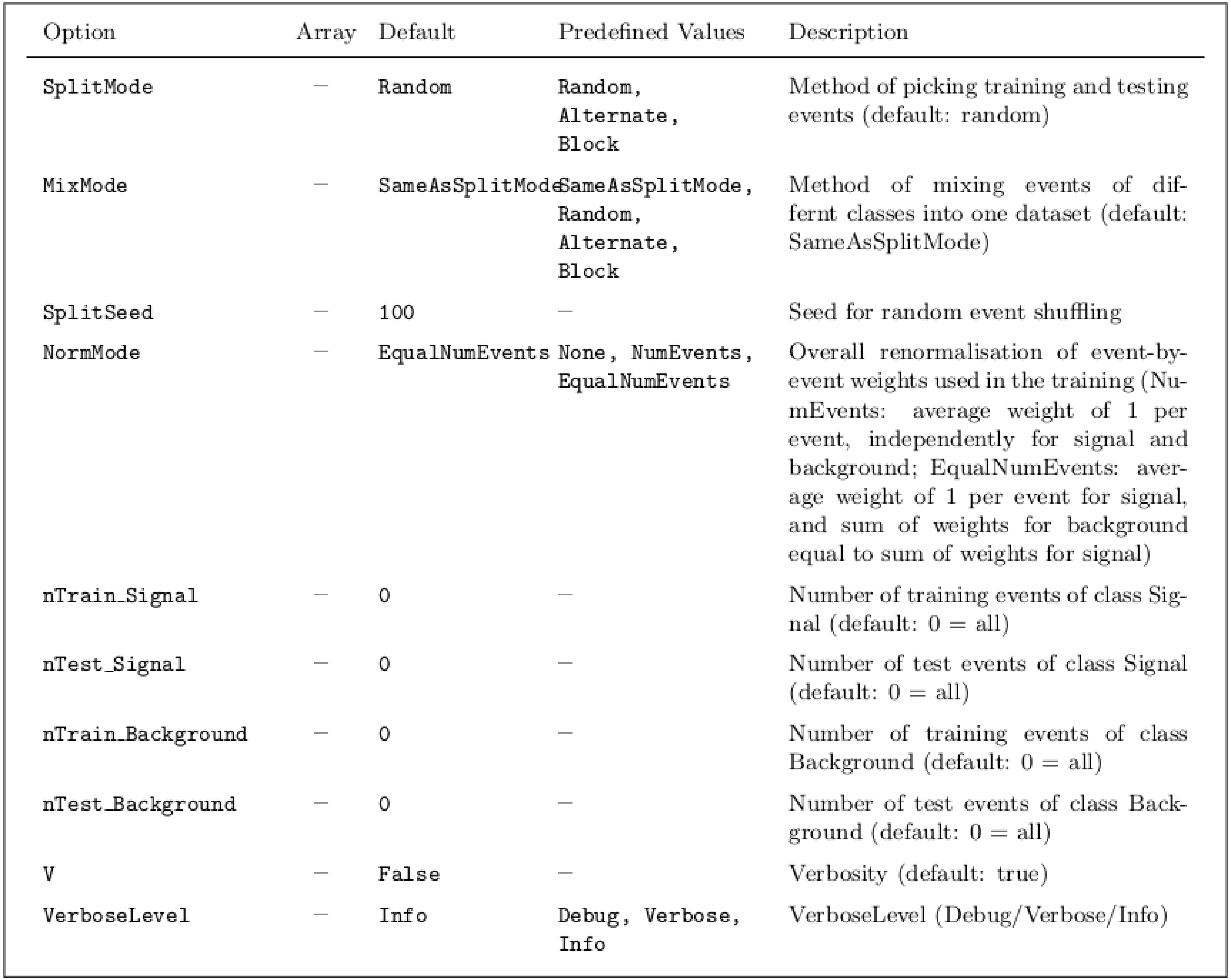


```
// Register the trees
factory->AddSignalTree    ( sigTreeTrain, sigWeight, TMVA::Types::kTraining);
factory->AddBackgroundTree( bkgTreeTrain, bkgWeight, TMVA::Types::kTraining);
factory->AddSignalTree    ( sigTreeTest,  sigWeight, TMVA::Types::kTesting);
factory->AddBackgroundTree( bkgTreeTest,  bkgWeight, TMVA::Types::kTesting);
```

TMVA Users Guide

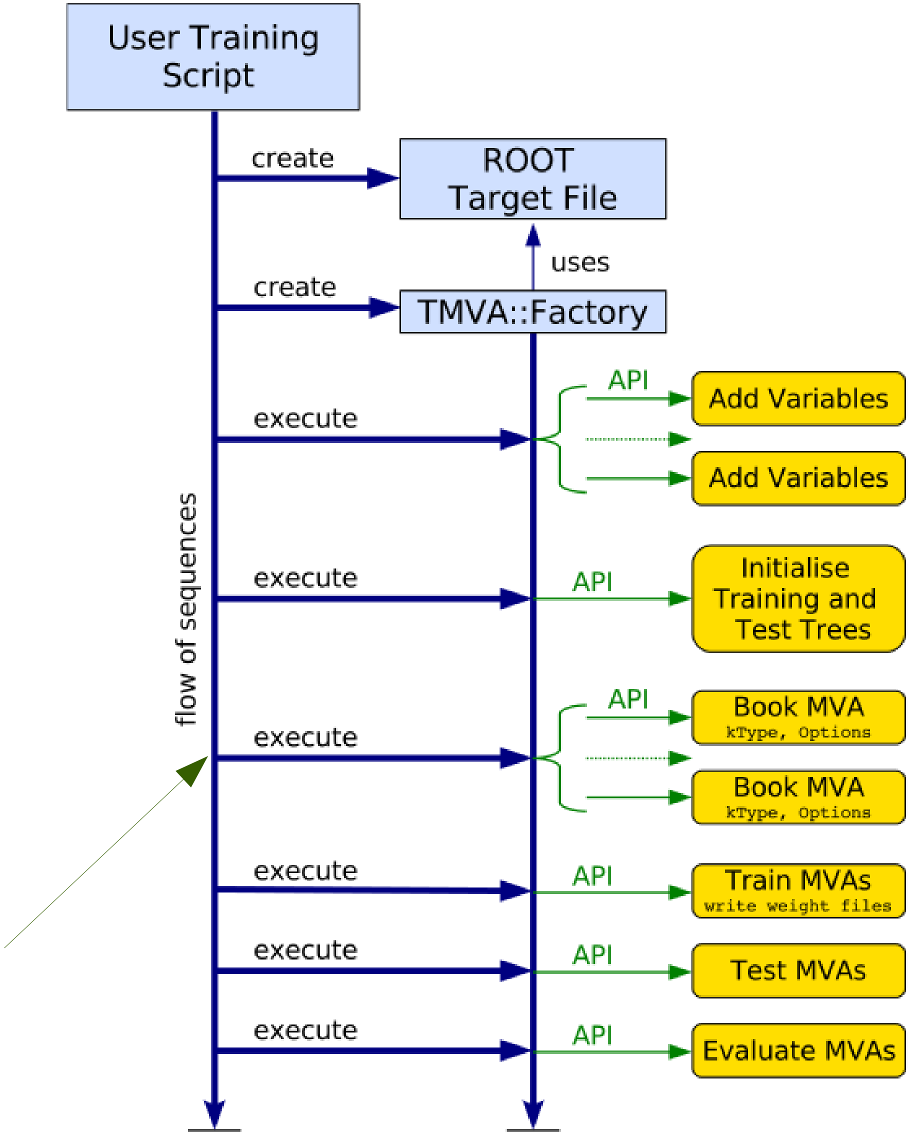
```
factory->PrepareTrainingAndTestTree( preselectionCut,
"<options>" );
```

TMVA Users Guide



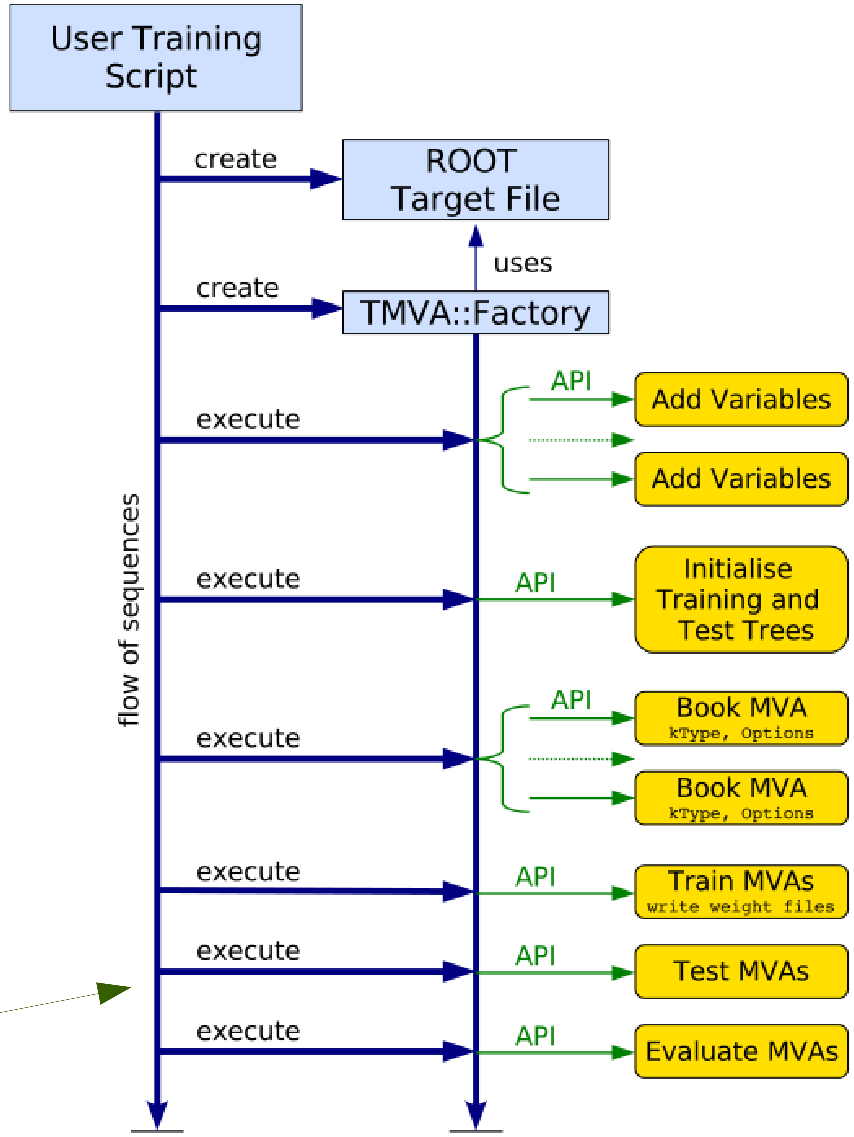
Option	Array	Default	Predefined Values	Description
SplitMode	–	Random	Random, Alternate, Block	Method of picking training and testing events (default: random)
MixMode	–	SameAsSplitMode	SameAsSplitMode, Random, Alternate, Block	Method of mixing events of differnt classes into one dataset (default: SameAsSplitMode)
SplitSeed	–	100	–	Seed for random event shuffling
NormMode	–	EqualNumEvents	None, NumEvents, EqualNumEvents	Overall renormalisation of event-by-event weights used in the training (NumEvents: average weight of 1 per event, independently for signal and background; EqualNumEvents: average weight of 1 per event for signal, and sum of weights for background equal to sum of weights for signal)
nTrain_Signal	–	0	–	Number of training events of class Signal (default: 0 = all)
nTest_Signal	–	0	–	Number of test events of class Signal (default: 0 = all)
nTrain_Background	–	0	–	Number of training events of class Background (default: 0 = all)
nTest_Background	–	0	–	Number of test events of class Background (default: 0 = all)
V	–	False	–	Verbosity (default: true)
VerboseLevel	–	Info	Debug, Verbose, Info	VerboseLevel (Debug/Verbose/Info)

```
TString outfileName = Form("TMVAoutput_
%s.root",Label.c_str());

TFile * outputFile = new
Tfile(outfileName,"RECREATE");

std::string factoryOptions( "!V:!
Silent:Transformations=I;P;G" );

TMVA::Factory *factory = new
TMVA::Factory( "TMVAClassificationCategory",
outputFile, factoryOptions );

factory->AddVariable("x",'F');

factory->AddVariable("y",'F');

factory->AddVariable("z",'F');

factory->AddSignalTree    (SigTree,1.);

factory->AddBackgroundTree(BkgTree,1.);

factory->PrepareTrainingAndTestTree
(mycuts,mycutb,"nTrain_Signal=0:nTrain_Background
=0:nTest_Signal=0:nTest_Background=0:SplitMode=R
andom:NormMode=NumEvents:!V");

factory->BookMethod(TMVA::Types::kFisher, "Fisher",
"H:!V:Fisher"); // factory->BookMethod( Types::kFisher,
"Fisher", "<options>" );

factory->TrainAllMethods();

factory->TestAllMethods();

factory->EvaluateAllMethods();
```



TMVA Users Guide

```
TString outfileName = Form("TMVAoutput_
%s.root",Label.c_str());
TFile * outputFile = new
Tfile(outfileName,"RECREATE");
std::string factoryOptions( "!V:!
Silent:Transformations=I;P;G" );
TMVA::Factory *factory = new
TMVA::Factory( "TMVAClassificationCategory",
outputFile, factoryOptions );
factory->AddVariable("x",'F');
factory->AddVariable("y",'F');
factory->AddVariable("z",'F');
factory->AddSignalTree    (SigTree,1.);
factory->AddBackgroundTree(BkgTree,1.);
factory->PrepareTrainingAndTestTree
(mycuts,mycutb,"nTrain_Signal=0:nTrain_Background
=0:nTest_Signal=0:nTest_Background=0:SplitMode=R
andom:NormMode=NumEvents:!V");
factory->BookMethod(TMVA::Types::kFisher, "Fisher",
"H:!V:Fisher"); // factory->BookMethod( Types::kFisher,
"Fisher", "<options>" );
factory->TrainAllMethods();
factory->TestAllMethods();
factory->EvaluateAllMethods();
```



User Training Script

create

ROOT Target File

uses

create

TMVA::Factory

flow of sequences

execute

API

Add Variables

Add Variables

execute

API

Initialise Training and Test Trees

execute

API

Book MVA kType, Options

Book MVA kType, Options

execute

API

Train MVAs write weight files

execute

API

Test MVAs

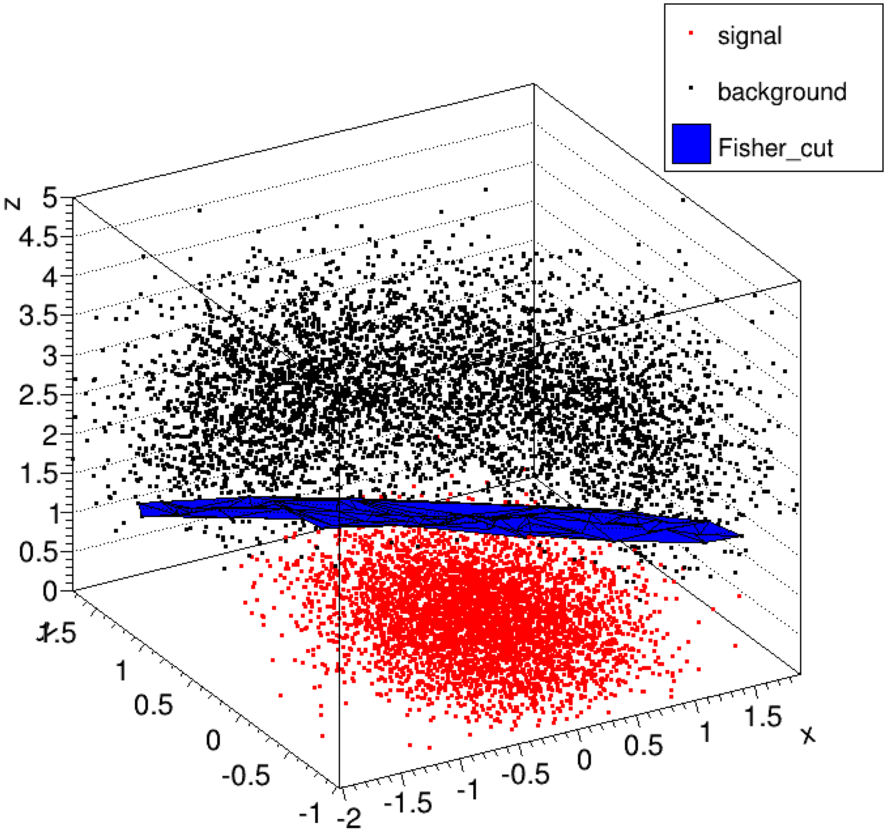
execute

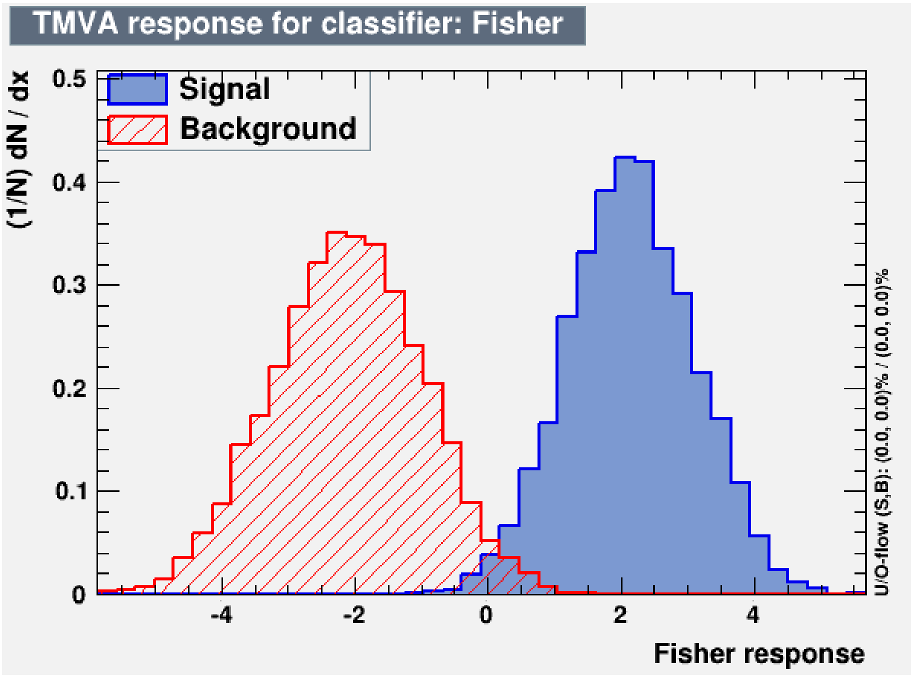
API

Evaluate MVAs

TMVA Users Guide

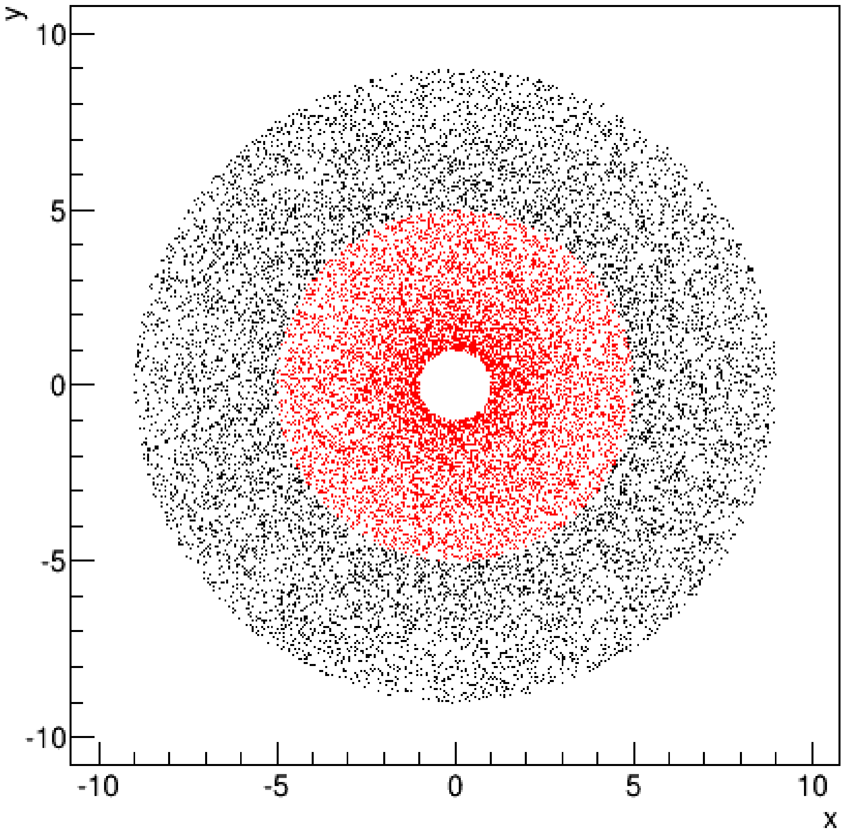
Output of TMVA



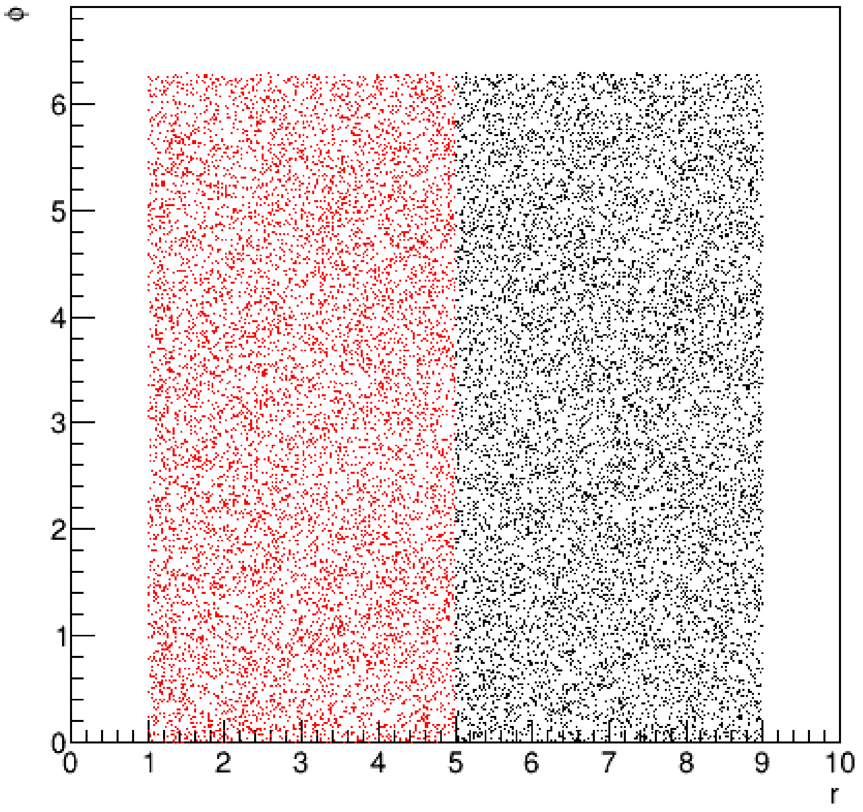


- 线性方法作为判别函数，方法简单，容易实现，计算量和数据存储量小

- 并不是所有样本都是线性可分的



- 广义线性判别函数
  - $x_1, x_2, \ldots, x_n \rightarrow g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x})$
  - $t(\mathbf{x})=\Sigma v_i g(\mathbf{x})_i + w_0$
- 比如：
  - $r = \mathrm{sqrt}(x^2+y^2)$
  - $\Phi = \tan^{-1}(y/x)$



人工神经网络

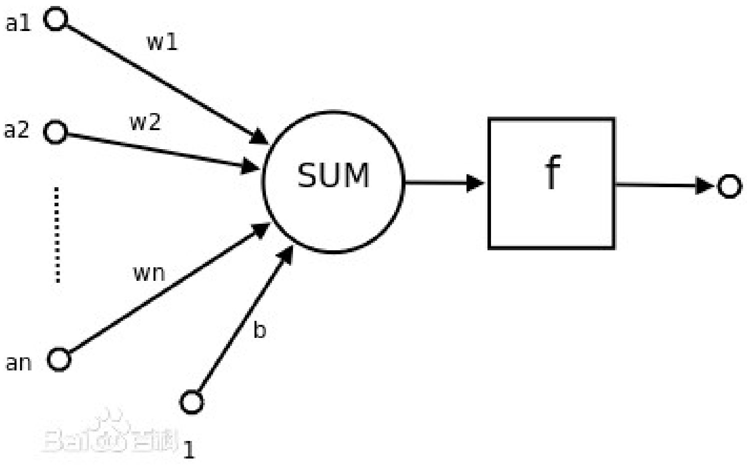
- 人工神经网络（ Artificial Neural Networks ）是对动物神经网络的结构、特性及功能进行理论抽象、简化、模拟而构建的一种信息处理系统
- 自适应非线性动态系统
- 研究内容相当广泛，应用于多个领域

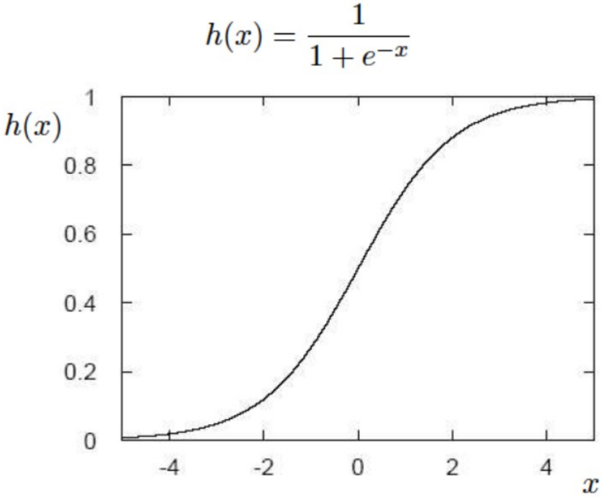
人工神经元

- 人工神经元是一个数学模型，模拟生物神经元的信息传递和处理功能
  - 多输入、单数出、非线性
  - 输出响应是所有输入的综合累加作用的结果，输入或输出分为兴奋型（正值）和抑制型（负值）两种
  - 输出强度可调节
- 1943 年由美国心理学家 McCulloch 和数学家 Pitts 提出的形式神经元模型（ MP 模型）

人工神经元

- 图中n个输入a_1, … , a_n相当与其他n个神经元对于神经元的输入值，n个权值w_1, … , w_n相当于突触的连接强度，SUM表示该神经元对于n个输入信号的累加，f表示神经元对于n个输入信号的响应，称为变换函数或激活函数。记x_0是该神经元的阈值，w_0=-1。采用如下记号
  - $\mathrm{SUM} = \Sigma_{i=1}^{n} w_i x_i - x_0 = \Sigma_{i=0}^{n} w_i x_i$
- 则神经元的输出值可表示为
  - y=f(SUM)
- 常见的变换函数有线性函数、符号函数、饱和函数、双曲线正切函数、阶跃函数、 simoid函数等
  - 要求：单调
  - 一些算法要求f可微

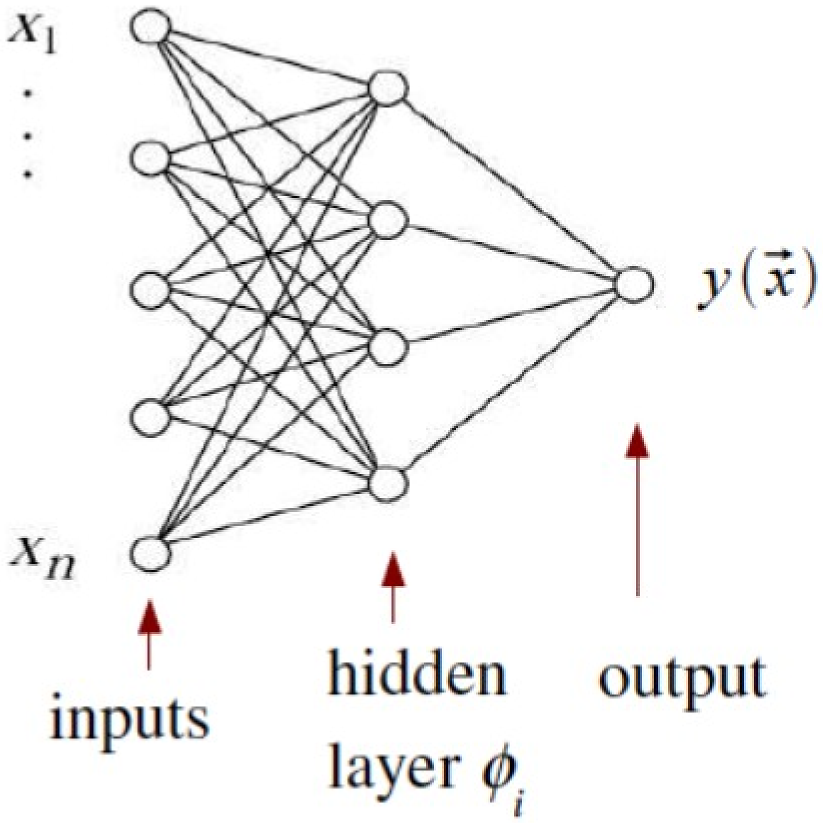




人工神经网络

- 由大量神经元相互连接而成的复杂系统
- 全互联型神经网络：每一个神经元都可以与其他所有神经元发生相互作用
- 阶层型神经网络：每一个神经元只能与相邻层的神经元发生相互作用，而与本层的神经元不发生信息传递

多输出单元感知器（ Multilayer perceptron , MLP ）



$$\varphi_i(\vec{x}) = h\left(w_{i0}^{(1)} + \sum_{j=1}^{n} w_{ij}^{(1)} x_j \right)$$

$$y(\vec{x}) = h\left(w_{10}^{(2)} + \sum_{j=1}^{n} w_{1j}^{(2)} \varphi_j(\vec{x}) \right)$$

G. Cowan

- “An MLP with a single hidden layer having a sufficiently large number of nodes can approximate arbitrarily well the optimal decision boundary.”
  - M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function,” Neural Networks, vol. 6, no. 6, pp. 861–867, 1993.

学习规则（ Network training ）

- 对于每一个训练样本，已知

$\vec{x}_a=(x_1,\ldots,x_n)$ the input variables, and

$t_a=0,1$ a numerical label for event type (“target value”)

- (1) 设置初值： 将连接权 w_{ij} 赋予 (-1, 1) 区间内的随机值
- (2) 输入每一个样本 x_a 和它的期望输出 t_a
- (3) 计算输出单元输出
- (4) 修正连接权各元素
- (5) 迭代 3-4 直到连接权不变为止

$$E(\mathbf{w})=\frac{1}{2}\sum_{a=1}^{N}|y(\vec{x}_a,\mathbf{w})-t_a|^2=\sum_{a=1}^{N}E_a(\mathbf{w})$$

$$\mathbf{w}^{(\tau+1)}=\mathbf{w}^{(\tau)}-\eta\nabla E(\mathbf{w}^{(\tau)})$$

- 容易陷入局部极小点，能否收敛取决于初始值
- 决定收敛速度的 η 的确定依赖于尝试和经验
- 收敛速度慢

学习规则——改进版

- 对于每一个训练样本，已知

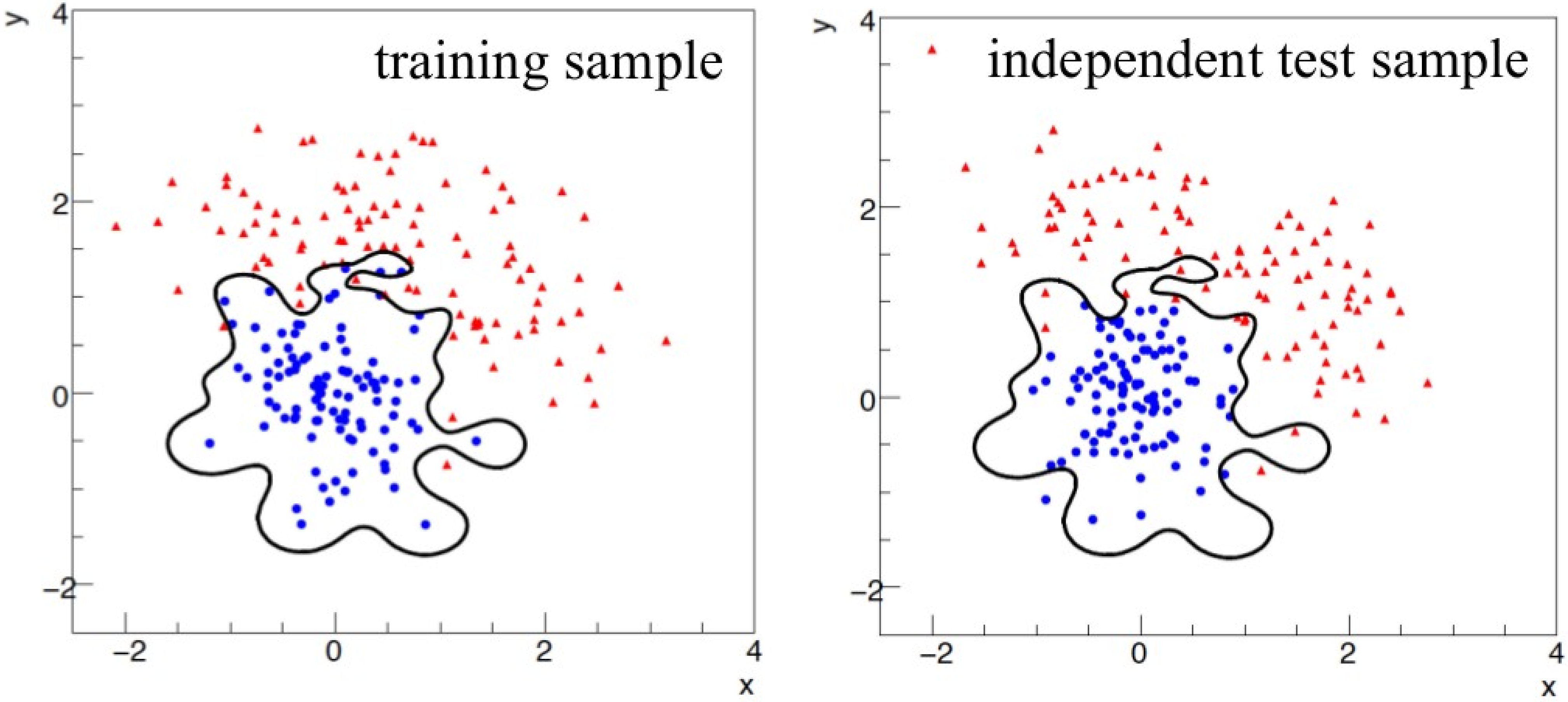
$\vec{x}_a=(x_1,\ldots,x_n)$ the input variables, and

$t_a=0,1$ a numerical label for event type (“target value”)

- (1) 设置初值： 将连接权 w_{ij} 赋予 (-1, 1) 区间内的随机值
- (2) 输入一个样本 x_a 和它的期望输出 t_a
- (3) 计算输出单元输出
- (4) 修正连接权各元素
- (5) 迭代 3-4 直到连接权不变为止
- (6) 迭代 2-5 直到连接权不变为止

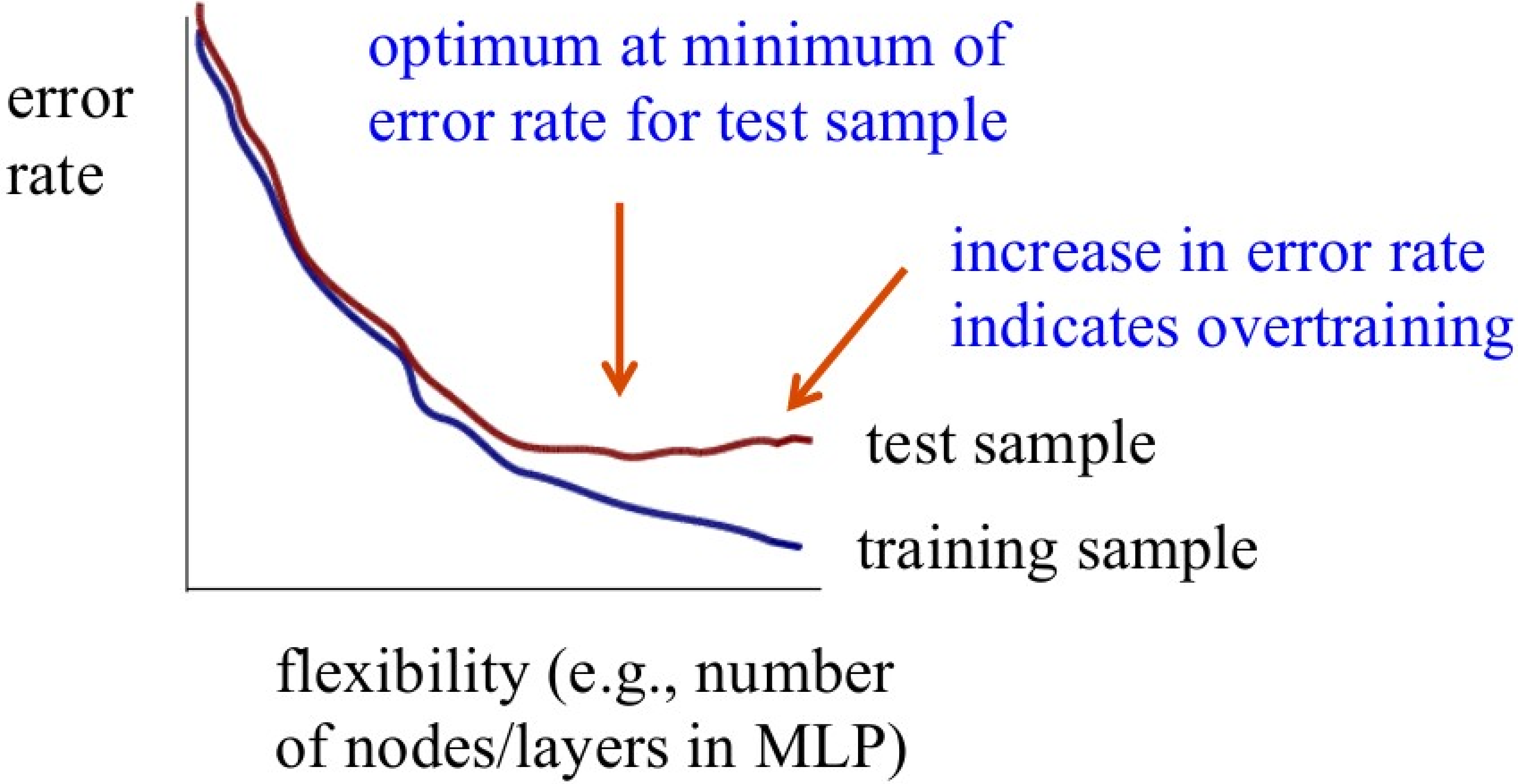
$$\boldsymbol{w}^{(\tau+1)}=\boldsymbol{w}^{(\tau)}-\eta\nabla E_a(\boldsymbol{w}^{(\tau)})$$

Overtraining



G. Cowan

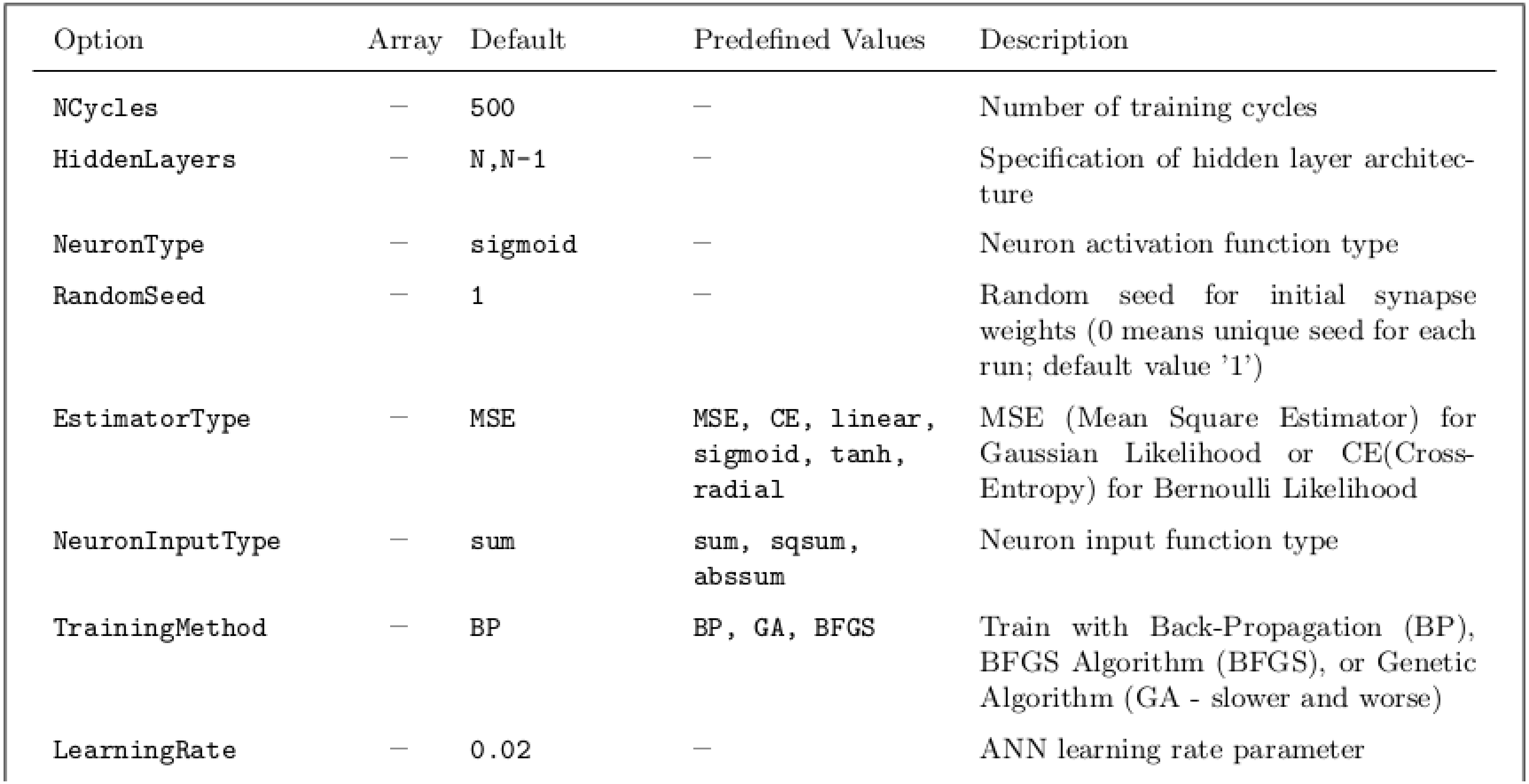
Monitoring overtraining



optimum at minimum of error rate for test sample

increase in error rate indicates overtraining

factory->BookMethod(Types::kMLP, "MLP_ANN", "<options>");

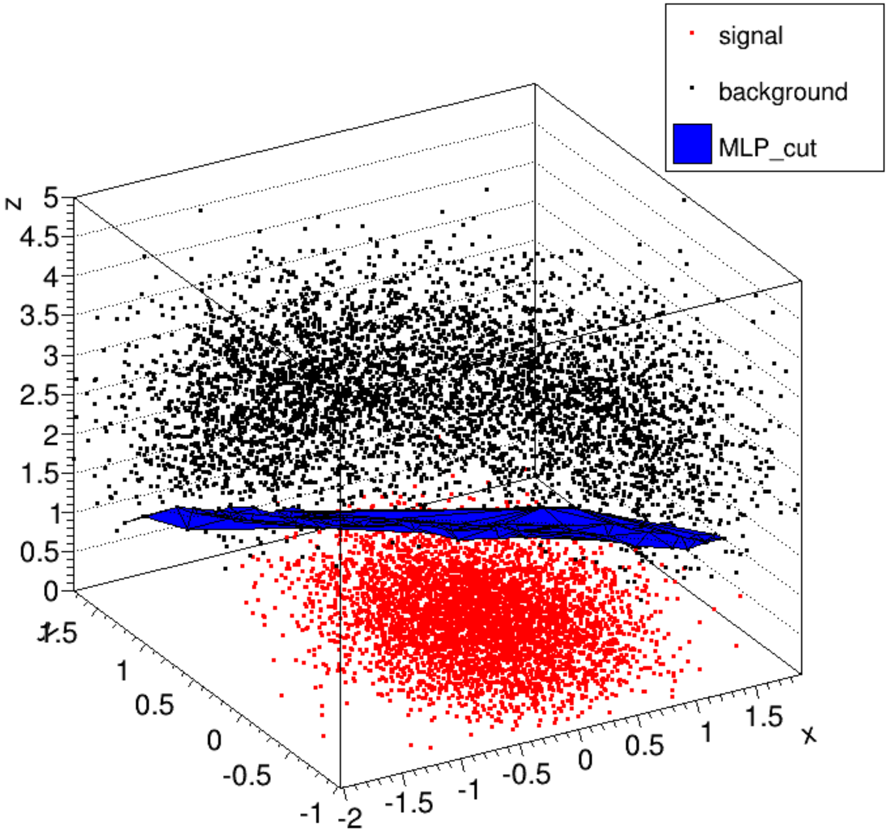


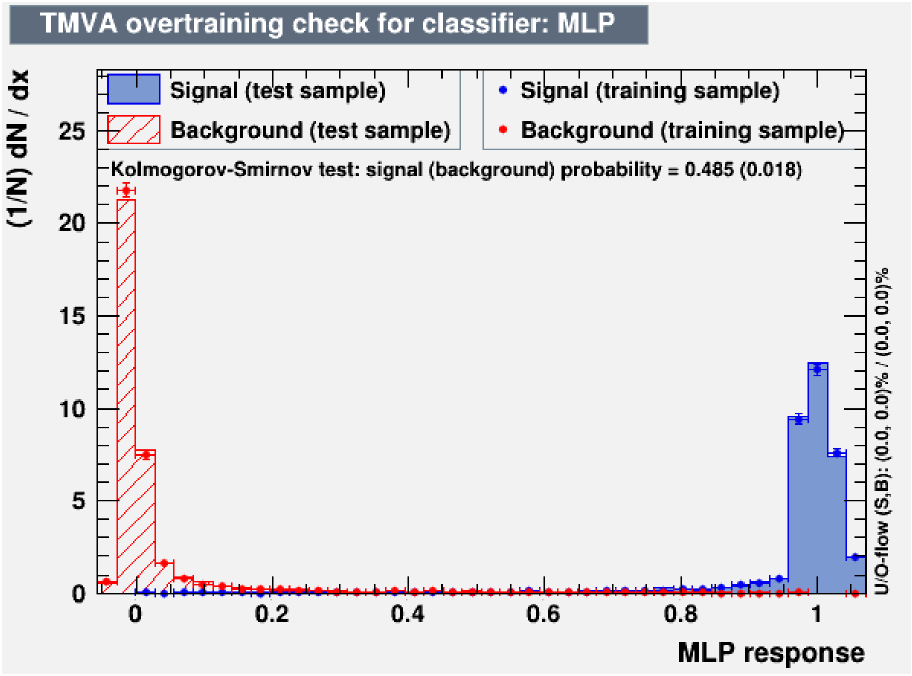
Option	Array	Default	Predefined Values	Description
NCycles	–	500	–	Number of training cycles
HiddenLayers	–	N,N-1	–	Specification of hidden layer architecture
NeuronType	–	sigmoid	–	Neuron activation function type
RandomSeed	–	1	–	Random seed for initial synapse weights (0 means unique seed for each run; default value '1')
EstimatorType	–	MSE	MSE, CE, linear, sigmoid, tanh, radial	MSE (Mean Square Estimator) for Gaussian Likelihood or CE(Cross-Entropy) for Bernoulli Likelihood
NeuronInputType	–	sum	sum, sqsum, abssum	Neuron input function type
TrainingMethod	–	BP	BP, GA, BFGS	Train with Back-Propagation (BP), BFGS Algorithm (BFGS), or Genetic Algorithm (GA - slower and worse)
LearningRate	–	0.02	–	ANN learning rate parameter

... ...

TMVA Users Guide

Output of TMVA

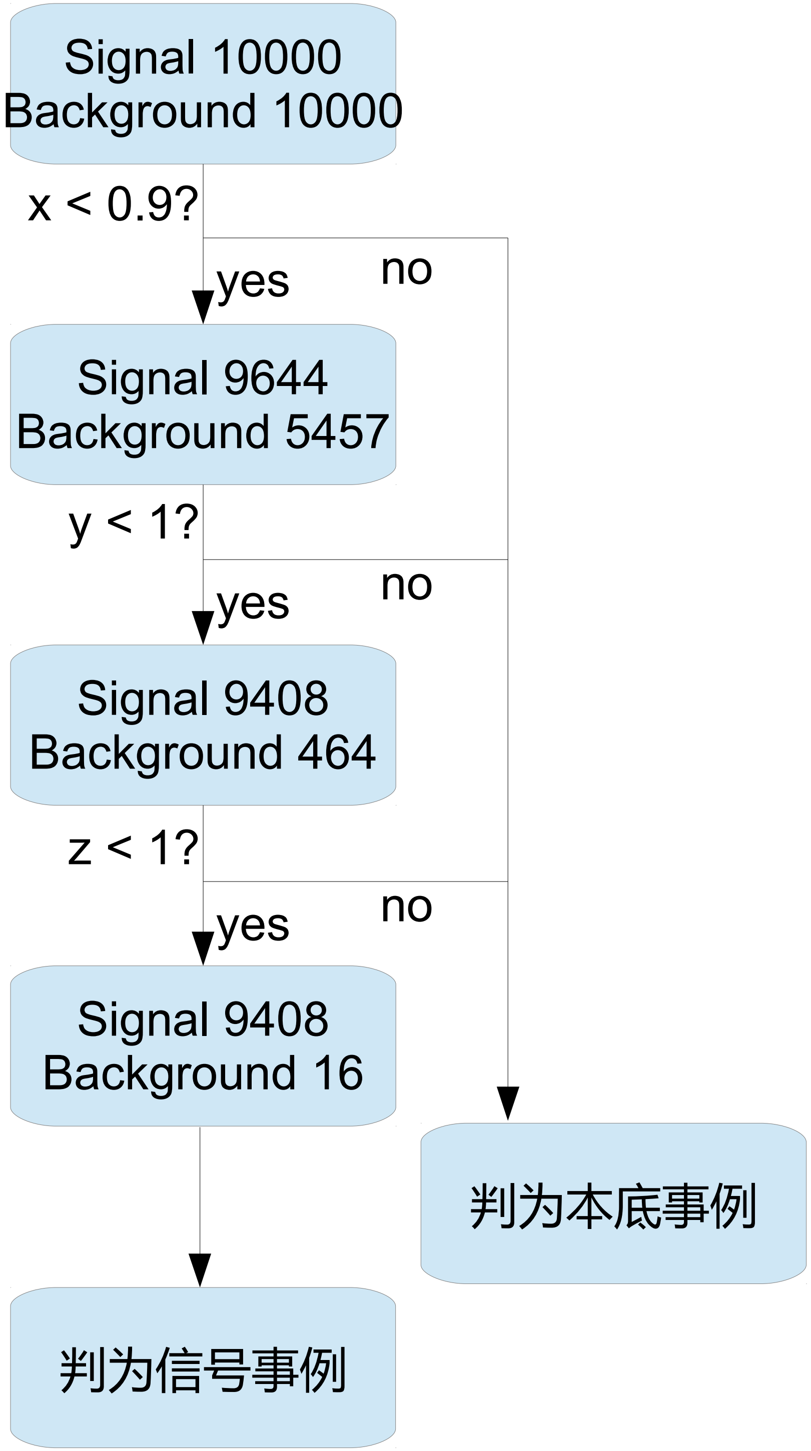




决策树判别

超长方体分割法

- 最简单的非线性判别方法
- 决策树方法的一种最简单的特例



Signal 10000
Background 10000

x < 0.9?

yes

no

Signal 9644
Background 5457

y < 1?

yes

no

Signal 9408
Background 464

z < 1?

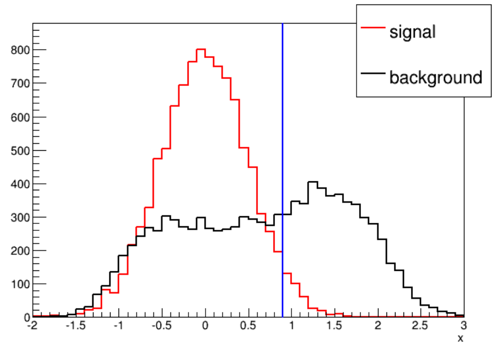
yes

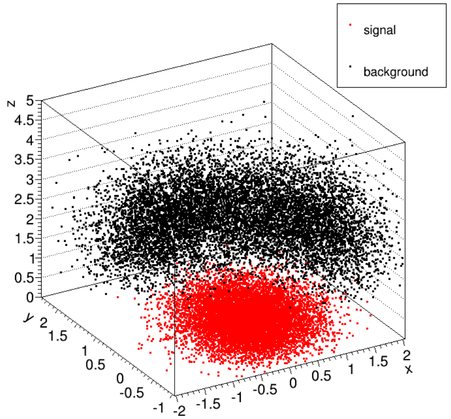
no

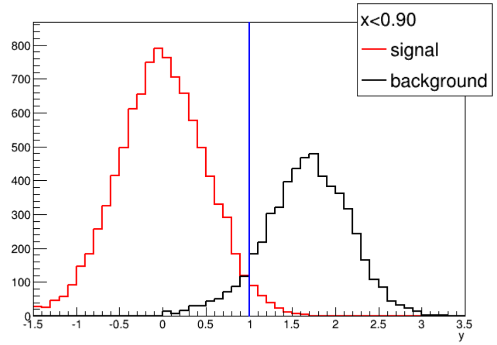
Signal 9408
Background 16

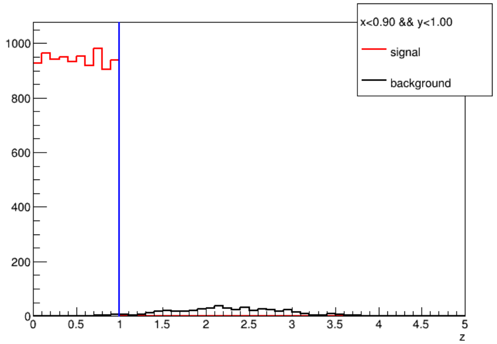
判为本底事例

判为信号事例

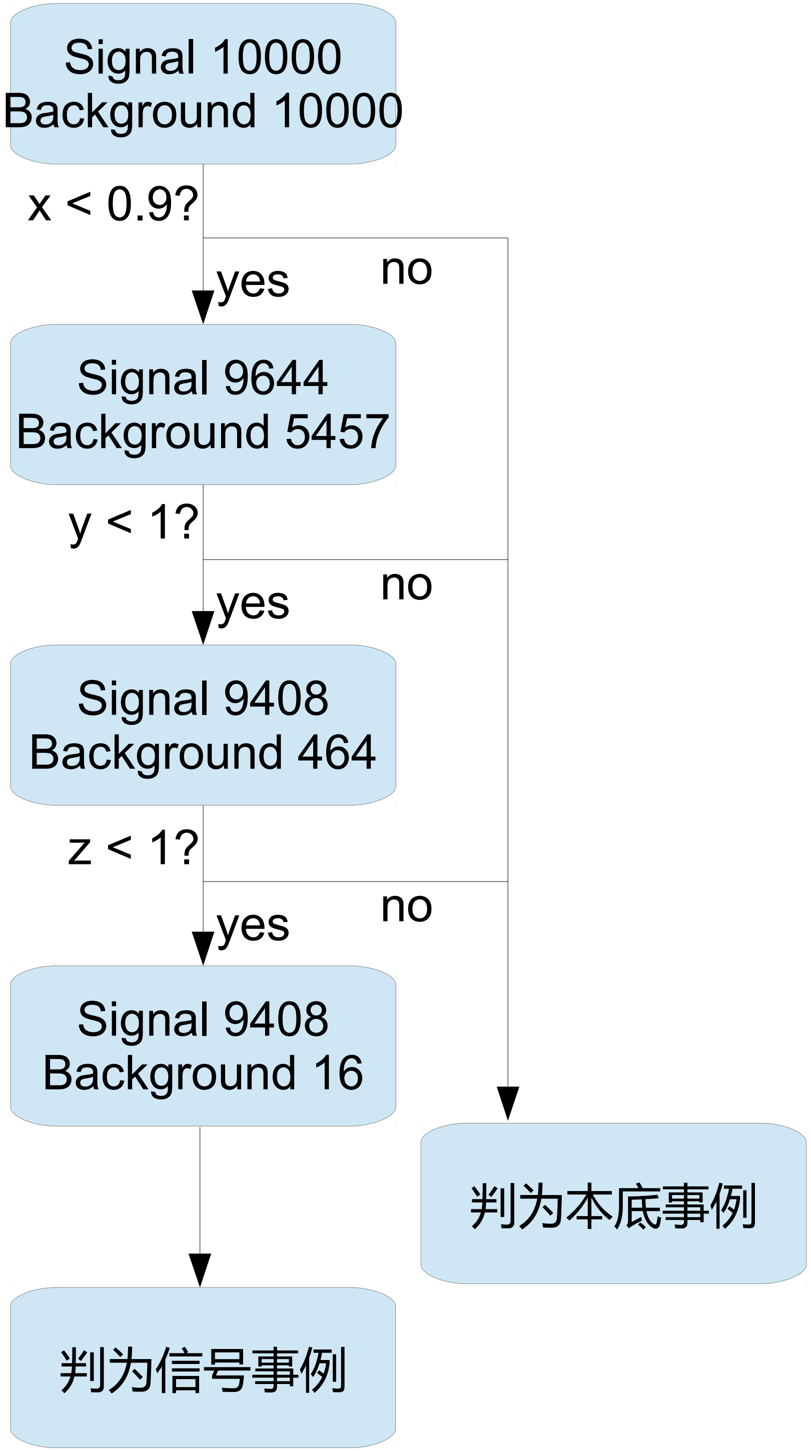








- 阈值的确定
  - 最大的信号显著性
  - $S_{sig} = n_{ss}/\mathrm{sqrt}(n_{ss} + n_{sb})$



Signal 10000
Background 10000

x < 0.9?

yes

no

Signal 9644
Background 5457

y < 1?

yes

no

Signal 9408
Background 464

z < 1?

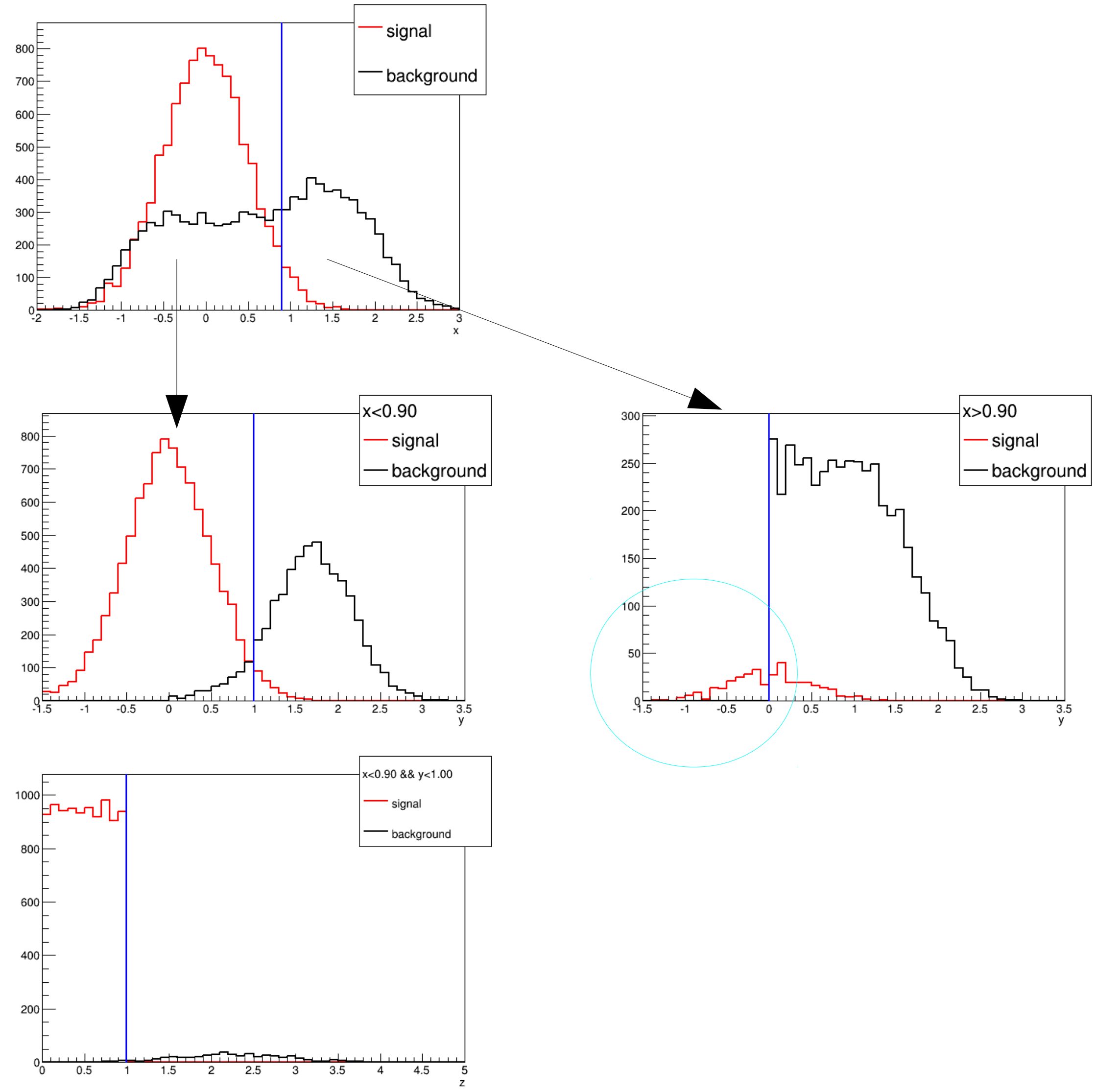
yes

no

Signal 9408
Background 16

判为本底事例

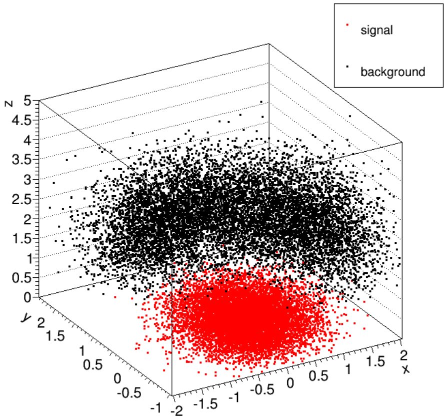
判为信号事例

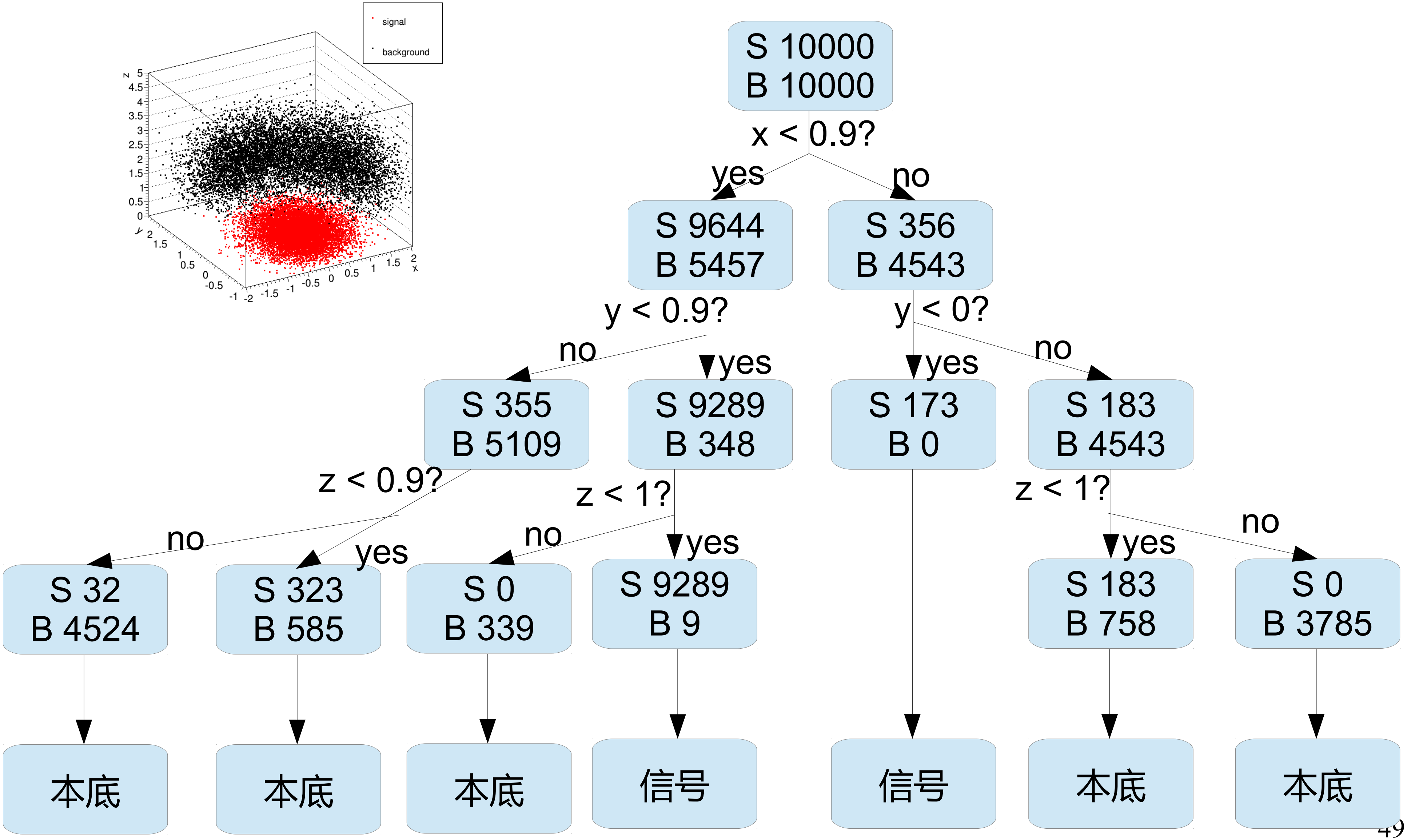


决策树（ decision trees ）

- 决策树或称树分类器
- 不是企图用一个决策规则把多个类别的样本一次分开，而是采用分级的方法，使分类问题逐步得到解决
- 1984. L.Breiman, J.H. Friedman, R.A.Olshen, C.J.Stone, “Classification and Regression Trees”, Wadsworth, 1984.

二叉决策树





S 10000
B 10000

x < 0.9?

yes

S 9644
B 5457

y < 0.9?

no

S 355
B 5109

z < 0.9?

no

S 32
B 4524

本底

yes

S 323
B 585

本底

yes

S 9289
B 348

z < 1?

no

S 0
B 339

本底

yes

S 9289
B 9

信号

no

S 356
B 4543

y < 0?

yes

S 173
B 0

信号

no

S 183
B 4543

z < 1?

yes

S 183
B 758

本底

no

S 0
B 3785

本底

决策树的构建

- 怎样评价每个节点选择的（变量+阈值）组合对于信号和本底的判别能力？
  - 判别指数 I
  - Gini指数 p(1-p)
  - 交叉熵 -plnp – (1-p)ln(1-p)
  - 误判误差 1-max(p, 1-p)
  - 统计显著性 $n_s/\mathrm{sqrt}(n_s+n_b)$
  - 其中 p 为信号事例纯度

$$P=\frac{\sum_{\text{signal}} w_i}{\sum_{\text{signal}} w_i+\sum_{\text{background}} w_i}$$

- 用子节点相对于母节点的判别指数的加权值的增量来衡量每个节点选择的（变量+阈值）组合对于信号和本底的判别能力
  - $\Delta I = n_{int} I – (n_1 I_1+n_2 I_2)$
  - 选择 ΔI 最大的作为本节点的判别变量

决策树的构建

- 迭代直到达到终止条件
  - 最大叶节点数
  - 最小事例数
  - 判别指数增量
  - …
- 训练完成后，输入事例数中信号事例占优的叶节点被指定为信号叶节点，输入事例数中本底事例占优的叶节点被指定为本底叶节点
- 当一个待分类的事例样本集输入这样构建的二叉树后，归入信号叶节点的事例被判为信号事例，归入本底叶节点的事例被判为本底事例

$$f(\boldsymbol{x}) = 1 \text{ if } \boldsymbol{x} \text{ in signal region, } -1 \text{ otherwise}$$

决策树的修剪

- 决策树的修剪 (pruning)：
  - 对于节点数达到极大值的决策树自下而上地减除对于有效地分辨信号 / 本底用处不大的节点
- 可以避免过度训练和计算量的有害增长

决策树林法 (boosted decision trees)

- 决策树法对于训练样本集的统计涨落具有不稳定性，可以通过决策树林法得到克服
- 构建第一棵决策树后，对该样本集中的每个事例按某种规则赋予新的权值，然后用具有新权值的样本构建第二棵决策树 ... 依次构建 K 棵决策树构成的树林
- 1996. Ref: Y. Freund, R.E.Schapire, “Experiments with a new boosting algorithm”, Proceedings of COLT, ACM Press, New York, 1996, pp. 209-217. 首次提出自适应 (AdaBoost) 算法

决策树林的构建

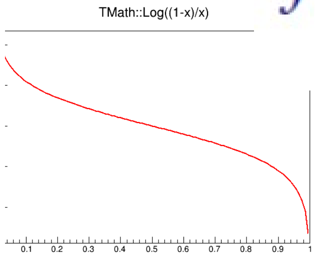
- 基本思想：在一棵决策树训练过程中被误判的所有事例在下一棵树的构建中赋以较高的权值，判别正确的事例则赋以较低的权值（或保持不变）
- 初值：

$\boldsymbol{x}_1,\ldots\ldots,\boldsymbol{x}_N$	event data vectors
$y_1,\ldots\ldots,y_N$	true class labels (+1 or -1)
$w_1^{(1)},\ldots\ldots,w_N^{(1)}$	event weights

  - 初始时所有事例的权值归一 $\sum_{i=1}^{N} w_i^{(1)}=1$

决策树林的构建

- 计算第 k 个决策树的错误率 $$\varepsilon_k=\sum_{i=1}^{N} w_i^{(k)} I(y_i f_k(\boldsymbol{x}_i)\leqslant 0)$$
  - 其中 I(x)=1, 当 x=true; I(x)=0, 当 x=false
- k 个决策树对事例的判别结果为 $$f(\boldsymbol{x})=\sum_{k=1}^{K} \alpha_k f_k(\boldsymbol{x}, T_k)$$
  - 其中 $$\alpha_k=\ln\frac{1-\varepsilon_k}{\varepsilon_k}$$


TMath::Log((1-x)/x)

- 则构建第 k+1 个决策树时，样本集中第 i 个事例的权值修改为

$$w_i^{(k+1)}=w_i^{(k)}\frac{e^{-\alpha_k f_k(\boldsymbol{x}_i) y_i/2}}{Z_k}$$

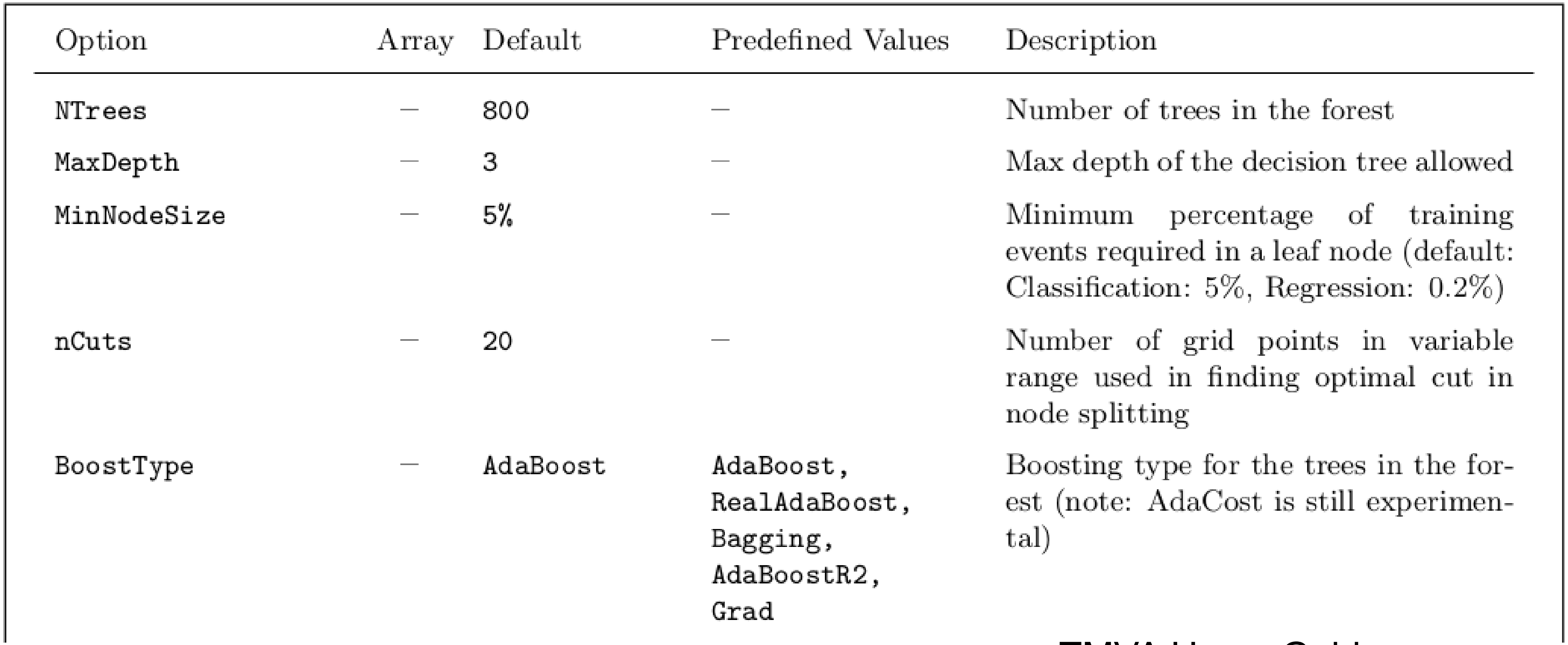
i = event index

k = training sample index

- 上海交通大学的杨海军老师等对决策树林的发展及其在粒子物理上的应用作出了开创性的贡献
- 2004/8/30, arXiv:physics/0408124, [Nucl.Instrum.Meth. A543 (2005) 577-584] Byron P. Roe, Hai-Jun Yang*, Ji Zhu, Yong Liu, Ion Stancu, Gordon McGregor, “Boosted Decision Trees as an Alternative to Artificial Neural Networks for Particle Identification”
- 2005/8/8, arXiv:physics/0508045, [Nucl.Instrum.Meth. A555 (2005) 370-385] Hai-Jun.Yang*, Byron P. Roe, Ji Zhu, “Studies of Boosted Decision Trees for MiniBooNE Particle Identification”
- 2006/10/31, arXiv:physics/0610276, [Nucl. Instrum. & Meth. A 574 (2007) 342-349] Hai-Jun Yang*, Byron P. Roe, Ji Zhu, “Studies of Stability and Robustness for Artificial Neural Networks and Boosted Decision Trees”
- 2007/8/27, arXiv:0708.3635, [JINST3:P04004,2008] Hai-Jun Yang*, Tiesheng Dai, Alan Wilson, Zhengguo Zhao, Bing Zhou, “A Multivariate Training Technique with Event Reweighting”

TMVA package

- factory->BookMethod(Types::kBDT, "BDT", "<options>");



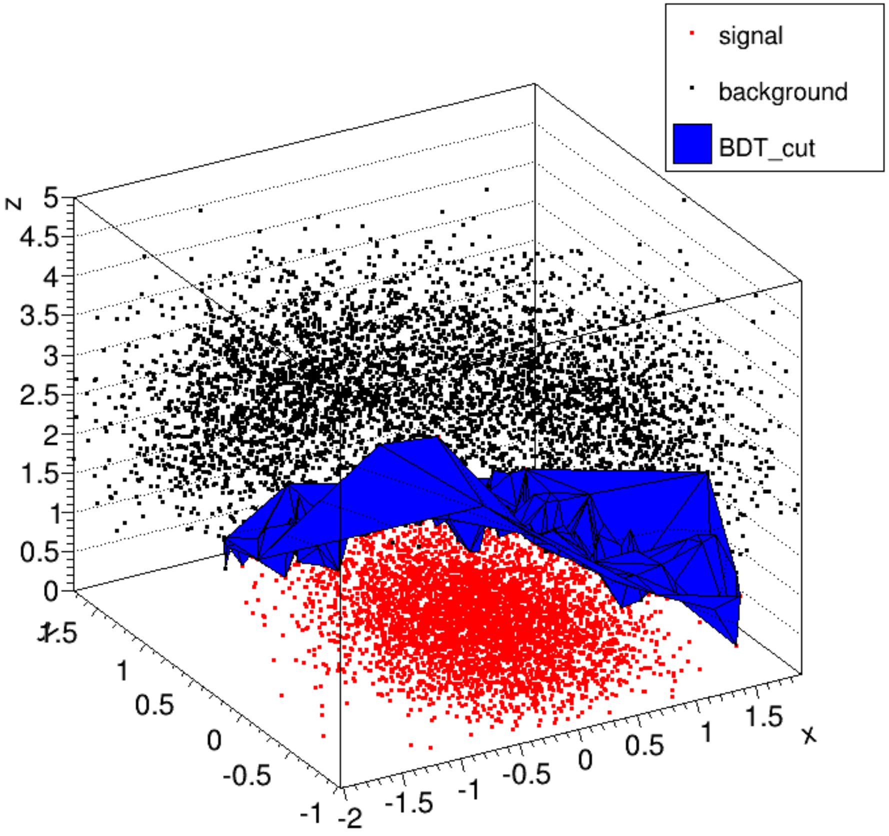
Option	Array	Default	Predefined Values	Description
`NTrees`	–	`800`	–	Number of trees in the forest
`MaxDepth`	–	`3`	–	Max depth of the decision tree allowed
`MinNodeSize`	–	`5%`	–	Minimum percentage of training events required in a leaf node (default: Classification: 5%, Regression: 0.2%)
`nCuts`	–	`20`	–	Number of grid points in variable range used in finding optimal cut in node splitting

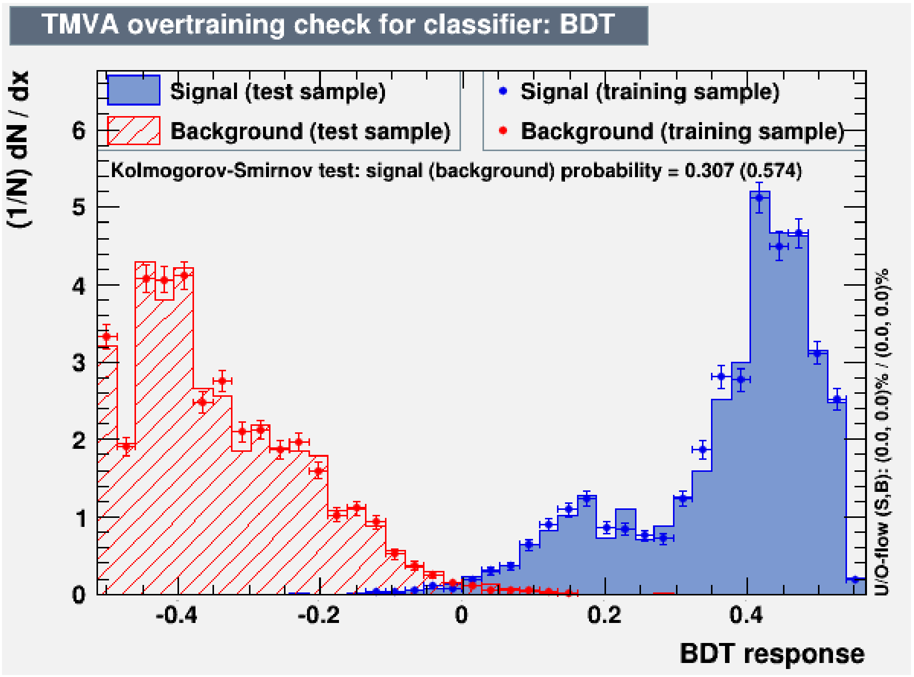
`BoostType`	–	`AdaBoost`	`AdaBoost,`
			`RealAdaBoost,`
			`Bagging,`
			`AdaBoostR2,`
			`Grad`
				Boosting type for the trees in the forest (note: AdaCost is still experimental)

……

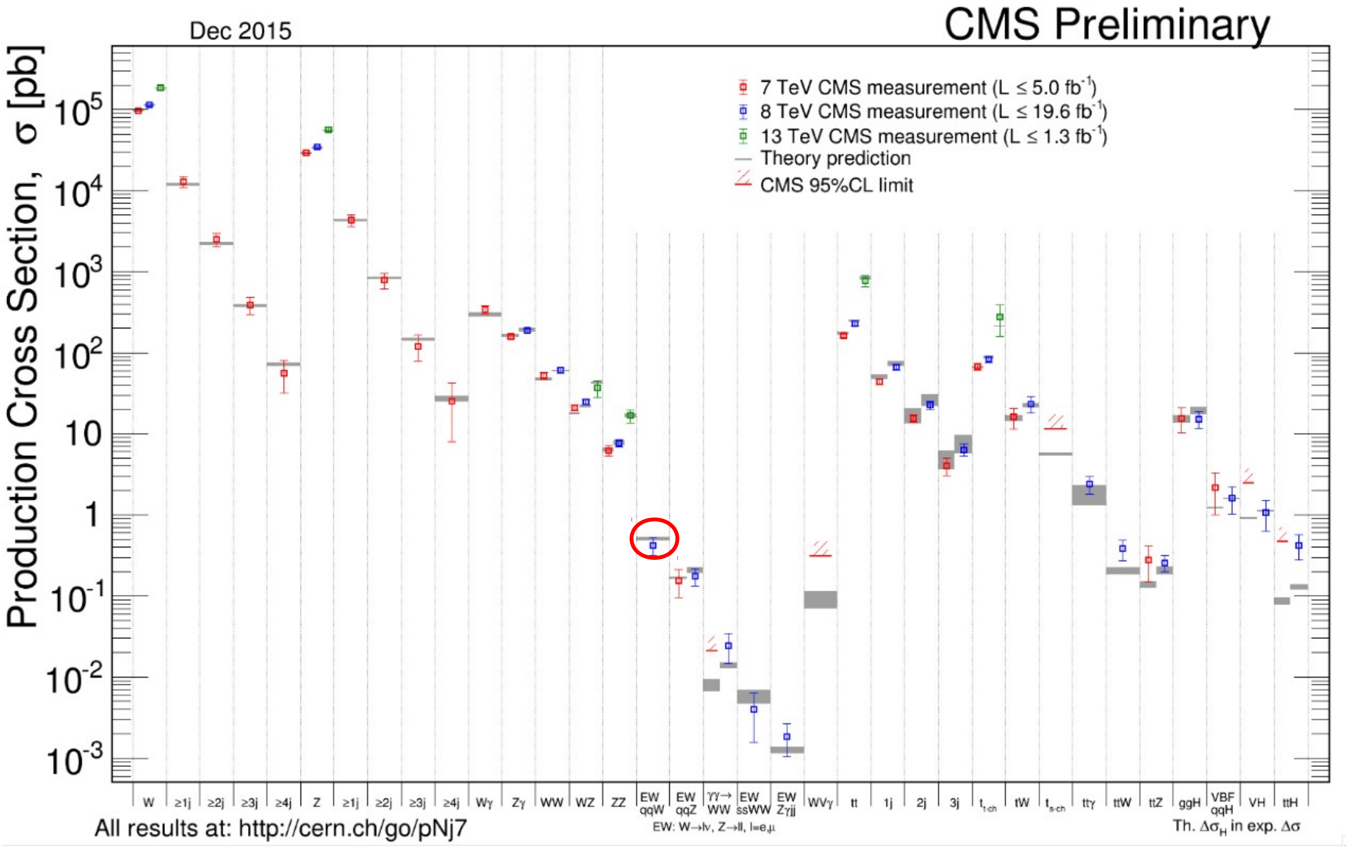
TMVA Users Guide

Output of TMVA





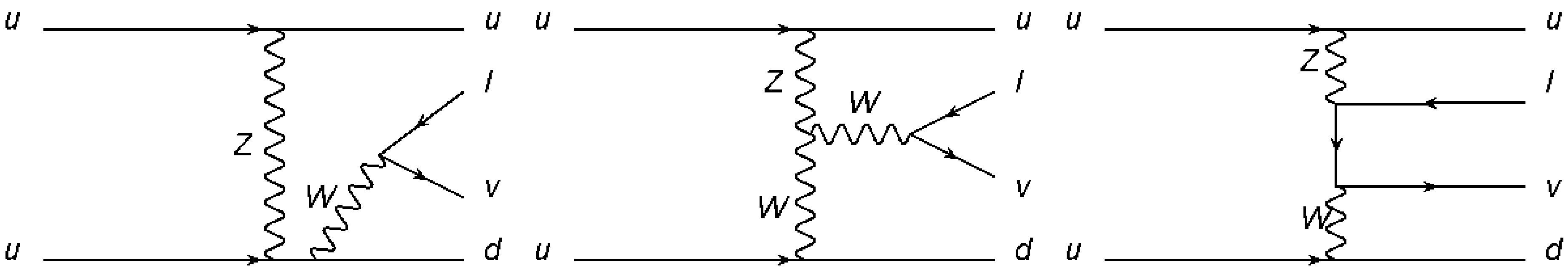
一个 BDT 在物理分析中的应用
—— EW W+2 jets arXiv:1607.06975



- **EWK W+2 jets**: first cross section measurement
- Submitted to JHEP
- arXiv1607.06975
- Sister analysis of EWK Z+2 jets analysis

Signal process

- EWK W+2 jets production, leptonic decay channel



Simplest VBF-like process with large cross sections

- Allows detailed tests of EWK production modeling (WWZ triple-gauge-boson couplings)
- Background of SM Higgs in VBF channel and in searching invisible Higgs

VBF Signature:

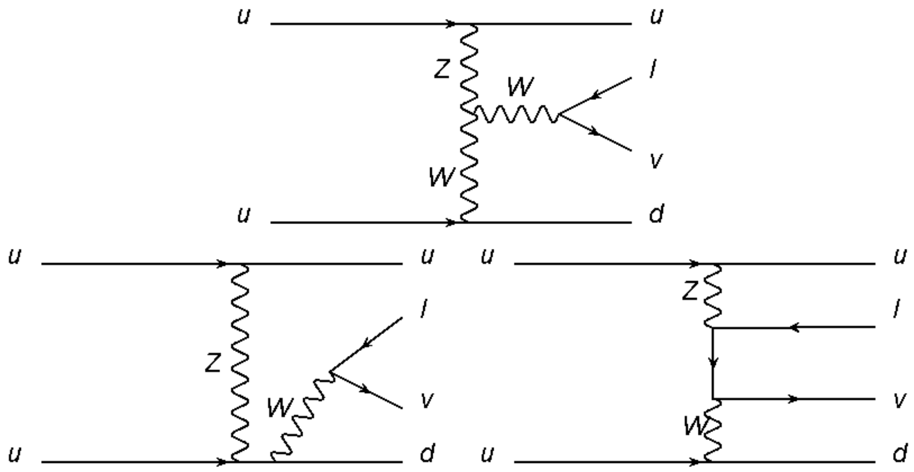
- Two high energetic hadronic jets (quarks) with wide rapidity separation
- Suppressed hadronic activity “between” the two jets (central region)
- High m_{jj} region enriched in EWK production

Signal and backgrounds

Signal:

EWK W+2 jets

- One lepton
- Two jets
- Significant missing transverse momentum (MET)



Background:

QCD W+jets

- The most dominant background

$t\bar{t}$

- Second dominant background

QCD Multijet:

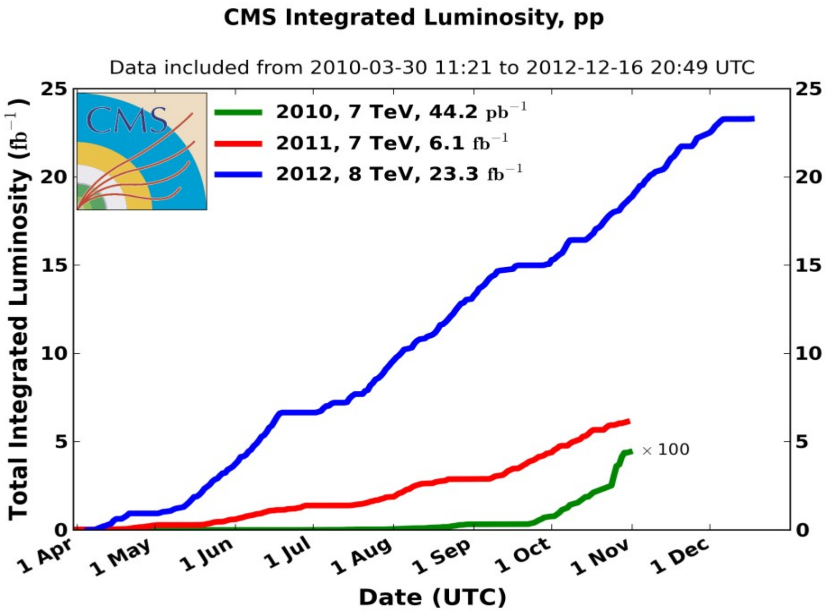
- Jets to be reconstructed as electrons

Drell-Yan Z/γ* +jets:

- Similar handronic activities

Dibosons (WW/WZ/ZZ), single-top

Data and simulated samples



Data:

- Collected by the CMS experiment in 2012
- √s = 8 TeV
- Integrated luminosity ~ 19.3 fb^{-1}

Simulation

- Event generation

  EWK W+2 jets: Madgraph5_aMC@NLO 2.1

  QCD W+jets, $t\bar{t}$, Drell-Yan: Madgraph5

  Single-top: POWHEG 1.0

  Diboson (WW, WZ, ZZ): PYTHIA 6.4

- PDF set: CTEQ6L1 (Signle-top CTEQ6M)
- Parton shower: PYTHIA 6
- GENT4-based CMS detector simulation
- QCD multijet events: data driven

Event selection

Online trigger

- Single-lepton triggers, p_T thresholds 27 (24) GeV for electrons (muons)

Primary vertex

- The vertex with the highest value of Σp_T^2 of all associated tracks
- Number of degrees of freedom (ndof) ≥ 4
- In the central detector region of |z| ≤ 24 cm and ρ ≤ 2 cm around the nominal interaction point.

Only one isolated lepton, muon or electron

- High quality lepton ID
- Isolation requirements
- Electrons: p_T > 30 GeV, |η| < 2.5, exclude the transition region between the barrel and endcap (1.44 < |η| < 1.57)
- Muons: p_T > 25 GeV, |η| < 2.1
- Second lepton veto

Event selection

At least two jets

- AK5 jet
- Jet energy corrections
- Cleaning from leptons (R=0.3)
- Remove pileup jets
- $p_T > 60$ GeV (leading) and $p_T > 50$ GeV (subleading), $|\eta| < 4.7$

Significant MET

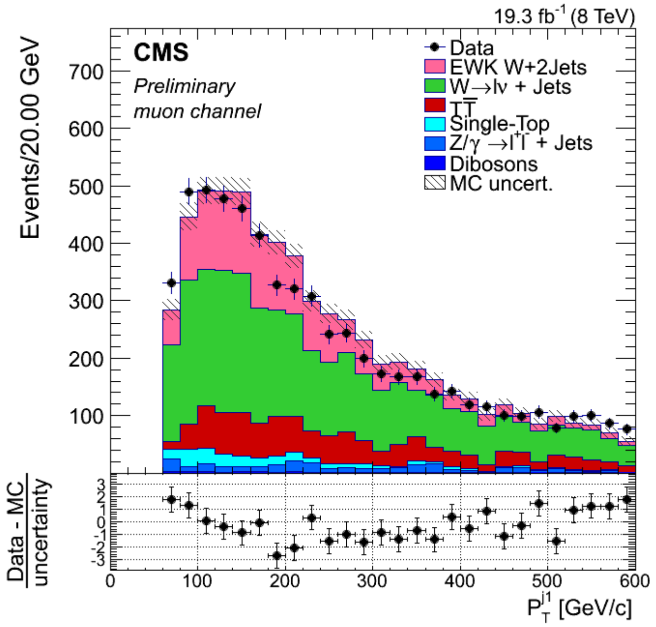
- Energy correction
- MET > 30(25) GeV for electron (muon) channel.

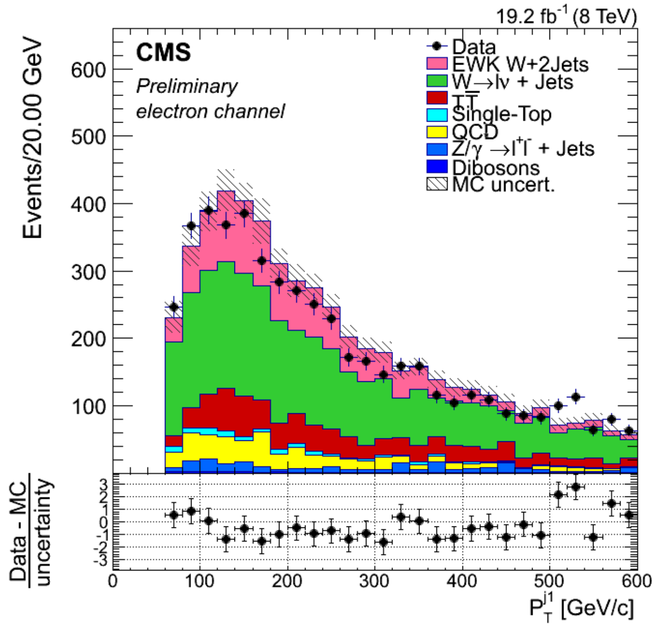
W boson reconstruction

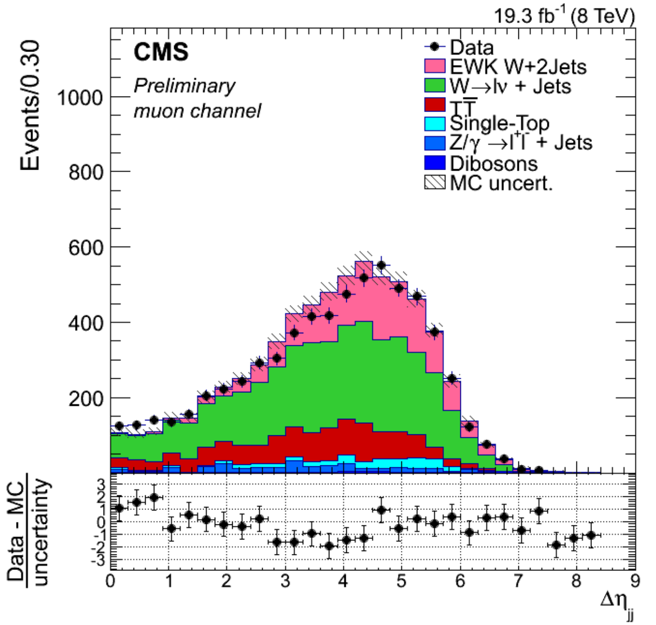
- Unmeasurable p_z component of the neutrino
- Assuming that the lepton and the MET arise from a W boson with $M_W = 80.4$ GeV → quadratic equation of p_z
- W transverse mass: $W_{mT} > 30$ GeV

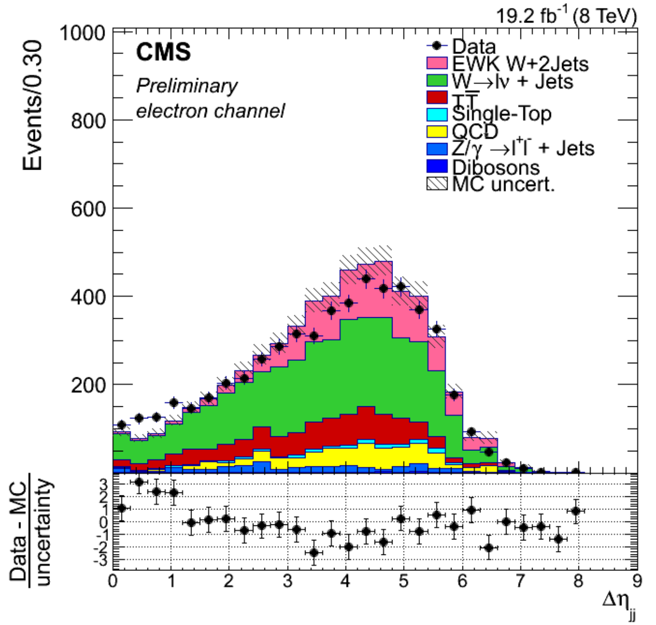
Additional requirements

- Zeppenfeld variable: $|y_W - (y_{jet1}+y_{jet2})/2.0| < 1.2$
- $m_{jj} > 1000$ GeV


19.3 fb^{-1} (8 TeV)
CMS
Preliminary
muon channel
Data
EWK W+2Jets
W→lν + Jets
$T\bar{T}$
Single-Top
Z/γ →l^+l^- + Jets
Dibosons
MC uncert.
Events/20.00 GeV
700
600
500
400
300
200
100
Data - MC
uncertainty
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600
P_T^{j1} [GeV/c]


19.2 fb^{-1} (8 TeV)
CMS
Preliminary
electron channel
Data
EWK W+2Jets
W→lν + Jets
$T\bar{T}$
Single-Top
QCD
Z/γ →l^+l^- + Jets
Dibosons
MC uncert.
Events/20.00 GeV
600
500
400
300
200
100
Data - MC
uncertainty
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600
P_T^{j1} [GeV/c]


19.3 fb^{-1} (8 TeV)
CMS
Preliminary
muon channel
Data
EWK W+2Jets
W→lν + Jets
$T\bar{T}$
Single-Top
Z/γ →l^+l^- + Jets
Dibosons
MC uncert.
Events/0.30
1000
800
600
400
200
Data - MC
uncertainty
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
6
7
8
9
$\Delta\eta_{jj}$


19.2 fb^{-1} (8 TeV)
CMS
Preliminary
electron channel
Data
EWK W+2Jets
W→lν + Jets
$T\bar{T}$
Single-Top
QCD
Z/γ →l^+l^- + Jets
Dibosons
MC uncert.
Events/0.30
1000
800
600
400
200
Data - MC
uncertainty
3
2
1
0
-1
-2
-3
0
1
2
3
4
5
6
7
8
9
$\Delta\eta_{jj}$

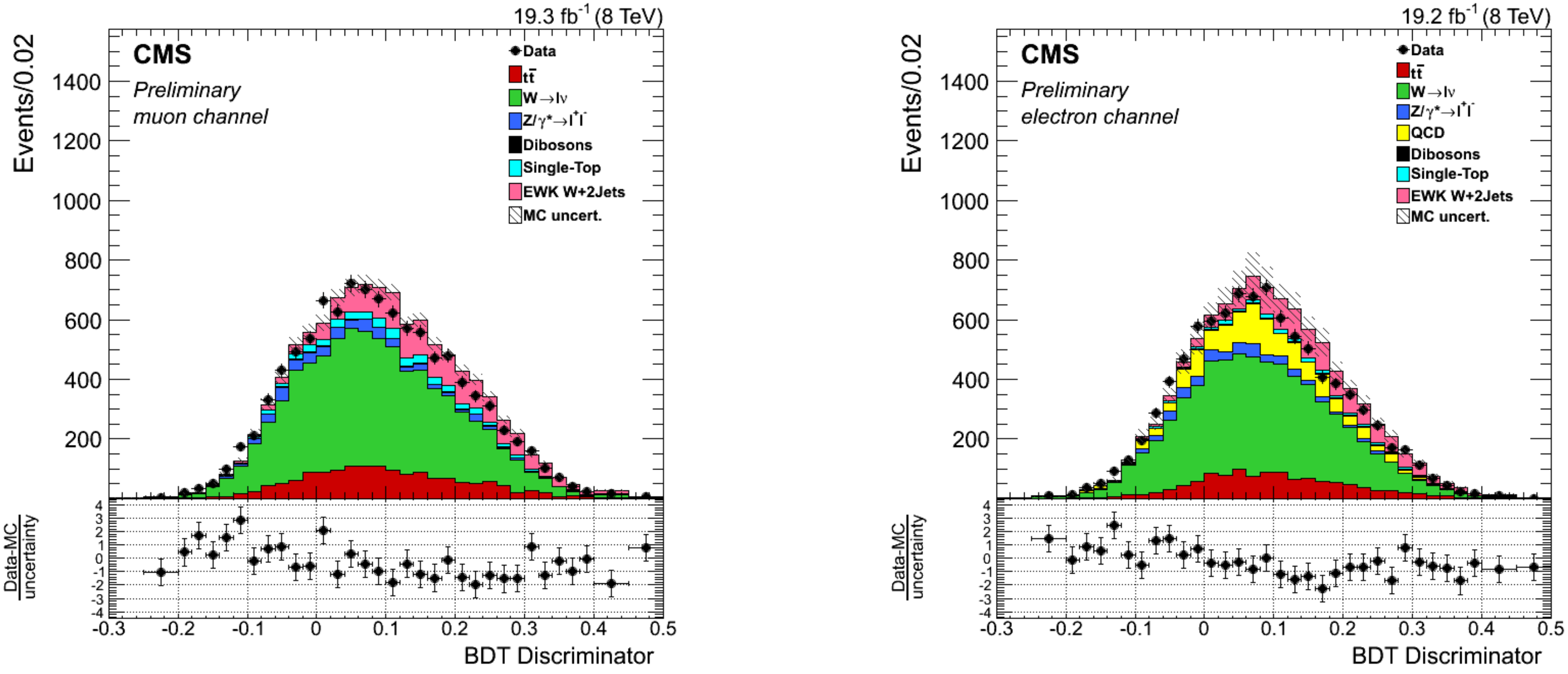
Background normalization evaluation

QCD W+jets production dominating the background and not well modeled by MC

→ Use data-driven methods to evaluate its normalization

Multivariate analysis:

- TMVA framework, ROOT package
- Boosted Decision Tree (BDT) technique, Adaptive Boost Algorithm
- Trained to distinguish between EWK W+2 jets and QCD W+jets events
- Input variables used: lepton η, $\Delta\eta$ between jets, $\Delta\eta$ and $\Delta\varphi$ between W boson and jets, and W boson p_T



- Singal events: higher BDT values
- Signal free region: BDT<0.1
- Overestimation of the QCD W+jets event yield
  - Normalization scale fectors: 0.71 ± 0.02 (stat.) for electron channel, 0.70 ± 0.02 (stat.) for muon channel

Signal estimation

EWK W+2 jets signal yield are extracted from an unbinned maximum likelihood fit to the m_{jj} distribution

Shape

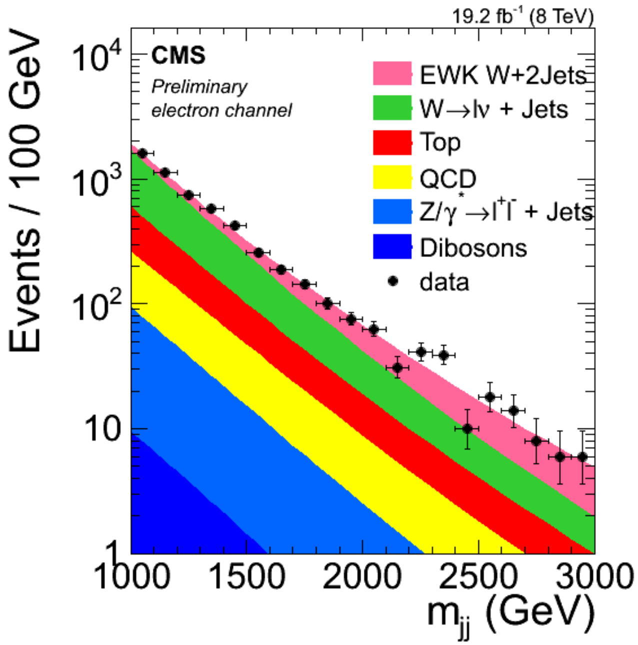
- Two parameter power law function

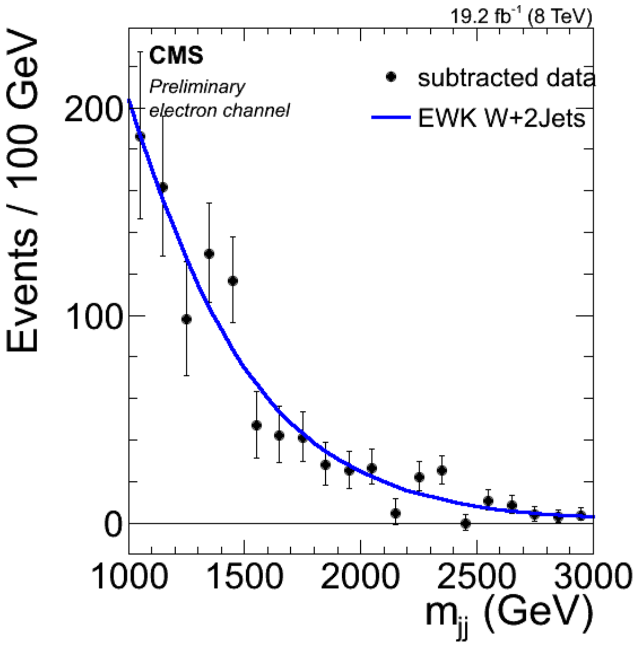
$$\mathcal{F} = \frac{1.0}{m_{jj}^{a_0 + a_1 log(m_{jj}/8000)}}$$

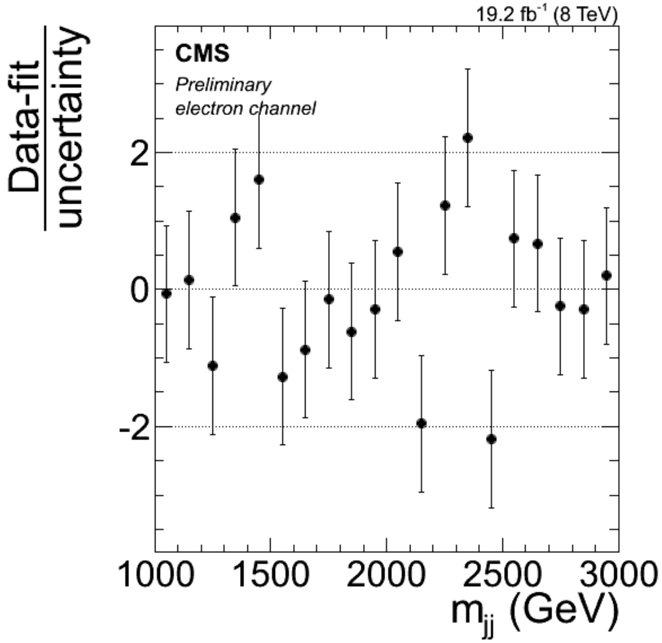
- Obtained from each component's MC sample (data-driven sample for QCD)
- Fixed except the QCD W+jets
- QCD W+jets shape: parameters floated, the initial values and their errors are obtained from MC

Normalization

- EWK W+2 jets: unconstrained
- QCD W+jets: evaluated from the BDT fit
- Top: NNLO cross section, Gaussian constrained ±7%
- Diboson, DY: Fixed to NLO cross section
- QCD multijet: Fixed to data driven-sample prediction



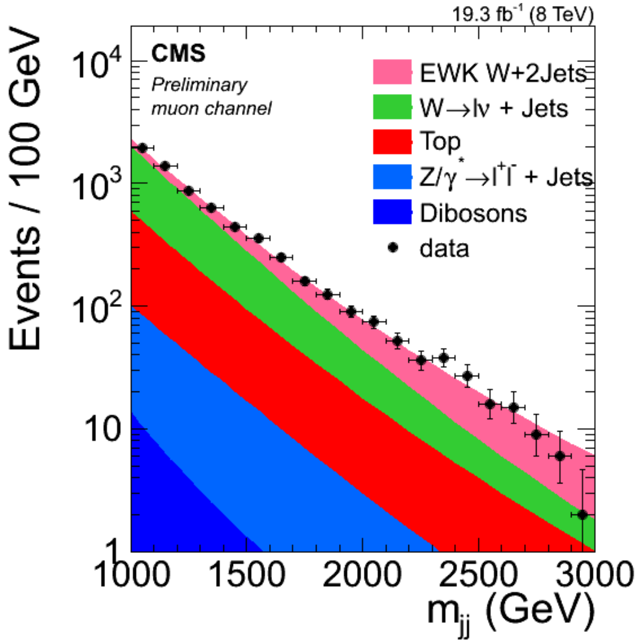


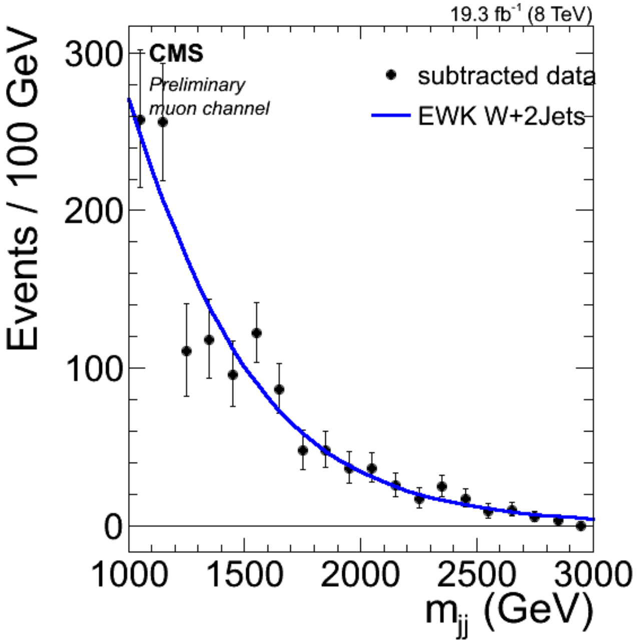


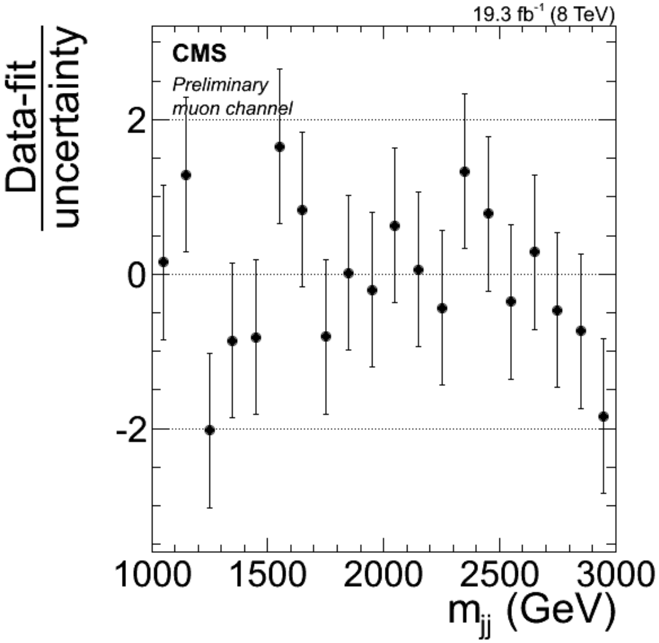
Electron channel

$r(\sigma/\sigma_{SM})$

$= 0.83 \pm 0.08$ (stat)





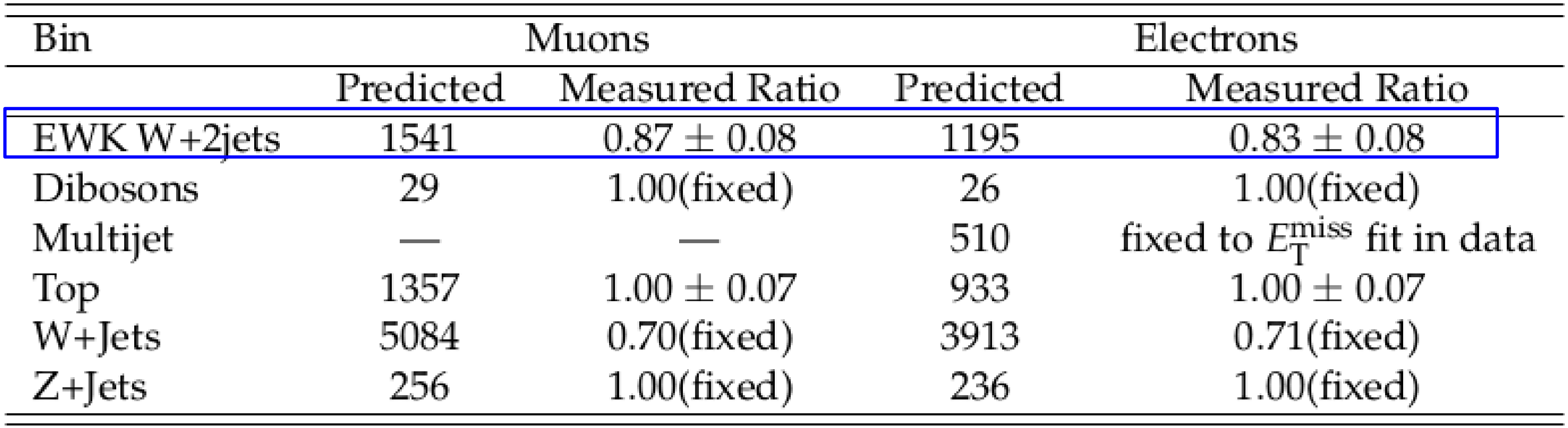


Muon channel

$r(\sigma/\sigma_{SM})$

$= 0.87 \pm 0.08$ (stat)

Expected event yields



Bin	Muons		Electrons	
	Predicted	Measured Ratio	Predicted	Measured Ratio
EWK W+2jets	1541	0.87 ± 0.08	1195	0.83 ± 0.08
Dibosons	29	1.00(fixed)	26	1.00(fixed)
Multijet	—	—	510	fixed to $E_{\mathrm{T}}^{\mathrm{miss}}$ fit in data
Top	1357	1.00 ± 0.07	933	1.00 ± 0.07
W+Jets	5084	0.70(fixed)	3913	0.71(fixed)
Z+Jets	256	1.00(fixed)	236	1.00(fixed)

Fiducial cross section

Fiducial region:

- W decays to μν or eν
- genjet: $p_T^{j1} > 60$ GeV, $|\eta^{j1}| <4.7$, $p_T^{j2} > 50$ GeV, $|\eta^{j2}| <4.7$, $m_{jj} > 1000$ GeV

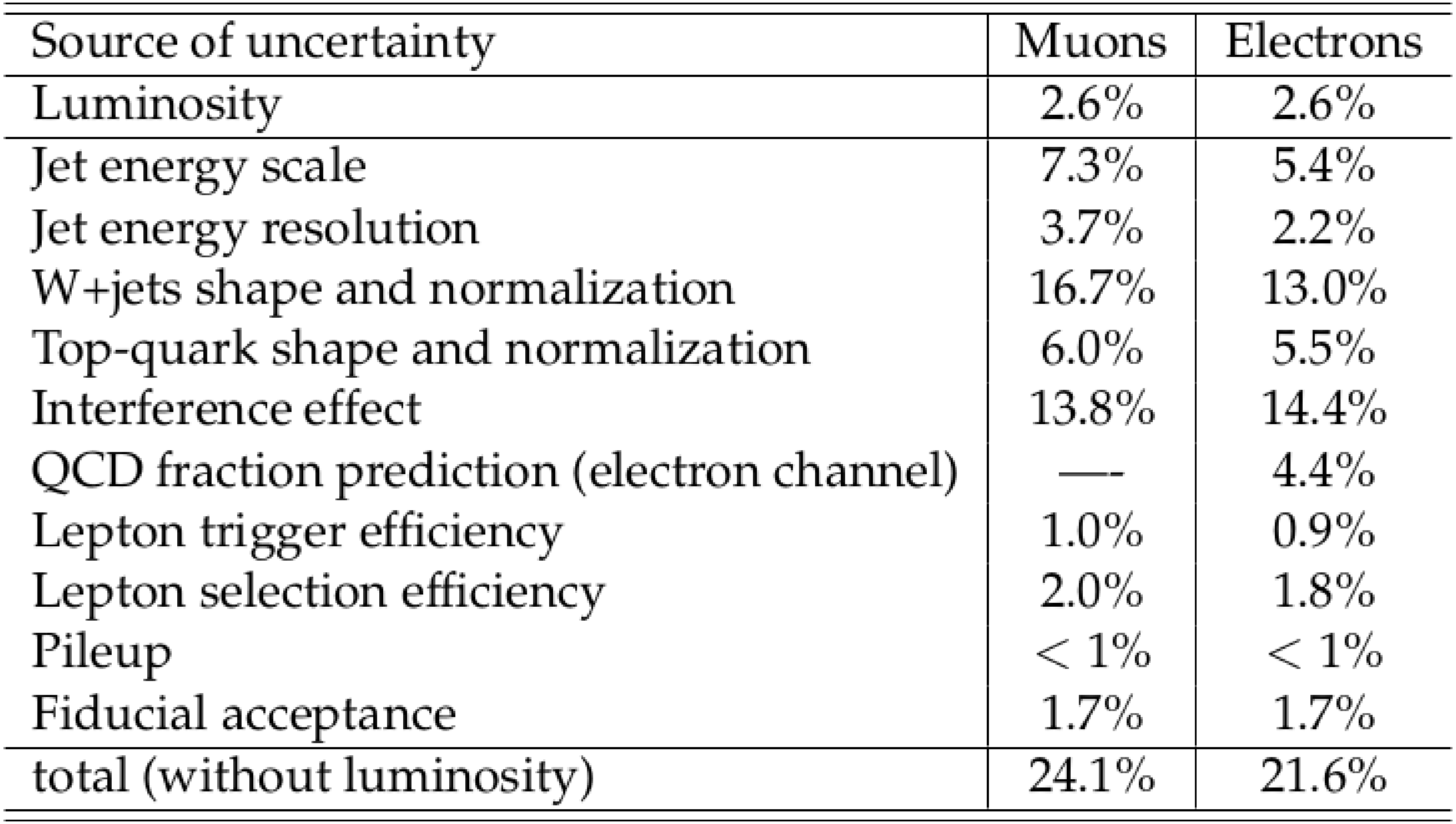
$\sigma_{\text{fiducial}} = \mu_{\text{signal strength}} * \sigma_{\text{generator}} * \varepsilon_{\text{generated to fiducial}}$

- $\sigma_{\text{generator}}$ = cross section in generator = 11.094 pb
- $\mu_{\text{signal strength}}$ = 0.83 (0.87) for electron (muon)
- $\varepsilon_{\text{generated to fiducial}}$ = (Simulated signal events in fiducial region)/(All signal MC events) = 0.0448

σ_{fiducial}:

- Electron channel: 0.41 pb
- Muon channel: 0.43 pb

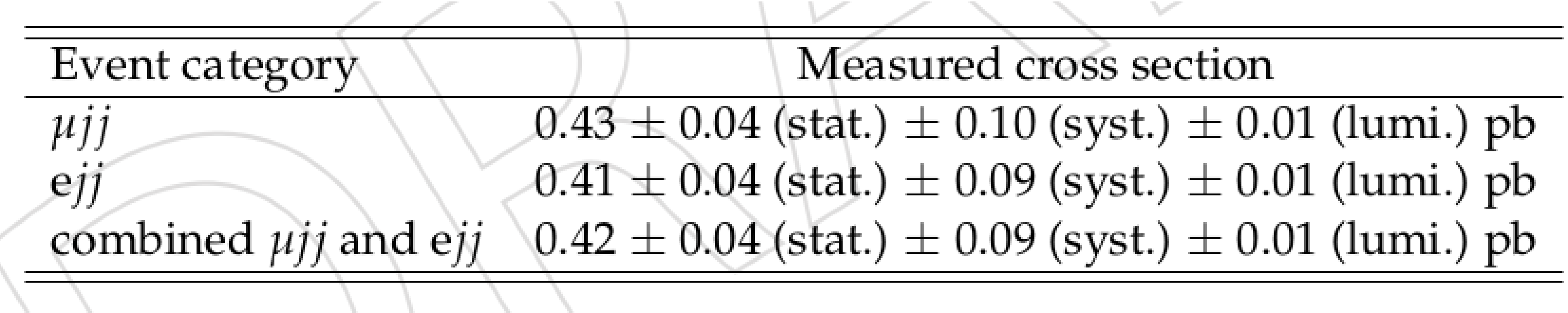
Systematic uncertainties



Source of uncertainty	Muons	Electrons
Luminosity	2.6%	2.6%
Jet energy scale	7.3%	5.4%
Jet energy resolution	3.7%	2.2%
W+jets shape and normalization	16.7%	13.0%
Top-quark shape and normalization	6.0%	5.5%
Interference effect	13.8%	14.4%
QCD fraction prediction (electron channel)	—-	4.4%
Lepton trigger efficiency	1.0%	0.9%
Lepton selection efficiency	2.0%	1.8%
Pileup	$< 1\%$	$< 1\%$
Fiducial acceptance	1.7%	1.7%
total (without luminosity)	24.1%	21.6%

Results

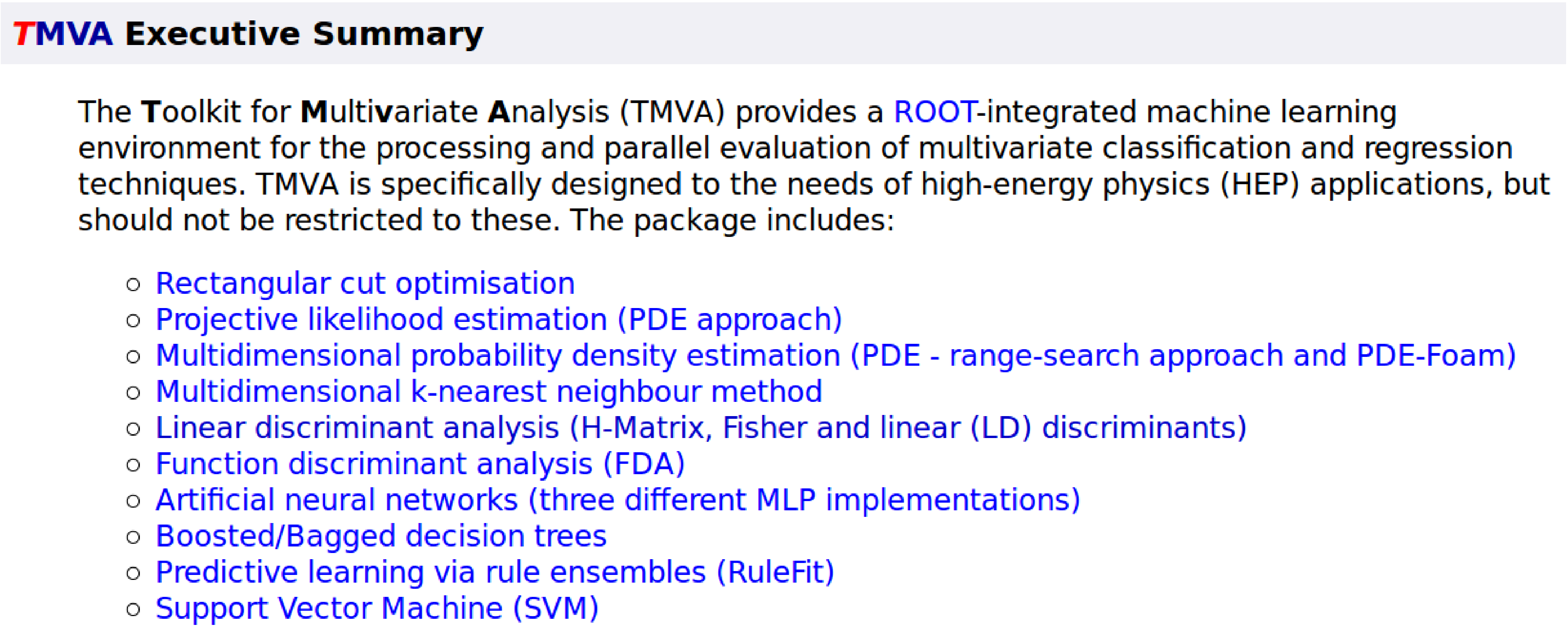
- A measurement of electroweak production cross section of a W boson in association with two jets events in pp collision is presented



Event category	Measured cross section
μjj	0.43 ± 0.04 (stat.) ± 0.10 (syst.) ± 0.01 (lumi.) pb
ejj	0.41 ± 0.04 (stat.) ± 0.09 (syst.) ± 0.01 (lumi.) pb
combined μjj and ejj	0.42 ± 0.04 (stat.) ± 0.09 (syst.) ± 0.01 (lumi.) pb

- Consistent with the SM prediction of 0.50 ± 0.02 (scale) ± 0.02 (PDF) pb obtained by MADGRAPH interfaced to PYTHIA 6.

其他判别方法



TMVA Executive Summary

The **T**oolkit for **M**ulti**v**ariate **A**nalysis (TMVA) provides a ROOT-integrated machine learning environment for the processing and parallel evaluation of multivariate classification and regression techniques. TMVA is specifically designed to the needs of high-energy physics (HEP) applications, but should not be restricted to these. The package includes:

- Rectangular cut optimisation
- Projective likelihood estimation (PDE approach)
- Multidimensional probability density estimation (PDE - range-search approach and PDE-Foam)
- Multidimensional k-nearest neighbour method
- Linear discriminant analysis (H-Matrix, Fisher and linear (LD) discriminants)
- Function discriminant analysis (FDA)
- Artificial neural networks (three different MLP implementations)
- Boosted/Bagged decision trees
- Predictive learning via rule ensembles (RuleFit)
- Support Vector Machine (SVM)

Matrix element method (MEM)

- Conditional probability (weight): P(**x**|**α**)
  - Theoretical information: **α**
  - Experimentally quantities: **x**
  - **y**: parton-level configuration
  - W(**x,y**): The evolution of **y** into x (transfer function)

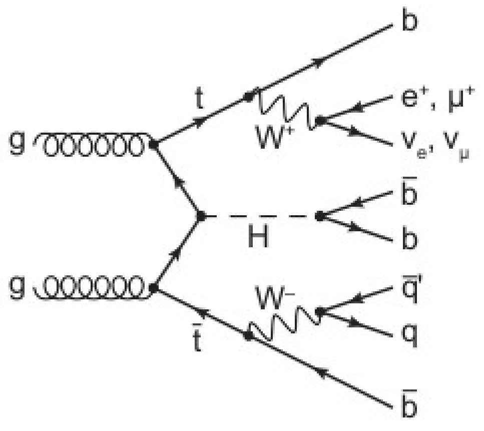
$$P(\boldsymbol{x}|\boldsymbol{\alpha}) = \int d\boldsymbol{y} P_\alpha(\boldsymbol{y}) W(\boldsymbol{x}, \boldsymbol{y}).$$

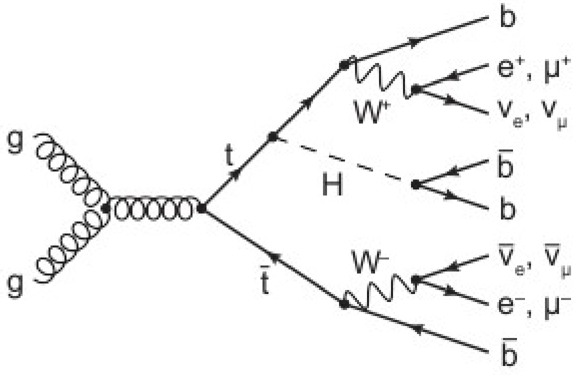
$$P(\boldsymbol{x}|\boldsymbol{\alpha}) = \frac{1}{\sigma_\alpha} \int d\Phi(\boldsymbol{y}) dq_1 dq_2 f_1(q_1) f_2(q_2) |M_\alpha|^2(\boldsymbol{y}) W(\boldsymbol{x}, \boldsymbol{y}) \, .$$

- The MEM discriminant is constructed as ratio of the probability density values of the signal and background hypothesis
- Makes maximal use of both experimental information and the theoretical model on an event-by-event basis
- Especially for complex final state

arXiv:1007.3300

$t\bar{t}H(\rightarrow b\bar{b})$ CMS-HIG-16-004





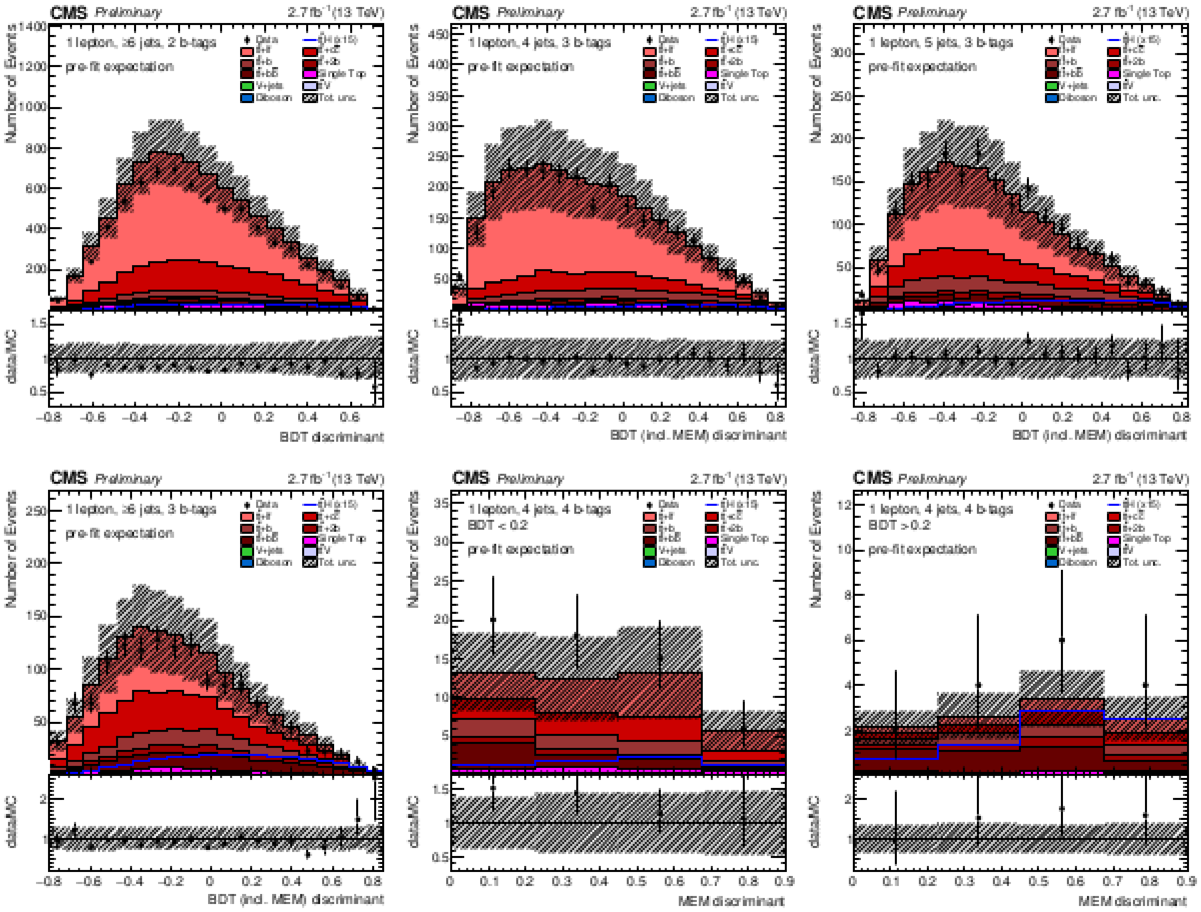


Figure 2: Final discriminant shapes in the different analysis categories in the lepton+jets channel before the fit to data. The expected background contributions (filled histograms) are stacked, and the expected signal distribution (line) for a Higgs-boson mass of $m_H = 125\,\text{GeV}$ is superimposed. Each contribution is normalized to an integrated luminosity of 2.7 fb^{-1}, and the signal contribution is additionally scaled by a factor of 15 for better readability. The distributions in data (markers) are also shown. In the top row the ≥ 6 jets 2 b-tag, the 4 jets 3 b-tag, and the 5 jets 3 b-tag category are shown. Below are the ≥ 6 jets 3 b-tag category, the 4 jets 4 b-tag category with low BDT output, and the 4 jets 4 b-tag category with high BDT output.

http://cds.cern.ch/record/2139578?ln=en

小结

- 线性判别方法：利用样本的线性函数作为样本类别的判别函数，方法简单，容易实现，计算量和数据存储量小；对于线性不可分样本，计算复杂，错分率较大，不宜采用
- 人工神经网络：对非线性复杂关联数据具有很强判别能力，基本思想简单；设计、训练复杂费时，计算量和存储量很大，需要有足够统计量的训练样本集
- 决策树方法：具有物理直观性，程序设计和调试简单，计算速度快；当信号和本底样本相互重叠或数据存在非线性关联时，判选效率下降。与人工神经网络相比，决策树方法计算量相对小，对训练样本量要求不是特别大；但理论最优性能略逊
- Toolkit for multivariate data analysis (TMVA) 是一个多元统计分析的工具性程序包
- 大型高能物理实验是典型的复杂大系统的科学研究工作，原始数据集合样本数量巨大，利用多元统计分析的方法对多维复杂数据集合进行科学的分析，可以帮助我们挖掘出隐藏在复杂海量数据中的规律和信息

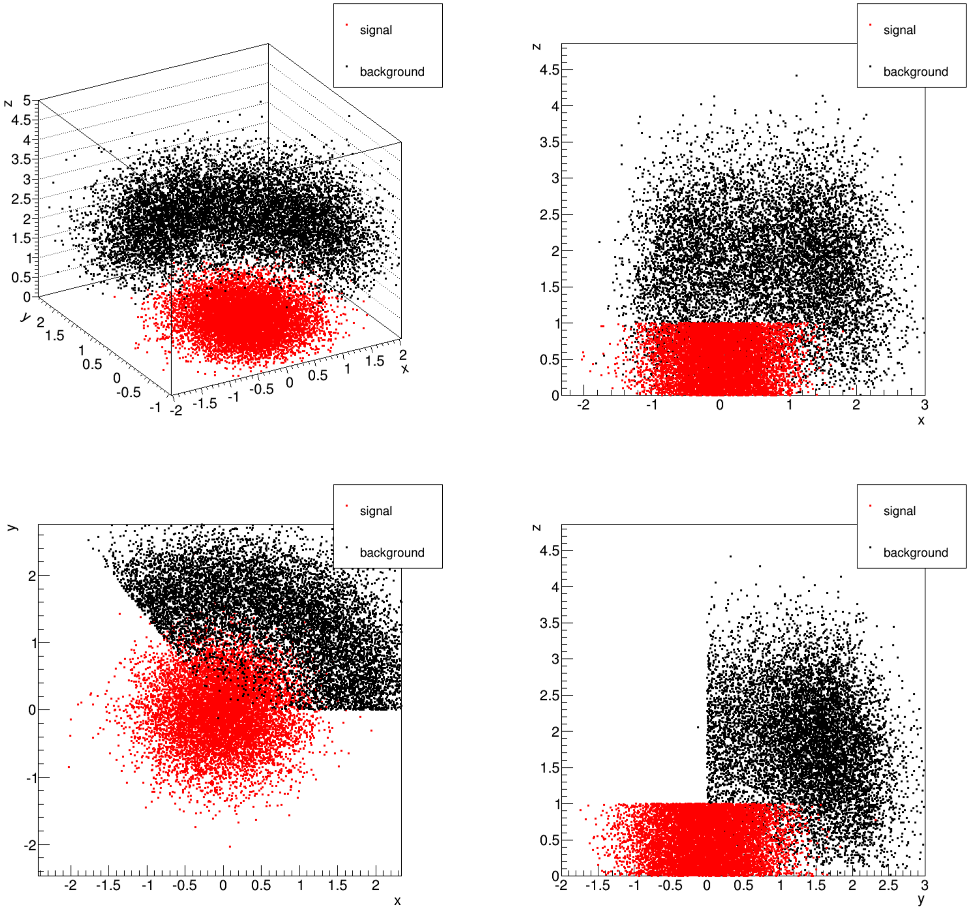
Thank You!

Backup

```
void random(double N, TString title){
  double sigma=0.5;
  double R0=2.;
  double r_ring;

  double x,y,z,r,theta,phi;

  TFile * f = new TFile(Form("%s
%.0f.root",title.Data(), N), "RECREATE");
  TTree * t = new TTree("t","t");
  t->Branch("x", &x, "x/D");
  t->Branch("y", &y, "y/D");
  t->Branch("z", &z, "z/D");
  t->Branch("r", &r, "r/D");
  t->Branch("theta", &theta, "theta/D");
  t->Branch("phi", &phi, "phi/D");
```

```
gRandom = new TRandom3();
gRandom->SetSeed(0);

if(title=="signal"){
  for(int i=0; i<N; i++){
    x=sigma*gRandom->Gaus();
    y=sigma*gRandom->Gaus();
    z=gRandom->Rndm();
    r=sqrt(x*x+y*y+z*z);
    theta=TMath::ACos(z/r);
    phi=TMath::ATan(y/x);
    t->Fill();
  }
}
```

```
  if(title=="background"){
    for(int i=0; i<N;){
      phi=0.7*PI*gRandom->Rndm();
      r_ring=R0+sigma*gRandom->Gaus();
      r=(R0+sigma*gRandom->Gaus())*sqrt(6)/2;
      x=r_ring*TMath::Cos(phi);
      y=r_ring*TMath::Sin(phi);
      if(r*r-x*x-y*y>=0){
        z=sqrt(r*r-x*x-y*y);
        i++;
      }
      else continue;
      theta=TMath::ACos(z/r);
      t->Fill();
    }
  }
  t->Write();
  f->Close();
}
```



```
void doTraining(TTree * SigTree, TTree * BkgTree, string Label){
  std::cout << std::endl << "==> Start TMVAClassificationCategory" << std::endl;

  TString outfileName = Form("TMVAoutput_%s.root",Label.c_str());
  TFile * outputFile = new TFile(outfileName,"RECREATE");

  std::string factoryOptions( "!V:!Silent:Transformations=I;P;G" );
  TMVA::Factory *factory = new TMVA::Factory( "TMVAClassificationCategory", outputFile, factoryOptions );

  factory->AddVariable("x",'F');
  factory->AddVariable("y",'F');
  factory->AddVariable("z",'F');
  //factory->AddVariable("r",'F');
  //factory->AddVariable("theta",'F');
  //factory->AddVariable("phi",'F');

  factory->AddSignalTree    (SigTree,1.);
  factory->AddBackgroundTree(BkgTree,1.);

  TCut mycuts = "";
  TCut mycutb = "";
  factory->PrepareTrainingAndTestTree(mycuts,mycutb,"nTrain_Signal=0:nTrain_Background=0:nTest_Signal=0:nTest_Background=0:SplitMode=Random:NormMode=NumEvents:!V");
  factory->BookMethod( TMVA::Types::kBDT, "BDT","!H:!V:NTrees=600:nCuts=20:MaxDepth=2:MinNodeSize=10%:BoostType=AdaBoost:SeparationType=GiniIndex:PruneMethod=NoPruning" );
  factory->BookMethod(TMVA::Types::kFisher, "Fisher", "H:!V:Fisher");
  factory->BookMethod(TMVA::Types::kMLP, "MLP", "H:!V:HiddenLayers=3");

  factory->TrainAllMethods();
  factory->TestAllMethods();
  factory->EvaluateAllMethods();

  outputFile->Close();
  std::cout << "==> Wrote root file: " << outputFile->GetName() << std::endl;
  std::cout << "==> TMVAClassificationCategory is done!" << std::endl;

  delete factory;

  if (!gROOT->IsBatch()) TMVAGui( outfileName );
}
```