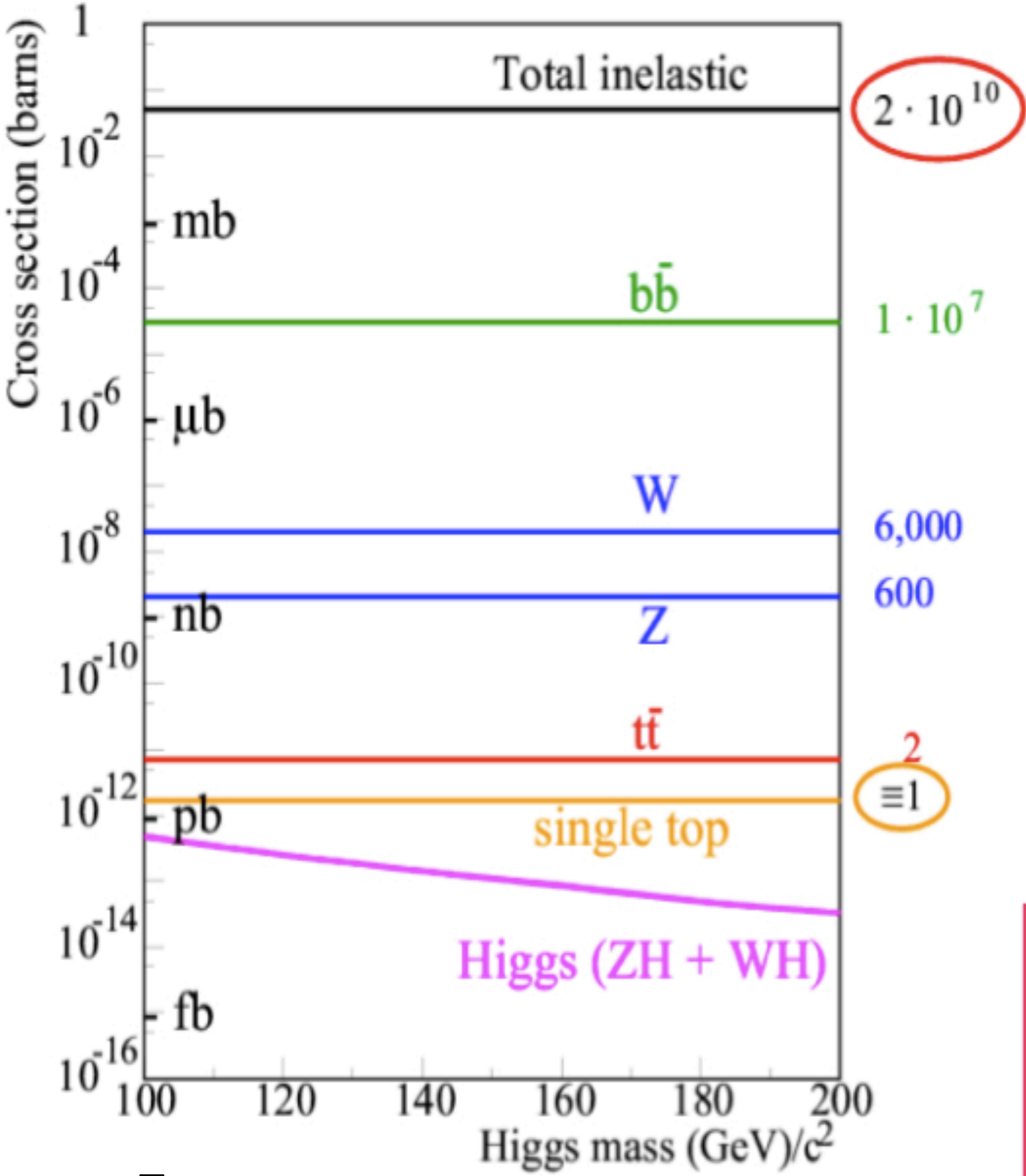
Advanced Analysis Methods

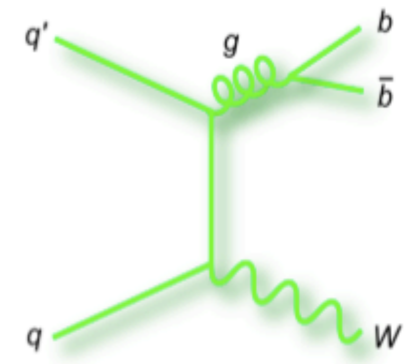
Tulika Bose

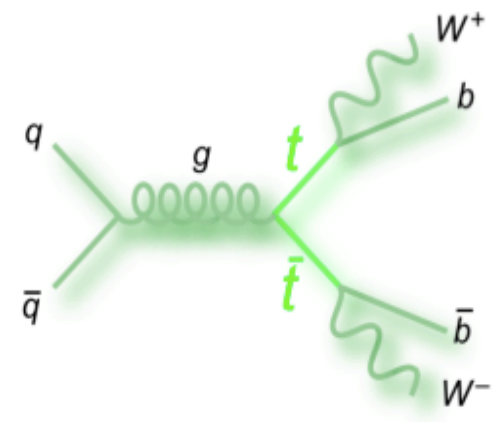
April 27th, 2009

Review of talk given by Reinhard Schwienhorst and others



Tevatron





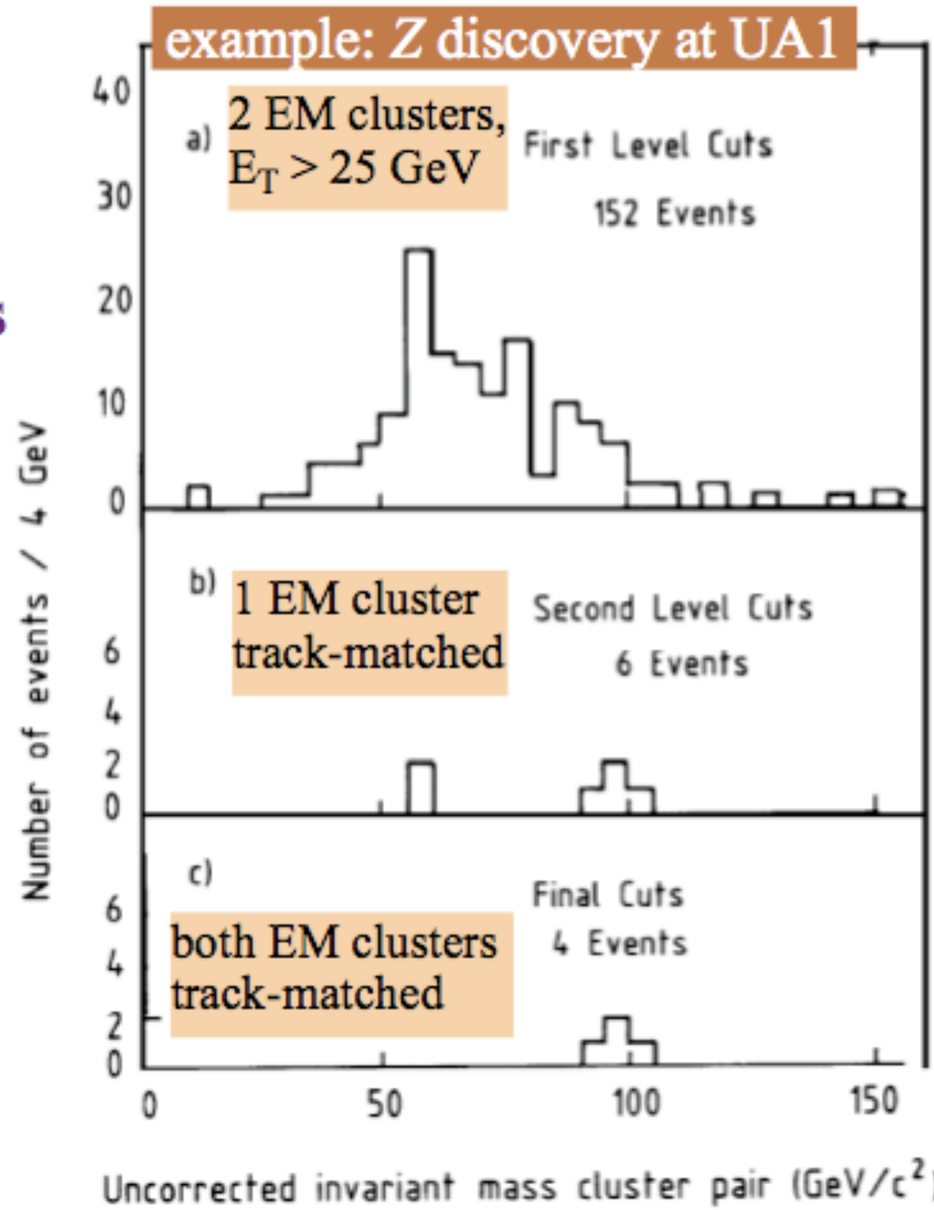
Simple counting experiment cannot extract the signal from the overwhelming background

Typical methods

- Cut-based event counting
- Peak in a characteristic distribution

Event counting

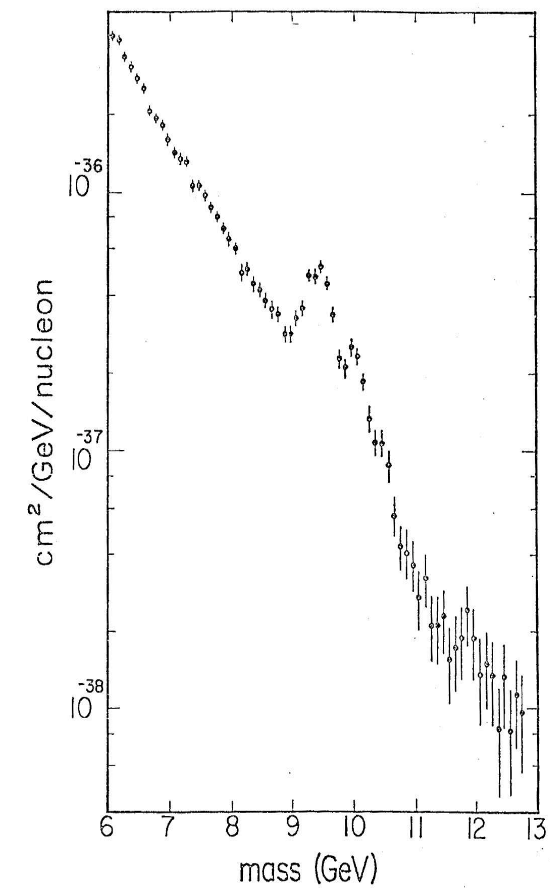
- Apply cuts to variables describing the event
  - Object identification
  - Kinematic cuts on objects
  - Event kinematics
- Goal: cut until the signal is visible
  - No background left
  - Or large $S/\sqrt{B}$
- Sensitive to any signal with this final state
- Requires understanding of background



Peak in a characteristic distribution

- Find a variable that has a smooth distribution for background
  - Typically invariant mass
- Measure this distribution over a large range of possible values
- Look for possible resonance peaks
- Sensitive to any resonance with this final state
- Background estimate for sidebands

Example: b-quark discovery at Fermilab

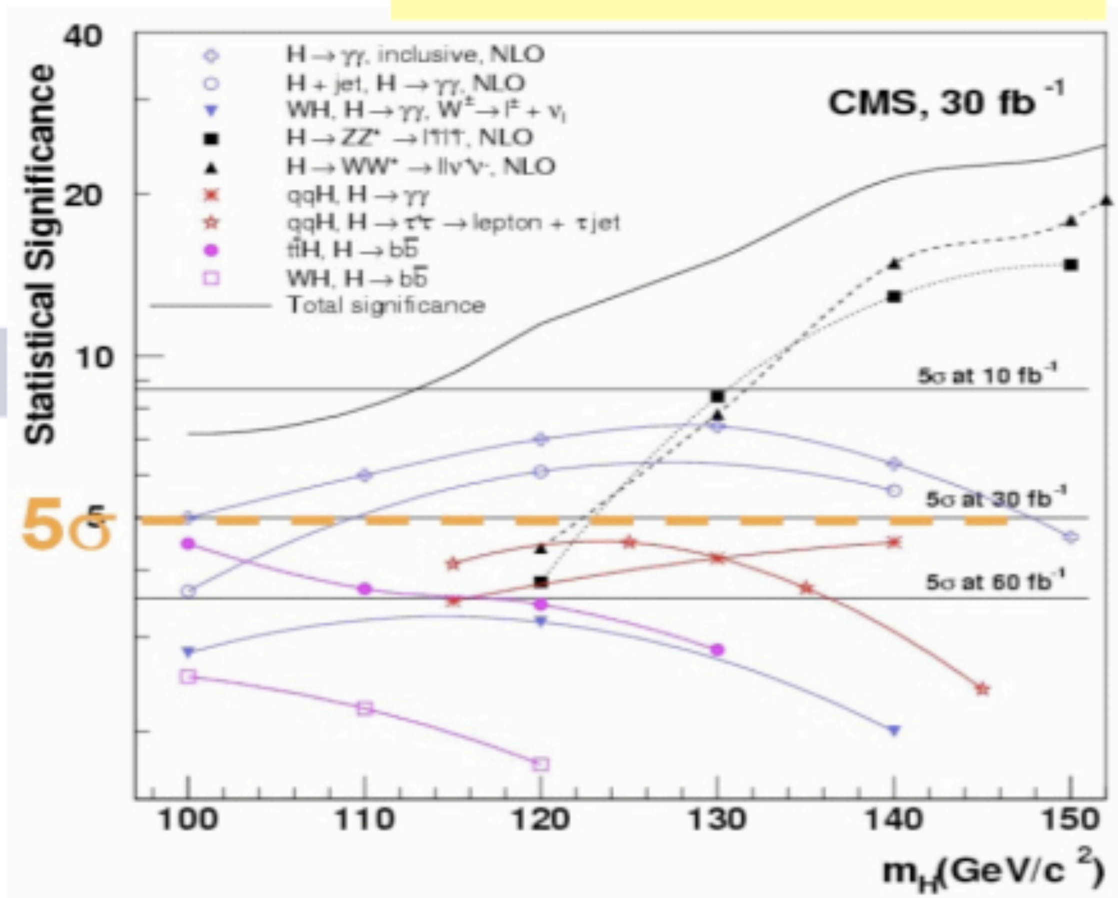


"Bump Hunting"

Searches at the energy frontier

- Searches for new particles, phenomena, couplings
  - Tevatron:
    - Single top quark production
    - Higgs boson search
    - SUSY
    - Extra dim
    - ...
  - LHC:
    - Higgs searches

LHC Higgs Sensitivity



Multivariate techniques will be required to reach this level of sensitivity

How to improve upon

Event Counting

And

Bump Hunting ?

Physics at the energy frontier

- Searches for new particles, phenomena, couplings
- First measurements of properties, couplings
- Multivariate techniques ↔ Adding more data

Making the most out of small samples of events

Bayesian limit

- For each analysis, there exists a fully optimized signal-background separation
  - Target function, also called Bayes discriminant or Bayesian limit

$$B(x) = \frac{L(S|x)}{L(B|x)}$$

$$r(\vec{x}) = \frac{p(s|\vec{x})}{p(b|\vec{x})} = \frac{p(x|s)p(s)}{p(x|b)p(b)}$$

Posterior probability

$$p(s|x) = \frac{r}{1+r} = \frac{p(x|s)}{p(x|b) + p(x|s)}$$

Bayesian limit

- For each analysis, there exists a fully optimized signal-background separation
  - Target function, also called Bayes discriminant or Bayesian limit

$$B(x) = \frac{L(S|x)}{L(B|x)}$$

- For a single discriminating variable, this ratio of signal and background likelihoods is easy to calculate
  - Monte Carlo procedure:
    - Generate signal and background MC events
    - Fill histograms for signal and background
    - Divide the two histograms

Bayesian Limit

- In case of more than one variable, this is not possible anymore
  - Not enough MC statistics to compute an multi-dimensional likelihood
  - Histogram data in M bins in each of the d feature variables
    - M^d bins
  - In high dimensions, we would either require a huge number of data points or most of the bins would be empty leading to an estimated density of zero.

- Curse of dimensionality

Optimized event analysis

Optimized = { Optimize signal-background separation
Exploit full event information
  Event kinematics, angular correlations, ...
Take all correlations into account

Goal: Reach the Bayesian limit

- Requires detailed understanding of signal and background
  - Only applicable to searches for a specific signal or measurements of a specific process

Optimized event analysis

Optimized = { Optimize signal-background separation
Exploit full event information
Event kinematics, angular correlations, ...
Take all correlations into account }

Goal: Reach the Bayesian limit

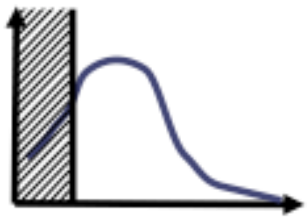
- Requires detailed understanding of signal and background
  - Only applicable to searches for a specific signal or measurements of a specific process
- Limited by background and signal modeling
  - MC statistics, MC model, background composition, shape,

<u>If signal model is wrong:</u> search is not sensitive

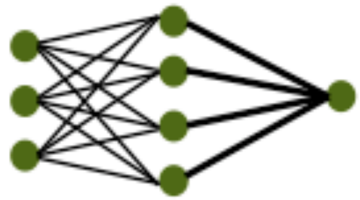
<u>If background model is wrong:</u> find something that isn't there

Event analysis techniques

Cut-Based

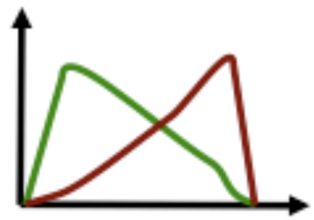


Neural networks



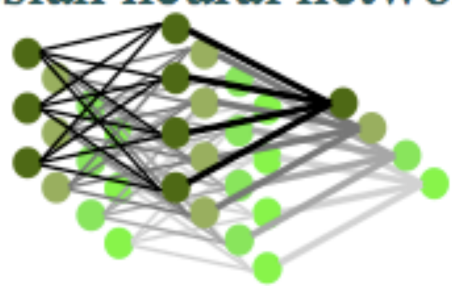
Decision trees

Likelihood



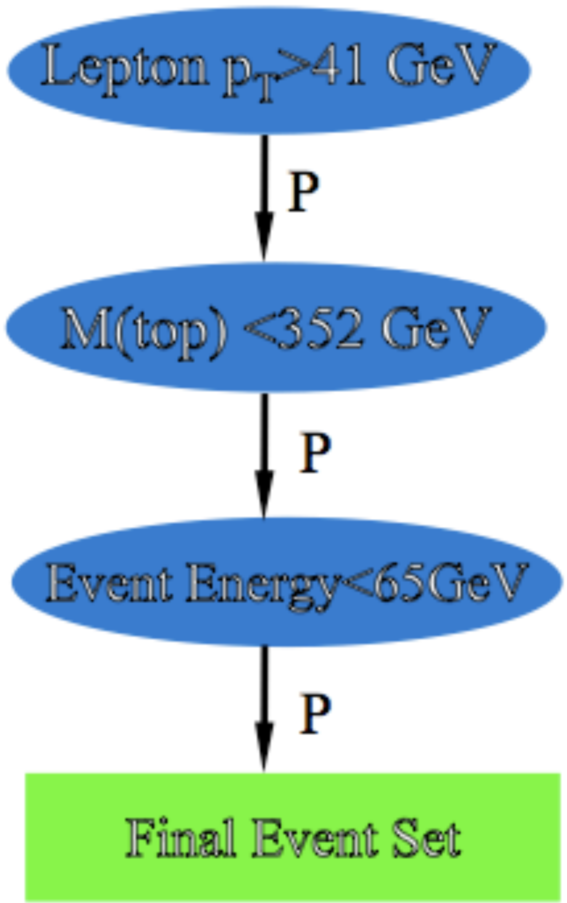
Boosted decision trees, random forest

Bayesian neural networks



Matrix Elements

Cut-based analysis



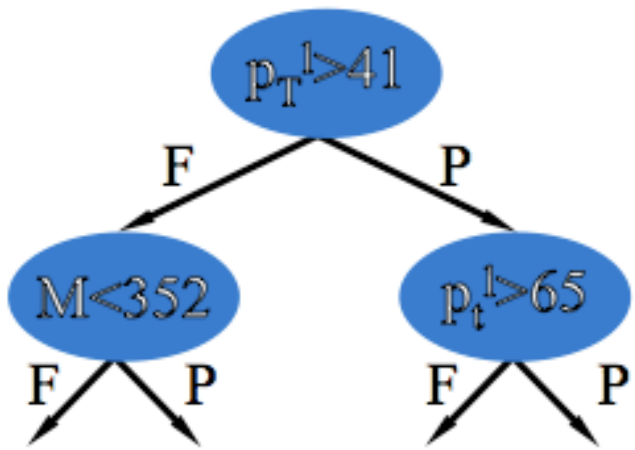
In the final event set

- Estimate background yield
- Compare to data
  $N_{obs} = N_{data} - N_B$
- Calculate signal acceptance
  $\sigma = N_{obs} / (A*L)$

Decision Trees

- Machine-learning technique, widely used in the social sciences
- Idea: recover events that fail criteria in cut-based analysis

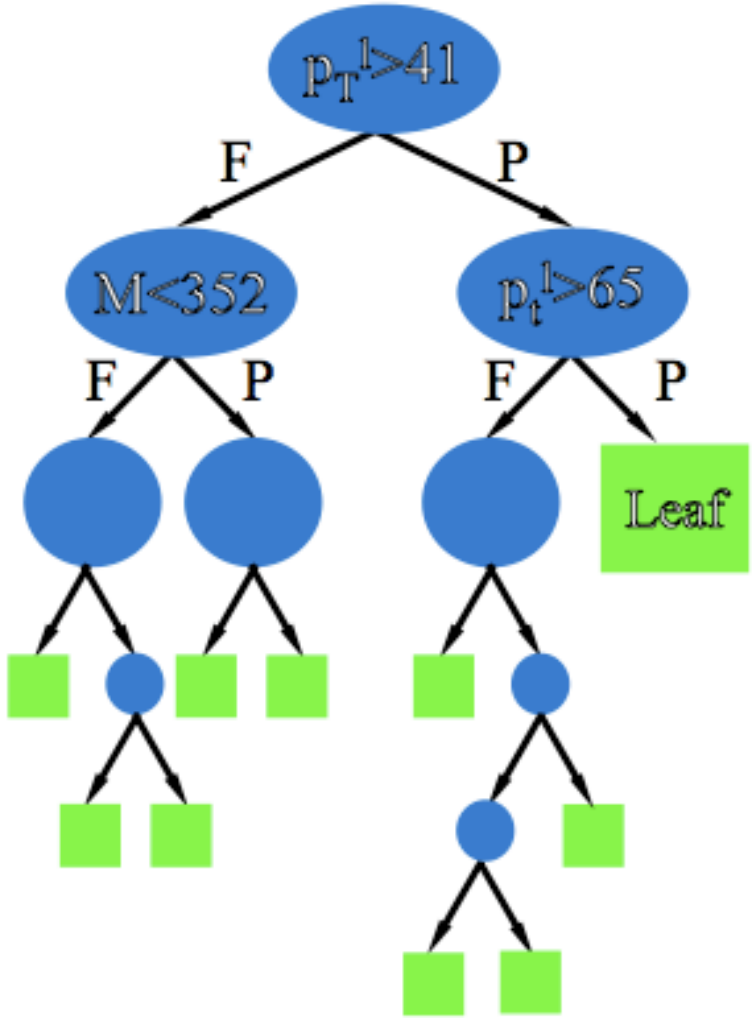
Including events that fail a cut



- Create a tree of cuts
- Divide sample into "pass" and "fail" sets
- Each node corresponds to a cut (branch)

- Start at first "node " with "training sample" of 1/3 of all signal and background events
  - For each variable, find splitting value with best separation between two children (mostly signal in one, mostly background in the other)
  - Select variable and splitting value with best separation to produce two "branches → " with corresponding events, (**F**)ailed and (**P**)assed cut

Trees and leafs



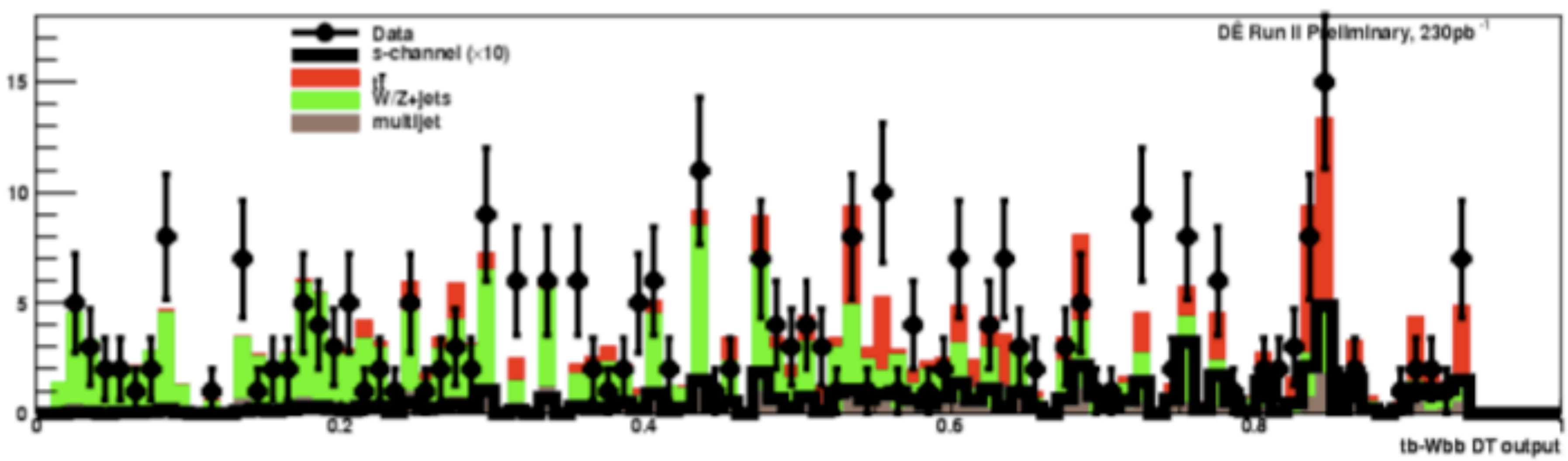
- Create a tree of cuts
- Divide sample into "pass" and "fail" sets
- Each node corresponds to a cut (branch)
- A leaf corresponds to an end-point
- For each leaf, calculate purity (from MC):
  purity = $N_S/(N_S+N_B)$

Repeat recursively on each node

Stop (terminate at leaf) when improvement stops or when too few events left

Decision tree output

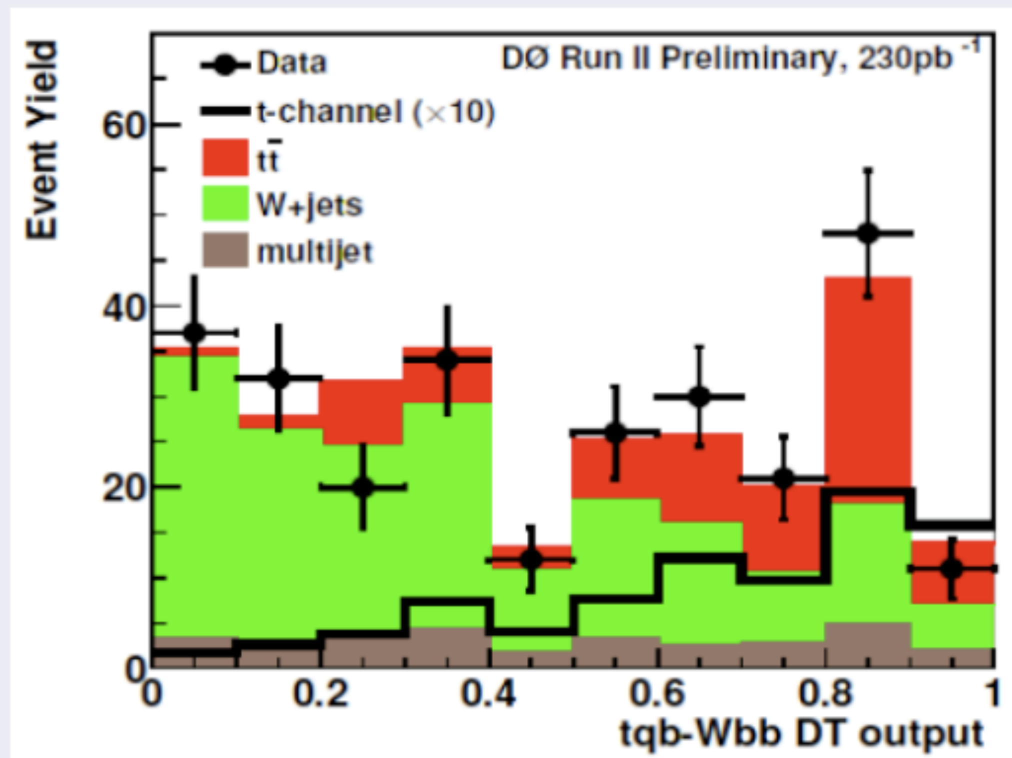
- Train on signal and background models (MC)
  - Stop and create leaf when $N_{MC}<100$
- Compute purity value for each leaf
- Send data events through tree
  - Assign purity value corresponding to the leaf to the event
- Result approximates a probability density distribution



Decision tree output for each event = leaf purity
Closer to 1 for signal and closer to 0 for background

Measure and Apply

- Take trained tree and run on independent simulated sample, determine purities.
- Apply to Data
- Should see enhanced separation (signal right, background left)
- Could cut on output and measure, or use whole distribution to measure.



Boosted Decision Trees

Boosting

- Recent technique to improve performance of a weak classifier
- Recently used on DTs by GLAST and MiniBooNE
- Basic principal on DT:
  - train a tree T_k
  - $T_{k+1} = \text{modify}(T_k)$

AdaBoost algorithm

- Adaptive boosting
- Check which events are misclassified by T_k
- Derive tree weight α_k
- Increase weight of misclassified events
- Train again to build T_{k+1}
- Boosted result of event i: $T(i) = \sum_{n=1}^{N_{\text{tree}}} \alpha_k T_k(i)$

- Averaging dilutes piecewise nature of DT
- Usually improves performance

Object Kinematics

- p_T(jet1)
- p_T(jet2)
- p_T(jet3)
- p_T(jet4)
- p_T(best1)
- p_T(notbest1)
- p_T(notbest2)
- p_T(tag1)
- p_T(untag1)
- p_T(untag2)

Angular Correlations

- ΔR(jet1,jet2)
- cos(best1,lepton)$_{\mathrm{besttop}}$
- cos(best1,notbest1)$_{\mathrm{besttop}}$
- cos(tag1,alljets)$_{\mathrm{alljets}}$
- cos(tag1,lepton)$_{\mathrm{btaggedtop}}$
- cos(jet1,alljets)$_{\mathrm{alljets}}$
- cos(jet1,lepton)$_{\mathrm{btaggedtop}}$
- cos(jet2,alljets)$_{\mathrm{alljets}}$
- cos(jet2,lepton)$_{\mathrm{btaggedtop}}$
- cos(lepton,Q(lepton)$\times z$)$_{\mathrm{besttop}}$
- cos(lepton,besttopframe)$_{\mathrm{besttopCMframe}}$
- cos(lepton,btaggedtopframe)$_{\mathrm{btaggedtopCMframe}}$
- cos(notbest,alljets)$_{\mathrm{alljets}}$
- cos(notbest,lepton)$_{\mathrm{besttop}}$
- cos(untag1,alljets)$_{\mathrm{alljets}}$
- cos(untag1,lepton)$_{\mathrm{btaggedtop}}$

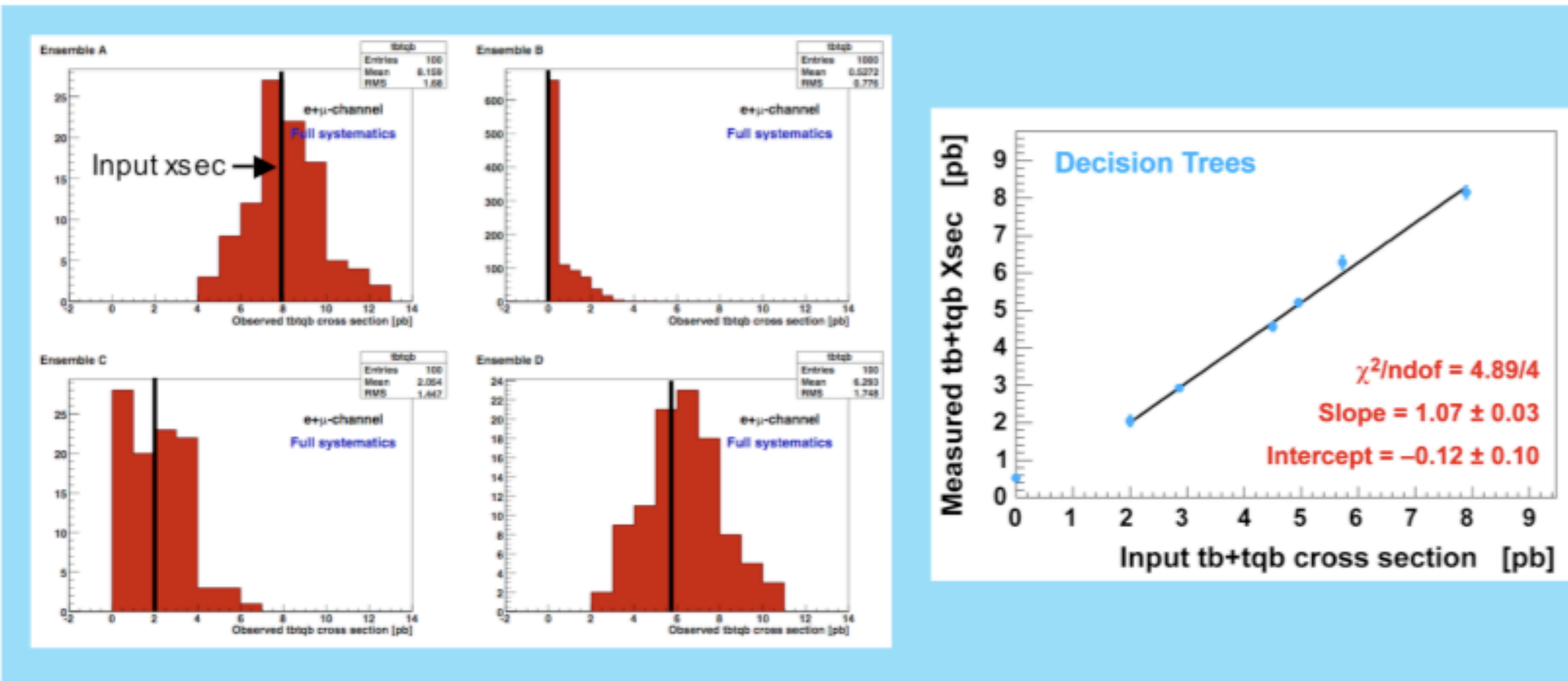
Event Kinematics

- Aplanarity(alljets,W)
- M(W,best1) ("best" top mass)
- M(W,tag1) ("b-tagged" top mass)
- H_T(alljets)
- H_T(alljets$-$best1)
- H_T(alljets$-$tag1)
- H_T(alljets,W)
- H_T(jet1,jet2)
- H_T(jet1,jet2,W)
- M(alljets)
- M(alljets$-$best1)
- M(alljets$-$tag1)
- M(jet1,jet2)
- M(jet1,jet2,W)
- M_T(jet1,jet2)
- M_T(W)
- Missing E_T
- p_T(alljets$-$best1)
- p_T(alljets$-$tag1)
- p_T(jet1,jet2)
- Q(lepton)$\times\eta$(untag1)
- $\sqrt{\hat{s}}$
- Sphericity(alljets,W)

- Adding variables does not degrade performance
- Tested shorter lists, lose some sensitivity
- Same list used for all channels

Decision Tree Verification

- Use "mystery" ensembles with many different signal assumptions
- Measure signal cross section using decision tree outputs
- Compare measured cross sections to input ones
- **Observe linear relation close to unit slope**

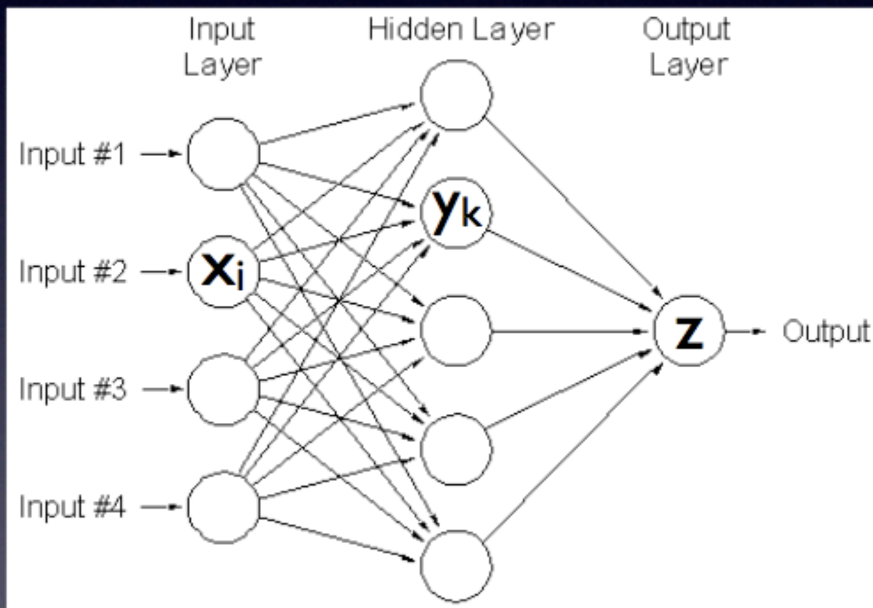


Random forest

- Average over many decision trees
  - Typically O(100)
- Each tree is grown using m variables
  - For N total variables, m<<N
- Very fast algorithm
  - Even with large number of variables
- Very few parameters to adjust
  - Typically only m

Neural Networks

Example Neural Network



Mathematics of Neural Networks

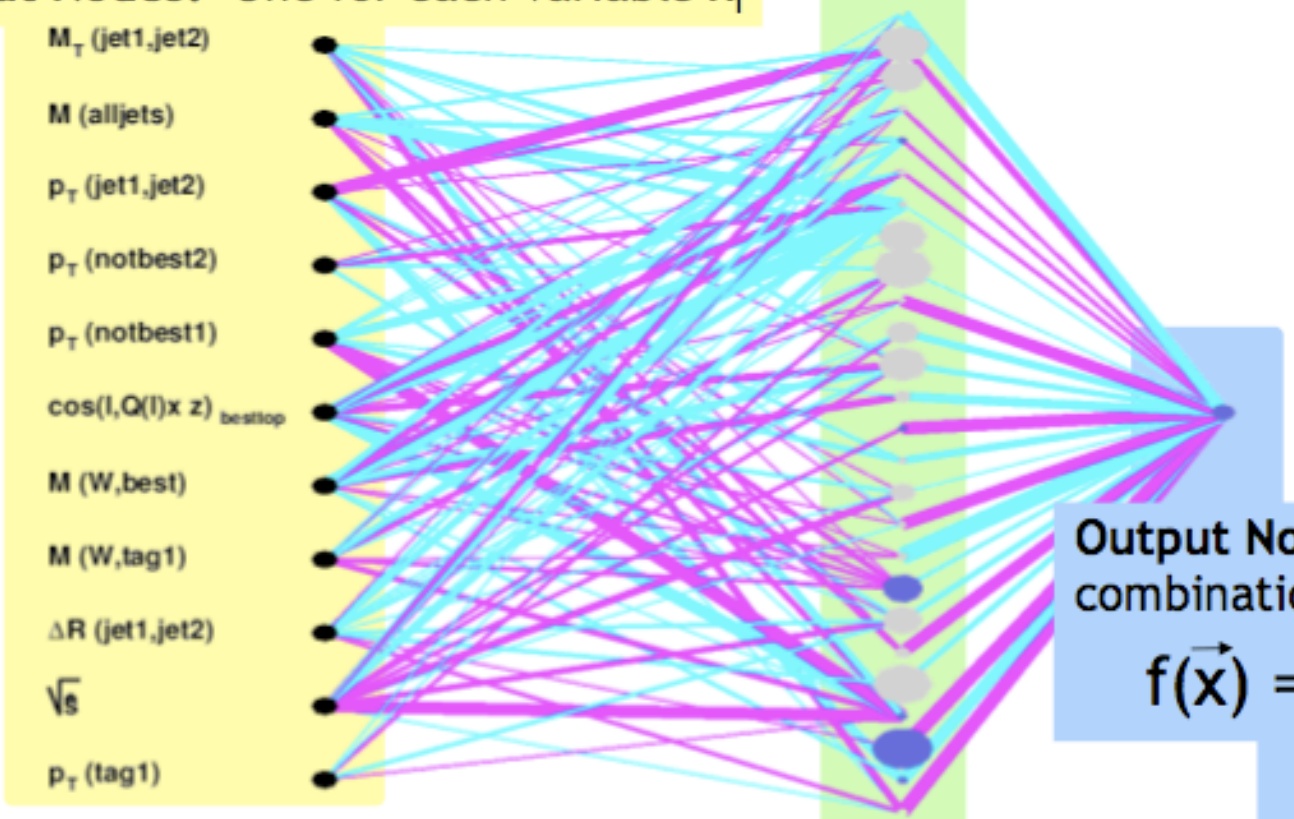
$$y_k = x_o' + \sum_i^n w_i \cdot x_i$$

$$z = x_o'' + \sum_k^m w_k \cdot y_k$$

- The activity of the input units represents the raw info that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

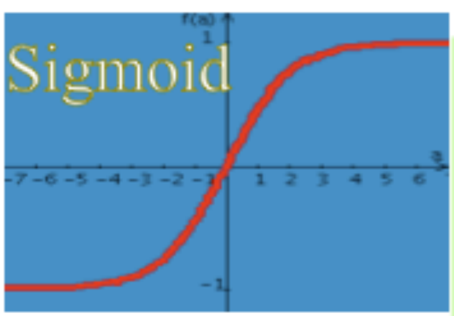
Neural networks

Input Nodes: One for each variable x_i



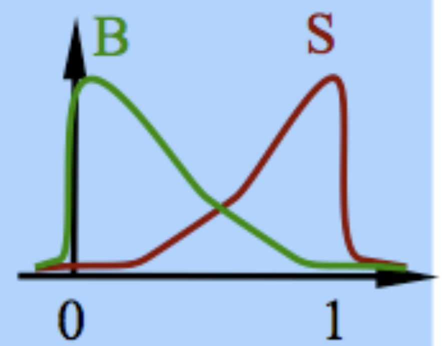
Output Node: linear combination of hidden nodes

$$f(\vec{x}) = \Sigma\, w'_k\, n_k(\vec{x},\vec{w}_k)$$



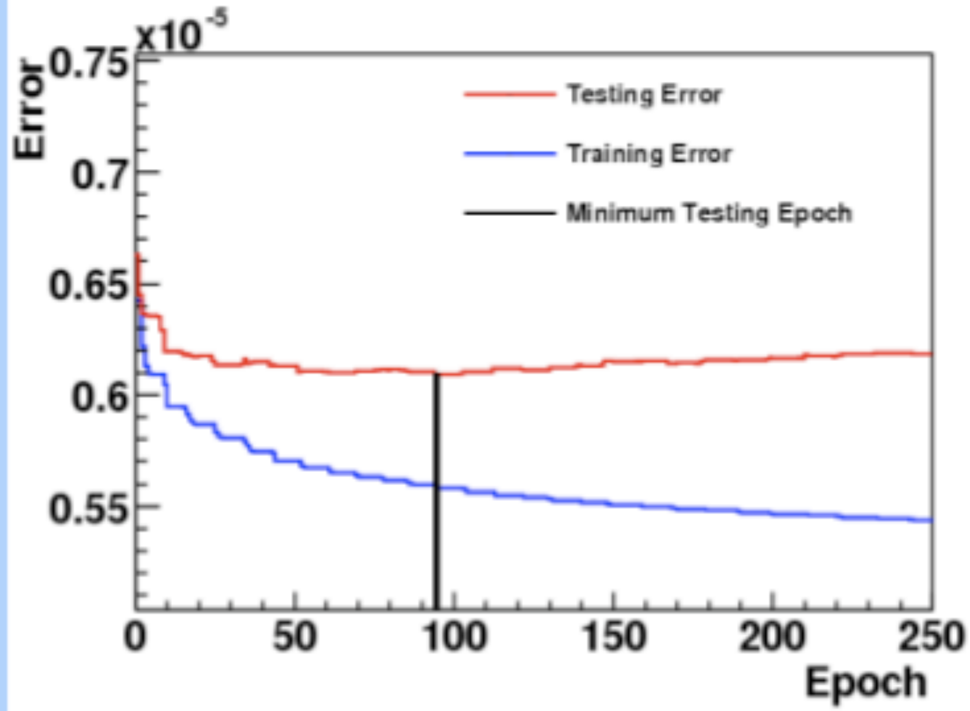
Hidden Nodes: Each is a sigmoid dependent on the input variables

$$n_k(\vec{x},\vec{w}_k) = \frac{1}{1 + e^{-\Sigma w_{ik} x_i}}$$



Neural Network Training

- Find optimum NN parameters on training signal/background events
- Apply NN to independent set of signal and background
  - Testing sample
- Stop training when error from testing sample starts increasing
  - Overfitting



DØ single top search

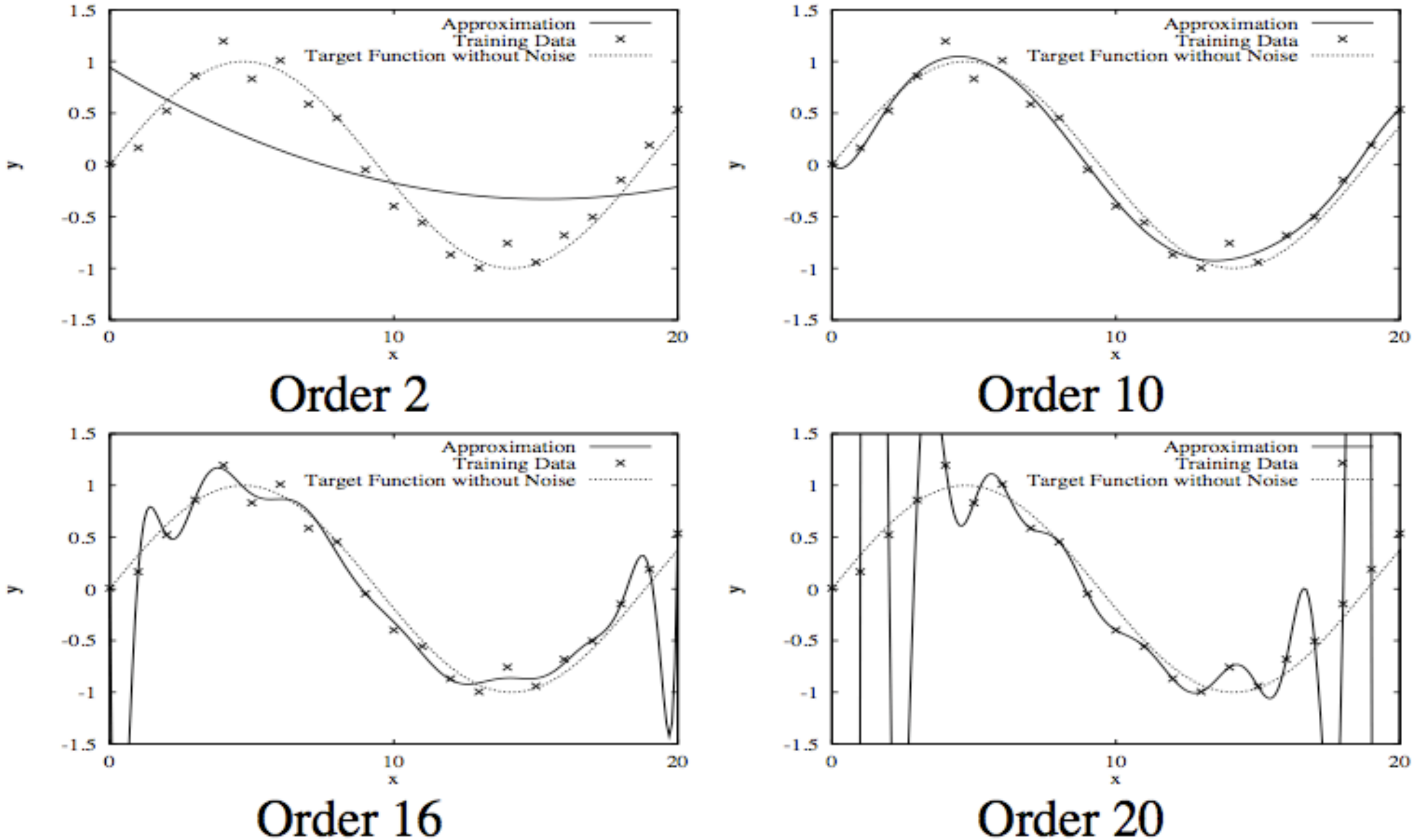
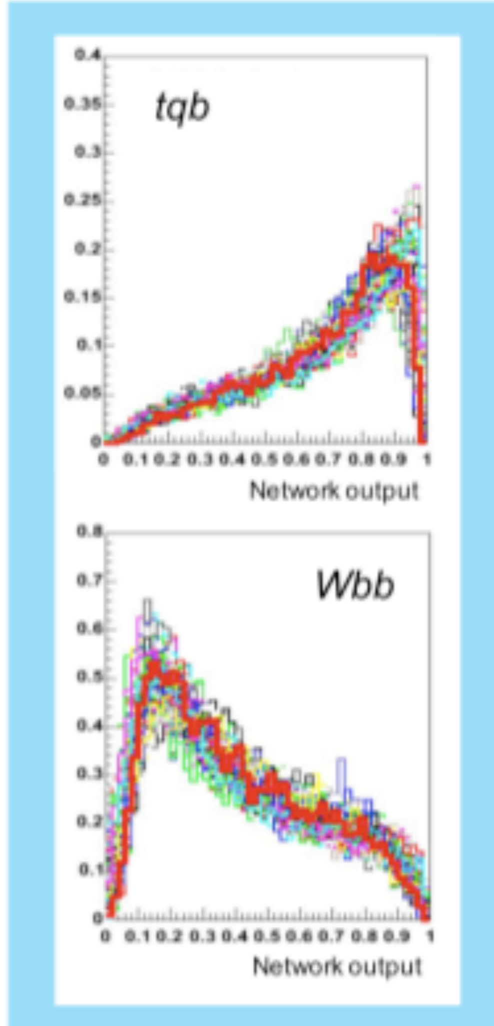


Figure 1. Polynomial interpolation of the function $y = \sin(x/3) + \nu$ in the range 0 to 20 as the order of the model is increased from 2 to 20. ν is a uniformly distributed random variable between -0.25 and 0.25. Significant overfitting can be seen for orders 16 and 20.

Signal-Background Separation using Bayesian Neural Networks

- Neural networks use many input variables, train on signal and background samples, produce one output discriminant
- **Bayesian neural networks improve on this technique:**
  - Average over many networks weighted by the probability of each network given the training samples
  - Less prone to over-training
  - Network structure is less important – can use larger numbers of variables and hidden nodes
- **For this analysis:**
  - 24 input variables (subset of 49 used by decision trees)
  - 40 hidden nodes, 800 training iterations
  - Each iteration is the average of 20 training cycles
  - One network for each signal (*tb+tqb, tb, tqb*) in each of the 12 analysis channels
- Bayesian neural network verification with ensembles shows good linearity, unit slope, near-zero intercept



Matrix Element Analysis

A matrix elements analysis takes a very different approach:

- Use the 4-vectors of all reconstructed leptons and jets
- Use matrix elements of main signal and background diagrams to compute an event probability density for signal and background hypotheses.
- Goal: calculate a discriminant:

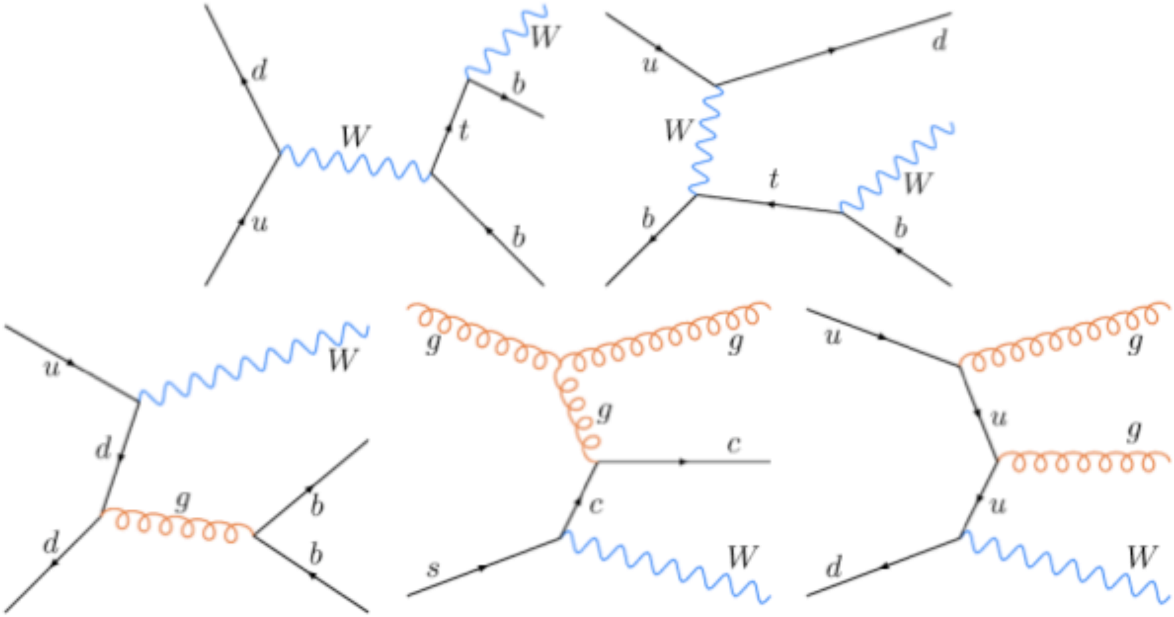
$$D_s(\vec{x}) = P(S|\vec{x}) = \frac{P_{Signal}(\vec{x})}{P_{Signal}(\vec{x}) + P_{Background}(\vec{x})}$$

- Define P_{Signal} as properly normalized differential cross section

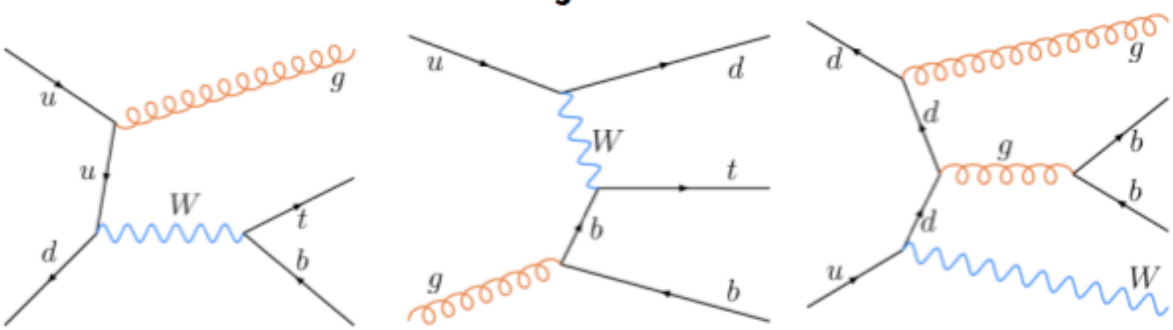
$$P_{Signal}(\vec{x}) = \frac{1}{\sigma_S} d\sigma_S(\vec{x}) \quad \sigma_S = \int d\sigma_S(\vec{x})$$

- Shared technology with mass measurement in $t\bar{t}$(eg. transfer functions)

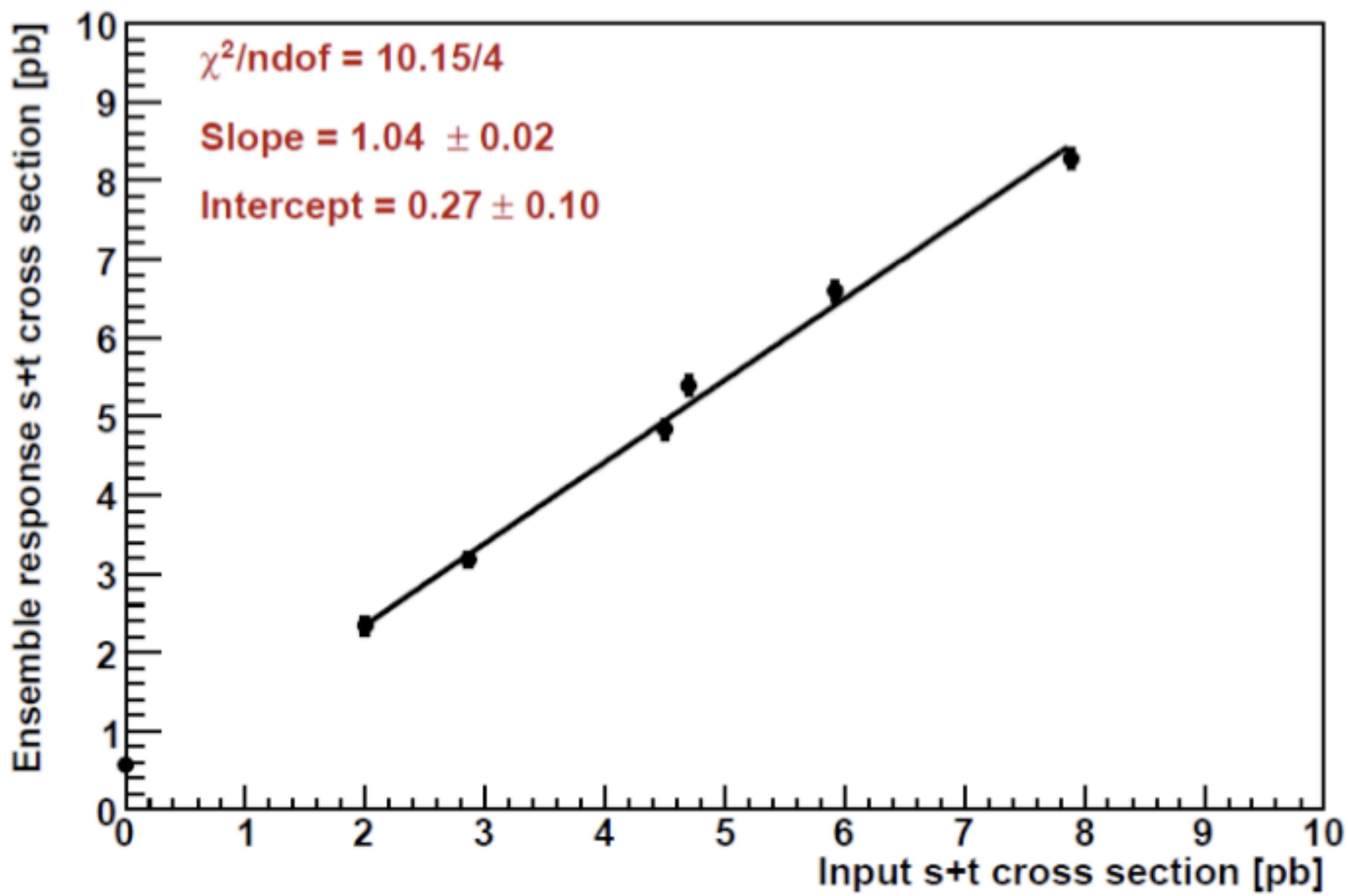
2-jets:


d u W t b W b
u d W t b W b
u W d g d b b
g g g c c s W
u g u g u d W

3-jets:

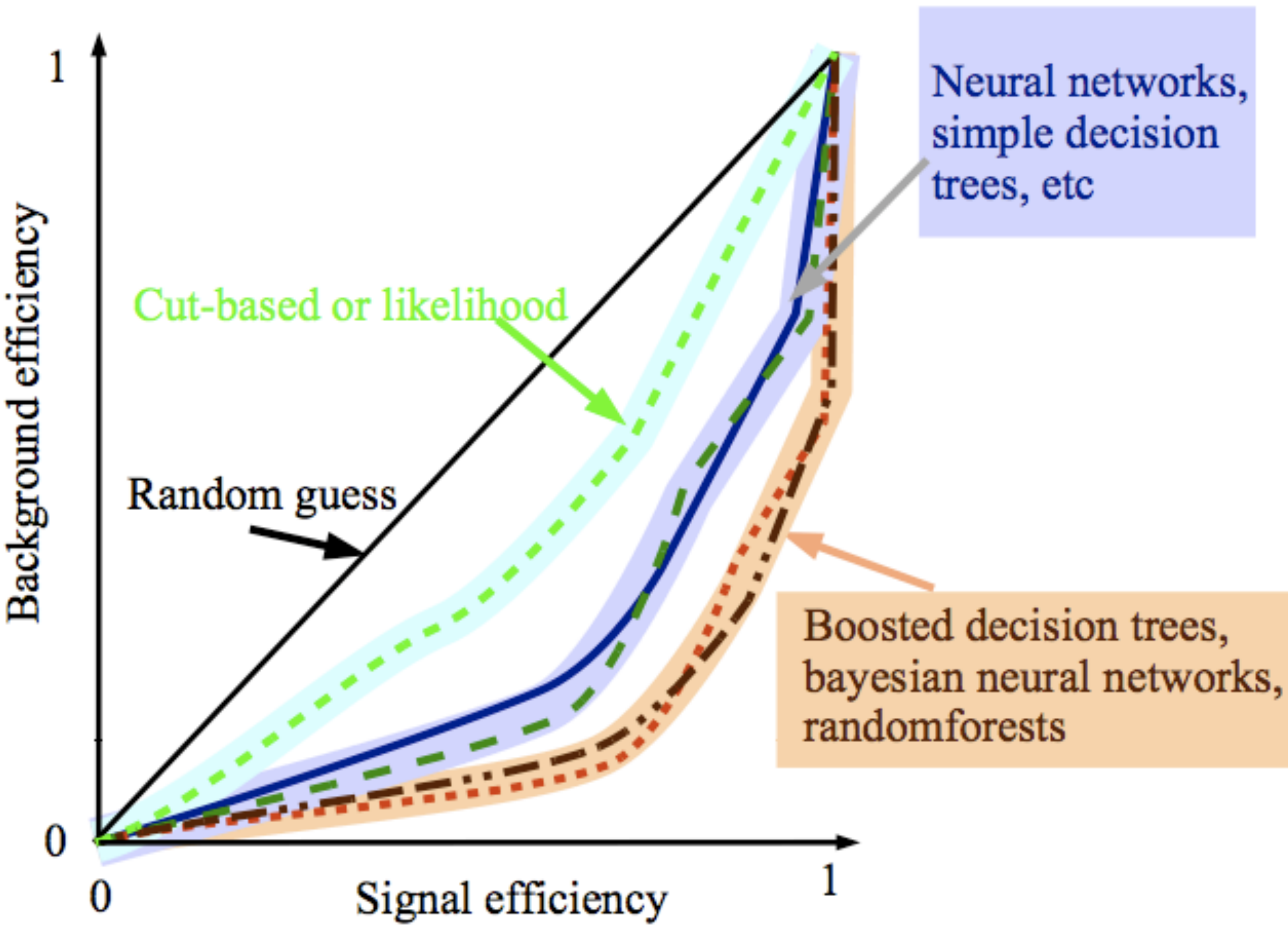

u g u W d t b
u d W t b g b
d g d g b b d u W

ME analysis



χ^2/ndof = 10.15/4
Slope = 1.04 ± 0.02
Intercept = 0.27 ± 0.10
Ensemble response s+t cross section [pb]
Input s+t cross section [pb]

Summary


1
0
Background efficiency
Signal efficiency
0
1
Random guess
Cut-based or likelihood
Neural networks, simple decision trees, etc
Boosted decision trees, bayesian neural networks, randomforests

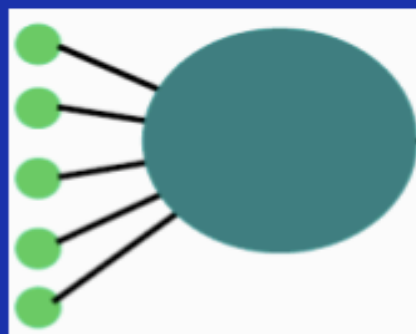
Resources

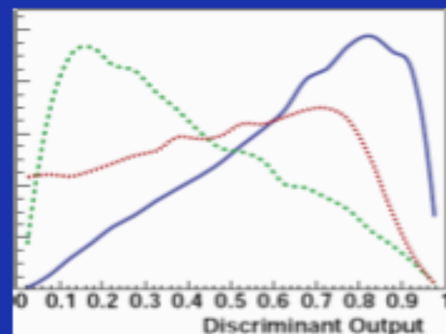
- PhyStat code repository
  https://plone4.fnal.gov:4430/P0/phystat/
- PhyStat 2007 conference
  http://phystat-lhc.web.cern.ch/phystat-lhc/
- Jim Linnemann's collection of statistics links:
  http://www.pa.msu.edu/people/linnemann/stat_resources.html
- Statistical analysis tool R
  http://www.r-project.org/
- TMVA (multivariate analysis tools in root)
  http://tmva.sourceforge.net/
- Neural Networks in Hardware
  http://neuralnets.web.cern.ch/NeuralNets/nnwInHep.html
- Boosted Decision Trees in MiniBoone
  http://arxiv.org/abs/physics/0508045
- Decision Tree Introduction
  http://www.statsoft.com/textbook/stcart.html
- GLAST Decision Trees
  http://scipp.ucsc.edu/~atwood/Talks%20Given/CPAforGLAST.ppt

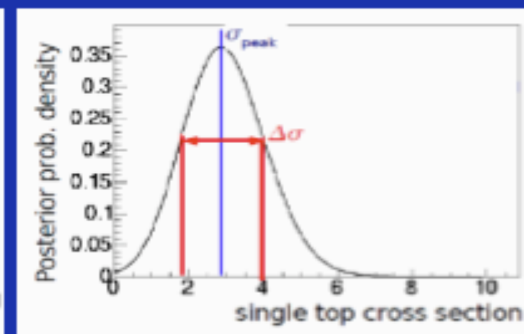
Analysis Strategy

Discriminating variables ➜ **Multivariate Classifier** ➜ **Signal Likelihood** ➜ **Statistical Analysis**

- Event kinematics
- Object kinematics
- constructed masses
- Angular correlations
- ...







Classifiers

- Likelihood Function (LF)
- Neural Network (NN)
- Bayesian Neural Networks(BNN)
- Boosted Decision Trees (BDT)
- Matrix Element (ME)

Build Bayesian posterior probability density to measure cross section

- Shape normalization and systematics treated as nuisance parameters
- Correlations between uncertainties properly accounted for
- Flat prior in signal cross section

Statistical Analysis

Before looking at the data, we want to know two things:

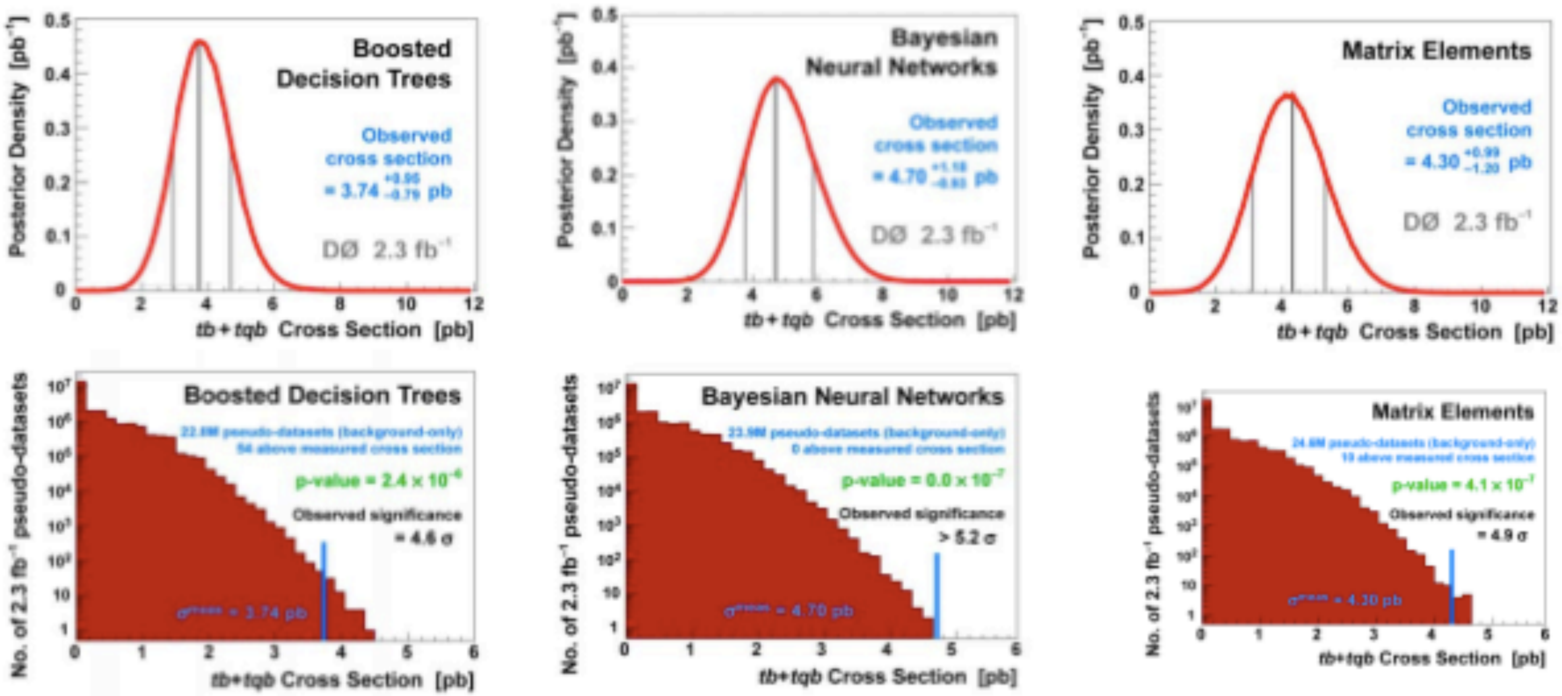
- **By how much can we expect to rule out a background-only hypothesis?**
  - Find what fraction of the ensemble of zero-signal pseudo-datasets give a cross section at least as large as the SM value, the "expected p-value"
  - For a Gaussian distribution, convert p-value to give "expected signficance"
- **What precision should we expect for a measurement?**
  - Set value for "data" = SM signal + background in each discriminant bin (non-integer) and measure central value and uncertainty on the "expected cross section"

With the data, we want to know:

- **How well do we rule out the background-only hypothesis?**
  - Use the ensemble of zero-signal pseudo-datasets and find what fraction give a cross section at least as large as the measured value, the "measured p-value"
  - Convert p-value to give "measured signficance"
- **What cross section do we measure?**
  - Use (integer) number of data events in each bin to obtain "measured cross section"
- **How consistent is the measured cross section with the SM value?**
  - Find what fraction of the ensemble of SM-signal pseudo-datasets give a cross section at least as large as the measured value to get "consistency with SM"

Cross Section Results

MVA	σ±Δσ(pb)	Expected Sensitivity	Observed Sensitivity
BDT	$3.74 \pm^{0.95}_{0.79}$	4.3 σ	4.6 σ
BNN	$4.70 \pm^{1.18}_{0.93}$	4.1 σ	5.2 σ
ME	$4.30 \pm^{0.99}_{1.20}$	4.1 σ	4.9 σ



Bayesian neural networks

- Bayesian idea:
  - Rather than finding one value for each weight, determine the posterior probability for each weight
- Form many networks by sampling from the posterior
- Typical case: ~100 individual neural networks
  - Each network gets a weight based on training performance
- Avoids overfitting
- But: very slow due to integration required to determine the posterior