Uses of Multivariate Analysis Methods

Yann Coadou
on behalf of the CDF and DØ collaborations

Simon Fraser University

International Workshop on Top Quark Physics
University of Coimbra, Portugal, 12-15 January 2006

Outline

CDF and DØ top groups techniques

Single variable techniques

- Cut-based
- Template methods

CDF and DØ top groups techniques

Single variable techniques

- Cut-based
- Template methods

Multivariate approaches

- Matrix elements
- Kernel density estimation
- Dynamic likelihood method
- Neural networks

Yesterday's talks

- Top pair production cross section (R. Rossin)
- Top decay properties (E. Varnes)
- Top mass in ℓ+jets channel (J. Cammin)
- Top mass in dilepton channel (B. Jayatilaka)
- Search for single top (M. Begel)

CDF and DØ top groups techniques

Single variable techniques

- Cut-based
- Template methods

Multivariate approaches

- Matrix elements
- Kernel density estimation
- Dynamic likelihood method
- Neural networks

Yesterday's talks

- Top pair production cross section (R. Rossin)
- Top decay properties (E. Varnes)
- Top mass in ℓ+jets channel (J. Cammin)
- Top mass in dilepton channel (B. Jayatilaka)
- Search for single top (M. Begel)

Why multivariate analyses?

Data is multivariate

- Relatively similar signal and background ⇒ simple cuts cannot separate them
- Few events ⇒ use all information available to keep as many signal events as possible

CDF and DØ top groups techniques

Single variable techniques

- Cut-based
- Template methods

Multivariate approaches

- Matrix elements
- Kernel density estimation
- Dynamic likelihood method
- Neural networks

Yesterday's talks

- Top pair production cross section (R. Rossin)
- Top decay properties (E. Varnes)
- Top mass in ℓ+jets channel (J. Cammin)
- Top mass in dilepton channel (B. Jayatilaka)
- Search for single top (M. Begel)

Why multivariate analyses?

Data is multivariate

- Relatively similar signal and background ⇒ simple cuts cannot separate them
- Few events ⇒ use all information available to keep as many signal events as possible

Illustration: techniques used in DØ single top group

- Likelihood discriminants
- Neural networks
- Decision trees
- Boosted decision trees

Datasets preparation

Advanced techniques are useless if inputs are not correct

Selection of events

- good object ID and resolution
- use basic criteria that keep events with particular final state

Generate realistic Monte Carlo events

- signal
- all backgrounds not extracted from data

Find discriminating variables (and their correlations)

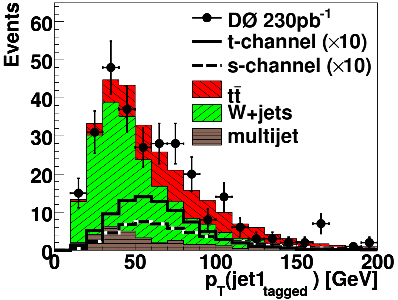
- key to analysis performance

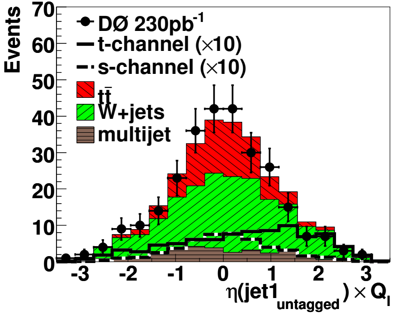
Most important: check models (bkg and/or signal) describe data

- overall normalization
- variables shapes

Cut-based analysis

- Reference for any "advanced" technique
- Example: DØ single top search





Cut-based analysis

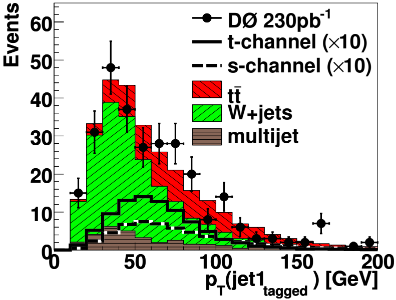
- Reference for any "advanced" technique
- Example: DØ single top search

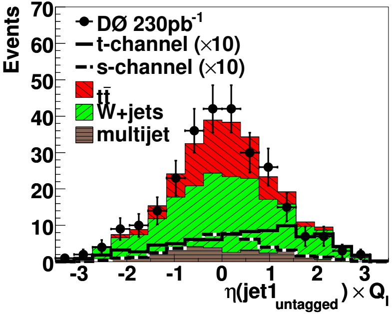
Random grid search

- Optimal cut (minimizes expected limit) on all variables
- Combine sets of variables and re-optimize
- Select set yielding lowest expected limit
- Set limits by counting events

Published limits (230 pb^{-1}) [observed (expected)]

	s-channel	*t*-channel
Preselection	13.0 (14.5) pb	13.6 (16.5) pb
Cut-based	10.6 (9.8) pb	11.3 (12.4) pb





Multivariate analysis methods

Likelihood discriminants

Likelihood for a vector of measurements $\vec{x} = x_i$

$$\mathcal{L}(\vec{x}) = \frac{\mathcal{P}_{signal}(\vec{x})}{\mathcal{P}_{signal}(\vec{x}) + \mathcal{P}_{background}(\vec{x})}$$

- $\mathcal{L}$ close to 0 for background and 1 for signal

- Probability Density Functions:

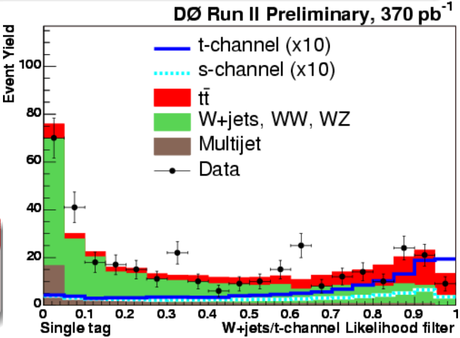
$$\mathcal{P}(\vec{x}) = \prod_{i}^{N_{variables}} P(x_i)$$

$P(x_i)$ = normalized x_i variable distribution

- Built likelihood for signal/W+jets and signal/$t\bar{t}$, with 7 to 10 variables
- Advantage: no training

Prelim. limits (370 pb^{-1}) [observed (expected)]

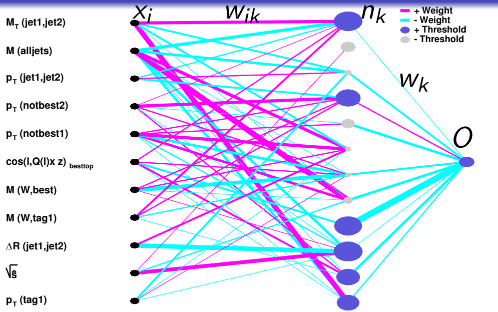
	s-channel	t-channel
Likelihood	5.0 (3.3) pb	4.4 (4.3) pb



Neural networks

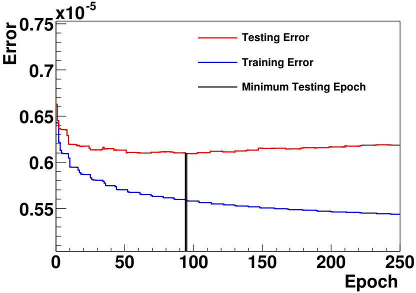
MultiLayer Perceptron

- MLPFit implementation
- Input layer nodes: variables x_i
- Hidden layer nodes:
  $n_k = \frac{1}{1+\exp^{-\sum w_{ik} x_i}}$
- Output node: $O = \sum w_k n_k$



Training method

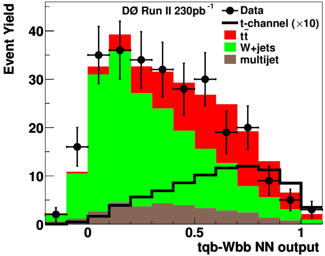
- Initialize weights, minimize error function on training sample, update weights $\Rightarrow$ first epoch
- Repeat procedure. After each epoch, apply NN on independent testing sample. Stop training when testing error increases (avoid overtraining)



Neural networks

Training for single top search

- Train signal/*Wbb* and signal/lepton + jets networks
- Train on 60% of events, test on remaining 40%
- Use logarithm of non-angular variables
- Use MLPfit hybrid method for error function minimization



Network optimization

- Optimize choice of input variables, using 11 out of 30 variables
- Optimize number of hidden nodes (found close to 30)
- Optimize number of training epochs (150-250)
- Very powerful technique but:
  - slow to train
  - set of weights sensitive to training events
  - sensitive to extra variables

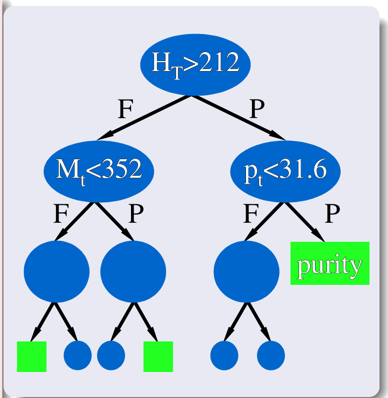
Published limits (230 pb^{-1}) [obs'd (exp'd)]

	s-channel	*t*-channel
Cut-based	10.6 (9.8) pb	11.3 (12.4) pb
Neural net	6.4 (4.5) pb	5.0 (5.8) pb

Decision trees

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

- Start with all events = first node
  - sort all events by each variable
  - 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, produce two branches with corresponding events ((F)ailed and (P)assed cut)
- Repeat recursively on each node
- Splitting stops: terminal node = leaf



- Run testing events and data through tree to derive limits

DT output = leaf purity

Ref: Breiman *et al*, "Classification and Regression Trees", Wadsworth (1984)

Tree construction parameters

Normalization of signal and background before training

- same total weight for signal and background events

Selection of splits

- list of questions ($variable_i > cut_i$?)
- goodness of split

Decision to stop splitting (declare a node terminal)

- minimum leaf size
- insufficient improvement from splitting

Assignment of terminal node to a class

- signal leaf if purity > 0.5
- background otherwise

Splitting a node

Impurity $i(t)$

- maximum for equal mix of signal and background
- symmetric in p_{signal} and $p_{background}$
- minimal for node with either signal only or background only
- strictly concave $\Rightarrow$ reward purer nodes

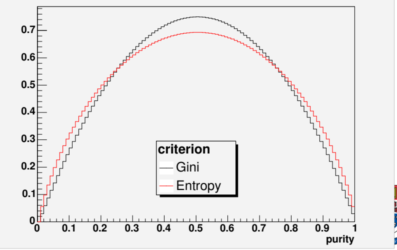
- Decrease of impurity for split s of node t into children t_L and t_R (goodness of split):
$\Delta i(s,t) = i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R)$
- Aim: find split s^* such that:

$$\Delta i(s^*,t) = \max_{s \in \{\text{splits}\}} \Delta i(s,t)$$

Examples

$Gini = 1 - \sum_{i=s,b} p_i^2 = \frac{2sb}{(s+b)^2}$

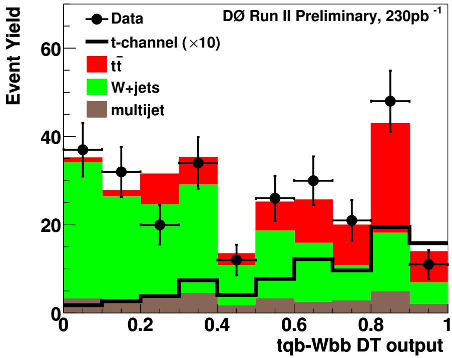
$entropy = -\sum_{i=s,b} p_i \log p_i$



- Maximizing $\Delta i(s,t) \equiv$ minimizing overall tree impurity

Decision tree output

- Followed same training strategy as NN analysis (different trees for different backgrounds)



Advantages

- DT has human readable structure (no black box)
- Training is fast
- Deals with discrete variables
- No need to transform inputs
- Resistant to irrelevant variables

Limitations

- Piecewise nature of output
- Instability of tree structure

Limits (230 pb^{-1}) [observed (expected)]

	s-channel	*t*-channel
Neural net	6.4 (4.5) pb	5.0 (5.8) pb
Decision tree	8.3 (4.5) pb	8.1 (6.4) pb

Similar sensitivity

Boosting a decision tree

Boosting

- Recent technique to improve performance of a weak classifier
- Recently used on decision trees in HEP by GLAST and MiniBooNE (Nucl. Instrum. Meth. A **543**, 577 (2005) [physics/0408124])
- Basic principal on DT:
  - train a tree T_k
  - minimize error function
  - $T_{k+1} = \text{modify}(T_k)$

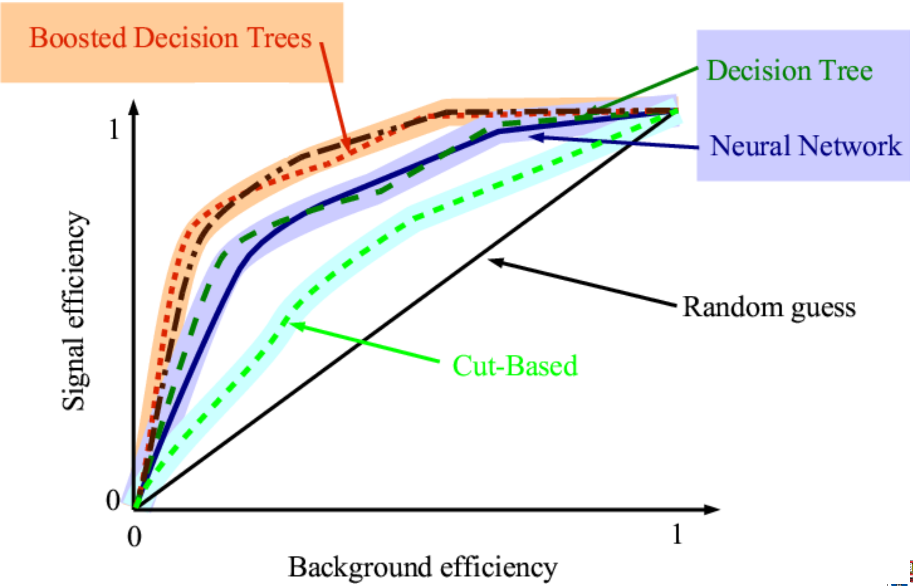
AdaBoost algorithm

- 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

Ref: Freund and Schapire, "Experiments with a new boosting algorithm", in *Machine Learning: Proceedings of the Thirteenth International Conference*, pp 148-156 (1996)

Comparison


Boosted Decision Trees
Decision Tree
Neural Network
Random guess
Cut-Based
Signal efficiency
Background efficiency
1
0
0
1

Summary and outlook

- Many different analysis techniques used by Tevatron top groups
  - single variable methods
  - multivariate approaches
- For all methods: need good inputs first
  - good reconstruction and identification of physics objects
  - realistic Monte Carlo events that describe data
- Advanced techniques useful for precision measurements, searches with small statistics
- Example: different techniques in DØ single top searches (likelihood discriminants, neural networks, decision trees, boosted decision trees)
- Ongoing:
  - improved results with more statistics and new strategies
  - boosted decision tree results soon
  - superNN: combining results of multiple NN into one