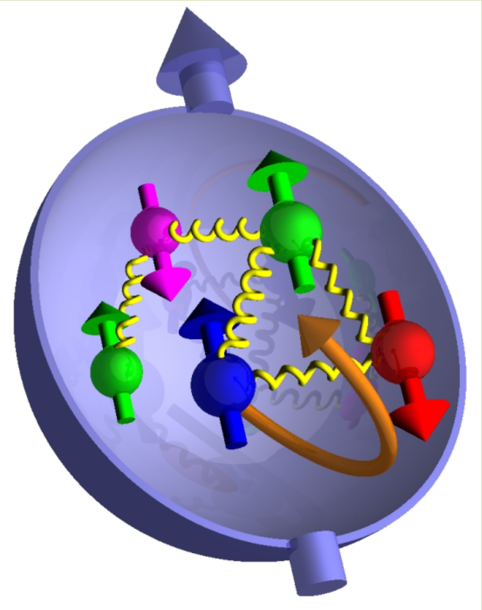
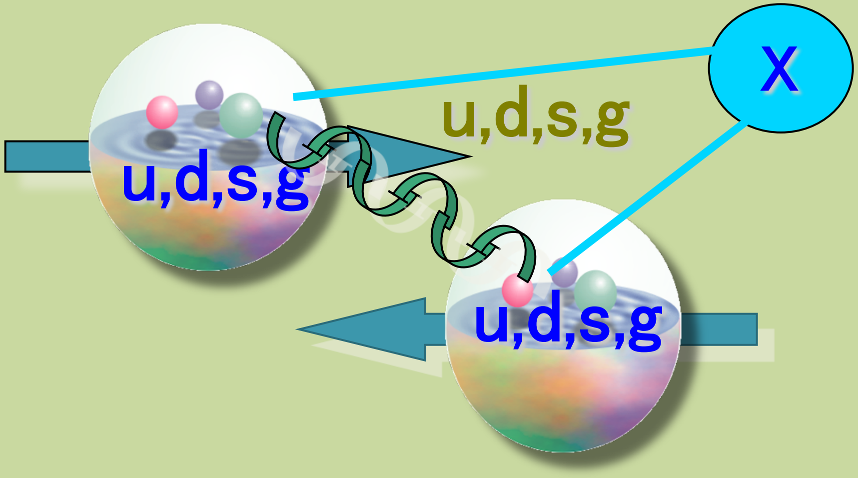




Photon-jet Coincidence Measurements in Polarized pp Collisions at 200 GeV at STAR

Seema Dhamija for the STAR Collaboration
Indiana University

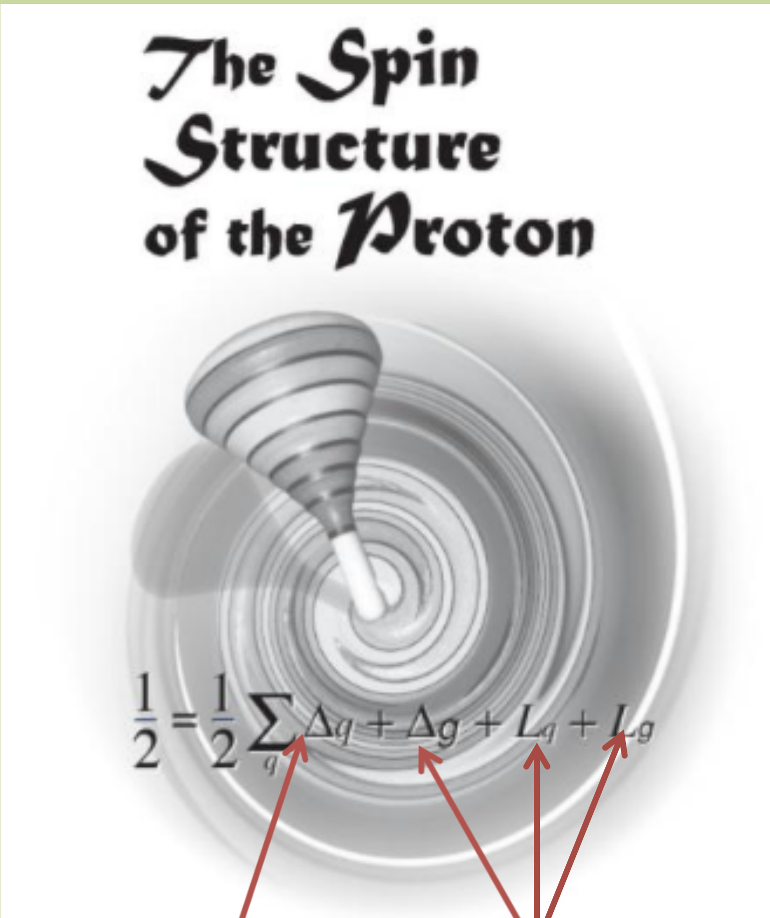
DNP Meeting
October 25, 2012

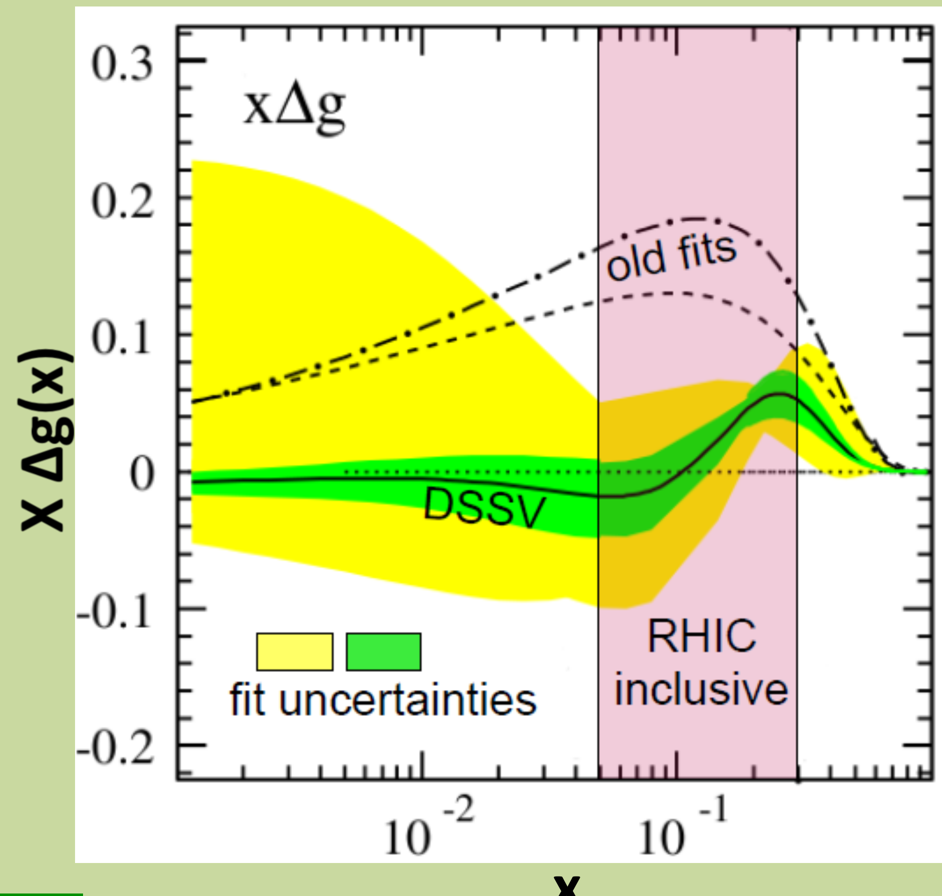
How does the proton spin?

In a simple model of the nucleon, the proton's spin structure can be decomposed into four parts:

$$S_z = \frac{1}{2}\Delta\sum + \Delta G + L_z^q + L_z^G = \frac{1}{2}$$

DSSV analysis includes : NLO pQCD calculations with fits to polarized (semi)-inclusive DIS and Inclusive RHIC data



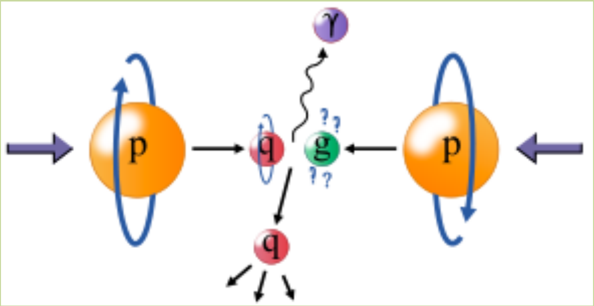


Polarized DIS : ~ 0.3

Poorly Constrained

de Florian et al., PRL 101, 072001

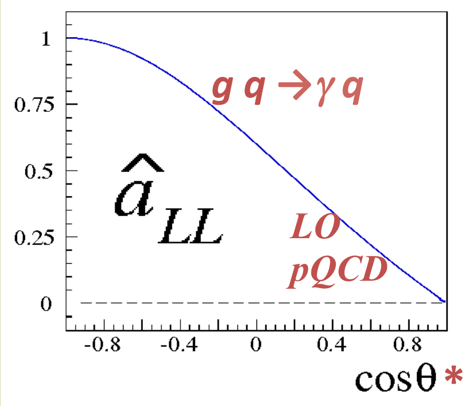
Why photon+jet channel?



➢ Direct γ dominated (~ 90% of yield) by QCD Compton process: q+g → q+γ, with large LO gluon spin sensitivity

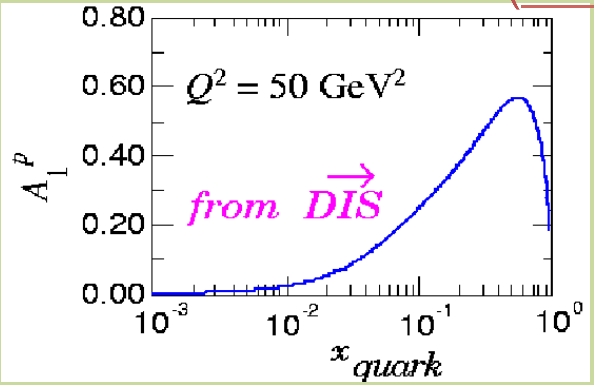
➢ Inclusive γ cannot compete statistically with incl. jet A_{LL} … but γ-jet conic. meas. a "golden channel"

$$A_{LL} = \frac{\sigma^{++} - \sigma^{+-}}{\sigma^{++} + \sigma^{+-}} \propto \frac{\Delta f_a \Delta f_b}{f_a f_b} \hat{a}_{LL}$$



➢ Select kinematics to optimize ΔG(x) sensitivity: high x_q ⇒ high $\Delta f_q / f_q$ (large quark polarization);

(cross section also peaks here!)



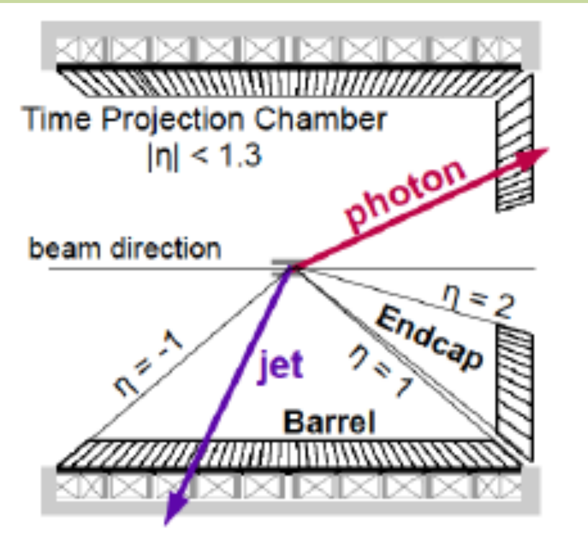
➢ For γ-jet coincidences, p_T^γ, η_γ, η_{jet} ⇒ x_1, x_2 and the angle θ* can be determined event-by-event.

➢ One uses high-x quarks (where most polarized) to probe low-x gluons (where they are abundant)

➢ above: very asymmetric collisions ⇒ γ's boosted into STAR Endcap EMC

How do we reconstruct photon+jet?

STAR Detector



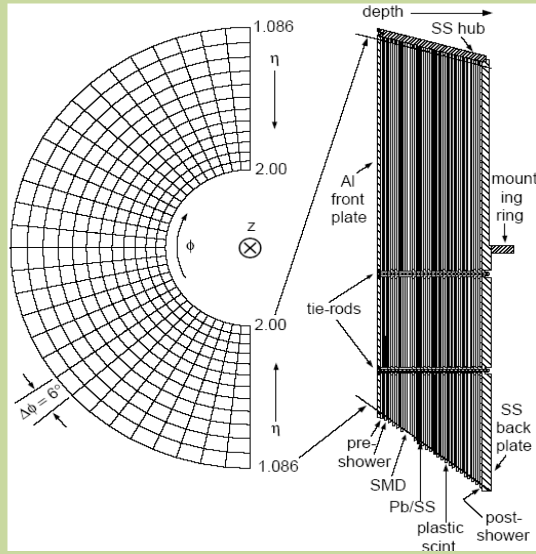
z vertex cut : $-100cm < z < 100cm$
photon : $1.08 < \eta_\gamma < 2$ $p_t^\gamma > 7$ GeV

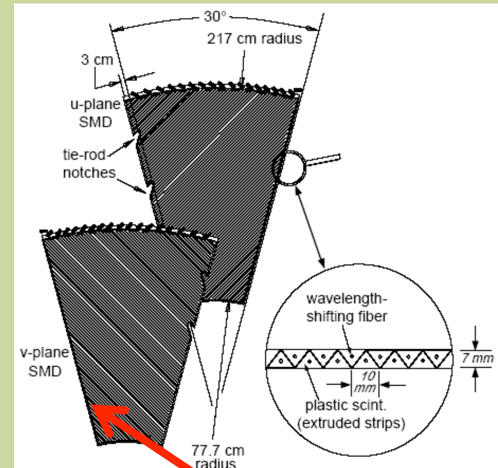
jet : $|\eta_{jet}| < 0.8$ $p_t^{jet} > 5$ GeV

photon and jet are back-to-back
$\cos(\varphi_{jet} - \varphi_\gamma) < -0.8$

Away side jet neutral fraction < 0.9

Endcap Electromagnetic Calorimeter





Discriminates between single photon vs photon pairs

- Coverage: $1.086 < \eta < 2.0$, $0 < \varphi < 2\pi$
- 12 sectors×5 subsectors×η-bins=720 towers
- 1 tower=24 layers:
  - Layer 1=preshower-1
  - Layer 2=preshower-2
  - Layer-24=postshower

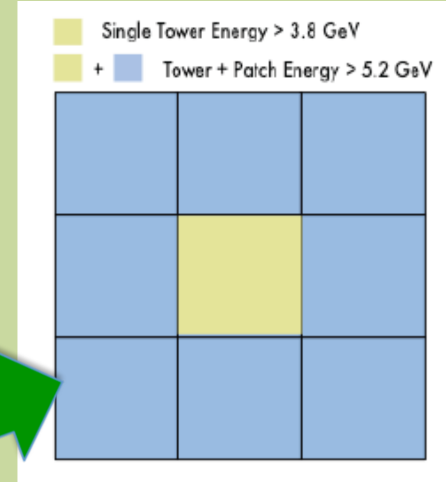
γ/π^0 discrimination

$e^\pm$ / hadron discrimination

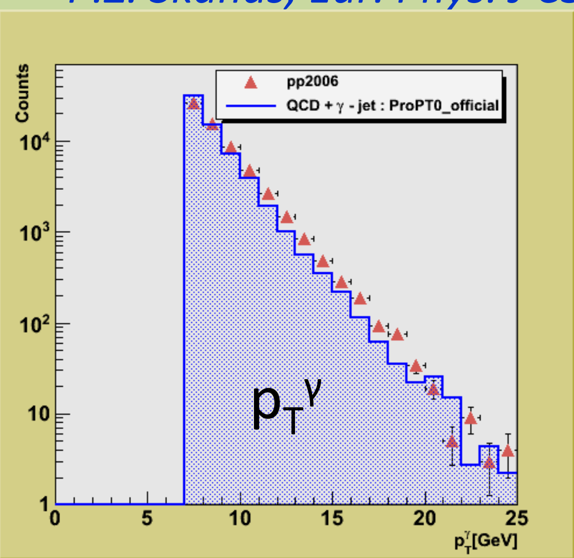
Run 6 photon+jet analysis

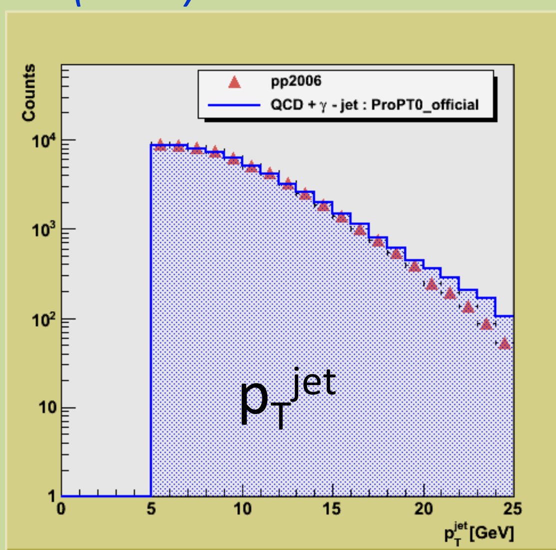
Data

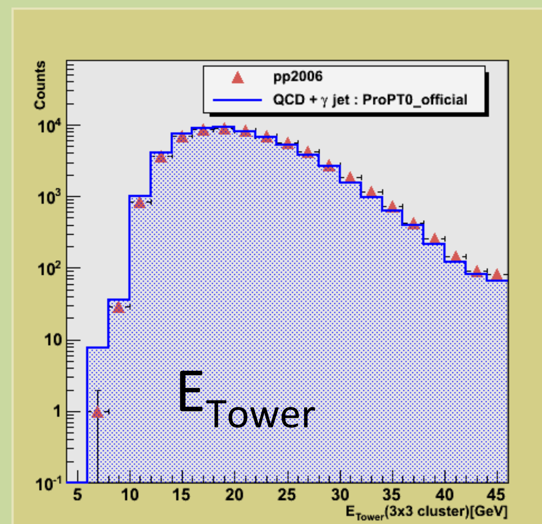
- 2006 Data : 3.1 pb^{-1} taken during RHIC Run 6
- Trigger : High tower with 3.8 GeV E_T and associated 3x3 patch with 5.2 GeV
- 2006 Simulation
  6 STAR MC productions for prompt photon and 4 STAR MC productions for QCD background with partonic p_T 2-35 GeV
- PYTHIA 6.410 ProPT$_0$ tune (pytune 329) T. Sjostrand and P.Z. Skands, Eur. Phys. J C39, 129 (2005).



3x3 tower "trigger patch" showing configuration of L2γ trigger condition



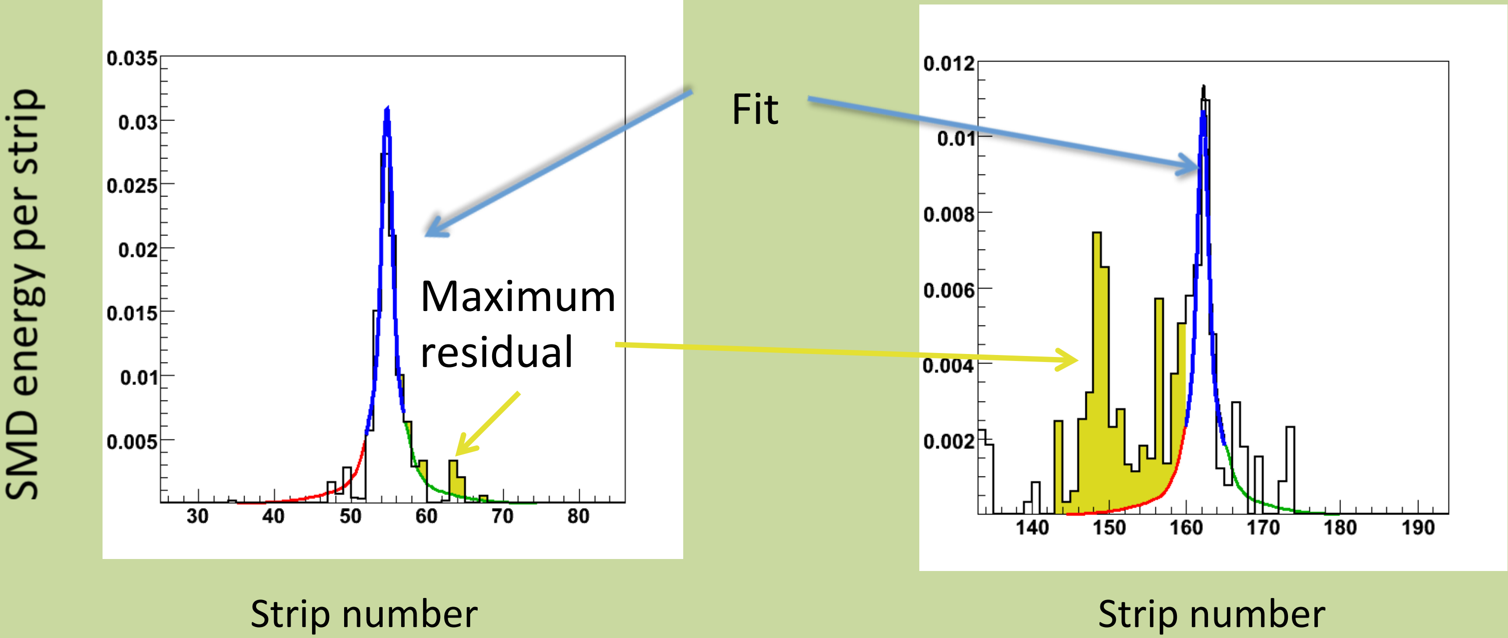




γ/π^0 discrimination in Endcap SMD :
Maximum Sided Residual

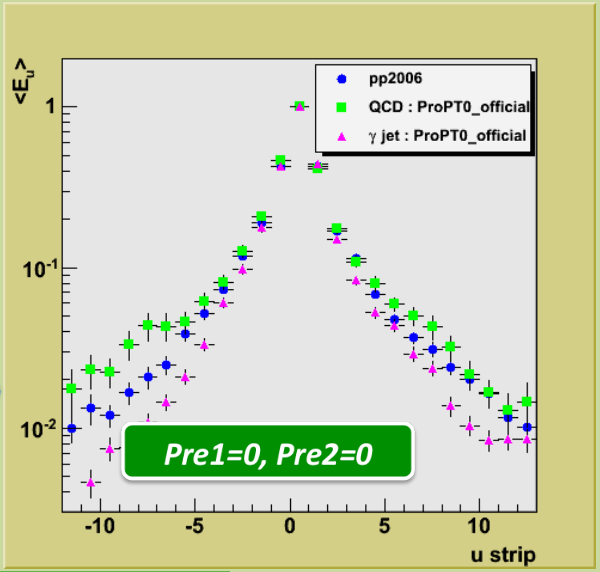
- *Basic idea:*
  - Look at transverse shower profile in the SMD
  - γ and e transverse shower profile ⇒ single peak
  - $\pi^0 \rightarrow \gamma\gamma$ ⇒ double peak structure
  - Fit main peak and compute residual=data-fit on each side of main peak ⇒ pick maximum residual
  - For given energy E, π^0 should have more residual than γ

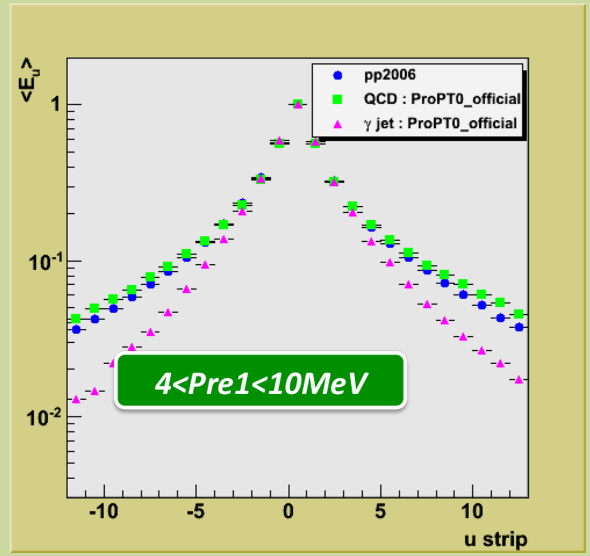
Single photon response ***Multi photon response***



Data/Simulation Run 6

Transverse shower profile in the shower maximum detector





Direct photon rich

Background dominated

Background : multi-photon production processes, such as π^0 -> YY

Events pre-sorted into four different categories :

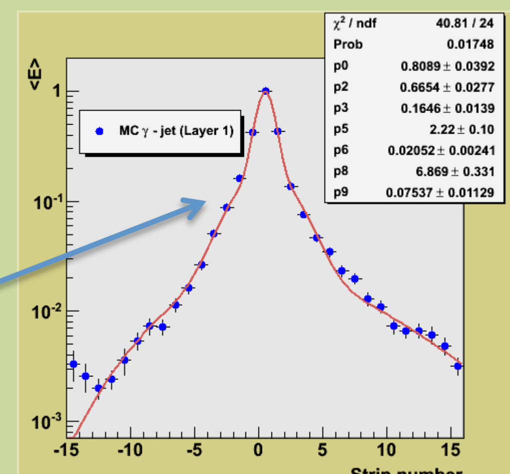
(a) $E_{pre1} = E_{pre2} = 0$(direct photon rich)

(b) $E_{pre1}=0$, $E_{pre2}>0$

(c) $0<E_{pre1}<4$MeV

(d) $4<E_{pre1}<10$ MeV
(background dominated)

Transverse shower profile fitted by the sum of three common-centroid gaussians with independent widths and scaling with some skewness added to the third gaussian



Seema Dhamija – DNP 2012

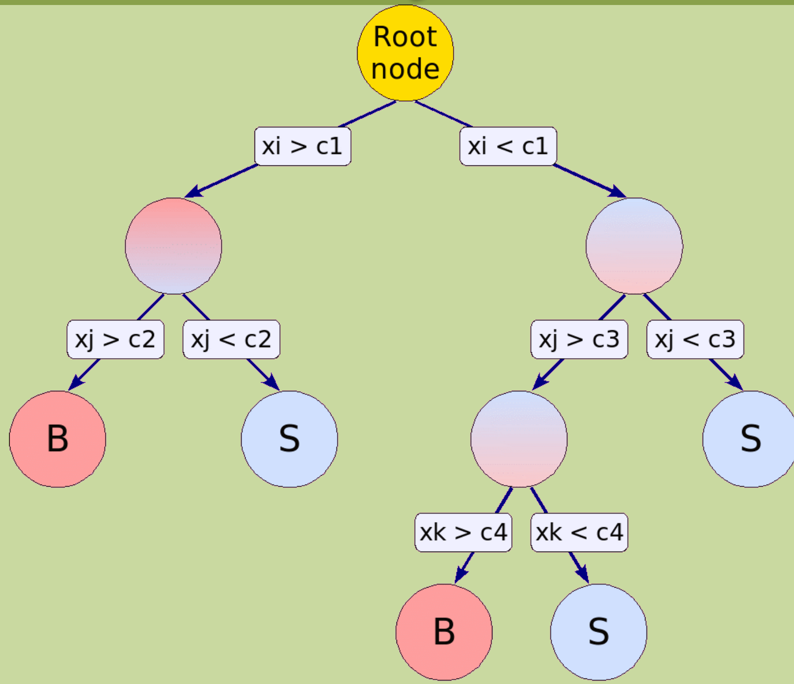
Y/π0 discrimination in Endcap

- ***Challenges:***
  - signal statistics, low Signal/Background ratio
  - To suppress more background & keep high Signal efficiency, rely on advanced techniques such as boosted decision trees

- ***Why not neural networks or linear discriminants?***
- ***ANN : Very good with non-linear correlations but black box, needs tuning***
- ***Fisher : Very fast and transparent but fails if PDFs have same mean, and if non-linear correlations***

- ***Boosted Decision Trees (BDT)***
  - Non-linear combination of input variables such as
    1. Energy deposited in different preshower, postshower layers
    2. EMC (Barrel, Endcap) towers fired around photon candidate
    3. Charged tracks around photon candidate (r=0.7)
    4. Energy fraction in the 3x3 tower and smaller clusters
    5. SMD shower shape response parameters
  - Great performance for large number of input variable
  - Powerful and stable by combining many decision trees to make a "majority vote"

Schematic View of a Decision Tree

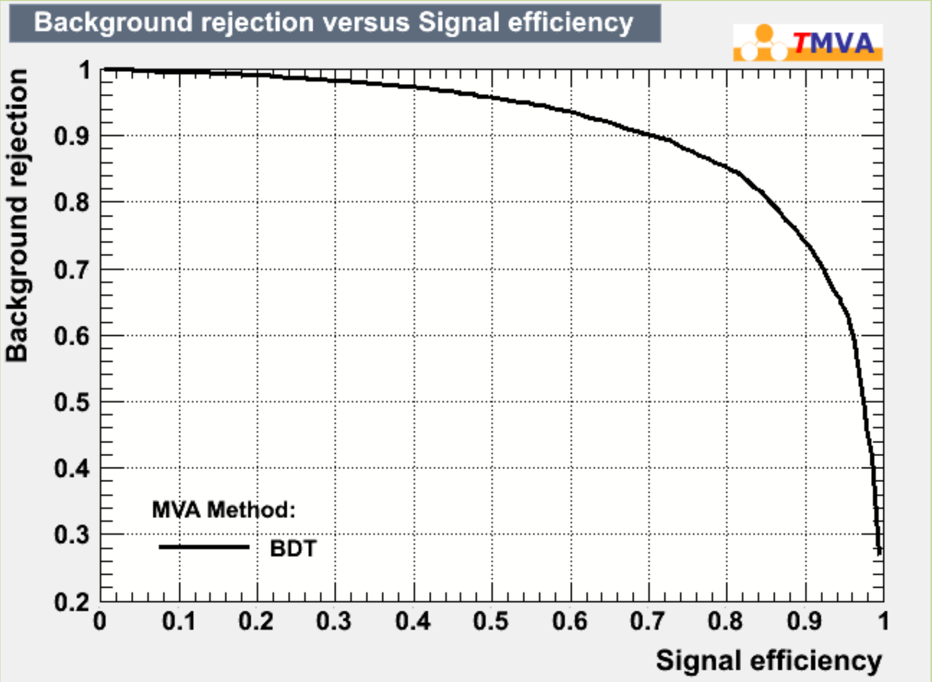


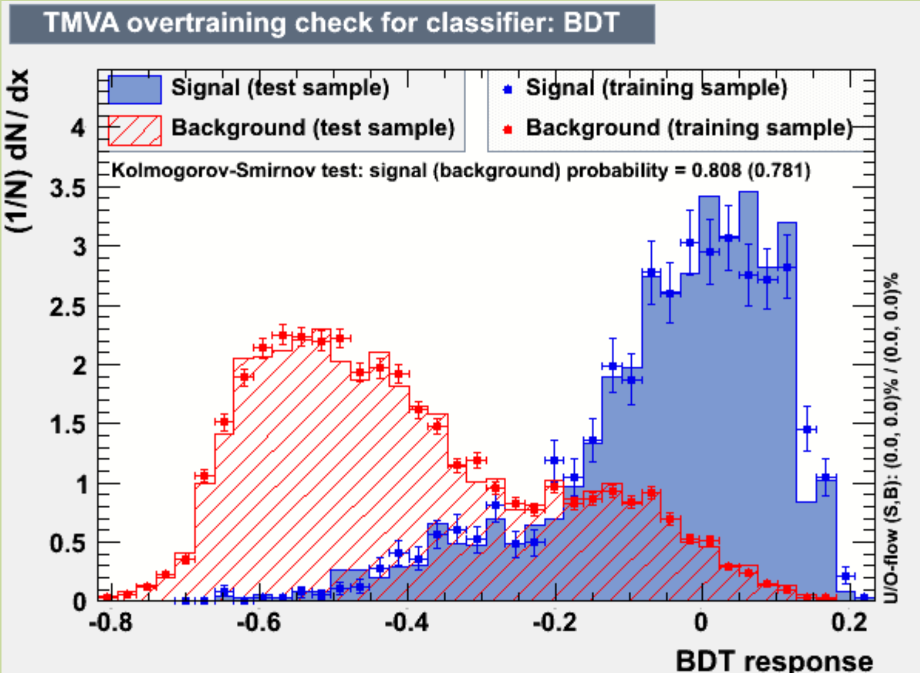
- Sequence of binary splits using the discriminating variables x_i
- Each split uses the variable that at this node gives the best separation between signal and background when being cut on
- Leaf nodes at the bottom labeled "S" and "B" on the majority of events in the respective nodes
- Boosting : Weights of misclassified events in current tree are increased, the next tree is built using the same events but with new weights

Boosted Decision Trees Output

- The MC samples are split into two halves, one for training, the other for test
- Training Events – selected randomly 1858 signal and 11857 background events
- 15 variables for training
- Testing Events – compare training and test samples (Kolmogorov-Simirnov test)

Good separation between signal and background





Seema Dhamija – DNP 2012

Conclusions

❖ Simulations reproduce the experimental conditions quite well

❖ Implementation of a BDT classifier, including (SMD) shower shape and other discriminating variables, provides us with a powerful tool for background rejection

What's next?

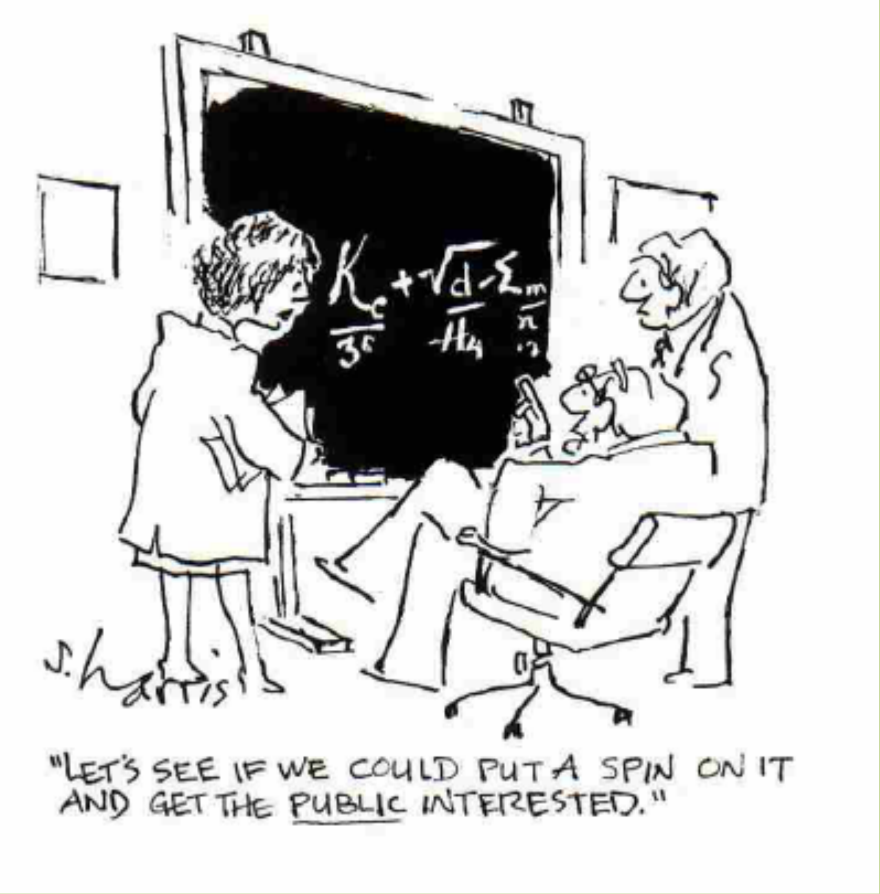
❖ Run BDT classifier on real data

❖ Refine purity/efficiency analysis with MC and data

❖ Extract signal/background vs. photon p_T, η etc.

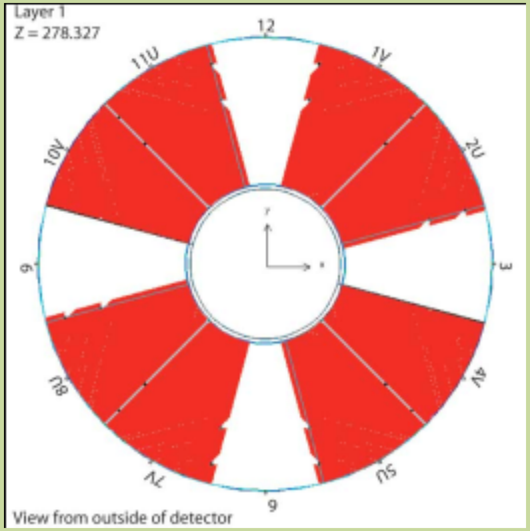
❖ Yields/photon-jet cross section

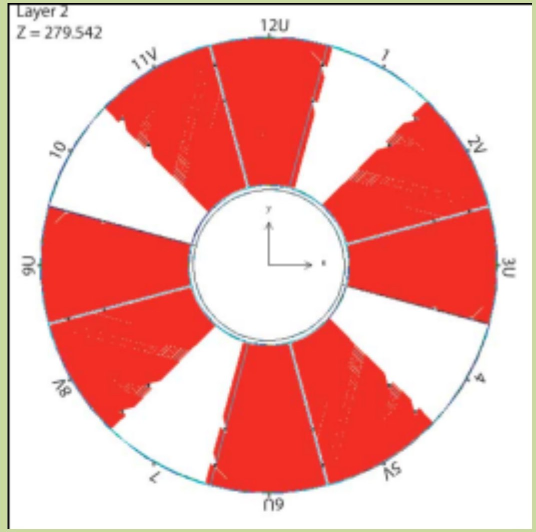
Thank You

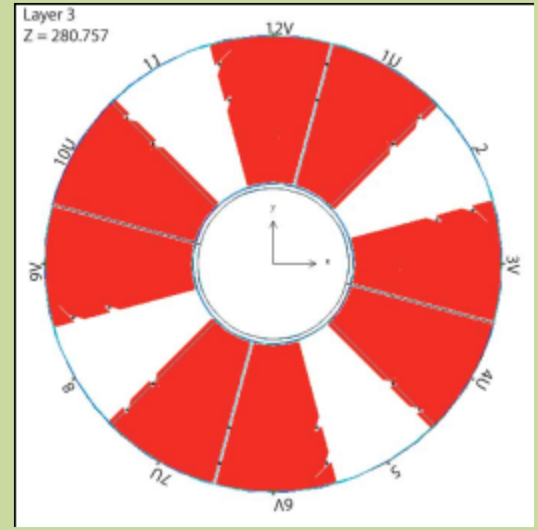

S. Harris
"LET'S SEE IF WE COULD PUT A SPIN ON IT AND GET THE PUBLIC INTERESTED."

BACKUP SLIDES

Asymmetric layout of the SMD sublayers







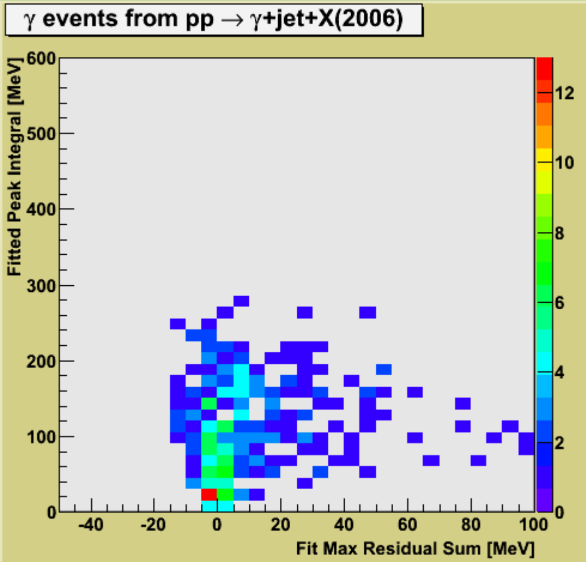
Sector-dependent structure of SMD in the EEMC

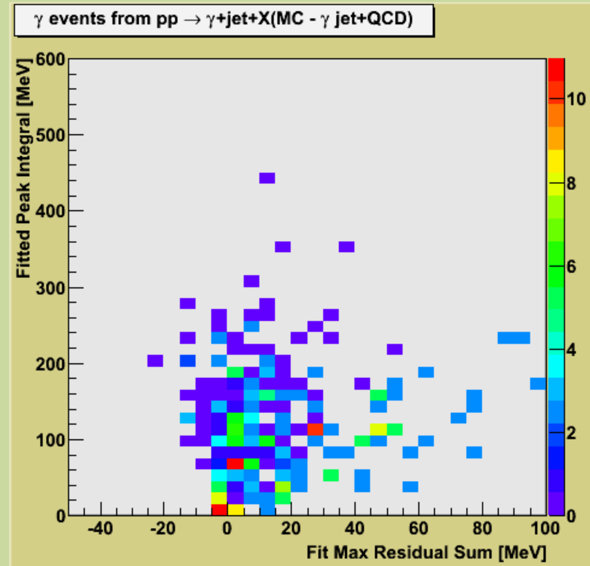
Sector	1	2	3	4	5	6	7	8	9	10	11	12
Layer1	V	U	Space	V	U	Space	V	U	Space	V	U	Space
Layer2	Space	V	U	Space	V	U	Space	V	U	Space	V	U
Layer3	U	Space	V	U	Space	V	U	Space	V	U	Space	V

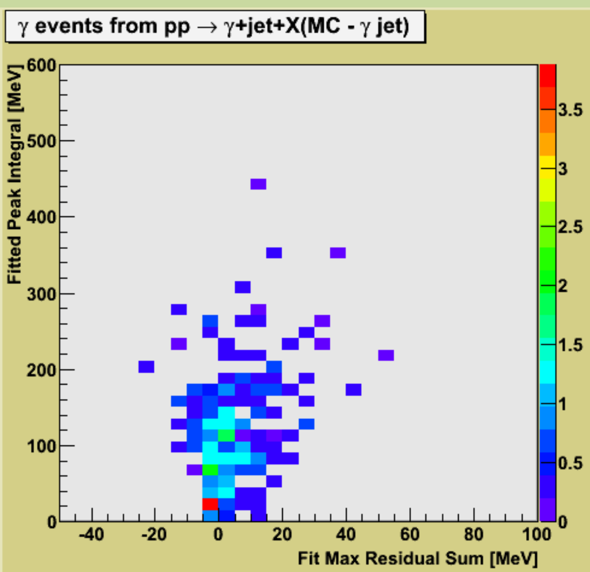
Layer 1 : u(2,5,8,11)+u(3,6,9,12)+v(1,4,7,10)
Layer 2 : v(2,5,8,11)+v(3,6,9,12)+u(1,4,7,10)

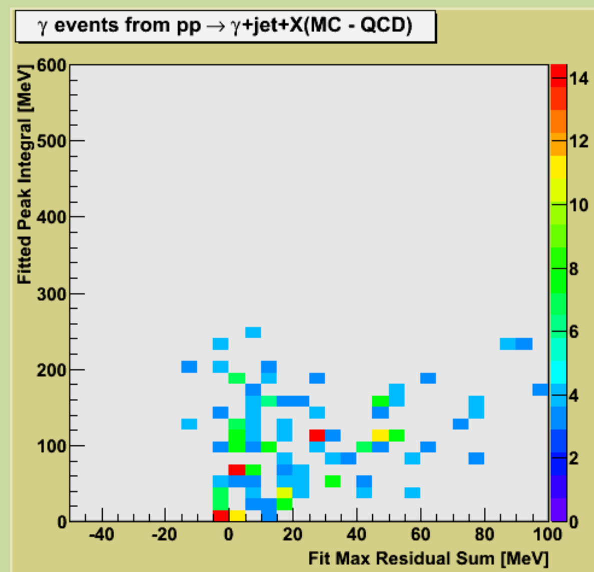
Y/π0 discrimination in Endcap SMD :

Maximum Sided Residual : first preshower bin









Decision Trees & Boosting Algorithms

- Decision Trees have been available about two decades, they are known to be powerful but unstable, i.e., a small change in the training sample can give a large change in the tree and the results.

  Ref: L. Breiman, J.H. Friedman, R.A. Olshen, C.J.Stone, "Classification and Regression Trees", Wadsworth, 1984.

- The boosting algorithm (AdaBoost) is a procedure that combines many "weak" classifiers to achieve a final powerful classifier.

  Ref: Y. Freund, R.E. Schapire, "Experiments with a new boosting algorithm", Proceedings of COLT, ACM Press, New York, 1996, pp. 209-217.

- Boosting algorithms can be applied to any classification method. Here, it is applied to decision trees, so called "Boosted Decision Trees", for the MiniBooNE particle identification.

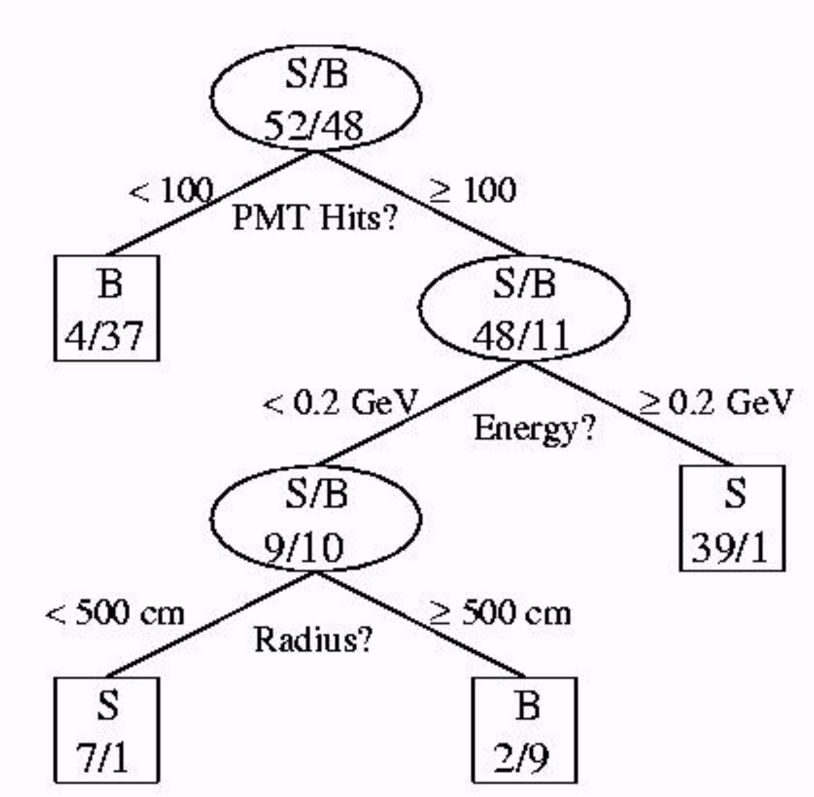
  * Hai-Jun Yang, Byron P. Roe, Ji Zhu, " Studies of boosted decision trees for MiniBooNE particle identification", physics/0508045, NIM A 555:370,2005

  * 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", NIM A 543:577,2005

  * Hai-Jun Yang, Byron P. Roe, Ji Zhu, "Studies of Stability and Robustness of Artificial Neural Networks and Boosted Decision Trees", NIM A574:342,2007

How to Build A Decision Tree?

1. Put all training events in root node, then try to select the splitting variable and splitting value which gives the best signal/background separation.
2. Training events are split into two parts, left and right, depending on the value of the splitting variable.
3. For each sub node, try to find the best variable and splitting point which gives the best separation.
4. If there are more than 1 sub node, pick one node with the best signal/background separation for next tree splitter.
5. Keep splitting until a given number of terminal nodes (leaves) are obtained, or until each leaf is pure signal/background, or has too few events to continue.



* If signal events are dominant in one leaf, then this leaf is signal leaf (+1); otherwise, backgroud leaf (score= -1).

Criterion for "Best" Tree Split

- Purity, *P,* is the fraction of the weight of a node (leaf) due to signal events.
- Gini Index: Note that Gini index is 0 for all signal or all background.

$$Gini = \left(\sum_{i=1}^{n} W_i\right) P(1-P)$$

- The criterion is to minimize

  Gini_left_node+ Gini_right_node.

Criterion for Next Node to Split

- Pick the node to maximize the change in Gini index. Criterion =

  $Gini_{parent\_node} - Gini_{right\_child\_node} - Gini_{left\_child\_node}$
- We can use Gini index contribution of tree split variables to sort the importance of input variables.
- We can also sort the importance of input variables based on how often they are used as tree splitters.

Signal and Background Leaves

- Assume an equal weight of signal and background training events.
- If event weight of signal is larger than ½ of the total weight of a leaf, it is a signal leaf; otherwise it is a background leaf.
- Signal events on a background leaf or background events on a signal leaf are misclassified events.

How to Boost Decision Trees ?

- For each tree iteration, same set of training events are used but the weights of misclassified events in previous iteration are increased (boosted). Events with higher weights have larger impact on Gini index values and Criterion values. The use of boosted weights for misclassified events makes them possible to be correctly classified in succeeding trees.
- Typically, one generates several hundred to thousand trees until the performance is optimal.
- The score of a testing event is assigned as follows: If it lands on a signal leaf, it is given a score of 1; otherwise -1. The sum of scores (weighted) from all trees is the final score of the event.

Weak → Powerful Classifier

- → The advantage of using boosted decision trees is that it combines many decision trees, "weak" classifiers, to make a powerful classifier. The performance of BDT is stable after few hundred tree iterations.

- → Boosted decision trees focus on the misclassified events which usually have high weights after hundreds of tree iterations. An individual tree has a very weak discriminating power.

Two Boosting Algorithms

- AdaBoost Algorithm:

1. Initialize the observation weights $w_i = 1/\text{n}$, i = 1, 2,..., n
2. For m = 1 to M:
   2.a Fit a classifier $T_m(x)$ to the training data using weights w_i
   2.b Compute
   $$err_m = \frac{\sum_{i=1}^{n} w_i I(y_i \neq T_m(x_i))}{\sum_{i=1}^{n} w_i}$$
   2.c Compute $\alpha_m = \beta \times log((1 - err_m)/err_m)$
   2.d Set $w_i \leftarrow w_i \times exp(\alpha_m I(y_i \neq T_m(x_i)))$, i=1, 2,...,n
   2.e Re-normalize $w_i = w_i / \sum_{i=1}^{n} w_i$
3. Output T(x) = $\sum_{m=1}^{M} \alpha_m T_m(x)$

I = 1, if a training event is misclassified; Otherwise, I = 0

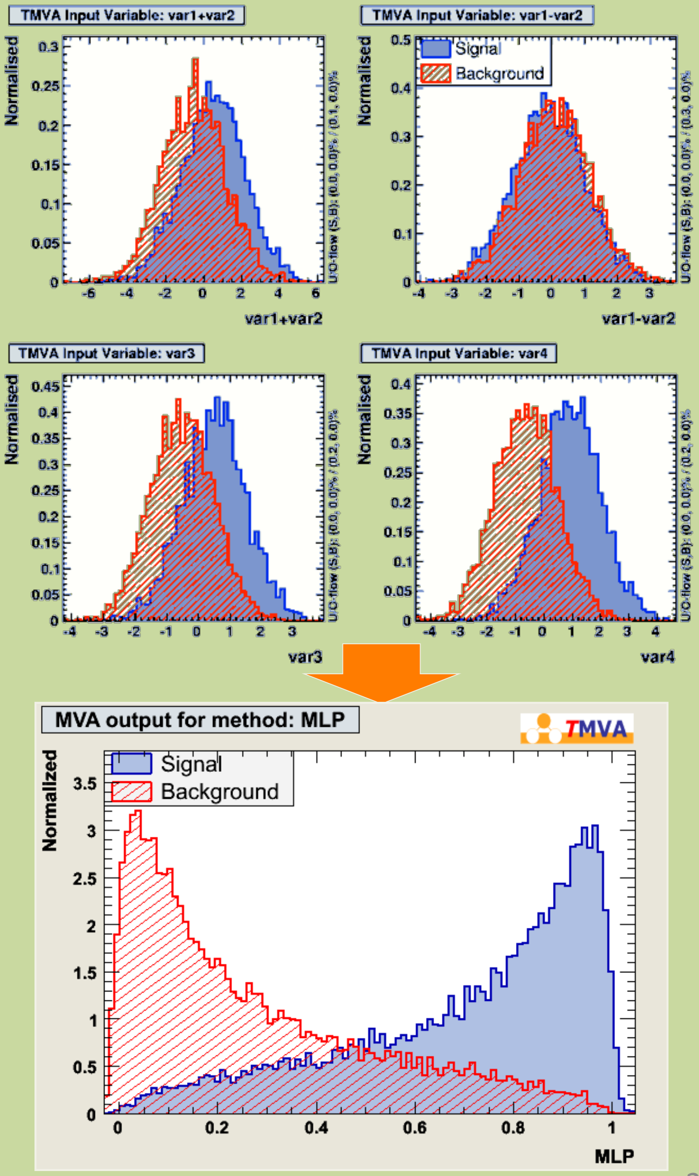
- $\epsilon-$boosting Algorithm:

1. Initialize the observation weights $w_i = 1/\text{n}$, i = 1, 2,..., n
2. For m = 1 to M:
   2.a Fit a classifier $T_m(x)$ to the training data using weights w_i
   2.b Set $w_i \leftarrow w_i \times exp(2\epsilon I(y_i \neq T_m(x_i)))$, i=1, 2,...,n
   2.c Re-normalize $w_i = w_i / \sum_{i=1}^{n} w_i$
3. Output T(x) = $\sum_{m=1}^{M} \epsilon T_m(x)$

What is a multi-variate analysis?

- “Combine“ all input variables into one output variable

- Supervised learning means learning by example: the program extracts patterns from training data



Seema Dhamija – DNP 2012

Typical multi-variate analysis steps

- Choice of input variables
- Define preselection

Physics input Is crucial

- Choice of MVA method
- Training the MVA method using samples with known signal/ background
- Choice of working point