Introduction to Multivariate Classification Problems

Byron P. Roe
University of Michigan
Ann Arbor, MI 48105
June 16, 2006

Use MiniBooNE as Example

- This experiment has many of the problems to be discussed in C (and some in A).
- MiniBooNE is looking for a small class of events $\nu_\mu \rightarrow \nu_e$
- Background is about 1000 times signal.
- Some 300 candidates for feature variables (FV). FV from reconstructed events.
- If new class exists, determine two parameters; if not set limits as functions of these parameters.

Classification problem

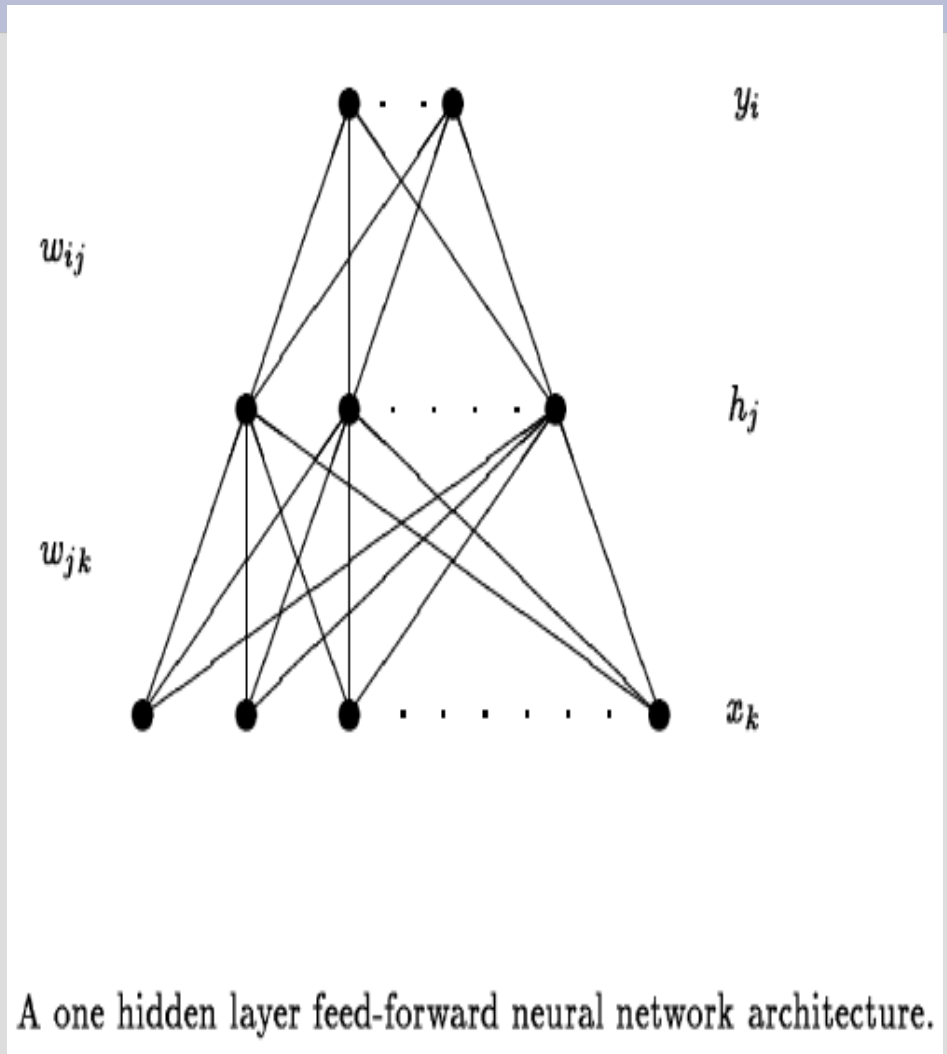
- Divide data into several categories given a number of feature variables with each event.
- Often used in particle physics with two categories—signal and background.

Older Methods

- Artificial Neural Net (ANN)
- Decision Trees

Neural Network Structure

- Combine the features in a non-linear way to a "hidden layer" and then to a "final layer"
- Use a training set to find the best w_{ik} to distinguish signal and background

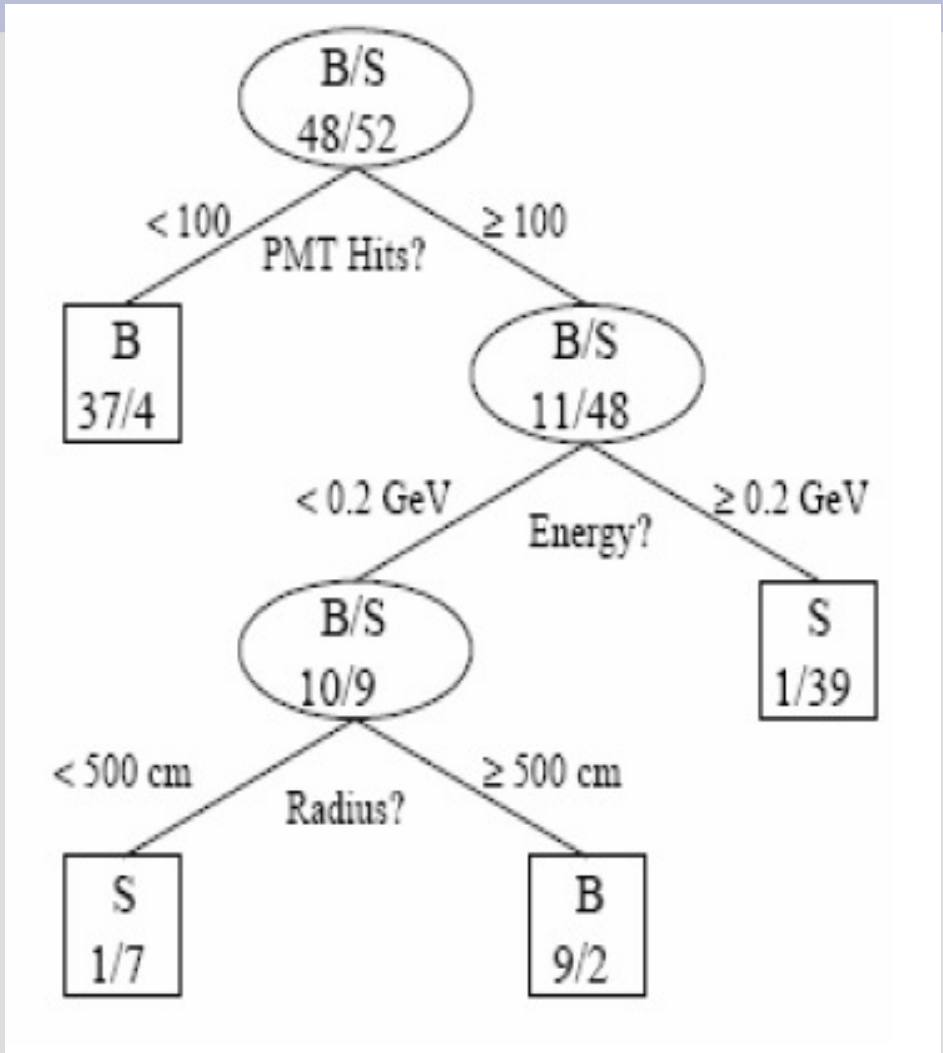


A one hidden layer feed-forward neural network architecture.

Decision Tree

Background/Signal

- Go through all feature variables and find best variable and value to split events.
- For each of the two subsets repeat the process
- Proceeding in this way a tree is built.
- Ending nodes are called leaves.



Select Signal and Background Leaves

- Assume an equal weight of signal and background training events.
- If more than ½ of the weight of a leaf corresponds to signal, 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.

One Criterion for "Best" Split

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

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

- The criterion is to minimize gini_left + gini_right of the two children from a parent node

Criterion for Next Branch to Split

- Pick the branch to maximize the change in gini.

$$\text{Criterion} = \text{gini}_{\text{parent}} - \text{gini}_{\text{right-child}} - \text{gini}_{\text{left-child}}$$

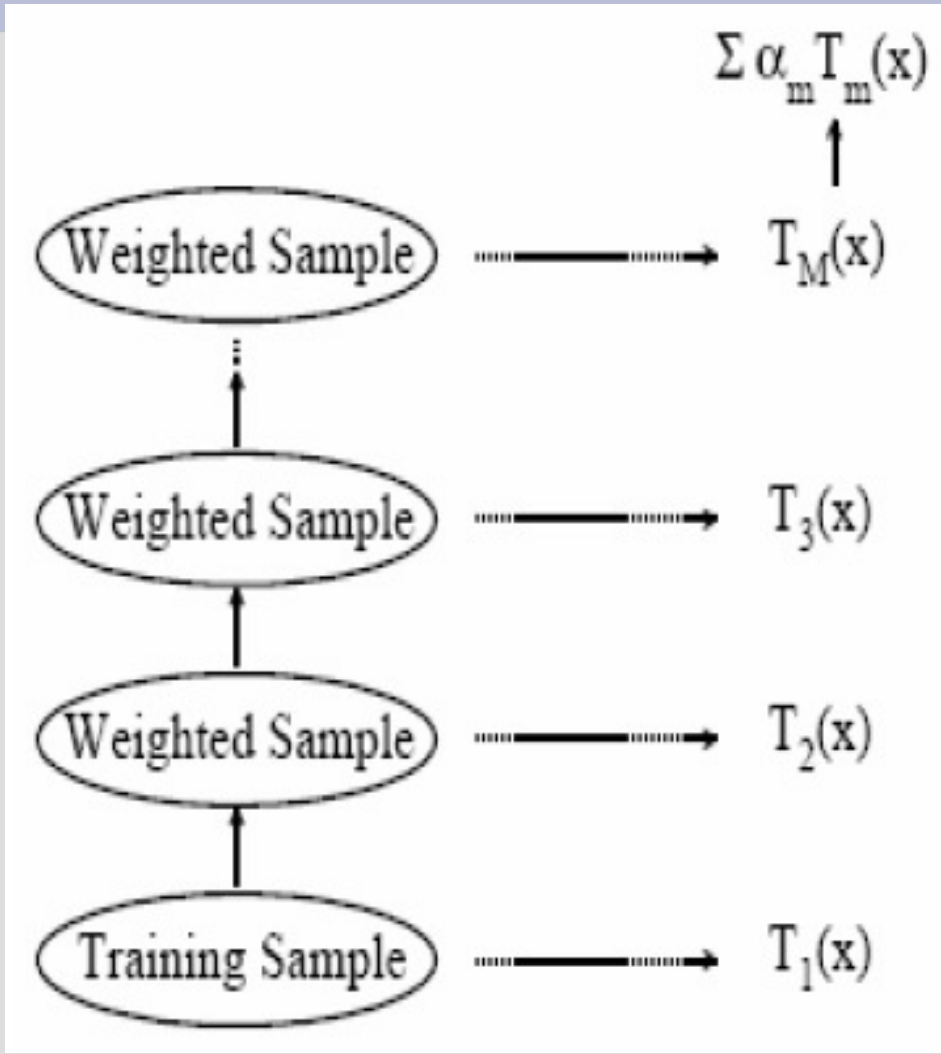
Problems with Older Methods

- ANN is not stable in many available versions
    i. If put variable in twice, answer often changes
    ii. If multiply one variable by two, answer often changes
    iii. If change order of variables, answer often changes
- Decision trees are also unstable.
- GO ON TO NEWER METHODS

Newer Methods

Boosting the Decision Tree

- Give the training events misclassified under this procedure a higher weight.
- Continuing build perhaps 1000 trees and do a weighted average of the results (1 if signal leaf, -1 if background leaf).



Many variants

- Change Gini criterion
- Several weight updating schemes
- Change scoring
- Don't change weights, but many trees with subsets of events (bagging, random forests)
- For neural nets Bayesian neural nets
- The basic point is to average over many trees in some way.
- Boosting can, in principle, be applied to many classification schemes—ANN..., but most use in physics from trees

Good Reference

- T. Hastie, R. Tibshirani, and J. Friedman, “The Elements of Statistical Learning. Data Mining, Inference and Prediction. “ Springer (2001).

Warning: Boost Use Different than in Many Statistics Articles

- 45 leaves (8 or less in many publications)
- 1000 trees
- Slightly modified scoring
- Use several sets of boosting trees. Make a cut with first set and then retrain on remainder. (Cascade boosting) OR train with several different backgrounds and then use boosting scores from each as additional feature variables for final training.

Rule Fit

- This is a variant of boosted decision trees of J. Friedman. Here each node of each tree can be thought of as a rule to select events. For 1000 trees with 45 leaves (89 nodes) apiece, this is 89,000 rules.
- The score is taken as a linear sum of the truth of the rules. An algorithm is used to optimize the weights of each rule with a regulator term to control the variations.

Support Vector Machines

- In the multidimensional space of the feature variables, find the borders of signal and background events. Use only the border region.
- Similar in a sense to boosting, which also gives the most weight to the hard to classify events, which are the border events.

Comparisons

- It is hard to generalize here. It is likely that the best method depends on the problem.
- Comparisons are not easy. The comparisons must be made with each method tuned. See for instance the note of J. Conrad and F. Tegenfeldt hep-ph/0605106 and the subsequent e-mails between Conrad and Haijun Yang.

Comparisons II

- In the comparisons we have made for mini-BooNE and some data from Babar, boosted decision trees worked as well as any method tried.
- B.P. Roe, H.J. Yang, J. Zhu, I. Stancu and G. McGregor, Nucl. Inst. and Meth. A543 (2005) 577
- H.J. Yang, B.P. Roe and J. Zhu, Nucl. Inst. and Meth. A555 (2005) 370-385

Can Statisticians Help Here?

- Are there different approaches to the data?
- Are there some useful graphical methods?
- There is a reluctance among some physicists to use modern classification methods because they are non-intuitive and because physicists worry about accurately modeling data in many dimensions. Are there suggestions from statisticians on these issues?

Number of Feature Variables

- In miniBooNE we would like to reduce from 300 to perhaps 150 feature variables
- a. Check if data distributions agree with Monte Carlo for individual variables and robustness vs small systematic changes in model
- b. Make short runs and look at:
    - i. Feature variables used most often OR
    - ii. Feature variables giving biggest change in Gini criterion OR
    - iii. Feature variables used first

Number of Feature Variables II

- To first approximation, equal results with each method, but each has problems. (Example: two variables looking at same thing. Boost may randomly pick one or the other, reducing use by factor of two.)
- Do statisticians have any suggestions concerning selection of feature variables?

Goodness of Fit

- First cut on boosting score to reduce sample size by a factor of more than hundred.
- Even in this cut sample, 2/3 or more are background events.
- For this cut sample: Take the boosting score as one variable and event energy as a second, do chi-square or log likelihood fit for best values of the two parameters of interest or, for upper limits of the size of the rare process as a function of the two parameters.

Systematic Errors

- Not easy to relate an assumed error in a parameter (e.g. Fraction of Cherenkov light) to the effect on the reconstructed event.
- Use Monte Carlo
- Unisim—One run for each systematic varied by one standard deviation. Compare c.v.
- Multisim—A number of MC runs, in each of which all systematic parameters are varied randomly. (See B. Roe technical note)
- Do statisticians have any suggestions here?

Chi-Square

- Use of data to further estimate systematic errors. (D. Stump et al., Phys. Rev. D65, 014012.) Ignore Bayes vs frequentist.
- Take the chi-square with only statistical errors and add a term for each systematic using the multidimensional correlated normal distribution assumed for the systematics
- N systematic parameters added, but, effectively N bins added so number df same.
- Runs into problems if more syst. than bins.

Log Likelihood Fits

- Effectively means using finer bins than can with chi-square. -2lnL approx chi-square fails past 90% CL in one example of our binning.
- Use Monte Carlo. If the two output parameters were really at the assumed values, what is the likelihood of lnL(best) – lnL(real val.) being at least as large as observed. Hard to get to the 4σ equivalent normal distribution level.
- Can statisticians suggest a better way?

Finally

- Physicists and statisticians are now starting to work together to the benefit of both groups.
- We can use all the help we can get!!

Backup

Feedforward Neural Network--I

$$F_i(\vec{x}) = g\left[\frac{1}{T}\sum_j \omega_{ij} g\left(\frac{1}{T}\sum_k \omega_{jk}x_k + \theta_j\right) + \theta_i\right].$$

This corresponds to a network where the x_k, the input layer, are combined with weights ω_{jk} and offsets θ_j to give a hidden layer $h_j = g\left(\frac{1}{T}\sum_k \omega_{jk}x_k + \theta_j\right)$, and the h_j then combined in a similar manner to give an output layer y_i. Sometimes there are several hidden layers, defined in the obvious way by iterating the procedure given in the previous equation. The hidden layer enables non-linear modeling of the input data.

Feedforward Neural Network--II

$$F_i(\vec{x}) = g\left[\frac{1}{T}\sum_j \omega_{ij} g\left(\frac{1}{T}\sum_k \omega_{jk} x_k + \theta_j\right) + \theta_i\right].$$

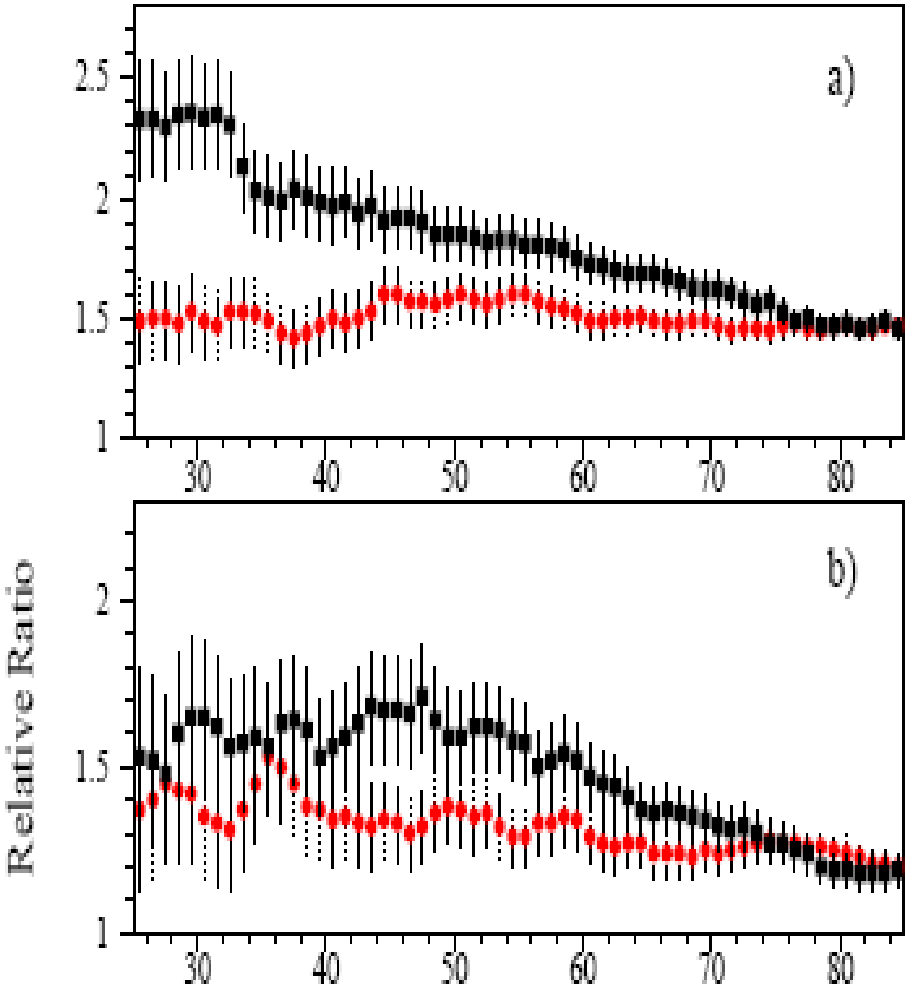
T is a system parameter which scales the size of F_i.

The weights ω_{ij} and ω_{jk} are the parameters to be fitted to the data distributions and $g(x)$ is the non-linear neuron activation function, typically of the "sigmoid" form,

$$g(x) = \frac{1}{2}[1 + \tanh(x)] = (1 + e^{-2x})^{-1}.$$

Comparison of Boosting and ANN

- Relative ratio here is ANN bkrd kept/Boosting bkrd kept. Greater than one implies boosting wins!
- A. All types of background events. Red is 21 and black is 52 training var.
- B. Bkrd is pi0 events. Red is 22 and black is 52 training variables

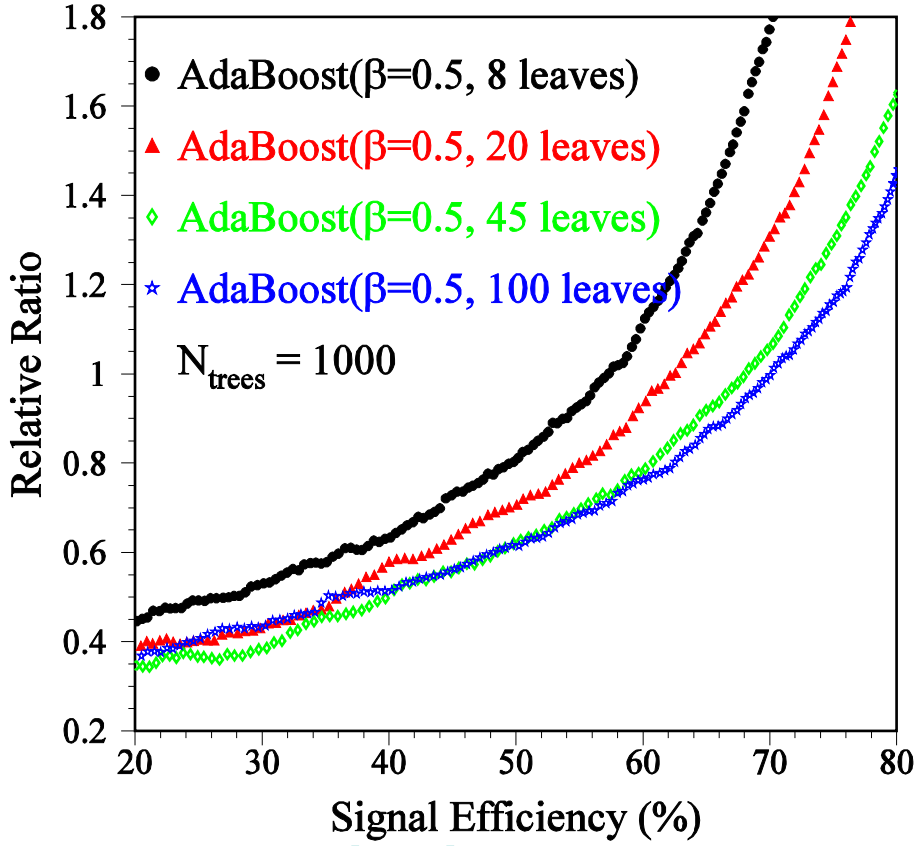


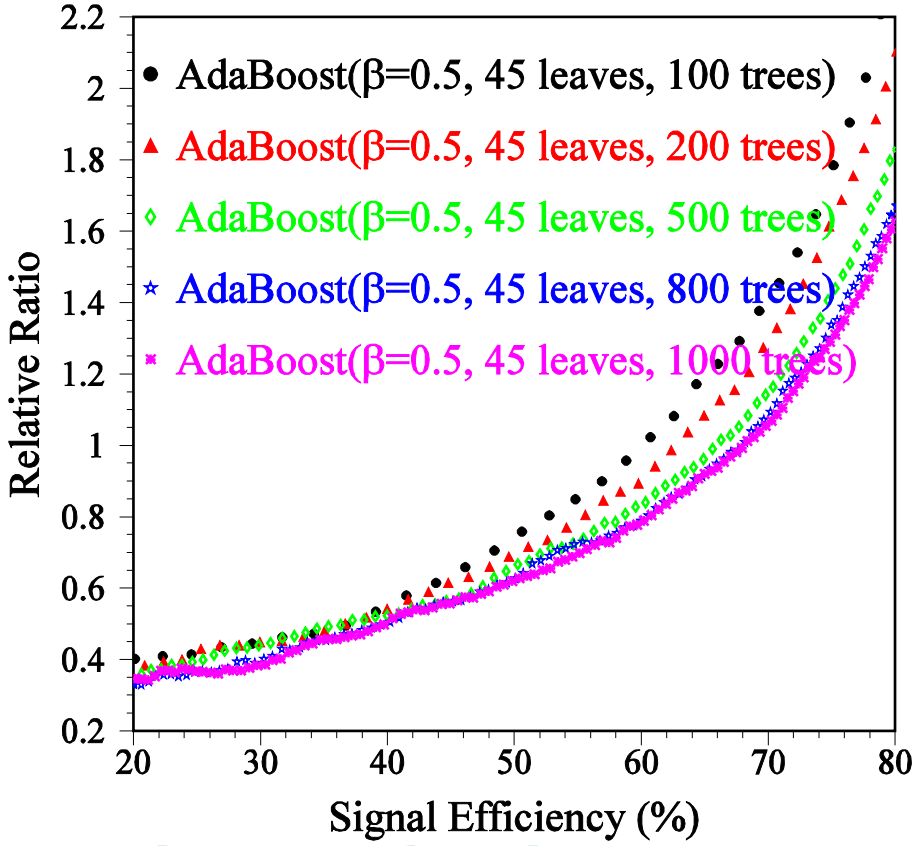
Percent nue CCQE kept

Boosting	Parameters	Relative ratios for given signal efficiencies					
Algorithms	β,ϵ (N_{leaves}, N_{trees})	30%	40%	50%	60%	70%	80%
AdaBoost	0.5 (8,1000)	0.53	0.63	0.81	1.11	1.78	3.21
AdaBoost	0.5 (8,5000)	0.50	0.60	0.74	0.98	1.40	2.52
ϵ-Boost	0.01 (8,1000)	0.49	0.55	0.71	0.93	1.40	2.44
ϵ-Boost	0.01 (8,5000)	0.51	0.55	0.66	0.86	1.17	1.82
ϵ-LogitBoost	0.01 (8,1000)	0.49	0.59	0.79	1.07	1.58	2.95
ϵ-LogitBoost	0.01 (8,5000)	0.52	0.57	0.68	0.89	1.22	2.01
ϵ-HingeBoost	0.01 (8,1000)	0.58	0.66	0.83	1.09	1.68	2.88
ϵ-HingeBoost	0.01 (8,5000)	0.61	0.69	0.82	1.05	1.48	2.49
ϵ-LogitBoost	0.01 (45,1000)	0.39	0.50	0.61	0.82	1.11	1.84
ϵ-HingeBoost	0.01 (30,1000)	0.77	0.80	0.86	0.96	1.20	1.80
LogitBoost	1.0 (45,130)	0.41	0.55	0.73	0.98	1.43	2.40
LogitBoost	0.1 (45,150)	0.44	0.52	0.62	0.82	1.23	2.00
Real AdaBoost	(45,1000)	0.47	0.57	0.69	0.82	1.10	1.60
Gentle AdaBoost	(45,1000)	0.47	0.54	0.67	0.83	1.05	1.56
Random Forests(RF)	(400,1000)	0.49	0.63	0.85	1.29	1.92	3.50
AdaBoosted RF	0.5 (100,1000)	0.48	0.56	0.66	0.81	1.04	1.58
AdaBoost	0.5 (45,1000)	0.38	0.50	0.62	0.78	1.06	1.63

Boosting	Parameters	Relative ratios for given signal efficiencies					
Algorithms	β,ϵ (N_{leaves}, N_{trees})	30%	40%	50%	60%	70%	80%
AdaBoost	0.3 (45,1000)	0.39	0.49	0.63	0.80	1.12	1.73
AdaBoost	0.5 (45,1000)	0.38	0.50	0.62	0.78	1.06	1.63
AdaBoost	0.8 (45,1000)	0.45	0.54	0.62	0.82	1.07	1.60
AdaBoost	1.0 (45,1000)	0.48	0.55	0.67	0.81	1.07	1.60
AdaBoost	0.5 (8,1000)	0.53	0.63	0.81	1.11	1.78	3.21
AdaBoost	0.5 (20,1000)	0.43	0.58	0.71	0.93	1.31	2.20
AdaBoost	0.5 (45,1000)	0.38	0.50	0.62	0.78	1.06	1.63
AdaBoost	0.5 (100,1000)	0.43	0.51	0.61	0.76	1.00	1.45
ϵ-Boost	0.005 (45,1000)	0.38	0.47	0.62	0.84	1.26	2.23
ϵ-Boost	0.01 (45,1000)	0.41	0.50	0.60	0.80	1.14	1.87
ϵ-Boost	0.02 (45,1000)	0.40	0.48	0.62	0.77	1.08	1.71
ϵ-Boost	0.03 (45,1000)	0.38	0.48	0.58	0.75	1.03	1.62
ϵ-Boost	0.04 (45,1000)	0.40	0.50	0.60	0.75	1.02	1.57
ϵ-Boost	0.05 (45,1000)	0.40	0.47	0.60	0.79	1.07	1.61
AdaBoost (b=0.5)	0.5 (45,1000)	0.39	0.47	0.60	0.76	1.06	1.58
ϵ-Boost (b=0.5)	0.01 (45,1000)	0.36	0.46	0.62	0.83	1.23	2.00
ϵ-Boost (b=0.5)	0.03 (45,1000)	0.38	0.45	0.58	0.76	1.06	1.65
ϵ-Boost (b=0.5)	0.05 (45,1000)	0.37	0.44	0.58	0.74	1.03	1.58

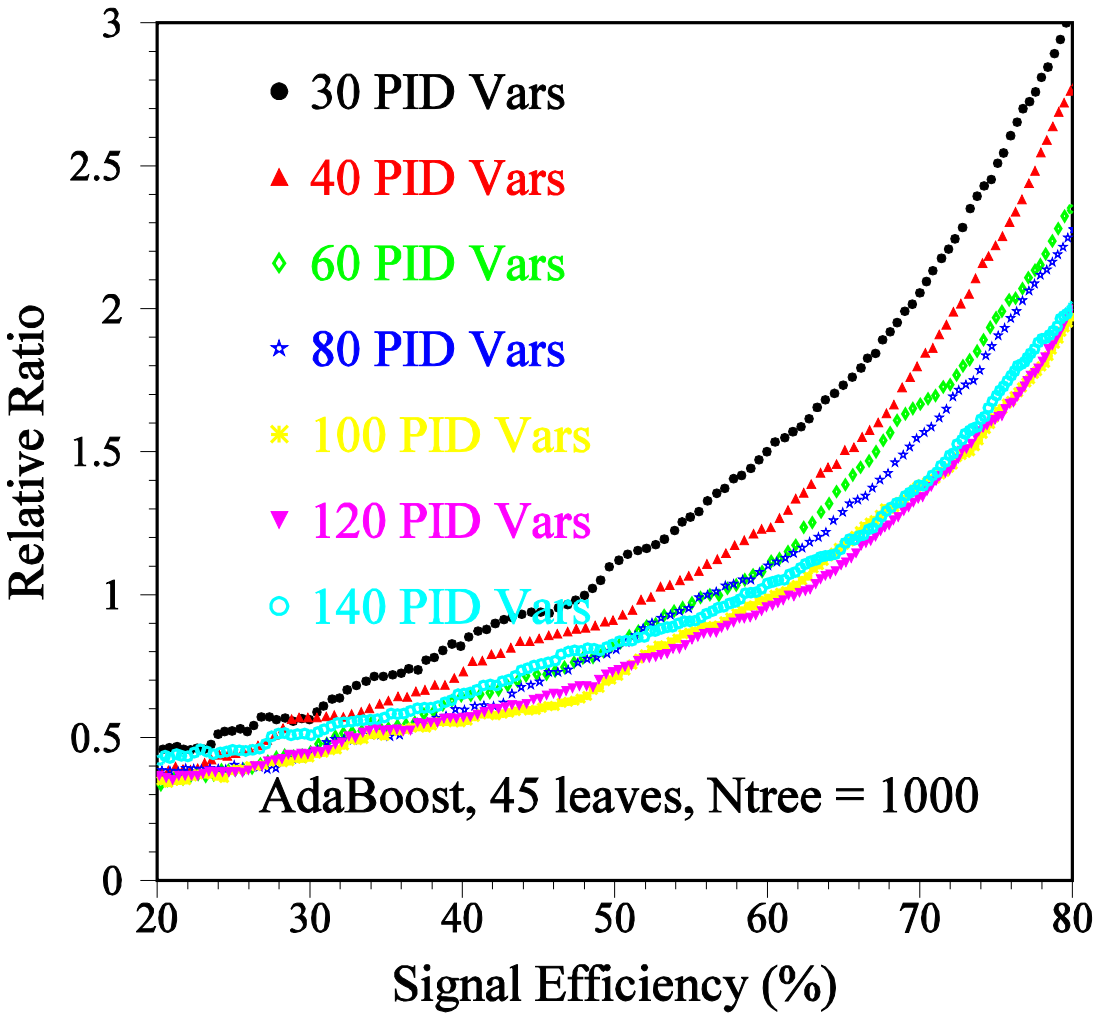
Effects of Number of Leaves and Number of Trees





Smaller is better! R = c X frac. sig/frac. bkrd.

Effect of Number of PID Variables



AdaBoost Optimization

$f_i(x)$ =classifier, with values $= +a_i$ or $-a_i$, with a_i a positive constant

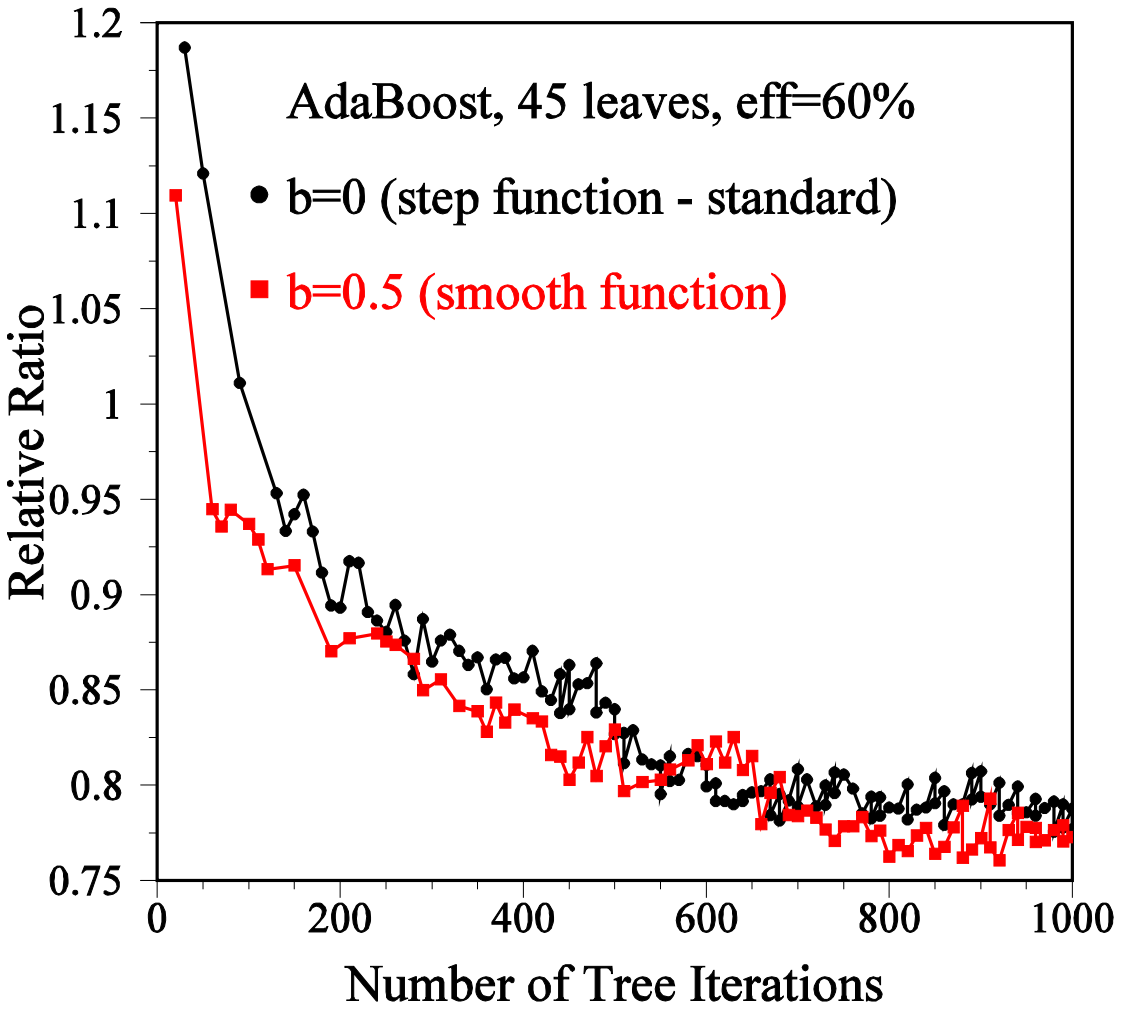
$F(x) = \sum_{i=1}^{N} f_i(x)$. (sum over trees.)

Can show that AdaBoost minimizes the expectation value, $E(e^{-yF(x)})$ by a series of Newton-like updates. Furthermore, the minimum value of $E(e^{-yF(x)})$ is $F(x) = \frac{1}{2}\ln\left(\frac{P(y=1|x)}{P(y=-1|x)}\right)$., which is $\frac{1}{2}$ the log-odds of the probabilty that $Y = 1$, given x. This minimization is closely related to maximizing the negative binomial log likelihood (cross-entropy). They can both be shown to have the same minimizer. Further, with y^* 1 or 0, and $p(x)$ = probability that $y^* = 1$ given x, then with $F(x) = \frac{1}{2}\ln\frac{p(x)}{1-p(x)}$ it can be shown that $e^{-yF(x)} = \frac{|y^*-p(x)|}{\sqrt{p(x)(1-p(x))}} = \chi$-statistic.

Can Convergence Speed be Improved?

- Removing correlations between variables helps.
- Random Forest (using random fraction[1/2] of training events per tree with replacement and random fraction of PID variables per node (all PID var. used for test here) WHEN combined with boosting.
- Softening the step function scoring: y=(2*purity-1); score = sign(y)*sqrt(|y|).

Performance of AdaBoost with Step Function and Smooth Function



AdaBoost Optimization

$f_i(x)$ =classifier, with values $= +a_i$ or $-a_i$, with a_i a positive constant

$F(x) = \sum_{i=1}^{N} f_i(x)$. (sum over trees.)

Can show that AdaBoost minimizes the expectation value, $E(e^{-yF(x)})$ by a series of Newton-like updates. Furthermore, the minimum value of $E(e^{-yF(x)})$ is $F(x) = \frac{1}{2}\ln\left(\frac{P(y=1|x)}{P(y=-1|x)}\right)$., which is $\frac{1}{2}$ the log-odds of the probabilty that $Y = 1$, given x. This minimization is closely related to maximizing the negative binomial log likelihood (cross-entropy). They can both be shown to have the same minimizer. Further, with y^* 1 or 0, and $p(x)$ = probability that $y^* = 1$ given x, then with $F(x) = \frac{1}{2}\ln\frac{p(x)}{1-p(x)}$ it can be shown that $e^{-yF(x)} = \frac{|y^*-p(x)|}{\sqrt{p(x)(1-p(x))}} = \chi$-statistic.

The MiniBooNE Collaboration

Y.Liu, I.Stancu
University of Alabama

S.Koutsoliotas
Bucknell University

R.A.Johnson, J.L.Raaf
University of Cincinnati

T.Hart, R.H.Nelson, M.Wilking, E.D.Zimmerman
University of Colorado

, A.A.Aguilar-Arevalo, L.Bugel, L.Coney, J.M.Conrad, J.M.Link, K.B.M.Mahn, J.Monroe, D.Schmitz, M.H.Shaevitz, M.Sorel, G.P.Zeller
Columbia University

D.Smith
Embry Riddle Aeronautical University

L.Bartoszek, C.Bhat, S.J.Brice, B.C.Brown, D.A.Finley, R.Ford, F.G.Garcia, P.Kasper, T.Kobilarcik, I.Kourbanis, A.Malensek, W.Marsh, P.Martin, F.Mills, C.Moore, E.Prebys, A.D.Russell, P.Spentzouris, R.J.Stefanski, T.Williams
Fermi National Accelerator Laboratory

D.C.Cox, T.Katori, H.Meyer, C.C.Polly, R.Tayloe
Indiana University

G.T.Garvey, A.Green, C.Green, W.C.Louis, G.McGregor, S.McKenney, G.B.Mills, H.Ray, V.Sandberg, B.Sapp, R.Schirato, R.Van de Water, N.L.Walbridge, D.H.White
Los Alamos National Laboratory

R.Imlay, W.Metcalf, S.Ouedraogo, M.O.Wascko
Louisiana State University

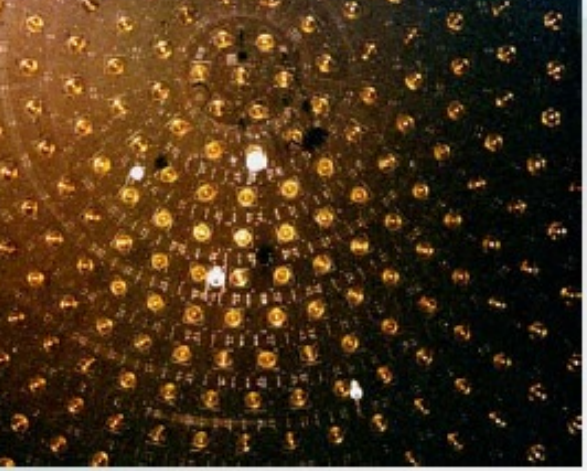
J.Cao, Y.Liu, B.P.Roe, H.J.Yang
University of Michigan

A.O.Bazarko, P.D.Meyers, R.B.Patterson, F.C.Shoemaker, H.A.Tanaka
Princeton University

P.Nienaber
Saint Mary's University of Minnesota

E.Hawker
Western Illinois University

A.Curioni, B.T.Fleming
Yale University

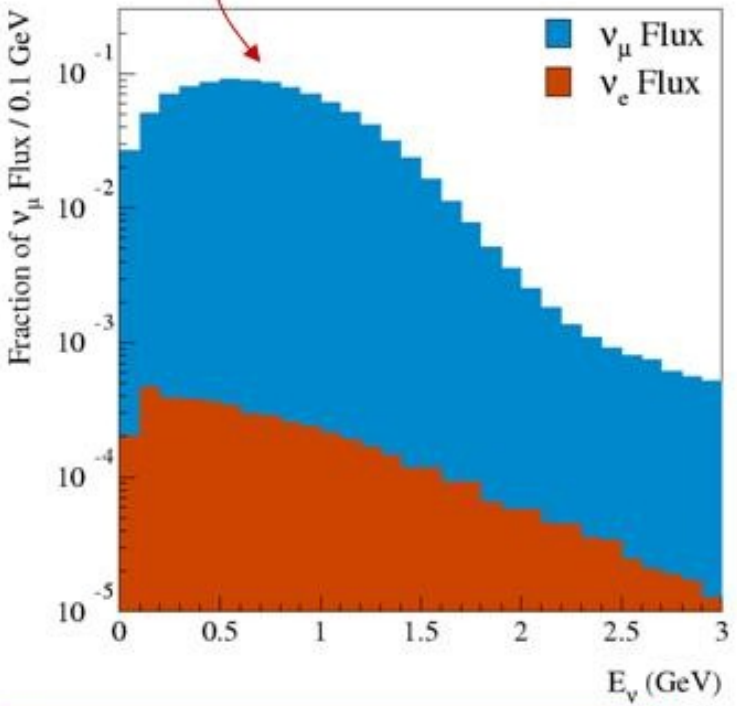


- 12 meter diameter sphere
- Filled with 950,000 liters (900 tons) of very pure mineral oil
- Light tight inner region with 1280 photomultiplier tubes
- Outer veto region wit 241 PMTs.
- Oscillation Search Method:

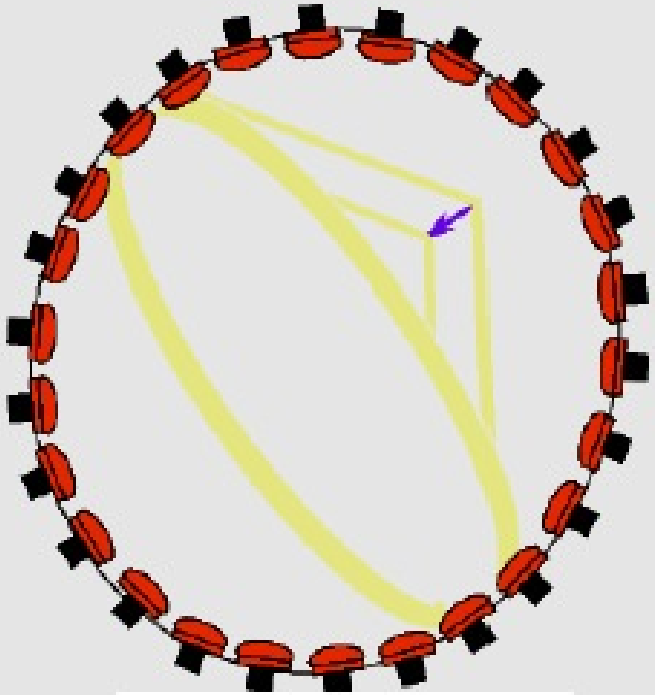
  Look for ν_e events in a pure ν_μ beam

Neutrino Beam

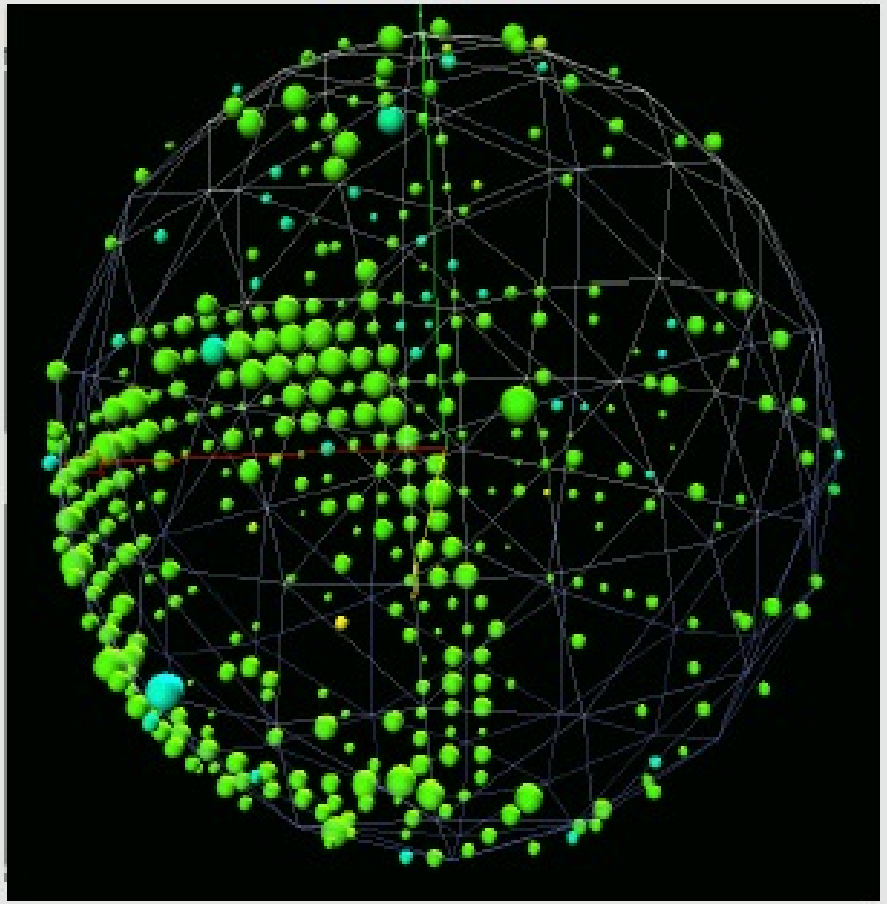
- Proton flux ~ 6E16 p/hr (goal 9E16 p/hr)
  - ~ 1 detected neutrino/minute
  - L/E ~ 1

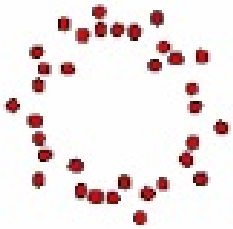


40’ D tank, mineral oil, surrounded by about 1280 photomultipliers. Both Cher. and scintillation light. Geometrical shape and timing distinguishes events



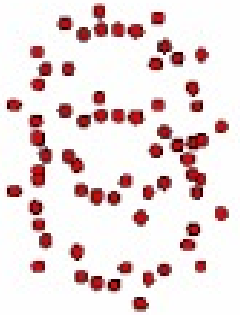
Stopping muon event





e^-

μ



π^0