Multivariate Machine Learning Methods: New Developments and Applications in HEP

Pushpa Bhat
Fermilab

August 6, 2015

No turning back!

- Over the past 25 years, Multivariate analysis (MVA) methods have gained gradual acceptance in HEP.
- In fact, they are now "state of the art"
- Some of the most important physics results in HEP, in the past two decades, have come from the use MVA methods.
- In 1990's, I'd have on my title slide

  "We are riding the wave of the future"
- That future is here, and MVA methods are here to stay!

Important Physics Results

- **From top quark to the Higgs**

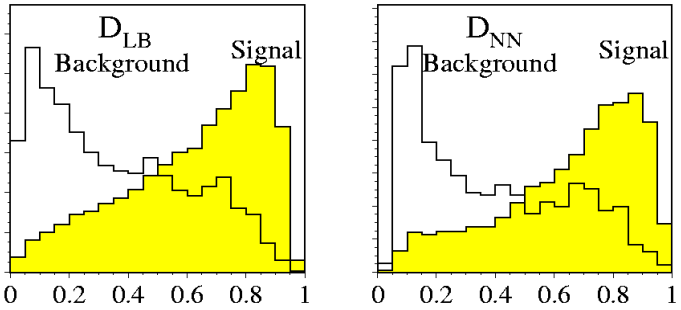
  … and many smart applications in object ID, energy corrections, as well

- Top-antitop event selection optimization – 1990-95 (D0)
- Top quark mass measurement -- 1996-97
- Top cross section measurements in all channels (1995 -)
- Top observation in the all-jets channel (D0) (1999)
- New particle/physics searches (1997 -)
- Observation of single top quark production (2009)
- Evidence for Higgs→ bb at the Tevatron (2012)
- **Higgs Discovery at the LHC in 2012**

MVA for Top quark in the mid-90's

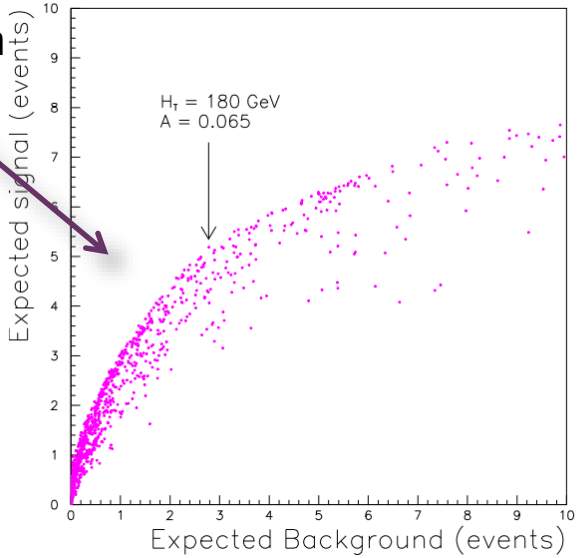
DØ Lepton+jets

The Discriminants

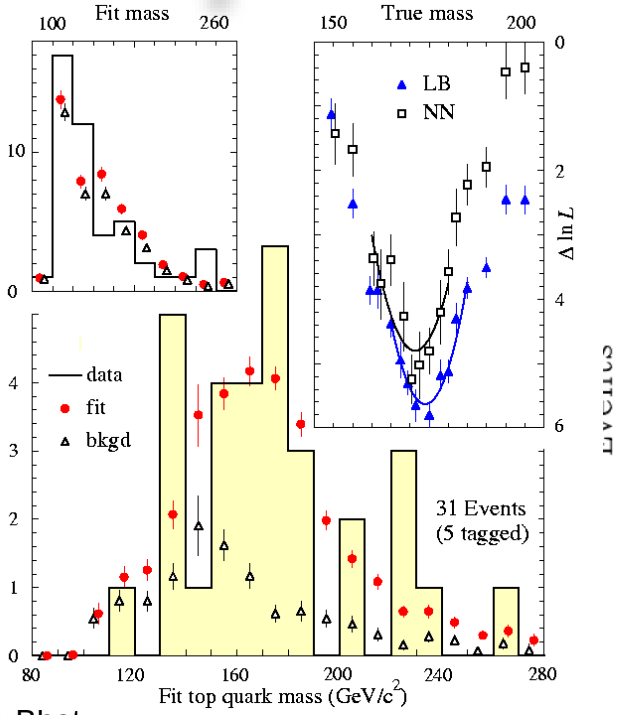


LB: Low-bias maximum likelihood
NN: Neural Networks

e+jets cut optimization for cross section measurement



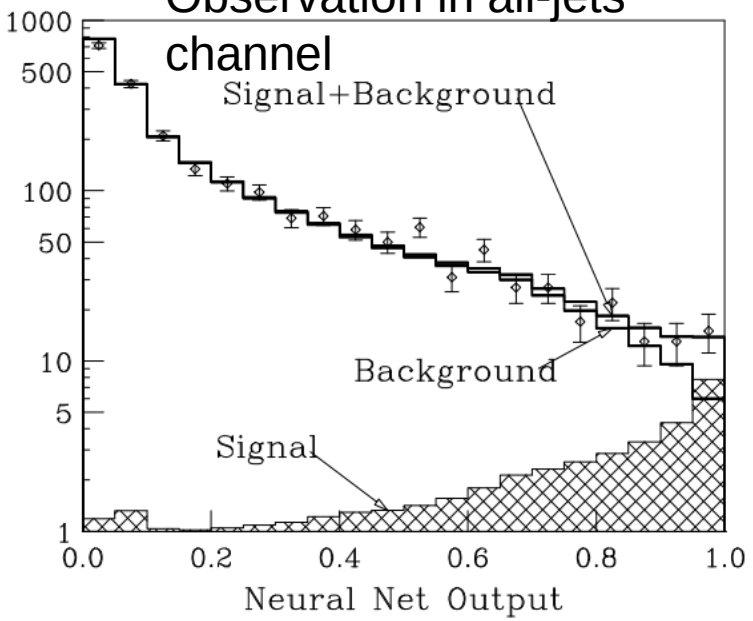
Top Quark Mass Measurement



m_t = 173.3 ± 5.6(stat.) ± 6.2 (syst.) GeV/c^2

Fit performed in 2-D: ($D_{LB/NN}$, m_{fit})

Observation in all-jets channel



MVA use in Higgs Discovery

- MVA used in every possible analysis aspect
  - Electrons/photons ID
  - MVA regression for EM cluster energy corrections
  - Vertex identification (diphotons)
  - b-tagging
  - S/B discrimination in all channels
    - $\gamma\gamma$, $ZZ \rightarrow 4l$, (WW, bb, $\tau\tau$)

Broad Categories of Analysis Tasks

- Classification
  - Object ID with high efficiency and low fake rates
    - Identification of electrons, photons, taus, b-quark jets, ..
  - signal/background discrimination
- Parameter Estimation
  - Measurement of quantities; observables ←→ parameters
- Function fitting
  - Energy correction functions, tag-rate functions, …

Mathematically, all of these are Functional Approximation problems.

Classification

- In classification, the function to be approximated is

$$f(x) = p(S \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid S)p(S)}{p(\boldsymbol{x} \mid S)p(S) + p(\boldsymbol{x} \mid B)p(B)}$$

  where *S* and *B* denote signal and background, respectively.
- In practice, it is sufficient to approximate the discriminant

$$D(x) = \frac{p(x \mid S)}{p(x \mid S) + p(x \mid B)}$$

  because $D(x)$ and $p(S|x)$ are related one-to-one:

$$p(S \mid x) = \frac{D(x)}{D(x) + [1 - D(x)] / A}$$

  where $A = p(S) / p(B)$ is the prior signal to background ratio

Multivariate Methods

A list of popular methods

- Random Grid Search
- Linear Discriminants
- Quadratic Discriminants
- Support Vector Machines
- Naïve Bayes (Likelihood Discriminant)
- Kernel Density Estimation
- Neural Networks
- Bayesian Neural Networks
- Decision Trees
- Random Forests
- Genetic Algorithms

Machine Learning

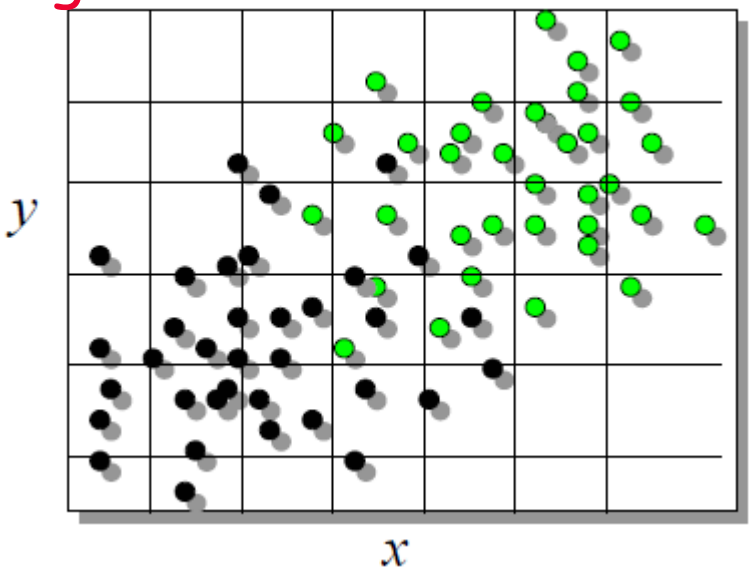
- Paradigm for automated learning from data, using computer algorithms
  - Has origins in the pursuit of artificial intelligence starting ~1960
- Requiring little *a priori* information about the function to be learned
- A method that can approximate a continuous non-linear function to arbitrary accuracy is called a universal approximator
  - e.g. Neural Networks

Machine Learning Approaches

- Supervised Learning
  - Supervised learning with a training data set containing feature variables (inputs) and target to be learned: {y,$\boldsymbol{x}$}
- Unsupervised Learning
  - No targets provided during training.
  - Algorithm finds associations among inputs.
- Reinforcement Learning
  - Correct outputs are rewarded, incorrect ones penalized.

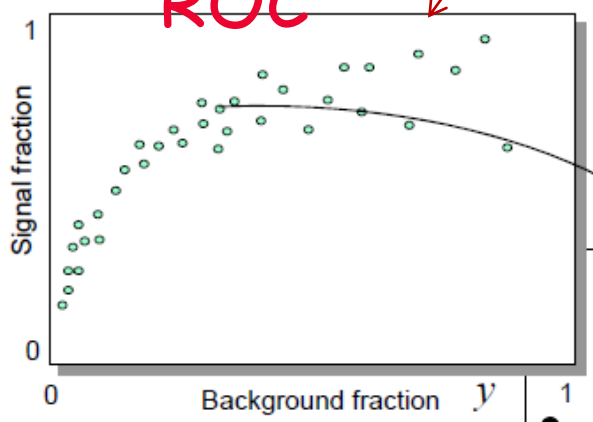
"Rectangular" Cuts

Regular Grid search



Signal eff. Vs bkgd. eff

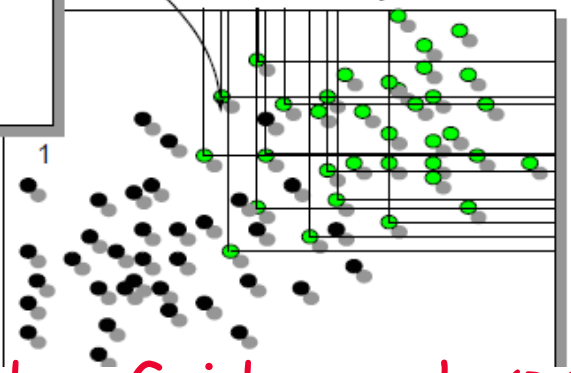
ROC



Take each point of the signal class as a cut-point

$x > x_i$

$y > y_i$



N_{tot} = # events before cuts
N_{cut} = # events after cuts
Fraction = N_{cut}/N_{tot}

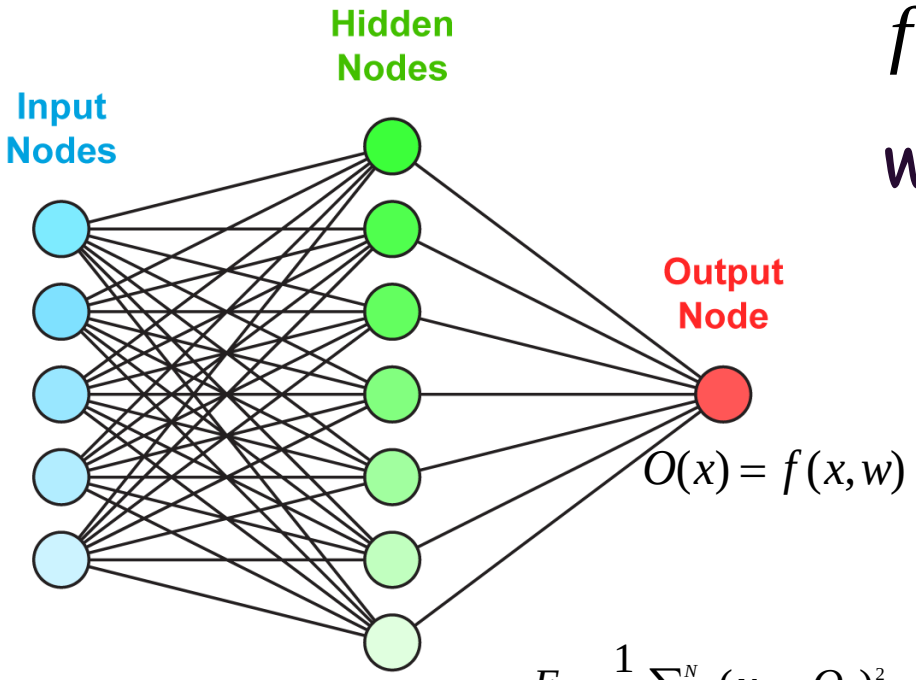
RGS can serve as a benchmark for comparisons of efficacy of variables, variable combinations, and classifiers

H.B.Prosper, P.Bhat, et al. CHEP' 95

Random Grid search (RGS)
Find "best" cuts

Neural Networks
The Bayesian Connection

- The output of a neural network can approximate the Bayesian posterior probability p(s|x):



$$f(\boldsymbol{x},\boldsymbol{w}) = g(\sum_j w_j h_j + \theta) = p(s \mid \boldsymbol{x})$$

where

$$h_j = g(\sum_i w_{ij} x_i + \theta_i);$$

$$g(a) = \frac{1}{1+e^{-a}}$$

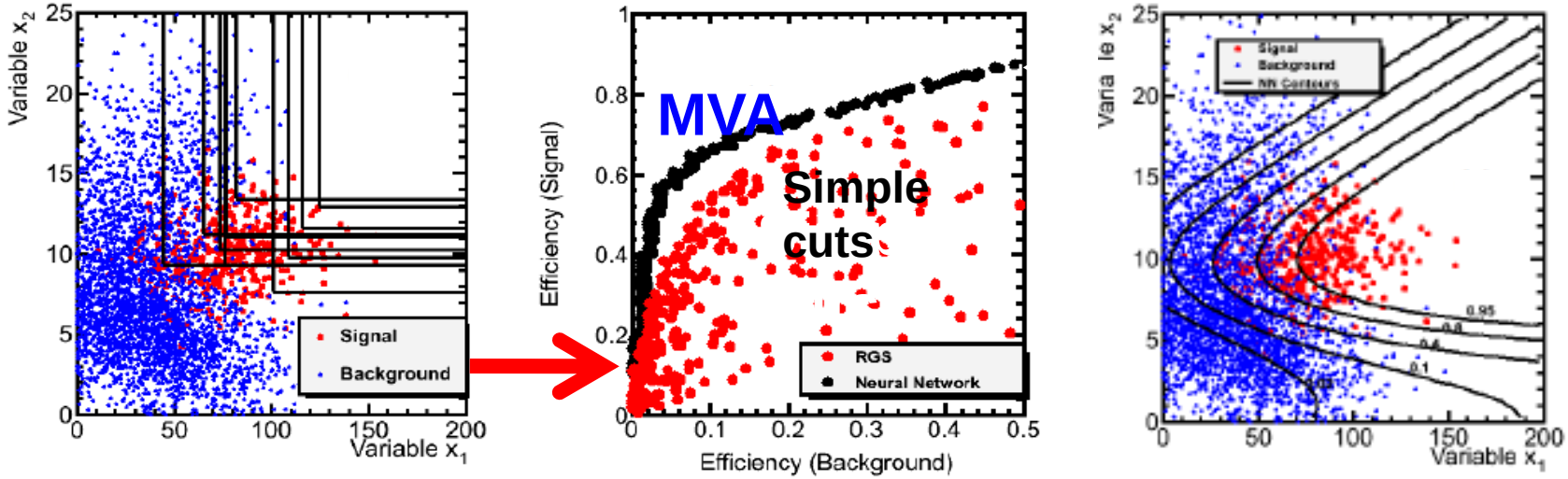
Flexible, non-linear model

$$E = \frac{1}{N}\sum_{p=1}^{N}(y_p - O_p)^2$$

RGS vs NN

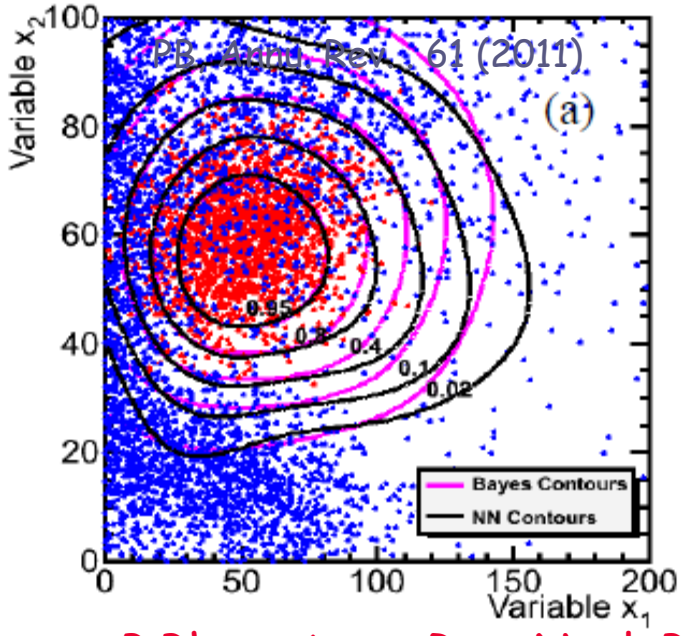
- Random Grid Search for "cut" optimization
  - The best "cut-based" analysis you can do!
- Notice that NN can provide significant gains even in this simple 2D analysis, at lower backgrounds which is the region of interest

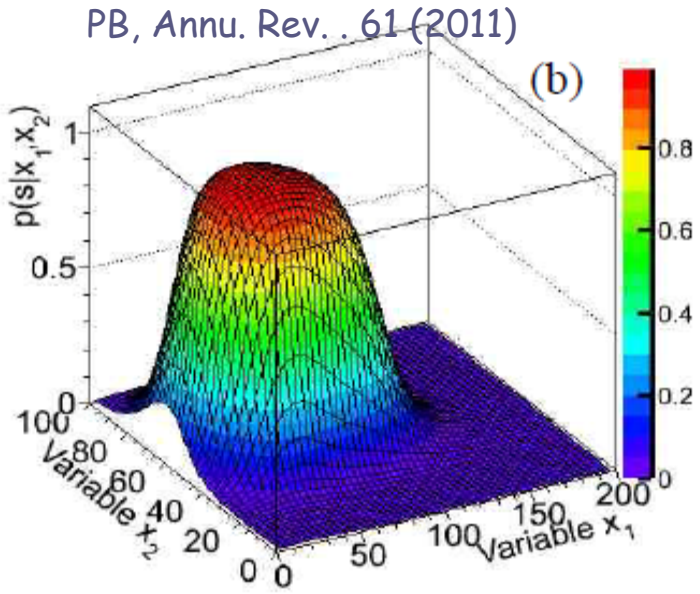
A simple illustration of MVA PB, *Annu. Rev. Nucl. Part. Sci*. 2011, **61**:281-309.



NN/MVA vs Bayes

NN (or any other fully multivariate technique) can provide discrimination close to the Bayes limit





P.Bhat, Annu. Rev. Nucl. Part. Sci. 61, 281-309 (2011)

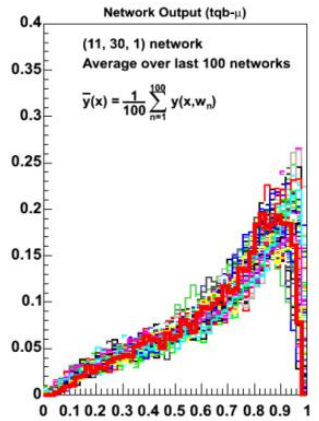
Bayesian Neural Networks

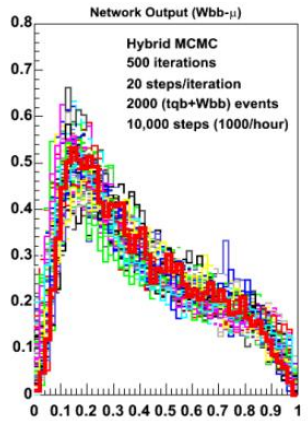
- Instead of attempting to find a single "best" network, i.e., a single "best" set of network parameters (weights), with Bayesian training we get a posterior density for the network weights, $p(w|\,T)$, $T \equiv$ Training data
- The idea here is to assign a probability density to each point w in the parameter space of the neural network. Then one takes a weighted average over all points, i.e., over all possible networks.

$$\tilde{y}(\boldsymbol{x}) = \int f(\boldsymbol{x},\boldsymbol{w})\, p(\boldsymbol{w}\,|\,T)\, d\boldsymbol{w}$$

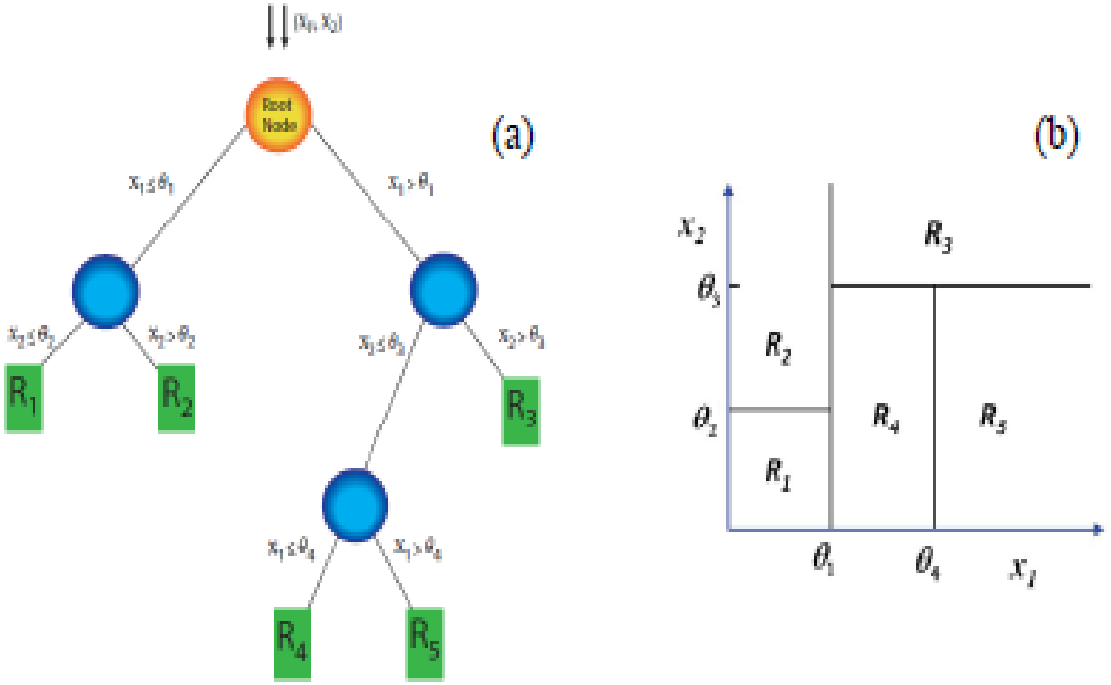
- Advantages:
  - Less likely to be affected by "over training"
  - No need to limit the number of hidden nodes
  - Good results with small training sample

P.C. Bhat, H.B. Prosper Phystat 2005, Oxford





Boosted Decision Tree (BDT)



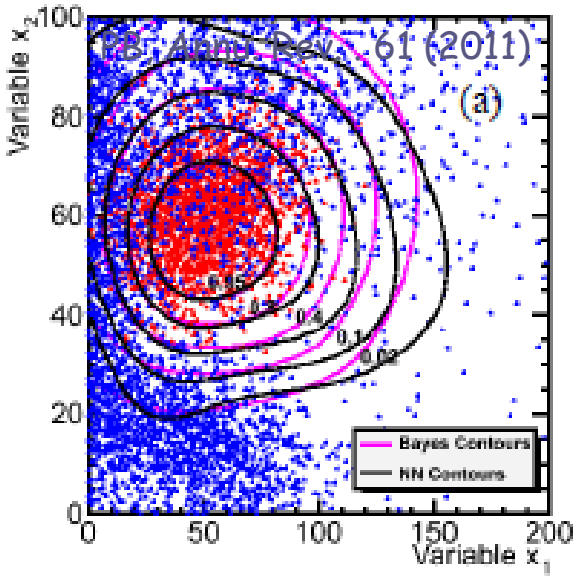
- A Decision Tree (DT) recursively partitions feature space into regions or bins with edges aligned with the axes of the feature space.
- A response value is attached to each bin,
  D(x) = s/(s+b)

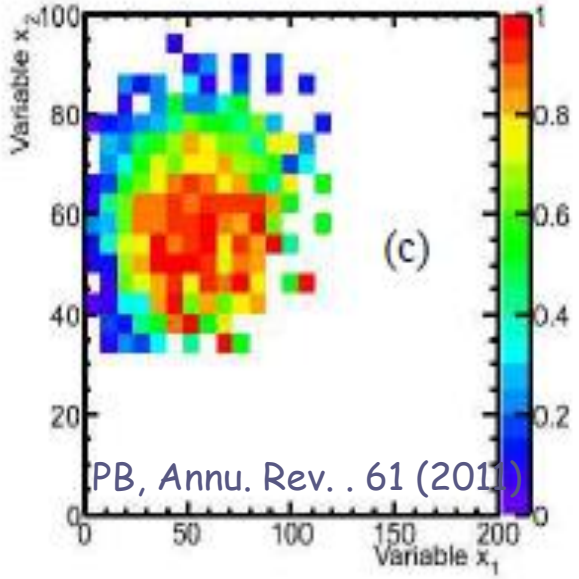
Boosting:
Make a sequence of M classifiers (DTs) that successively handle "harder" events and take a weighted average ➔ BDT

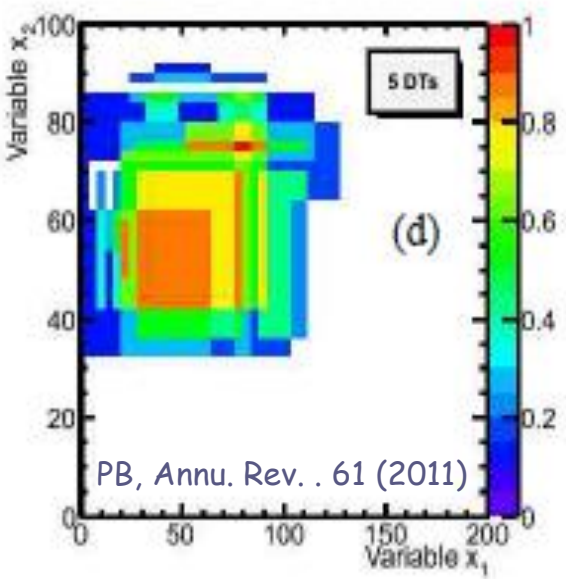
$$y(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x}, \boldsymbol{w_m})$$

$$\alpha_{\boldsymbol{m}} = \ln\left[\frac{1-\varepsilon_m}{\varepsilon_m}\right]$$

An Example







P.Bhat, Annu. Rev. Nucl. Part. Sci. 61, 281-309 (2011)

What method is best?

- The "no free lunch" theorem tells you that there is no one method that is superior to all others for all problems.

- In general, one can expect Bayesian neural networks (BNN), Boosted decision trees (BDT) and random forests (RF) to provide excellent performance over a wide range of problems.

- BDT is popular because of robustness, noise resistance (and psychological comfort!)

The Buzz about Deep Learning

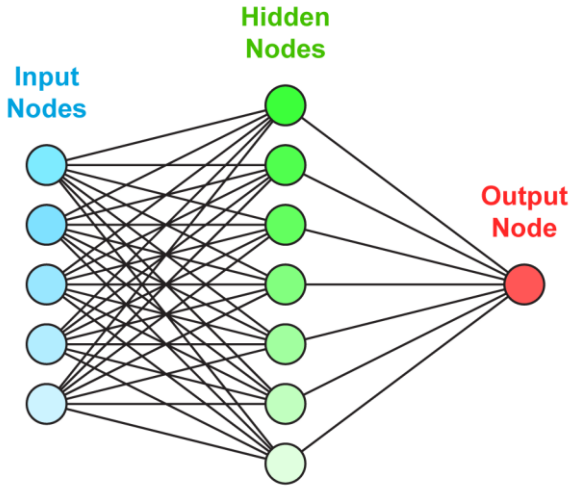
- A lot of excitement about "Deep Learning" Neural Networks (DNN) in the Machine Learning community
  - Spreading to other areas!
  - Some studies already in HEP!
- Multiple non-linear hidden layers to learn very complicated input-output relationships
- Huge benefits in applications in computer vision (image processing/ID), speech recognition and language processing

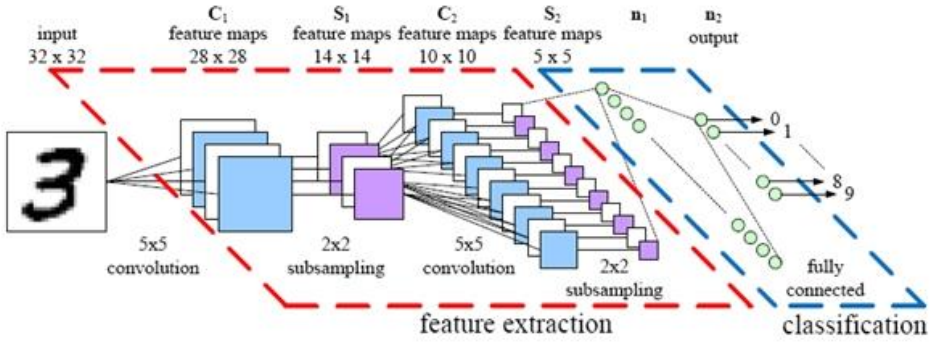
Deep Learning NN

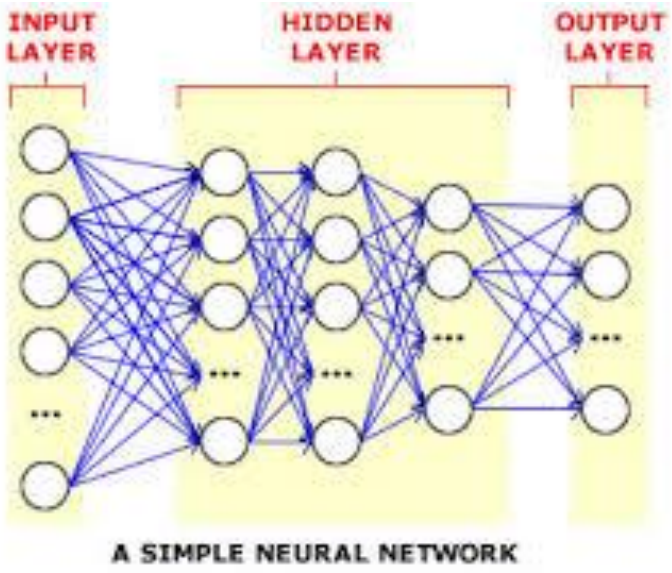
- Use raw data inputs instead of derived "intelligent" variables (or use both)
  - Pre-processing or feature extraction in the DNN
- Pre-train initial hidden layers with unsupervised learning
- Multi-scale Feature Learning
  - Each high-level layer learns increasingly higher-level features in the data
- Final learning better than shallow networks, particularly when inputs are unprocessed raw variables!
- **However, need a lot of processing power (implement in GPUs, time (and training examples)**

Deep Learning

Single hidden layer NN

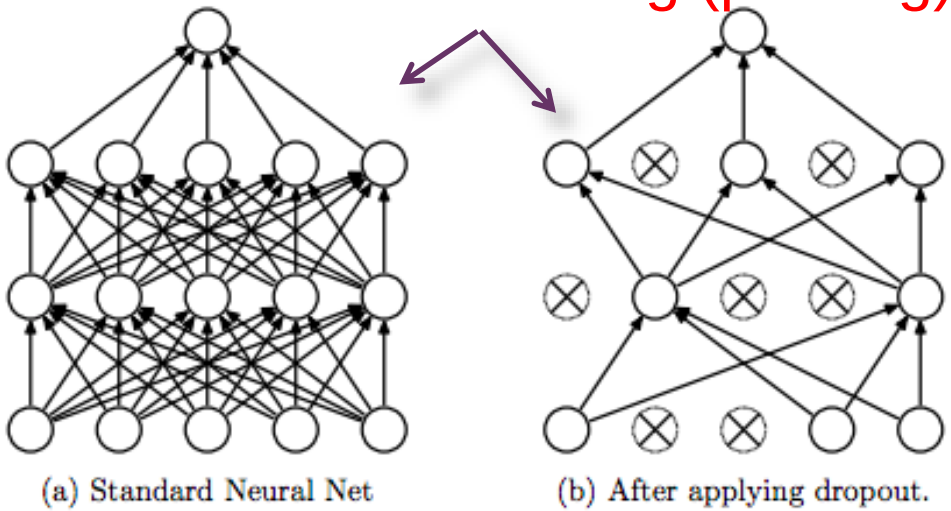






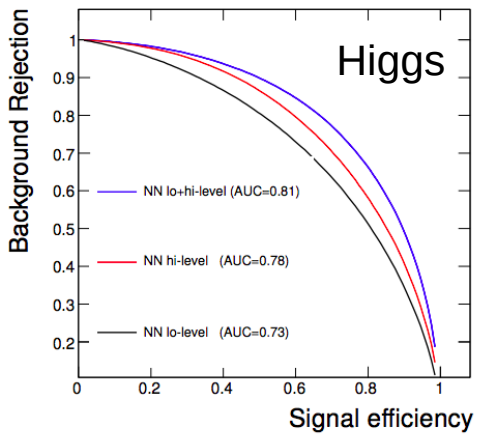
Multiple hidden layer NN

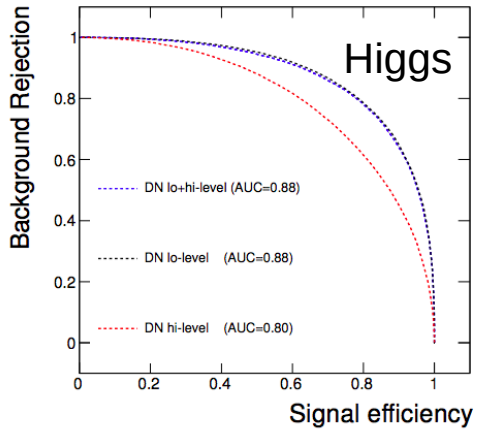
"Dropout" algorithm
to avoid overfitting (pruning)

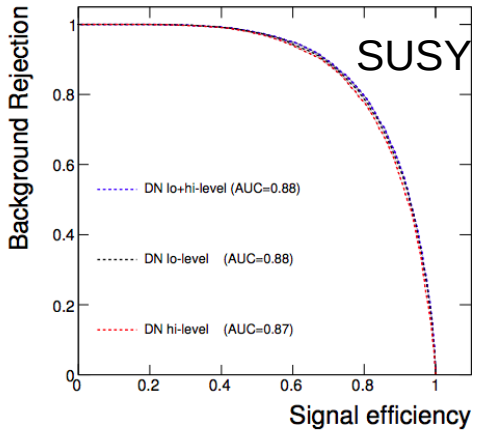


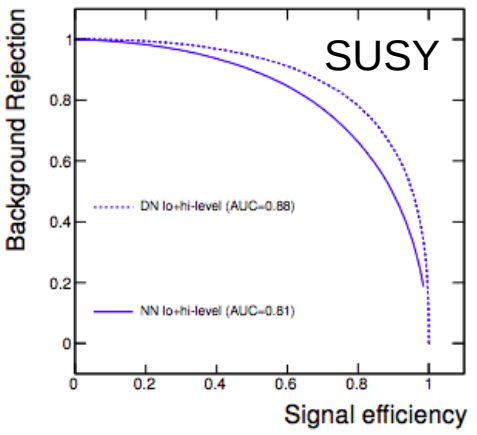
Deep Neural Networks for HEP

- Baldi, Padowski, Whiteson arXiv:1402.4735v2
- Studied two benchmark processes
  - Charged Higgs vs ttbar events
  - SUSY: Chargino pairs vs WW events into dilepton+MET final state









Significant improvement in Higgs case, not so dramatic in case of SUSY

Exotic Higgs

Discovery significance

Technique	Low-level	High-level	Complete
NN	2.5σ	3.1σ	3.7σ
DN	4.9σ	3.6σ	5.0σ

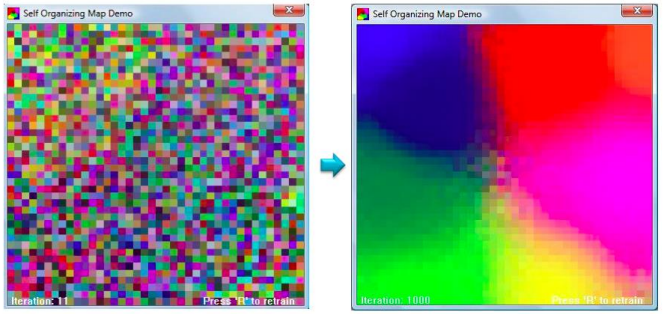
SUSY Study

Discovery significance

Technique	Low-level	High-level	Complete
NN	6.5σ	6.2σ	6.9σ
DN	7.5σ	7.3σ	7.6σ

Unsupervised Learning

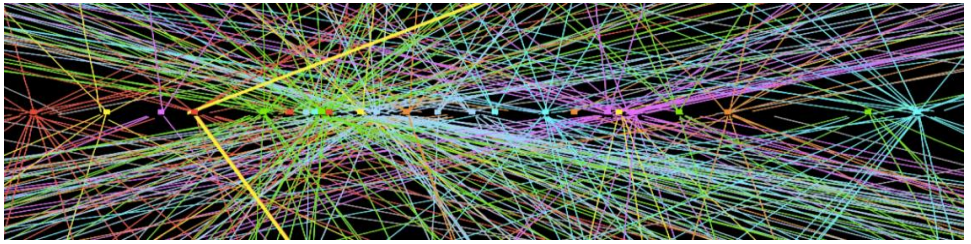
- The most common approach is to find clusters or hidden patterns or groupings in data
- Common and useful methods
  - K-Means clustering
  - Gaussian mixture models
  - Self-organizing maps (SOM)



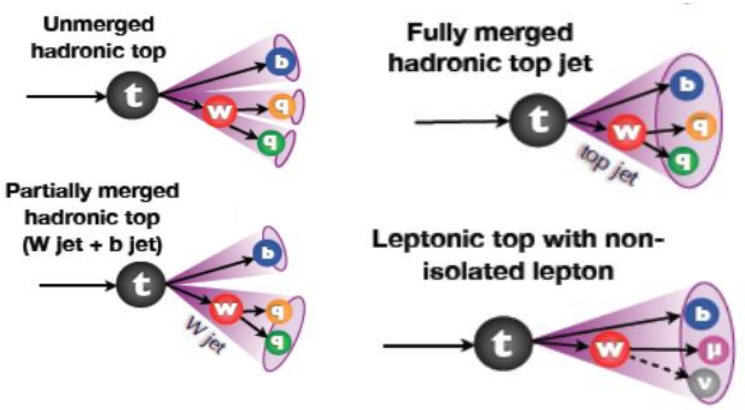
http://chem-eng.utoronto.ca/~datamining/Presentations/S

- We have not tapped these methods for identifying unknown components in data, unsupervised classification, for exploratory data analysis
- Could be useful in applications for topological pattern recognition
  - Use in Jet-substructure, boosted jet ID

Challenges in LHC Run 2 and beyond



- Challenges:
  - Pile-Up mitigation!
    - <PU>~40 in Run2
      - Associating tracks to correct vertices
      - Correcting jet energies, MET, suppressing fake "pileup" jets,
      - Lepton and photon isolation
  - Boosted Objects
    - Complicates Object ID
    - W, Z, Higgs, top taggers!
      - Provides new opportunities
      - Use jet substructure
  - High energy Lepton ID
  - Signals of BSM could be very small
    - Small MET in SUSY signatures (compressed, stealth,…)



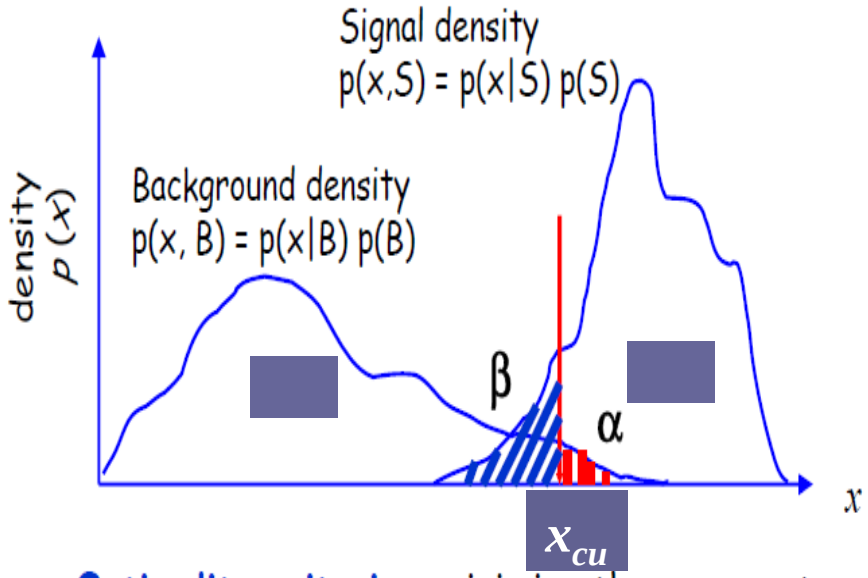
- Need new algorithms, approaches for reco and analysis
- New ideas in triggering and data acquisition

Summary

- Multivariate methods brought a paradigm shift in HEP analysis ~20 years ago. Now they are state of the art.
- Applications of new ideas/algorithms such as deep learning should be explored, but the resources involved may not justify the use in every case.
- Revived emphasis on unsupervised learning is good and should be exploited in HEP.
- Well established techniques of the past – single hidden layer neural networks, Bayesian neural networks, Boosted Decision Trees should continue to be the ubiquitous general purpose MVA methods.

Extra slides

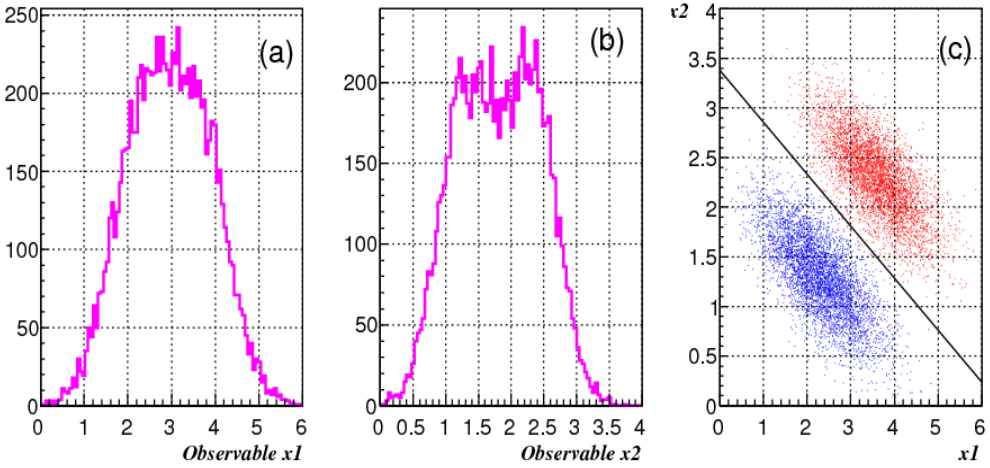
Optimal Discrimination



Optimality criterion: minimize the error rate:

min α + β

- More dimensions can help!
- One dimensional distributions are marginalized distributions of multivariate density.
- **f(x1)=∫g(x1,x2,x3, ..)dx2dx3..**



← Minimize the total **misclassification** error

$$\alpha = \int_{x_{cut}}^{\infty} p(x,B)dx \qquad \beta = \int_{-\infty}^{x_{cut}} p(x,S)dx$$

Significance level

1-β: Power

Minimizing Loss of Information .. And Risk

- General Approach to functional approximation
- Minimize Loss function:

$$L\{y, f(\mathbf{x}, \mathbf{w})\}$$

- It is more robust to minimize average loss over all predictions

$$R(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L\{y_i, f(\mathbf{x}_i, \mathbf{w})\}.$$

A common Risk function

$$R(\mathbf{w}) = E(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i, \mathbf{w}))^2$$

or a cost (or error) function: $C(\mathbf{w}) = R(\mathbf{w}) + \lambda Q(\mathbf{w})$

- There are many approaches/methods
-

Calculating the Discriminant

$$p(S \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid S)\,p(S)}{p(\boldsymbol{x} \mid S)\,p(S) + p(\boldsymbol{x} \mid B)\,p(B)}$$

$$D(x) = \frac{p(x \mid S)}{p(x \mid S) + p(x \mid B)}$$

- Density estimation, in principle, is simple and straightforward.
- Histogramming:
  - **Histogram** data in M bins in each of the d feature variables

    ➔M^d bins ← **Curse Of Dimensionality**
  - In high dimensions, we would need a huge number of data points or most of the bins would be empty leading to an estimated density of zero.
  - But, the variables are generally correlated and hence tend to be restricted to a sub-space Therefore, **Intrinsic Dimensionality << d**
- There are more effective methods for density estimation