FERMILAB-PUB-10-425-PPD

Advanced Analysis Methods in Particle Physics

Pushpalatha C. Bhat

Fermi National Accelerator Laboratory, Batavia, IL 60510, USA

email: pushpa@fnal.gov

(Submitted to *Annual Review of Nuclear and Particle Science*)

"*That is positively the dopiest idea I have heard*." - Richard Feynman, when he signed on to work on the Connection Machine, at the *Thinking Machines Corp.*, in the summer of 1983.

Key Words

Advanced analysis, multivariate methods, neural networks, Bayesian inference, Tevatron, Large Hadron Collider (LHC)

Abstract

Each generation of high energy physics experiments is grander in scale than the previous – more powerful, more complex and more demanding in terms of data handling and analysis. The spectacular performance of the Tevatron and the beginning of operations of the Large Hadron Collider, have placed us at the threshold of a new era in particle physics. The discovery of the Higgs boson or another agent of electroweak symmetry breaking and evidence of new physics may be just around the corner. The greatest challenge in these pursuits is to extract the extremely rare signals, if any, from huge backgrounds arising from known physics processes. The use of advanced analysis techniques is crucial in achieving this goal. In this review, I discuss the concepts of optimal analysis, some important advanced analysis methods and a few examples. The judicious use of these advanced methods should enable new discoveries and produce results with better precision, robustness and clarity.

Operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy

Contents

1 INTRODUCTION

The ambitious goal of understanding Nature at the most fundamental scale has led to the development of particle accelerators and detectors at successively grander scale. The revolutionary discoveries at the beginning of the twentieth century opened up the quantum world. By mid-century, the Standard Model (SM) of particle physics (1-6) was being built and by the turn of the century, the last quark (7, 8) and the last lepton (9**)** of the Standard Model had been found. Despite this spectacular success, a vital part of the Standard Model, the "Higgs mechanism" (10-13), still awaits experimental evidence. And there are indications that the SM particles and forces might be telling us only a part of the story. Since the SM accounts for only 4% of what makes up the universe, the rest must be explained in terms of matter and phenomena we have yet to uncover. The evidence for dark matter in the universe, the evidence for an accelerating universe, the discovery of neutrino oscillations, and the persistent discrepancies in some of the precision measurements in SM processes, are some of the strong indicators of the existence of new physics beyond the SM. It appears that new physics is inevitable at the TeV energy scale. We might be at the threshold of what might prove to be another extraordinary century.

Since the discovery of the top quark in 1995 (7, 8, 14), the pursuit of the Higgs boson and searches for new physics beyond the SM have taken center-stage. The luminosity upgrades of the Fermilab Tevatron (15) in the past decade have produced unprecedented amounts of proton-antiproton collision data at the center of mass energy ($\sqrt{s}$) of 1.96 TeV. This, in conjunction with the use of advanced analysis methods, has enabled the observation of the electroweak production of single top quarks (16,17) and sensitive searches for the Higgs boson and physics beyond the SM. The Large Hadron Collider (LHC) (18), with the design energy of $\sqrt{s}$ = 14 TeV, will open new energy frontiers that might help answer some of the most pressing particle physics questions of today.

The investments in the accelerator facilities and experiments – intellectual and monetary – and the total time span of the undertakings are so great that they cannot be easily replicated. Therefore, it is of the utmost importance to make the best use of the output of this investment – the data we collect. While the advances in computing technology have made it possible to

handle vast amounts of data, it is crucial that the most sophisticated techniques be brought to bear in the analysis of these data at all stages of the experiment. The instrumentation has, over the past century, advanced from photographic detectors to those integrated with ultra-fast electronics that produce massive amounts of digital information each second. The data analysis, likewise, has progressed from visual identification of particle production and decays to hunting for bumps in invariant mass spectra of exclusive final state particles to event counting in inclusive data streams. The rates of interactions and the number of detector channels to be read out have grown by orders of magnitude over the course of the last few decades. We can no longer afford to write out data to storage media based on simple interaction criteria. But, the events that we seek to study are extremely rare. So, today, data analysis in high energy physics (HEP) experiments starts when a high energy interaction or an *event* occurs. The electronic data from the detectors need to be transformed into useful "physics" information in real-time. The *trigger system* is expected to select interesting events for recording and discard the background or uninteresting events. Information from different detector systems is used to extract event features such as the number of tracks, high transverse momentum objects, and object identities. The extracted features are then used to decide whether the event should be recorded. At the LHC, the event rate will be reduced from 40 MHz collision rate to ~200 HZ for recording. This online processing of data is performed with a combination of hardware and software components.

More detailed analysis of the recorded data is performed offline. The common offline data analysis tasks are: charged particle tracking, energy/momentum measurements, particle identification, signal/background discrimination, fitting, the measurement of parameters, and the derivation of various correction and rate functions. The most challenging of the tasks is identifying events that are rare, and obscured by the wide variety of processes that can mimic the signal. This is a veritable case of "finding needles in a hay-stack" for which the conventional approach of selecting events using cuts on individual kinematic variables can be far from optimal.

The power of computers coupled with important developments in *machine learning* algorithms, particularly the back-propagation algorithm for training neural networks, brought a revolution in multivariate data analysis by the late 1980s. There was much skepticism about these ideas in the

early 1990s when these methods were brought into HEP analyses (19-22). However, after several successful applications (23-29), particle physicists have largely accepted the use of neural networks and other multivariate methods. It is also now evident that without these powerful techniques, many of the important physics results that we have today would not have been achievable using the available datasets. My goal, in this paper, is to provide an introduction to the concepts that underlie these advanced analysis methods and describe a few popular methods. I will also briefly discuss some analysis examples and prospects for future applications.

2 OPTIMAL ANALYSIS CONCEPTS

"*Keep it simple, as simple as possible, not any simpler*" Albert Einstein

The goal in data analysis is to extract the best possible results. Here I discuss the types of analysis tasks we perform, why the sophistication of multivariate methods is necessary to obtain optimal results, introduce the concepts and the general framework that underlie the popular methods.

The broad categories of analysis tasks are: (a) classification (b) parameter estimation and (c) function fitting. Mathematically, in all these cases, the underlying task is that of functional approximation. Classification of objects or events is, by far, the most important analysis task in HEP. Common examples of classification are identification of electrons, photons, τ-leptons, *b*-quark jets, etc., and discriminating signal events from those arising from background processes. It is necessary to identify objects with good purity and to isolate events arising from specific physics processes before further studies can be undertaken. Optimal discrimination is crucial if one wishes to make the best use of data and provide signal-enhanced samples for precision physics measurements. Parameter estimation is essentially regression or fitting a model to the data. Measurements of track parameters, vertices, physical parameters such as production cross sections, branching ratios, masses and other properties are examples of regression. Some examples of function fitting are the derivation of correction functions, tag rate functions and fake rate functions.

These categories of tasks are also referred to as *pattern recognition* problems.[1]

2.1 Multivariate Treatment of Data

Data characterizing an object or an event generally involve multiple quantities referred to as *feature variables.* These may be, for example, the four-vectors of particles, energy deposited in calorimeter cells, deduced kinematic quantities of objects in the event, or global event characteristics. The variables, generally, are also correlated in some way. Therefore, to extract results with maximum precision and minimum bias it is necessary to treat these variables in a fully multivariate way. Consequently, the methods for an *optimal analysis* are necessarily multivariate.

Each multivariate datum of an object or an event can be represented by a vector $\boldsymbol{x} = (x_1, x_2, ..., x_d)$ in a *d*-dimensional *feature space.* The objects or events of a particular type or class can be expected to occupy specific contiguous regions in the feature space. When correlations exist between variables, the *effective dimensionality* of the problem is smaller than *d.* (The kinematic variables in HEP events are, generally, smooth functions and highly correlated across objects in the event.)

Diligent "pre-processing" of data is the first step in an analysis. This is also referred to as *feature extraction* or *variable selection.* Having selected a set of variables, one might apply a transformation to the variables to yield a representation of the data that exhibits certain desirable properties. This could be simple scaling of the variables or a more sophisticated transformation. In some applications this pre-processing might be the only necessary multivariate treatment of the data. In others, it serves as the starting point for more refined analysis. Given $\boldsymbol{x}$, the goal is to construct a function $y = f(\boldsymbol{x})$ with properties that are useful for subsequent decision-making and inference. That is, we would like to extract a map $f : \Re^d \rightarrow \Re^N$, preferably with $N << d$. ($\Re^m$: real vector space of dimension *m*.) We try to approximate the desired function with $\tilde{y} = f(\boldsymbol{x}, \boldsymbol{w})$, where $\boldsymbol{w}$ are some adjustable parameters. I will discuss the general approach for obtaining the functional mapping in later sections.

[1] Pattern recognition also encompasses *knowledge discovery* by data exploration which deals with data-driven extraction of features, and deriving empirical rules via data-mining.

The power of multivariate analysis is illustrated by a simple two-dimensional example. **Figure 1(a), (b)** show distributions of two variables $x1$ and $x2$ arising from two bivariate Gaussian distributions shown in **Figure 1(c)**. The one-dimensional projections (**Figure 1(d,e)**), i.e., marginalized densities $f(x1) = \int G(x1, x2)dx2$ and $f(x2) = \int G(x1, x2)dx1$ have considerable overlap and there are no obvious cuts on the variables $x1$ and $x2$ that would separate the two classes. But, when we examine the data in 2-dimensions, we see that the two classes are largely separable. Therefore, a cut applied to the linear function (30), $\tilde{y} = ax1 + bx2$ (called a linear discriminant) plotted in **Figure 1(f)** can provide *optimal discrimination* of the two classes. (The linear function shown in **Figure 1(c)** is a simple example of a decision boundary.) By optimal discrimination we mean a procedure that minimizes the probability of mis-classification.

Machine Learning:

Machine Learning is the paradigm for automated *learning from data* using computer algorithms. It has origins in the pursuit of Artificial Intelligence, particularly, in Frank Rosenblatt's creation of the Perceptron around 1960 (31).

2.2 Machine Learning

The availability of vast amounts of data, challenging scientific and industrial problems characterized by multiple variables paved the way to the development of automated algorithms for *learning from data*. The primary goal of learning is to be able to respond correctly to future data. In conventional statistical techniques, one starts with a mathematical model and finds parameters of the model either analytically or numerically using some optimization criteria. This model then provides predictions for future data. In machine learning, an approximating function is inferred automatically from the given data without requiring a priori information about the function.

In machine learning, the most powerful approach to obtain the approximation $f(\boldsymbol{x}, \boldsymbol{w})$, of the unknown function $f(\boldsymbol{x})$, is *supervised learning*, in which a training data set, comprising feature vectors (inputs)[2] and the corresponding targets (or desired outputs), is used. The training data

[2] I use feature vectors and inputs, interchangeably.

set $\{y, \boldsymbol{x}\}$, where *y* are the targets (from the true function $f(\boldsymbol{x})$), encodes information about the input-output relationship to be learnt. In HEP, the training data set generally comes from Monte Carlo simulations. The function $f(\boldsymbol{x})$ is discrete for classification ({0,1} or {-1,1} for binary classification) and is continuous for regression. (Thus the distinction between discrimination and regression is not fundamental.) The goal of learning (or training) is to find $\boldsymbol{w}$ the parameters of our "model" for the desired input-output map.

In all approaches to functional approximation (or function fitting), the information loss incurred in the process has to be minimized. The information loss is quantified by a loss function $L(y, f(\boldsymbol{x}, \boldsymbol{w}))$. In practice, the minimization is more robust if one minimizes the loss function averaged over the training data set. A learning algorithm, therefore, directly or indirectly, minimizes the average loss, called the *risk*, quantified by a risk function $R(\boldsymbol{w})$ that measures the cost of mistakes made in the predictions, and finds the best parameters $\boldsymbol{w}$. The empirical risk is defined as the average loss over all (*N*) predictions,

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

A common risk function used is the mean square error,

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

which represents the discrepancy between the desired function and its approximation. If the optimization has to take into account any constraint $Q(\boldsymbol{w})$, it can be added to the risk function to give a cost function to be minimized, given by,

$$C(\boldsymbol{w}) = R(\boldsymbol{w}) + \lambda Q(\boldsymbol{w}), \qquad 3.$$

where λ is an adjustable parameter that determines the strength of the constraint imposed. The cost function in the case of a mean square error is the well known constrained χ^2 fit. The function $f(\boldsymbol{x}, \boldsymbol{w})$ obtained by the procedure converges, in the limit of a large training data set, to the function $f(\boldsymbol{x})$ that minimizes the true risk function.

The risk minimization can be done using many algorithms, each of which essentially attempt to find the global minimum of the cost function or error hypersurface in the parameter space. The generic method is that of gradient descent. Other popular methods include Levenberg-Marquardt (32), simulated annealing (33) and genetic algorithms (34). The constraint in the cost function is typically used to control model complexity (or "over-fitting"), and is called regularization. The performance of the classifier or estimator is generally evaluated using a test data set independent of the training set.

A method that is able to approximate a continuous nonlinear function to arbitrary accuracy is called a *universal approximator*. Neural networks are examples of universal approximators.

Two other types of learning approaches are unsupervised and reinforcement learning. In the former, no targets are provided and the algorithm finds associations among the feature vectors. In the latter approach, correct outputs are rewarded and incorrect ones are penalized. These methods will not be further discussed here.

2.3 The Bayesian Framework

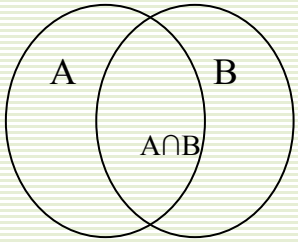
"Today's posterior distribution is tomorrow's prior." – David Lindley

The Bayesian approach to statistical analysis is that of inductive inference. It allows the use of prior knowledge and new data to update probabilities. Therefore, it is a natural paradigm for learning from data. It is an intuitive and rigorous framework for handling classification and parameter estimation problems. At the heart of Bayesian inference (35) is Bayes theorem,

$$p(B \mid A) = \frac{p(A \mid B)\, p(B)}{p(A)}, \qquad 4.$$

where the conditional probabilities $p(B \mid A)$ and $p(A \mid B)$ are referred to as the *posterior probability* and *likelihood*, respectively, $p(B)$ is the *prior probability* of B, and the denominator is simply the total probability of A,

Conditional Probabilities:



$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)}$$

Bayes theorem can be readily derived from these expressions.

$p(A) = \int p(A \mid B)\, p(B) dB$. If B is discrete, the integral is replaced by a sum.

Let us consider a binary classification problem where an event has to be classified either as due to a signal process s, or due to a background process b. This is achieved by placing a cut on the ratio of the probabilities for the two classes,

$$r(\boldsymbol{x}) = \frac{p(s \mid \boldsymbol{x})}{p(b \mid \boldsymbol{x})} = \frac{p(\boldsymbol{x} \mid s)\, p(s)}{p(\boldsymbol{x} \mid b)\, p(b)}, \qquad 5.$$

where $p(\boldsymbol{x} \mid s)$ and $p(\boldsymbol{x} \mid b)$ are the likelihoods of the data for signal and background classes, respectively; $p(s)$ and $p(b)$ are the prior probabilities. The discriminant '*r*' is called the Bayes discriminant, where $r(\boldsymbol{x})=constant$ defines a decision boundary in the feature space. The Bayes rule is to assign a feature vector to the signal class if $p(s \mid \boldsymbol{x}) > p(b \mid \boldsymbol{x})$. This rule minimizes the probability of misclassification. Any classifier which minimizes the misclassification rate is said to have reached the *Bayes limit*. The problem of discrimination, then, mathematically reduces to that of calculating the Bayes discriminant $r(\boldsymbol{x})$ or any one-to-one function of it.

The posterior probability for the desired class *s*, becomes,

$$p(s \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid s)\, p(s)}{p(\boldsymbol{x} \mid s)\, p(s) + p(\boldsymbol{x} \mid b)\, p(b)} = \frac{r}{1+r}. \qquad 6.$$

There are parametric and non-parametric methods to estimate $p(\boldsymbol{x} \mid s)$ and $p(\boldsymbol{x} \mid b)$ that I will discuss in the next section. If one minimizes the mean square error function (Equation 2) where the targets are {0,1}, then $f(\boldsymbol{x},\boldsymbol{w})$, if flexible enough, will directly approximate the posterior probability, $p(s \mid \boldsymbol{x})$. Neural networks, being universal approximators, are one such class of functions.

When $p(s)$ and $p(b)$ are not known, which is typically the case, one can calculate the discriminant function,

$$D(\boldsymbol{x}) = \frac{s(\boldsymbol{x})}{s(\boldsymbol{x}) + b(\boldsymbol{x})}, \qquad 7.$$

where $s(\boldsymbol{x}) = p(\boldsymbol{x} \mid s)$ and $b(\boldsymbol{x}) = p(\boldsymbol{x} \mid b)$. The posterior probability for the signal class is related to this discriminant function by,

$$p(s \mid \boldsymbol{x}) = \frac{D(\boldsymbol{x})}{[D(\boldsymbol{x}) + (1 - D(\boldsymbol{x}))/k]}, \qquad 8.$$

where $k = p(s)/p(b)$. The discriminant $D(\boldsymbol{x})$ is often referred to as the likelihood discriminant in HEP. The discriminating power of $D(\boldsymbol{x})$, which is a one-to-one function of $p(s \mid \boldsymbol{x})$, is the same as that of $p(s \mid \boldsymbol{x})$.

When many classes $C_k\,(k = 1,2,...,N)$ are present, the Bayes posterior probability can be written as,

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k)\,p(C_k)}{\sum p(\boldsymbol{x} \mid C_k)\,p(C_k)}. \qquad 9.$$

The Bayes rule for classification is to assign the object to the class with highest posterior probability. This is also the criterion in hypothesis testing.

In problems of parameter estimation, the posterior probability for a model parameter θ is,

$$p(\theta \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid \theta)\,p(\theta)}{p(\boldsymbol{x})}, \qquad 10.$$

where $p(\theta)$ is the prior probability of θ. Thus in the Bayesian approach, one has a probability distribution of possible values for the parameter θ, while in conventional machine learning methods one calculates a maximum likelihood estimate for θ. However, the two approaches are closely related. The minimization of the error or cost function in the machine learning approach is equivalent to maximizing the Bayesian posterior probability.

3 POPULAR METHODS

I discuss here several methods that are particularly relevant and popular in high energy physics – from the simplest to the most sophisticated multivariate methods with minimal, essential,

mathematics. The interested reader can consult many excellent books for details of these methods and algorithms (36-40).

3.1 Grid Searches

The conventional approach to separating signal from background is to apply a set of cuts such as $x_1 > z_1, x_2 > z_2 \ldots$ where $(z_1, z_2 \ldots z_d)$ forms a cut-point in the *d*-dimensional feature space. These "rectangular" cuts are usually arrived at by a process of trial and error informed by common sense and physics insight. Unfortunately, there is no guarantee that this procedure will lead to optimal cuts (as illustrated by the example in section 2). One can obtain the best set of rectangular cuts by a systematic search over a grid in feature space. A search over a regular grid, however, is inefficient: a lot of time can be spent scanning regions of feature space that have few signal or background points. Moreover, the number of grid points grows like M^d, which increases rapidly with bin count *M* and dimensionality *d*, a problem known as the "curse of dimensionality". A better way is to use a "Random Grid Search" (41) where a distribution of points, which form a random grid, is used as the set of cut-points. The cut-points could be obtained, for example, from signal events generated by a Monte Carlo simulation. The results can be plotted as efficiency for retaining signal versus efficiency for background for each of the cuts. The optimal cuts are those that maximize signal efficiency for desired background efficiency.[3]

The random grid search can be used for a rapid search for the best rectangular cuts, to compare the efficacy of variables or to serve as a benchmark for more sophisticated multivariate analyses.

3.2 Linear Methods

In grid searches, the decision boundaries are lines or planes parallel to the axes of the feature space. As illustrated in **Figure 1**, optimal separation of classes might require decision boundaries rotated relative to the axes of the original feature space.

In a linear model, the mapping can be written as,

[3] The plot is akin to the ROC (Receiver Operating Characteristic) curve, first invented in the 1950s to study radio signals in the presence of noise and used in signal detection theory.

$$\tilde{y}(\boldsymbol{x}) = w_0 + w_1 x_1 + w_2 x_2 + ... = \sum_i w_i x_i = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} \,, \qquad 11.$$

where, $\boldsymbol{w}^{\mathrm{T}}$ is the vector of weights[4].

Sir Ronald Fisher (30) pioneered the earliest successful applications of linear discriminants. Fisher's approach to discrimination between classes was to find a linear combination of input variables that maximizes the ratio of between-group variance to within-group variance. If we consider two sets of feature vectors $\boldsymbol{x_s}, \boldsymbol{x_b}$ from signal and background classes, with means $\boldsymbol{\mu}_s, \boldsymbol{\mu}_b$, and variances σ_s, σ_b, the Fisher criterion is to maximize

$$F(\boldsymbol{w}) = \frac{(\boldsymbol{\mu}_s - \boldsymbol{\mu}_b)^2}{\sigma_s^2 + \sigma_b^2}, \qquad 12.$$

which yields, for the parameters $\boldsymbol{w}$,

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_s - \boldsymbol{\mu}_b), \qquad 13.$$

where $\boldsymbol{\Sigma}$ is the common covariance matrix for the classes. The Fisher discriminant can also be derived from Bayes discriminant starting with Gaussian density for each class,

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}\left|\boldsymbol{\Sigma}\right|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]. \qquad 14.$$

Then, taking the logarithm of the Bayes discriminant (Equation 5), we obtain,

$$D(\boldsymbol{x}) = \log\frac{p(s\,|\,\boldsymbol{x})}{p(b\,|\,\boldsymbol{x})} = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_b)^T \boldsymbol{\Sigma}_{\boldsymbol{b}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_b) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_s)^T \boldsymbol{\Sigma}_{\boldsymbol{s}}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_s) + \frac{1}{2}\log\frac{\left|\boldsymbol{\Sigma}_{\boldsymbol{s}}^{-1}\right|}{\left|\boldsymbol{\Sigma}_{\boldsymbol{b}}^{-1}\right|} + \log\frac{p(s)}{p(b)}. \qquad 15.$$

This is the general form of the Gaussian classifier, which after omitting non-essential terms that are independent of $\boldsymbol{x}$, can be written as,

$$F = D(\boldsymbol{x}) = \frac{1}{2}(\chi_b^2 - \chi_s^2), \qquad 16.$$

[4] I will use weights and parameters, interchangeably.

where, $\chi^2 = (\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$. If the covariance matrices are equal, i.e., $\boldsymbol{\Sigma}_s = \boldsymbol{\Sigma}_b = \boldsymbol{\Sigma}$, then one obtains Fisher's linear discriminant. If the covariance matrices are not equal, Equation 16 represents a quadratic function in the input variables. However, if we consider the augmented feature space with variables x_1, x_2, x_1^2, x_2^2, and $x_1 x_2$, the quadratic discriminant function in the original space becomes a linear discriminant corresponding to linear decision boundaries in the augmented 5D space.

The Gaussian classifier is also referred to as the *H*-matrix method, where $H = \boldsymbol{\Sigma}^{-1}$ and is used in electron identification in DØ (see Refs. 23, 42).

So far, we discussed Gaussian densities as the relevant models. In case of non-Gaussian densities, one can still use linear methods such as Support Vector Machines (38), provided that the data are mapped into a space of sufficiently high dimensions.

3.3 Naïve Bayes or Likelihood Discriminant

When the feature variables are statistically independent, the multivariate densities can be written as products of one dimensional densities, without loss of information. In this case, the discriminant in Equation 7 becomes,

$$D(\boldsymbol{x}) = \frac{\Pi_i s_i(x_i)}{\Pi_i s_i(x_i) + \Pi_i b_i(x_i)}, \qquad 17.$$

where $s_i(x_i)$ and $b_i(x_i)$ are the densities of the i^{th} variable from signal and background classes, respectively. When the statistical dependence is not great, this method is useful since the univariate densities can be readily estimated by simple parametrizations (or by non-parametric methods discussed below). To simplify, one can parametrize the likelihood ratio of the individual variables $L_i = s_i / b_i$ and calculate the discriminant as $D(x) = L/(1+L)$ where $L = \exp \sum L_i$ (see Ref. 24).

3.4 Kernel-based Methods

When the multivariate densities cannot be factorized as above, it is necessary to estimate them to calculate the discriminant function. In principle, multivariate densities can be estimated simply by histogramming the multivariate data $\boldsymbol{x}$ in *M* bins in each of the *d* feature variables. The

fraction of data points that fall within each bin yields a direct estimate of the density at the value of the feature vector ***x***, say at the center of the bin. The bin width (and therefore the number of bins *M*) has to be chosen such that the structure in the density is not washed out (due to too few bins) and the density estimation is not too spiky (due to too many bins). Unfortunately, this method suffers from the curse of dimensionality as in the case of the standard grid search. We would need a huge number of data points in order to fill bins with a sufficient number of points.

More efficient methods for density estimation are based on sampling neighborhoods of data points. Let us take the simple example of a hypercube of side *h* as the kernel function in a *d*-dimensional space. Such a hypercube can be placed at each point $\boldsymbol{x}_n$, counting the number of points that fall within it and dividing that by the volume of the hypercube and the total number of points, i.e.,

$$\tilde{p}(\boldsymbol{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{h^d}H\left(\frac{\boldsymbol{x}-\boldsymbol{x}_n}{h}\right), \tag{18.}$$

where *N* is the total number of points, and *H(u)*=1 if ***x*** is in the hypercube, 0 otherwise.

The method is essentially histogramming, but with overlapping bins (hypercubes) placed around each data point. Smoother and more robust density estimates can be obtained by using smooth functional forms for the kernel function *H*. A common choice is a multivariate Gaussian,

$$H(u) = \frac{1}{(2\pi h)^{d/2}}\exp\left(-\frac{\left\|\boldsymbol{x}-\boldsymbol{x}_n\right\|^2}{2h^2}\right), \tag{19.}$$

where the width of the Gaussian acts as a smoothing parameter, the bandwidth, to be chosen appropriately for the problem. If the kernel functions satisfy,

$$H(u) \geq 0; \int H(u)du = 1, \tag{20.}$$

then, the estimator satisfies $\tilde{p}(x) \geq 0$ and $\int \tilde{p}(\boldsymbol{x})d\boldsymbol{x} = 1$.

In the standard kernel methods, the parameter *h* is the same for all points and consequently the density estimation can be over-smoothed in some regions and spiky in some others. Choosing appropriate width is a critical aspect of this algorithm. This problem is addressed by use of adaptive kernels or the K-nearest neighbor approach.

Adaptive Kernels: The basic idea is to have the kernel width depend on the local density of data points. So we can define the local kernel width $h_i = \lambda_i h$ where *h* is the global width and λ_i is a scaling factor determined by the local density, a simple ansatz being that λ_i is inversely proportional to the square root of the density of sample points in the locality. Even here setting the global width is an issue, especially for multiple dimensions.

K-Nearest Neighbor Method: In this method, a kernel, say a hypersphere, is placed at each point $\boldsymbol{x}$ and instead of fixing the volume *V* of the hypersphere and counting the number of points that fall within it, we vary the volume (i.e., the radius of the hypersphere) until a fixed number of points lie within it. Then, the density is calculated as,

$$\tilde{p}(\boldsymbol{x}) = \frac{K}{NV}. \qquad 21.$$

This estimated density can be used to calculate the discriminant from Equation 7.

The probability density estimation (PDE) method (see for example, Ref. 43) using kernels has been used in both discrimination and regression problems.

3.5 Neural Networks

Feed-forward neural networks, also known as Multilayer Perceptrons (MLP), are the most popular and widely used of the multivariate methods. A schematic of a feed-forward neural network (NN) is shown in **Figure 2**. An MLP consists of an interconnected group of neurons or nodes arranged in layers; each node processes information received by it with an activation (or transformation) function, and passes on the result to the next layer of nodes. The first layer, called the input layer, receives the feature variables, followed by one or more hidden layers of

nodes and the last layer outputs the final response of the network. Each of the interconnections is characterized by a weight, and each of the processing nodes can also have a bias or a threshold. The weights and thresholds are the network parameters, often collectively referred to as weights, whose values are found during the training phase. The activation function is generally a non-linear function that allows for flexible modeling. It has been shown that neural networks with one hidden layer are sufficient to model the posterior probability to arbitrary accuracy. In the schematic shown in **Figure 2** with one hidden layer of nodes and a data set with *d* input feature variables $\boldsymbol{x} \equiv \{x_1, x_2, ...x_d\}$, the output of the network is,

$$O(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{w}) = g(\theta + \sum_j w_j h_j) = p(s \mid \boldsymbol{x}), \tag{22.}$$

where h_j is the output from the hidden nodes,

$$h_j = g(\theta_j + \sum_i w_{ij} x_i). \tag{23.}$$

The non-linear activation function g is commonly taken as a sigmoid

$$g(a) = \frac{1}{1+e^{-a}}. \tag{24.}$$

If $g(a) \sim a$, the outputs h_j at the hidden layer would be linear combinations of the inputs and the network with a single layer of adaptive weights would be a linear model. The logistic sigmoid function is linear close to $a \sim 0$, nonlinear for higher values of *a* and saturates for large values; it maps the input interval $(-\infty, \infty)$ onto (0,1). Therefore, a network with sigmoidal activation function contains a linear model as a special case. The function g is usually chosen to be a logistic sigmoid for classification while for regression it is taken as a linear function. The network parameters are determined by minimizing an empirical risk function, usually the mean square error between the actual output O_p and the desired (target) output y_p,

$$E = \frac{1}{N} \sum_{p=1}^{N} (y_p - O_p)^2 \tag{25.}$$

over all the data in the training sample, where *p* denotes a feature vector.[5] As mentioned in section 2.3, a network trained for signal/background discrimination with y_p=1 for the signal class

[5] Note that Equation 25 is essentially same as Equation 2.

and y_p=0 for the background can directly approximate the Bayesian posterior probability. **Figure 2** (right) shows an example, in a 2-variable feature space, of non-linear decision boundaries obtained with cuts placed on the discriminant (NN output).

There are several heuristics that are helpful in the construction of neural networks. Since the hidden nodes are critical in the modeling of the function, the number needed depends on the density of the underlying data. Too few nodes lead to under-fitting and too many leads to over-fitting. To avoid over-fitting, one can employ *structure stabilization* (optimizing the size of the network) and *regularization.* In the former, one starts with large networks and *prunes* connections or starts with small networks and adds nodes as necessary. In regularization, one penalizes complexity by adding a penalty term to the risk function. It is thought useful to scale the inputs appropriately. The standard advice is to scale the magnitude of the input quantities such that they have mean around zero and a standard deviation of one. Generally, it suffices to make sure that the inputs are not >>1. The starting values of weights are chosen randomly. When using standard scaled inputs as suggested above, the starting weights can be chosen randomly in the range -0.7 to 0.7. A network is trained cycling through the training data hundreds or thousands of times. The performance of the network is periodically tested on a separate set of data. The training is stopped when the error on the test data starts increasing.

3.6 Bayesian Neural Networks

In the conventional methods for training neural networks, one attempts to find a single "best" network, i.e., a single "best" set of network parameters (weights). Bayesian training provides a posterior density for the network weights, $p(\boldsymbol{w} \mid \textit{training data})$. The idea behind Bayesian neural networks (BNN) is to assign a probability density to each point $\boldsymbol{w}$ in the parameter space of the neural network. Then, one performs a weighted average over all points, that is, over all possible networks. Given the training data T=$\{y, \boldsymbol{x}\}$, the probability density assigned to point $\boldsymbol{w}$, that is, to a network, is given by Bayes' theorem

$$p(\boldsymbol{w} \mid T) = \frac{p(T \mid \boldsymbol{w})\, p(\boldsymbol{w})}{p(T)}. \qquad 26.$$

Then, for a given input vector, the posterior distribution of weights will give rise to a distribution over the outputs of the network,

$$\tilde{y}(\boldsymbol{x}) = \int f(\boldsymbol{x}, \boldsymbol{w}) p(\boldsymbol{w} \mid T) d\boldsymbol{w}. \qquad 27.$$

Implementation of Bayesian learning is far from trivial since the dimensionality of the parameter space is typically very large. Currently, the only practical way to perform the high-dimensional integral in Equation 27 is to sample the density $p(\boldsymbol{w} \mid T)$, in some appropriate way, and to approximate the integral using the average

$$\tilde{y}(\boldsymbol{x}) \approx \frac{1}{K} \sum_{k=1}^{K} f(\boldsymbol{x}, \boldsymbol{w}_k), \qquad 28.$$

where K is the number of points $\boldsymbol{w}$ sampled. An algorithm using a Markov Chain Monte Carlo (MCMC) method has been developed and implemented by Neal (44).

There are several advantages to Bayesian neural networks over conventional feed-forward neural networks (45, 46). Each point $\boldsymbol{w}$ corresponds to a different neural network function in the class of possible networks and the average is an average over networks. Therefore, one expects to produce an estimate of the signal class probability $p(s \mid \boldsymbol{x})$ that is less likely to be affected by "over-training." Moreover, in the Bayesian approach, there is less need to severely limit the number of hidden nodes because a low probability density will be assigned to points $\boldsymbol{w}$ that correspond to unnecessarily large networks, in effect, pruning them away. The network can be as large as is computationally feasible so that the class of functions defined by the network parameter space includes a subset with good approximations to the true mapping.

One of the issues in the training of a BNN is to check that the Markov chain has converged. There are many heuristics available. But, in practice, one runs many chains or a single long chain and checks that the results are stable. Also, every Bayesian inference requires the specification of a prior. The choice, in this case, is not obvious. However, a reasonable class to

choose from is the class of Gaussian priors centered at zero that favors smaller rather than larger weights. Smaller weights yield smoother fits to data.

3.7 Decision Trees

Decision trees (46, 47) employ sequential cuts as in the standard grid search to perform the classification (or regression) task, but with a critical difference. At each step in the sequence, the best cut is searched for and used to split the data and this process is continued recursively on the resulting partitions until a given terminal criterion is satisfied. Geometrically the procedure amounts to recursively partitioning the feature space into hypercubic regions or bins with edges aligned with the axes of the feature space. So, essentially, a DT creates M disjoint regions or a d-dimensional histogram with M bins of varying bin-sizes. A response value is assigned to each of these bins. We can assign a value based on which class contributes most to the bin or assign the discriminant $D(\boldsymbol{x}) = s/(s+b)$, where s and b are the signal and background counts in the bin. As the training data set becomes arbitrarily large, and the bin sizes approach zero, the predictions of a DT approaches that of the target function, provided the number of bins also grow arbitrarily large (but at a rate slower than the size of the data set). A DT gives a piece-wise constant approximation to the function being modeled, say, the discriminant $D(\boldsymbol{x})$.

The DT algorithm is applicable to discrimination of n-classes. But, we will keep to the binary decision tree used in 2-class signal/background discrimination. An illustration of a binary decision tree for a problem characterized by two variables and the resulting partition of the feature space is shown in **Figure 3**.

The DT algorithm starts at the so-called *root node,* with the entire training data set containing signal and background events. At each iteration of the algorithm, and for each node, one finds the best cut for each variable and then the best cut overall. The data are split using the best cut thereby forming two branch nodes. One stops splitting when no further reduction in impurity is possible (or the number of events is judged to be too small to proceed further). The measure that is commonly used to quantify impurity is the so called the *Gini* index. The Gini index is given by,

$$Gini = (s+b).P(1-P) = \frac{s.b}{s+b}, \qquad 29.$$

where $P = s/(s+b)$ is the signal purity ($\equiv D(\boldsymbol{x})$, in our definition). The splitting at a branch node is terminated if the impurity after the split is not reduced, and the node then becomes a terminal node or a *leaf* and an output response $D(\boldsymbol{x}) = s/(s+b)$, for example, is assigned to the leaf.

The decision trees are very popular because of the transparency of the procedure and interpretation. They also have some other advantages: (i) tolerance to missing variables in the training data and test data; (ii) insensitivity to irrelevant variables since the best variable on which to cut is chosen at each split and therefore ineffective ones do not get used; (iii) invariance to monotone transformation of variables and hence preprocessing of data is not necessary. But, decision trees also have serious limitations: (i) instability with respect to the training sample (a slightly different training sample can produce a dramatically different tree); (ii) sub-optimal performance due to the piece-wise constant nature of the model, i.e., the predictions are constant within each bin (region represented by a leaf) and discontinuous at its boundaries; (iii) poor global generalization because the recursive splitting results in the use of fewer and fewer training data per bin and only a small fraction of the feature variables might be used to model the predictions for individual bins or leaves.

Most of these limitations, however, have been overcome with the use of ensemble learning techniques such as bagging, boosting or random forests.

3.8 Other Methods

Matrix Element Method:

All of the physics information about a high energy event is contained in the matrix element describing the collision process. The probability to observe data $\boldsymbol{x}$ from a given physics process can be written as

$$p(\boldsymbol{x} \mid process_i) = \frac{1}{\sigma_i}\frac{d\sigma_i}{d\boldsymbol{x}}, \qquad 30.$$

where $d\sigma_i \propto |\mathcal{M}|^2$. Here $d\sigma_i$ is the differential cross-section and $\mathcal{M}$ is the matrix element. The differential cross-section is a convolution of the cross-section for the process, the parton distribution functions (PDFs) and the response function of the detector,

$$\frac{d\sigma}{d\boldsymbol{x}} = \sum_j \int |\mathcal{M}|^2 f(q_1) f(q_2) \xi(\boldsymbol{y}, \boldsymbol{x}) d\boldsymbol{y}. \qquad 31.$$

The sum is over all possible configurations that contribute to the final state, $f(q)$ are the PDFs, $\boldsymbol{y}$ are the partonic variables and $\xi(\boldsymbol{y}, \boldsymbol{x})$ is the response, or transfer, function that gives the probability for partonic variables $\boldsymbol{y}$ to give rise to the observation $\boldsymbol{x}$ in the detector after event reconstruction. The Matrix Element method is a semi-analytical calculation of the probability densities $p(\boldsymbol{x} \mid \boldsymbol{s})$, $p(\boldsymbol{x} \mid \boldsymbol{b})$ from which a discriminant can be computed using Equation 7 in the usual way.

In case of parameter estimation, the event probability is built using

$$P_{event}(\boldsymbol{x}, \theta) = \sum_{i=process} f_i P_i(\boldsymbol{x} \mid \theta), \qquad 32.$$

where the process could be the expected signal and each of the possible backgrounds giving rise to the observed event. One then uses either a Bayesian or maximum likelihood fit to extract the parameters of interest.

The method is computationally very demanding because of the need to perform a multi-dimensional integration for each feature vector.

Genetic Algorithms

While neural networks are inspired by the workings of the human brain, Genetic Algorithms (GA) are inspired by ideas from evolutionary biology and genetics. Genetic algorithms evolve a population of candidate solutions for a problem using principles that mimic those of genetic variation and natural selection, such as crossover, inheritance, mutation, and survival of the

fittest. These algorithms can be used to determine the parameters of a model in functional approximation.

The steps involved in a GA are as follows – (1) randomly generate an initial population of candidate solutions (or parameters $\boldsymbol{w}$) , (2) compute and save the fitness for each individual solution in the current population, (3) generate *n* off-springs of the members of the population by crossover (i.e., swap some of the parameter values between candidate vectors) with some probability and mutate the off-springs with some probability, (4) replace the old population with the new one, which gives the new generation. The procedure is repeated until a set of sufficiently fit candidates have emerged.

Genetic algorithms can be applied to any optimization problem. One such algorithm is Neuroevolution (48), which allows both the NN structure and the NN parameters (weights and thresholds) to be evolved.

3.9 Ensemble Learning

We have discussed several methods to perform functional approximation. The goal is to minimize an appropriate cost function and create approximations that provide best predictive performance and incorporate the correct tradeoff between bias and variance. Bias in a predictor[6] comes from differences between the learned function and the true function, while variance is a measure of the sensitivity of the learned function to inputs. Averaging over multiple predictors has been shown to provide the best compromise between bias and variance, while providing generalization error that can be much smaller than that of an individual predictor. The fundamental insight is that it is possible to build highly effective classifiers from predictors of modest quality.

Here I briefly outline a few of these ensemble techniques (49, 50).

Bagging: Bagging (Bootstrap Aggregating) is a simple average of the outputs of *n predictors*, usually classifiers, where each is trained on a different bootstrap sample (i.e., a randomly selected subset) drawn from a training sample of N events.

[6] A predictor is a discriminant, a classifier or an estimator.

Boosting: The idea behind boosting is to make a sequence of classifiers that work progressively harder on increasingly "difficult" events. Instead of seeking one high performance classifier, one creates an ensemble of classifiers, albeit weak, that collectively have a "boosted" performance. For an ensemble of *M* classifiers, one can write, for the predictions of the final classifier,

$$\tilde{y}(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m y_m(\boldsymbol{x}, \boldsymbol{w_m}), \qquad 33.$$

where $\boldsymbol{w_m}$ the parameters of the m^{th} classifier. The weighting coefficients $\alpha_{\boldsymbol{m}}$ are defined and determined differently in each algorithm. In the case of bagging $\alpha_{\boldsymbol{m}} = 1/M$. In AdaBoost, the first successful high performance boosting algorithm, the underlying functions are decision trees (as is the case for bagging and random forests). $\tilde{y}(\boldsymbol{x})$, in that case, is a boosted decision tree (BDT). The coefficients are taken as $\alpha_{\boldsymbol{m}} = \ln\left[\frac{1-\varepsilon_m}{\varepsilon_m}\right]$ where $\varepsilon_{\boldsymbol{m}}$ is the (event-weighted) mis-classification error for the m^{th} decision tree. The BDTs, unlike single DTs, have been found to be very robust. A striking feature of AdaBoost is that the misclassification rate on the training set approaches zero exponentially as the number of trees increases but the error rate on an independent test sample remains essentially constant. This resistance of the AdaBoost to over-fitting is not yet fully understood.

Random Forests: In principle, this algorithm, like the other two described above, can be applied to any predictors whose construction can incorporate randomization. In practice, however, random forests use decision trees. Many classifiers are trained, each with a randomly chosen subset of feature variables at each split providing a random forest of decision trees. The output for each event is the average output of all trees in the random forest. Further randomization can be introduced through the use of bootstrap samples as in the case of bagging.

3.10 Tools

There are many easy-to-use packages that implement methods discussed above and others. Some of them are specific neural network implementations such as Jetnet (51), MLPFit (52) and FBM (53) for Bayesian Networks. There are general multivariate analysis packages such as TMVA (54) in ROOT and StatPatternRecognition (55) that have many methods implemented. The TMVA software enables the user

to easily try out different methods and compare their efficacies directly. An example is shown in **Figure 4.**

4 ANALYSIS EXAMPLES

Because of their demonstrated power, advanced analysis methods are becoming common tools in several aspects of high energy physics analysis – most notably, in particle identification (electrons, photons, tau-leptons, *b*- jets) and signal/background discrimination.

In this section, I have chosen to discuss briefly a few important physics analyses that illustrate both the potential of the methods and the challenges. I discuss the first precision measurement of the top quark mass at DØ. Then, I discuss the recent observation of the single top quark production which was an important milestone. This observation is important not only because it provides further validation of the SM but because the single top production rate is particularly sensitive to new physics beyond the SM. And, it provides an analysis test-bed for what has become the "holy grail" of particle physics, namely, the search for the Higgs boson. I will make some comments on the Higgs boson searches and end with a brief discussion of an interesting application in fitting the parton distribution functions using neural networks and genetic algorithms.

4.1 An Early Successful Example: The Top Quark Mass

The top quark mass measurement was the first important physics result that benefitted from multivariate methods. The DØ experiment did not have a silicon vertex detector (SVX) during the first Run (Run I) of the Tevatron. Instead, *b*-tagging relied on the presence of soft muons from the decay of *b*-quarks, the efficiency for which was only 20% in the lepton + ≥4-jets channel ($t\bar{t} \to W^+bW^-\bar{b} \to l\nu bq\bar{q}\bar{b}$ process) compared to approximately 53% at CDF which had the ability to tag *b*-jets with its SVX. Nonetheless, in spite of this technical disadvantage, DØ was able to measure the top quark mass with a precision approaching that of CDF, by using multivariate techniques for separating signal and background.

Two multivariate methods, (1) a variant of the likelihood discriminant technique (the LB method) and (2) a feed forward neural network (NN method), were used to compute a

discriminant $D \equiv p(top \mid \boldsymbol{x})$ for each event. A likelihood fit, based on a Bayesian method (56), of the data to discrete sets of signal and background models in the $[\, p(top \mid \boldsymbol{x}), m_{fit}\,]$ plane was used to extract the top quark mass. (m_{fit} is the mass from a kinematic fit to the $t\bar{t}$ hypothesis.) The distributions of variables and the discriminants are shown in **Figure 5**. Combining the results of the fits from the two methods, DØ measured $m_t = 173.3 \pm 5.6(stat) \pm 5.5(syst) GeV/c^2$ (24), which was a factor of two better than the result obtained using conventional methods. This example underscores that even very early in the life of an experiment, huge gains can be had through a judicious, but advanced, treatment of a few simple variables.

Most of the measurements of the top quark mass at CDF and DØ, since this first successful application of a multivariate approach, have used some kind of multivariate method – neural networks, matrix element or likelihood, etc. The current measured world average top quark mass is $m_t = 173.1 \pm 1.3\, GeV/c^2$ (57).

4.2 Single Top Quark Production at the Tevatron

The top quark was discovered in 1995 through the pair production process $p\bar{p} \to t\bar{t}$ via the strong interaction. The SM predicts electroweak production of a single top quark along with a *b*-quark or a *b*- and a light quark with a cross section $\sigma_t \sim 3$ pb ($\sigma_{t\bar{t}} \sim 6.8$ pb, assuming $m_t = 175$ GeV/c^2). While the top quark discovery was in hand with data sets corresponding to an integrated luminosity of ~ 50 pb^{-1}, the single top quark observation required about 50 - 60 times more luminosity (DØ:2.3 fb^{-1}, CDF:3.2 fb^{-1}) and came fourteen years later (58, 59). What makes single top quark events so extremely difficult to extract from data is the fact that the final state contains fewer features than in $t\bar{t}$ to exploit for the purpose of discriminating signal from the overwhelming background of W+jets and QCD multijet production (wherein a jet is misidentified as a lepton). The use of multivariate methods was indispensable in the analyses in both experiments.

Single top quarks are produced at the Tevatron through the s-channel $q\bar{q} \to t\bar{b}$ ($\sigma \sim 0.95$ pb) and t-channel $q'g \to tqb$ ($\sigma \sim 2.05$ pb) processes (60). The top quark decays to a *W* boson and a *b*-

quark nearly 100% of the time (as per the SM). Final state channels involving leptonic decays of the *W* boson and at least one *b*-tagged jet are considered by both experiments, in order to have better signal to background ratio from the outset. Both experiments use neural networks to enhance the *b*-tag efficiency and purity.

After initial selection criteria, requiring a high p_T lepton and and high p_T jets, and large missing transverse energy, both experiments estimate a very similar overall signal to background ratio, *s/b* ~ 0.05, (CDF: 0.053, DØ: 0.048). CDF observes 4,726 events while expecting 4780±28 background and 255 ± 21 signal events, while DØ observes 4,519 events with an expected background of 4651±234 and signal of 223 ± 30 events. At this point in the analysis, the signal, in both cases, is smaller than the uncertainties in the background estimates.

The single top signal is further discriminated from the backgrounds using many multivariate techniques. DØ performs three independent analyses using (1) Bayesian Neural Networks (BNN), the first such application in HEP, (2) Boosted Decision Trees (BDT) and (3) the Matrix Element (ME) method, while CDF in addition to these methods also uses the likelihood discriminant method. Since the results from these methods are not completely correlated, the discriminant outputs are further combined into a single discriminant (called the Combination Discriminant by DØ, and the Super Discriminant by CDF). The final discriminant is then used to extract the cross section for single top quark production and the signal significance. The signal to background ratio in the signal region of the final discriminants, *s/b* >5, is about a factor of 100 larger with respect to the *s/b* in the base samples. The cross sections are measured to be 2.3±0.5 pb by CDF (at m_t=175 GeV/c^2) and 3.94±0.88 pb by DØ (at m_t = 170 GeV/c^2), using the final discriminants and a Bayesian technique. The significance of the signal is 5.0 standard deviations in both results.

The analyses, depending on the channel, use anywhere from 14 up to 100 variables. In order to ensure that the background is modeled correctly, both experiments compared thousands of distributions of the data sample with the modeled backgrounds. The output discriminant modeling was also verified at various stages with control samples from known physics processes.

4.3 Searches for the Higgs Boson

The Higgs boson has been the most sought after particle in the past decade and a half. The intense searches by the four experiments (ALEPH, DELPHI, L3 and OPAL) at the e^+e^- collider LEP at CERN ($\sqrt{s}$ = 189 - 209 GeV) before it was decommissioned, resulted in 95% Confidence Level (C.L.) lower bound on the Higgs boson mass of 114.4 GeV/c^2 (61-63). In 2000, studies of the Higgs discovery reach at the Tevatron (64, 65) led to the conclusion that the use of multivariate methods can significantly enhance the potential for its discovery at the Tevatron with the planned upgrades for Run II. The Tevatron experiments, have, with the help of several fb^{-1} of data accumulated and with the help of advanced analysis techniques, reached the sensitivity levels to find hints or to exclude certain masses beyond the range of LEP exclusion.

The predicted cross sections for the production of SM Higgs at the Tevatron are more than an order of magnitude smaller than for single top in the mass regions of interest. The dominant production process at the Tevatron is $gg \to H$, with cross sections between 1 pb and 0.2 pb in the mass range of 100-200 GeV/c^2. The cross sections are between 0.5 pb and 0.03 pb for $q\bar{q}' \to WH$ or ZH and 0.1 pb – 0.02 pb for $q\bar{q} \to q\bar{q}H$ in the same mass range. The dominant decay channels are $H \to b\bar{b}$ for m_H < 135 GeV/c^2 and H → WW* for m_H > 135 GeV/c^2 (W* is off-shell if m_H < 160 GeV/c^2). The $gg \to H \to b\bar{b}$ channel suffers from very large QCD multijet background. Therefore, for m_H < 135 GeV/c^2, the *WH* and *ZH* production channels are used for the searches. For m_H > 135 GeV/c^2, gg→ H → WW* is the most promising channel.

The searches or the SM Higgs boson have been done in 90 mutually exclusive final states (36 for CDF and 54 for DØ). The analysis channels are sub-divided based on lepton-type, number of jets and number of *b*-tags. The most important aspects that can help discriminate Higgs signal from background are efficient *b*-tagging and good dijet mass resolution (in low mass Higgs searches). To achieve high *b*-tag efficiency, both experiments use a neural network to combine outputs of simpler discriminants based on secondary vertex and decay track and jet information. CDF constructs two separate networks to discriminate *b*-jets from c-quarks jets and *b*-jets from light-quark jets. DØ builds an NN *b*-tagger to discriminate *b*-jets from all other types of jets. The DØ NN *b*-tagger gives significantly higher efficiencies compared to that of the next best method based on the JLIP (Jet Lifetime Probability) algorithm (66). It has been estimated that

the benefit of the *NN* tagger is equivalent to a doubling of the luminosity (67) in SM Higgs boson searches. CDF has also developed a multivariate approach for *b*-jet energy correction and demonstrated improved di-jet mass resolution that in turn helps Higgs search sensitivity (68).

Both CDF and DØ use neural networks, boosted decision trees and other multivariate discriminants in all analyses. CDF finds in the case of $H \to WW^*$ analysis, that the multivariate techniques provide a gain factor of 1.7-2.5 (depending on m_H) in effective integrated luminosity over an optimized cut-based selection. Some example NN discriminants are shown in **Figure 6**. The combined results from the two experiments provide 95% C.L. upper limits on Higgs boson production that are a factor of 2.7 (0.94) times the SM cross-section for $m_H = 115(165) GeV/c^2$. The combination of results from the two experiments has, as of December 2009, using data sets of luminosities of up to 5.2 fb^{-1}, yielded a 95% C.L. exclusion for a SM Higgs for 163 $GeV/c^2 < m_H < 166$ GeV/c^2 (69-71).

4.4 Determination of Parton Distribution Functions

One of the exciting applications of multivariate methods is in the parametrization of parton distribution functions with neural networks by the *NNPDF* collaboration (72). A parton distribution function (PDF) is the probability density of finding a parton (a quark, an antiquark or a gluon) inside a hadron with a certain fraction *x* of the hadron's longitudinal momentum at momentum transfer Q^2. The PDFs are essential inputs in making predictions for the SM and beyond the SM physics processes at hadron colliders. The PDFs are determined by fitting the theoretical predictions to various sets of experimental measurements, primarily from deep inelastic scattering of leptons on hadrons (or nuclei). The Tevatron experiments have produced numerous results on a variety of hard interaction processes providing precision tests of the SM akin to the LEP and SLC electroweak measurements. The tests of these results as well as predictions for searches beyond the SM demand very precise determination of the PDFs. The PDF uncertainties are sometimes the dominating uncertainties, and it is, therefore, important to have reliable estimates of these uncertainties.

The standard approach to fitting PDFs is to assume a specific parameterized functional form for the PDFs $f(x,Q_0^2) = x^{\alpha}(1-x)^{\beta}P(x)$ and determine the parameters and the associated errors from a fit to the data by minimizing χ^2. The choice of a specific functional form, as we have discussed, results in an inflexible model that introduces unnecessary systematic errors (bias in the region of sparse or no data) and uncertainties that are surely underestimated. One way to build more flexible models for PDFs is to rely on the fact that neural networks are universal approximators.

In order to train the neural networks that model the PDFs, an ensemble of Monte Carlo (MC) data-sets which are "replicas" of the original experimental data points are generated. The MC data sets have points that are Gaussian distributed about the experimental data points, with errors and covariance equal to the corresponding measured quantities. The MC set thus gives a sampling of the probability distribution of the experimental data. The *NN* architecture uses two inputs (x and $\log x$), two hidden layers with two neurons each, and one output, $f(x,Q_0^2)$ at a reference scale Q_0^2. Generic Algorithms are used for optimization, yielding a set of *NN* parameters for each replica. The mean value of the parton distribution at the starting scale for a given ***x*** is found by averaging over all the networks and the uncertainty is given by the variance of the values. The errors on the *PDF* s from the *NNPDF* fits are larger than those from other global fitting methods, possibly indicating that the latter have underestimated the errors, as noted earlier.

5 OPEN ISSUES

Over the past two decades, a lot of experience has been gained in the use of advanced multivariate analysis methods in particle physics and spectacular results have been obtained because of their use. However, there are still some important open issues which I outline below.

- **Choosing the Variables:** How do we choose the best set of feature variables so that no more than a prescribed amount of information is lost? Even though ranking the efficacy of individual variables for a given application is straightforward, the best way to decide which combination of variables to use can only be done, currently, by evaluating the performance of different sets in the given application.

- **Choosing a Method:** The "No free lunch theorem" states that there is no one method that is superior to all others for all problems, which prompts the question: is there a way to decide which method is best for which problem? Here, again, one needs to try out different methods for a given application and compare performance. In general, however, one can expect Bayesian neural networks, boosted decision trees and random forests to provide excellent performance over wide range of problems.
- **Optimal Learning:** How can one test convergence of training i.e., know when the training cannot be improved further? The practice is to stop training when prediction error on an independent test data set begins to increase. But, how can one verify that a discriminant is close to the Bayes limit?
- **Testing the Procedures:** For complicated analyses, with lots of input variables, and small signals, it is necessary to validate the procedure itself or, in fact, the whole chain of analysis. But, since this is computationally demanding, are there alternative and reliable methods of validation? If not, it is important that an algorithm be computationally efficient so that an analysis can be repeated for many scenarios to ensure the robustness of the results.
- **Modeling of Backgrounds:** By far, the most important issue of any non-trivial analysis is how to ensure the correctness of modeling of backgrounds (and signal) in the training data.

  However good a learning method is, if the training data are faulty, the results will be unreliable. When we use a large number of variables, how do we verify the modeling? How many arbitrary functions of the variables do we need to check? Say, we use 100 variables in a multivariate analysis, how can we check the modeling of the 100-dimensional density? The larger the number of input variables used, the higher is the burden of verifying the correctness of the modeling. In simple applications such as in particle identification, data from well-understood physics processes can be used to cross-check results. But, in discriminating new signals from very large backgrounds, the task of verifying a multivariate density in high dimensions is a daunting one. The number of combinations of variables and functions thereof that one needs to check grows rapidly with the number of feature variables used. In fact, only an infinitely large number of

arbitrary functions can guarantee that all correlations have been verified. But, the practical question is how many or what checks are needed to achieve a specified level of confidence in the validity of the results?

6 SUMMARY & PROSPECTS

Advanced analysis methods that match the sophistication of the instruments used in high energy physics research and meet the challenges that vast data sets and extremely rare signals impose are imperative. The field already has several high profile results that simply would not have been possible without such methods. Clearly, there is no going back!

In this article, I have provided an overview, with a unified perspective, of the concepts and methods of optimal analysis. I have discussed a range of methods: from the simple to the sophisticated, in particular, those that make use of multivariate universal approximators. I have discussed some useful heuristics and outlined open issues. I have presented a few examples of successful applications of these methods in the past decade and a half. There are other examples from the Tevatron, as well as from LEP, HERA (73), the b-factories (74) and neutrino experiments (75).

The LHC experiments (76) are planning to use advanced methods in many analyses. But, there is some concern about whether their use in the early data-taking period is appropriate due to the expected lack of good understanding of the detectors and systematic effects. These are valid concerns. Nevertheless, there are ample opportunities for using advanced methods safely:

- Where it is possible to ascertain the correctness of modeling using well known physics processes such as Z boson decays, QCD $b\bar{b}$ events, etc.
- When one has arrived at a set, albeit small, of well understood variables.

Moreover, the following points should be kept in mind:

- Even two or three variables treated in a multivariate manner can provide significant gains over cuts applied to the variables directly.
- Combining simple classifiers based on a few variables can help cross check the modeling more easily and significantly boost the final performance and precision of the results.

- One can make use of the available easy-to-use analysis kits to try two or more methods to ensure that there are no bugs in the procedure or bias due to possible incorrect use of a method. For example, one could use a feed-forward neural network, Bayesian neural network and boosted decision trees and check the consistency of the results.
- One can use data as the background model in channels where signal to background ratio is initially very small. One advantage of this approach is that the data (necessarily) models both physics and instrumental backgrounds.

The bar for the quality of the analyses, especially when a potential discovery is at stake, should be (and almost certainly will be) set very high. The advanced methods I have described need to be used in every step of the data analysis chain, if possible, to reap maximum benefits. But, as is true of all scientific methods and tools, these methods should be used with a great deal of diligence and thought. We would be well served to follow the principle of Occam's razor, which in this context can be stated thus: if we have two analyses of comparable quality we should choose the simpler one. I am sure Einstein would agree.

ACKNOWLEDGMENTS

My research is supported in part by the U.S. Department of Energy under contract number DE-AC02-07CH11359. I would like to thank my DØ and CDF colleagues for extensive work on the applications of the advanced analysis methods over the years. My special thanks to Harrison Prosper, Serban Protopopescu, Mark Strovink and Scott Snyder for delightful collaboration on the early multivariate analyses at DØ. I would like to thank Paul Grannis, Dan Green, Rob Roser and Harrison Prosper for reading the manuscript and providing very useful feedback. I thank Chandra Bhat and Shreyas Bhat for useful discussions and comments and Jenna Caymaz for preparing schematic diagrams in figures 2 and 3.

LITERATURE CITED

1. Glashow S. *Nucl. Phys.* 22:579 (1961)
2. Weinberg S. *Phys. Rev. Lett.* 19:1264 (1967)
3. Salam A. *Elementary Particle Physics,* Almquvist and Wiksells, Stockholm, (1968)
4. Glashow S, Iliopoulos J, and Maiani L. *Phys. Rev.* D2:1285 (1970)
5. Gross D, Wilczek F. *Phys. Rev.* D8: 3633 (1973); *ibid., Phys. Rev. Lett.* 30:1343 (1973)
6. Politzer HD. *Phys. Rev. Lett.* 30:1346 (1973)
7. CDF Collaboration (Abe F, et al.), *Phys. Rev. Lett.* 74:2626 (1995); CDF Collaboration (Abe F, *et al.*), Phys. Rev. D50:2966 (1994)
8. DØ Collaboration (Abachi S, et al.), *Phys. Rev. Lett.* 74:2632 (1995)
9. DONUT Collaboration (Kodama K, et al.), *Phys.Lett.*B504:218 (2001)
10. Higgs PW. *Phys. Lett.* 12:132 (1964)
11. Higgs PW. *Phys. Rev. Lett.* 13:508 (1964)
12. Guralnik GS, Hagen CR and Kibble TWB. *Phys. Rev. Lett.* 13:585 (1964)
13. Anderson PW. *Phys. Rev.* 130:439 (1963)
14. Wimpenny SJ, Winer BL. *Annu. Rev. Nucl. Part. Sci.* 46:149 (1996); Campagnari C, Franklin M. *Rev. Mod. Phys.* 69:137(1997); Bhat PC, Prosper HB, Snyder SS. *Int. J. Mod. Phys. A* 13: 5113 (1998)
15. Bhat PC, Spalding WJ. *Proc. 15th Topical Conf. on Hadron Collider Physics*, East Lansing, Michigan, 2004, *AIP Conf.Proc.*753:30 (2005)
16. DØ Collaboration (Abazov VM et al.), *Phys. Rev. Lett.* 103:092001 (2009)
17. CDF Collaboration (Aaltonen T et al.), *Phys. Rev. Lett.* 103:092002 (2009)

18. *The Large Hadron Collider*, http://public.web.cern.ch/public/en/lhc/lhc-en.html

19. Denby B. *Comp. Phys. Comm.* 49:429 (1988)

20. Lönnblad L, Peterson C, Rögnvaldsson T. *Phys. Rev. Lett*. 65:1321 (1990)

21. Bhat PC, Lonnblad L, Meier K, Sugano K. *Research Directions for the Decade: Proc. of 1990 Summer Study on High Energy Physics*: Snowmass, CO, p. 168 (1990)

22. Bhat PC (for the DØ Collaboration), *Proc. Meeting of the American Physical Society, Division of Particles and Fields*, Albuquerque, NM, p. 705 (1994)

23. Bhat PC (for the DØ Collaboration), *Proc. pbar-p Collider Workshop, AIP Conf. Proc*.357:308, (1996) e-Print: hep-ex/9507007

24. DØ Collaboration (Abbott B, et al.), *Phys. Rev. Lett.* 79:1197 (1997); *ibid Phys.Rev*. D58: 052001 (1998)

25. Bhat PC et al. DØ Note 3061, unpublished (1997)

26. DØ Collaboration (Abbott B, et al.), *Phys. Rev. Lett.*, 80:2051 (1998); *ibid Phys. Rev. Lett.* 79:4321 (1997)

27. DØ Collaboration (Abbott B, et al.), *Phys. Rev. Lett*. 83:1908 (1999)

28. Bhat PC. *Proc. Mtg. American Physical Society, Division of Particles and Fields*, Columbus, OH, 2000, *Int. J. Mod. Phys*.A16S1C:1122 (2001); Bhat PC. *Proc. Int. Workshop on Adv. Comp. Anal. Tech, in Phys. Res.* (ACAT 2000), Batavia, IL, 2000, *AIP Conf. Proc*. 583, p. 22 (2001)

29. McNamara PA III, Wu SL. *Rep. Prog. Phys*. 65:465,(2002), and references therein.

30. Fisher R. *Annals of Eugenics*, 7:179 (1936)

31. Rosenblatt F. *Psych. Rev*. 65:386 (1958)

32. Levenberg K. *The Qtrly. Appl. Math.* 2:164 (1944); Marquardt D. *SIAM J. Appl. Math.* 11:431 (1963)

33. Kirkpatrick S, Gelatt CD, Vecchi MP. *Science. New Series* 220 (4598):671 (1983)

34. Goldberg DG. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison Wesley, (1989)

35. O'Hagan A. *Kendall's Advance Theory of Statistics: Volume 2B, Bayesian Inference,* Oxford University Press, New York (2002)

36. Bishop CM. *Neural Networks for Pattern Recognition,* Oxford University Press (1995)

37. Bishop CM. *Pattern Recognition and Machine Learning,* pp. 738. New York: Springer Science+Business Media (2007)

38. Vapnik VN. *Statistical Learning Theory*, York: Springer-Verlag (2000)

39. Duda RD, Hart PE, Stork DG. *Pattern Recognition*, pp. 654. United States of America: Wiley Interscience (2000)

40. Hastie T, Tibshirani R, Friedman JH. *The elements of statistical learning : Data mining, inference, and prediction*, pp. 533, New York: Springer Verlag (2001)

41. Amos NA, Stewart C, Bhat PC, Cretsinger C et al. *Proc. Int. Conf. Computing in High Energy Physics* (CHEP 95), Rio de Janeiro, Brazil, p. 215 (1995)

42. Engelman R, et al. *Nucl. Inst. Meth*. 216:45 (1983); Raja R, DØ Note 1192, unpublished (1991)

43. Holmstrom L, Sain R, Miettinen HE. *Comput. Phys. Commun*. 88:195 (1995)

44. Neal RM. *Bayesian Learning for Neural Networks* (Lecture Notes in Statistics), New York: Springer Verlag (1996)

45. Bhat PC, Prosper HB. *Proc. PHYSTATO5: Statistical Problems in Particle Physics, Astrophysics and Cosmology*, Oxford, England, UK, Imperial College Press, p. 151 (2005); Prosper HB, *Proc. Adv. Comput. Anal. Tech.*, Erice, Italy, *PoS(ACAT08)010* (2008)

46. Brieman L, Friedman J, Olshen R, Stone C. *Classification and Regression Trees,* Wadsworth, (1984)

47. Roe BP et al. *Nucl. Inst. Meth. in Physics Research A* 543:577(2005)

48. Yao X. *Proc. IEEE* 87:1423 (1999)

49. Brieman L, *Machine Learning*, 26:123(1996); *ibid* 45:5 (2001)

50. Freund Y, Schapire RE. *J. Comput. Sys. Sci*. 55:119 (1997); Friedman J, Hastie T, Tibshirani R. *Annals of Stat.* 28:337 (2000)

51. Peterson C, Rögnvaldsson T. *JETNET 3.0—A Versatile Artifical Neural Network Package*, Rep. No. CERNTH. 7135/94 (1994)

52. Schwindling J. http://schwind.home.cern.ch/schwind/MLPfit.html (2000)

53. Neal R. http://www.cs.toronto.edu/~radford/fbm.software.html (2004)

54. Hoecker A, Speckmayer P, Stelzer J, Therhaag J, vonToerne E, Voss H. *TMVA - Toolkit for Multivariate Data Analysis*, Rep. No. CERN-OPEN-2007-007 TMVA version 4.0.1 (2009)

55. Narsky I. arXiv:physics/0507143v1 (2005)

56. Bhat. PC, Prosper HB, Snyder SS. *Phys. Lett.* B407:73 (1997)

57. The Tevatron Electroweak Group, http://arxiv.org/pdf/0903.2503 (2009)

58. DØ Collaboration (V.M. Abazov, et al.) *Phys. Rev. Lett*.103:092001 (2009); *ibid*, *Phys. Rev.* D78:012005 (2008)

59. CDF collaboration (Aaltonen T et al.) *Phys. Rev. Lett*. 103:092002 (2009)

60. Harris BW et al. *Phys. Rev. D*. 66:054024 (2002)

61. Barate R et al. (LEP Higgs Working Group), *Phys. Lett*. B565:61 (2003); ALEPH Collaboration (Barate R et al.) *Phys. Lett.* 495:1 (2000), OPAL Collaboration, (Abbiendi G) *Phys. Lett. B*499:38 (2001); L3 Collaboration, (Achard P) *Phys. Lett. B*517:319 (2001); DELPHI Collaboration, (Abreu P) *Eur. Phys. J. C*17:187 (2000)

62. Bhat PC, *Proc. Int. Workshop on Adv. Comp. and Anal. Tech. Phys. Res.* Moscow, Russia, 2002, *Nucl. Inst. And Meth. in Phys. Research* A 502:327 (2003)

63. Kado & Tully CG. *Annu. Rev. Nucl. Part. Sci*. 52:65 (2002)

64. Bhat PC, Gilmartin R, Prosper HB. *Phys. Rev. D*62:074022 (2000)

65. Carena M et al., e-Print: hep-ph/0010338 (2000)

66. Scanlon T. *Ph.D. Thesis*, Imperial College, London, UK (2006); DØ Collaboration. arXiv:1002.4224v1 [hep-ex]

67. DØ Reference Place Holder

68. CDF Reference Place Holder

69. CDF and the DØ Collaborations, *Phys. Rev. Lett*. 104:061802 (2010)

70. CDF Collaboration, *Phys. Rev. Lett*. 104:061803 (2010)

71. DØ Collaboration, *Phys. Rev. Lett*. 104:061804 (2010), http://www-d0.fnal.gov/ DØ Note 6008-CONF (2009)

72. NNPDF Collaboration, (Ball RD, et al.) *Nucl. Phys. B* 809:1 (2009)

73. Kiesling C et al. *Proc. Int. Workshop on Adv. Comp. Anal. Tech, in Phys. Res.* (ACAT 2000), Batavia, IL, 2000, *AIP Conf. Proc*. 583, p. 36 (2001)

74. Babar Collaboration, (Aubert B, et al.) *Phys. Rev. Lett.* 99:021603 (2007); *ibid*, 87:091801 (2001)

75. Shaevitz MH, et al. (for MiniBooNE Collaboration) *J. Phys.: Conf. Ser.* 120:052003 (2008); Yang HJ, Roe BP, Zhu J. *Nucl. Inst. Meth A*555:370 (2005);MINOS Collaboration (Adamson P et al.) *Phys.Rev.Lett*. 103:261802 (2009)

76. http://cms.web.cern.ch/cms/ ; http://atlas.ch/

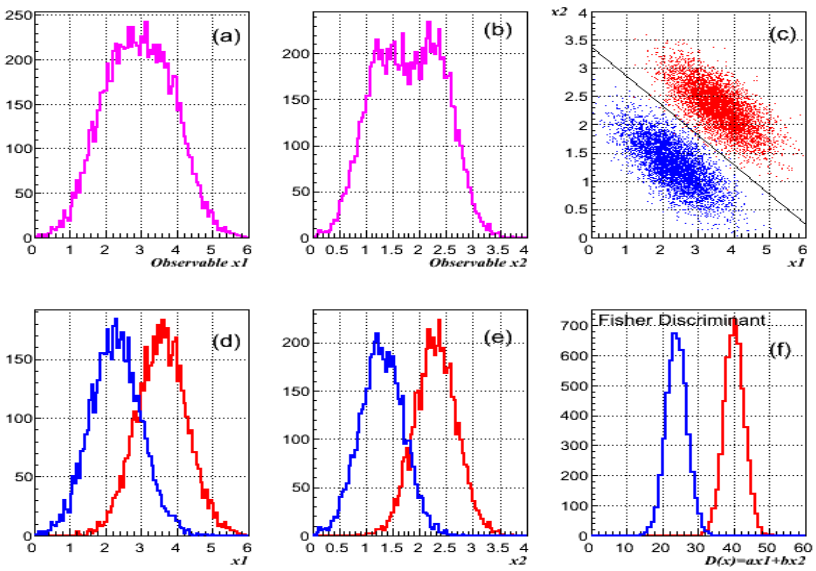


Figure 1 (a,b) Distributions of two hypothetical observables x1 and x2 arising from a mixture of two classes with bivariate Gaussian densities; (c) bivariate densities of the two classes (d,e) 1D marginalized densities and (f) a linear discriminant function f(x1,x2) that reveals two distinct distributions. An optimal cut placed on the discriminant results in the linear decision boundary shown in (c).

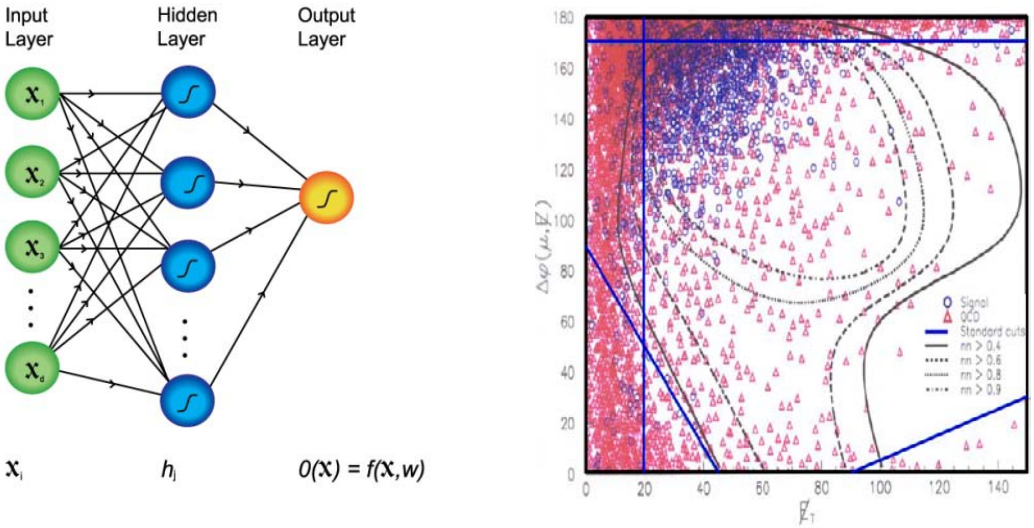


Figure 2 (Left) A schematic representation of a three-layer feed-forward neural network. (Right) Discrimination in two-variable space using a neural network: the contours are the decision boundaries corresponding to network output values indicated in the legend.

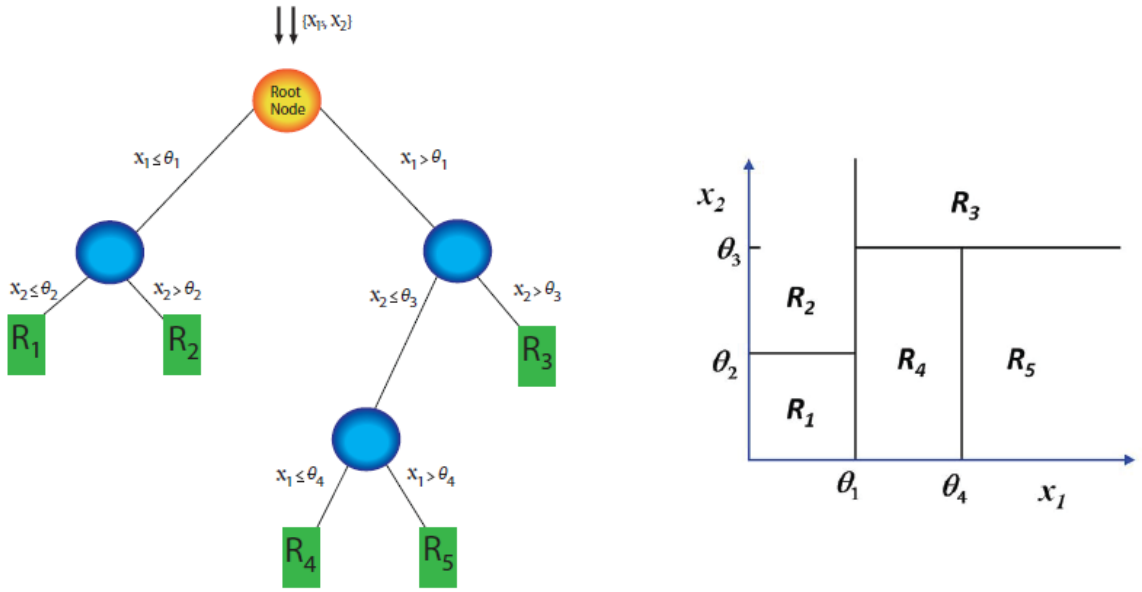


Figure 3 (Left) A schematic of a binary decision tree using two variables x_1 and x_2 and (Right) an illustration of the corresponding partitions of the 2D input space (see text for details).

.

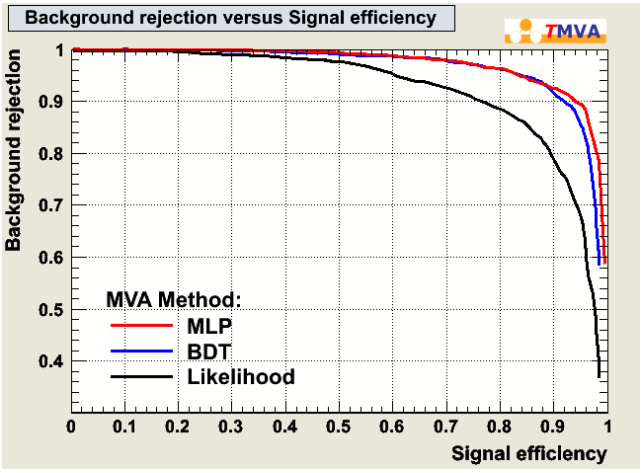


Figure 4. Comparison of the performance of neural networks (MLP), boosted decision tree (BDT) and Likelihood discriminant using TMVA in ROOT in an example discrimination problem. (Hypothetical signal: First generation scalar leptoquark pair events with $m_{LQ}=240$ GeV/c^2 , and background: top-antitop events.)

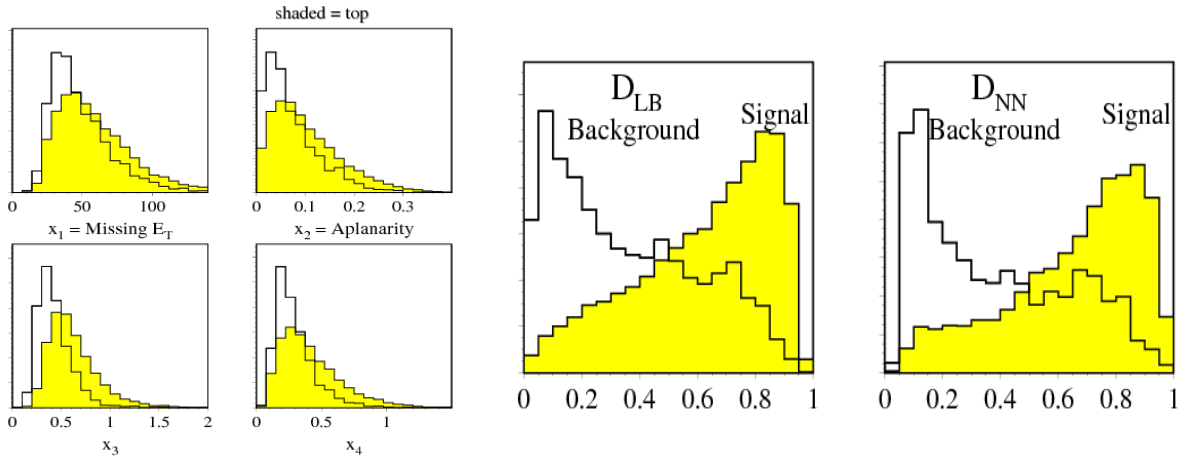


Figure 5 (Left) Distributions of discriminant variables x_1, x_2 , x_3 , x_4 (see Ref. 24 for definitions) used in the first direct precision measurement of the top quark mass at DØ and (right) the distributions of the final multivariate discriminants. The filled histograms are for signal and unfilled ones are for background. All histogram areas are normalized to unity.

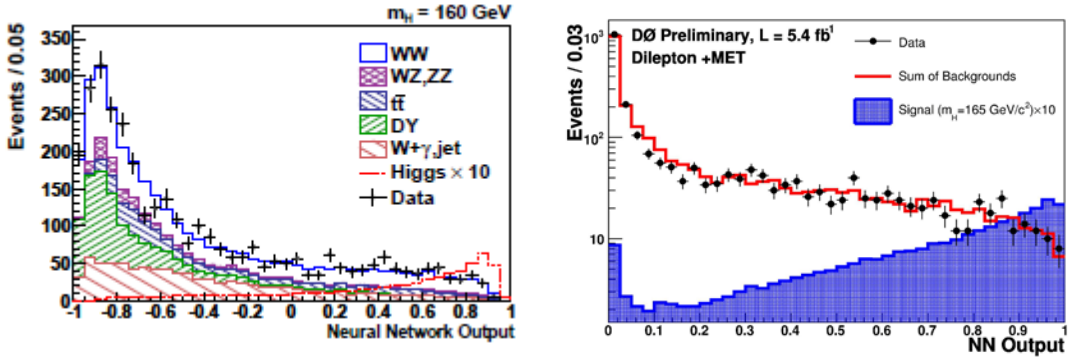


Figure 6. Neural network output distributions from H→ WW* analyses at the Tevatron. (Left) CDF results showing data compared with total and individual backgrounds. Also shown is the expected distribution for the SM Higgs signal for $m_H = 160$ GeV/c^2. (Right) DØ results comparing data with total background in the dilepton + missing transverse energy channel. Here, the Higgs signal distribution is shown for $m_H = 165$ GeV/c^2. In both cases, the signal is scaled up by a factor of ten relative to the SM prediction.

FIGURE CAPTIONS:

Figure 1 (a,b) Distributions of two hypothetical observables x1 and x2 arising from a mixture of two classes with bivariate Gaussian densities; (c) bivariate densities of the two classes (d,e) 1D marginalized densities and (f) a linear discriminant function f(x1,x2) that reveals two distinct distributions. An optimal cut placed on the discriminant results in the linear decision boundary shown in (c).

Figure 2 (Left) A schematic representation of a three-layer feed-forward neural network. (Right) Discrimination in two-variable space using a neural network: the contours are the decision boundaries corresponding to network output values indicated in the legend.

Figure 3 (Left) A schematic of a binary decision tree using two variables x_1 and x_2 and (Right) an illustration of the corresponding partitions of the 2D input space (see text for details).

Figure 4. Comparison of the performance of neural networks (MLP), boosted decision tree (BDT) and Likelihood discriminant using TMVA in ROOT in an example discrimination problem. (Hypothetical signal: First generation scalar leptoquark pair events with $m_{LQ}=240$ GeV/c^2 , and background: top-antitop events.)

Figure 5 (Left) Distributions of discriminant variables x_1, x_2 , x_3 , x_4 (see Ref. 24 for definitions) used in the first direct precision measurement of the top quark mass at DØ and (right) the distributions of the final multivariate discriminants. The filled histograms are for signal and unfilled ones are for background. All histogram areas are normalized to unity.

Figure 6. Neural network output distributions from H→ WW* analyses at the Tevatron. (Left) CDF results showing data compared with total and individual backgrounds. Also shown is the expected distribution for the SM Higgs signal for m_H = 160 GeV/c^2. (Right) DØ results comparing data with total background in the dilepton + missing transverse energy channel. Here, the Higgs signal distribution is shown for m_H = 165 GeV/c^2. In both cases, the signal is scaled up by a factor of ten relative to the SM prediction.