Generalizing multivariate analysis

Information Theory, Statistical Learning Theory, & Projection Pursuit

Kyle Cranmer
University of Wisconsin-Madison

June 5, 2003
Cracow School of Theoretical Physics

Outline:

- Introduction
- Three Points of View
- Projection Pursuit
- Examples
- Conclusions

Multivariate Analysis is becoming an increasingly popular tool within High Energy Physics.

The techniques are borrowed from a variety of different fields, which leads to an incoherent picture.

In this talk I will try to outline some general considerations and clarify the different approaches.

Consider the search for a new particle.

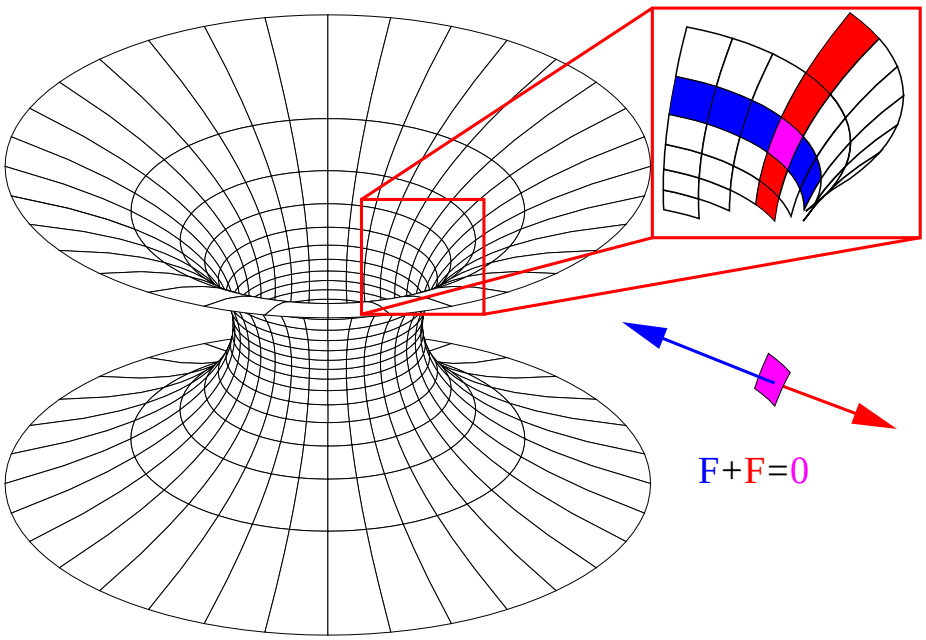
Goal	Type of Problem	Local/Global
Discover a new particle	Statistical	Global
Select candidate events	Classification	Local

These two problems are similar but distinct.

Define:

- Local = Decision based on a single event
- Global = Decision based on entire data set

Consider a Soap film stretched between two rings, there are two approaches to the solution:



Local: The total forces cancel, thus the curvature in the two (appropriate) orthogonal directions should be equal and opposite.

Global: The surface wishes to take the shape with the least area because that minimizes the stored energy.

These two approaches are equivalent. The Variational Problem $\frac{\delta A[f]}{\delta f} = 0$ leads to a local condition. *E.g.* Minimal Surfaces have Zero Mean Curvature everywhere.

We shall find similar equivalences in the Statistical and Classification problems which follow.

What is the *classification* problem?

Consider an *event* with some data $x \in I$ which can be associated with one of a set of classes $\mathcal{C}$.

Based only on x we wish to determine to which $i \in \mathcal{C}$ the event belongs.

In addition we may be able to estimate the conditional probabilities $P(x|i)$ from independent observations, theory, or Monte Carlo.

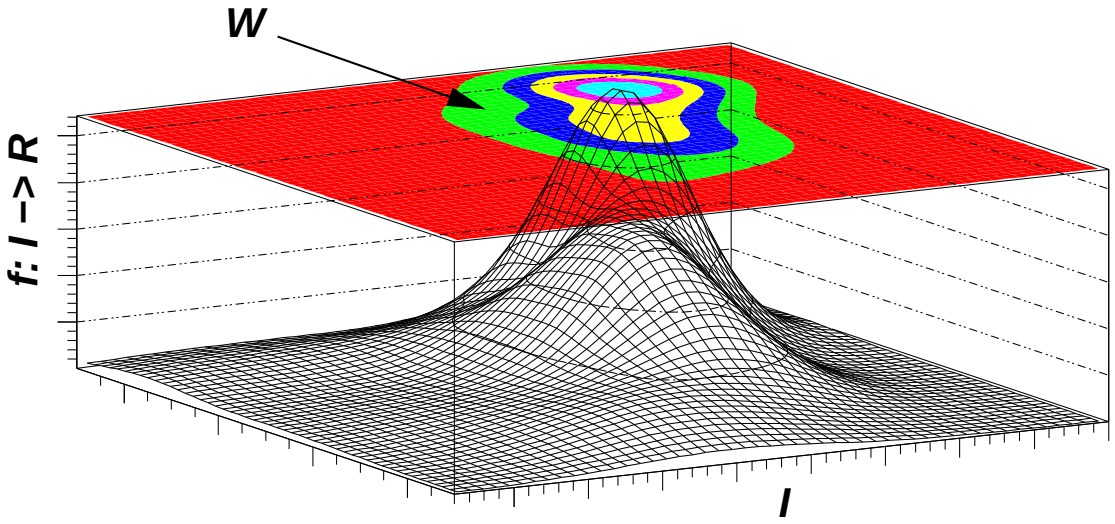
Note: for a given x, there may be several $i \in \mathcal{C}$ such that $P(x|i) > 0$. *I.e.* The classes are not disjoint.

Often in Particle Physics we are more interested in Hypothesis testing or Parameter Estimation.

Consider a set of N *observations* which make up some data set $\{x_i \in I\}$ with $i \in \{0 \ldots N\}$. Also consider a set of *Hypotheses* $\{H_j\}$ which predict the probabilities of observing x viz. $P(x|H_j)$.

Based only on $\{x_i\}$ we wish to reject or accept one of the Hypotheses with some confidence.

For both problems, we are essentially interested in defining some *Critical Region* $W \subset I$ such that if $x \in W$ we accept the event as a candidate event or associate it with some class.



It is very common that this critical region is a contour of some function $f : I \to \mathbb{R}$

Note: The function f is a local concept,
but the critical region W is a global concept.

Essentially Multivariate Analysis is a family of techniques which produce some function f on a multi-dimensional space.

The different approaches to multivariate analysis correspond to:

- techniques to construct the function f from training data
- properties of an abstract f necessary or sufficient to satisfy some condition
- limits on the performance of f
- behavior of f in the presence of uncertainty, noise, or error

These approaches include:

- Statistics
- Artificial Intellegence & Statistical Learning Theory
- Signal Processing & Information Theory

The Statistical Point of View

Consider a single Hypothesis H from which we can predict the probabilities of observing x viz. $P(x|H)$.

In 1925 Fisher developed the idea of a *pure signficance test* based on a set of observations $\{x_i\}$ in which record the probability

$$p = \prod_i P(x_i \notin W|H) \tag{1}$$

However, the choice of W was left up to the experimenter and was not uniquely determined by the p-value.

In 1928-1938 Neyman & Pearson developed a theory in which one must consider competing Hypotheses:

- the Null Hypothesis H_0
- the Alternate Hypothesis H_1

Given some probability that we wrongly reject the Null Hypothesis

$$\alpha = P(x \notin W | H_0) \tag{2}$$

Find the region W such that we minimize the probability of wrongly accepting the H_0 (when H_1 is true)

$$\beta = P(x \in W | H_1) \tag{3}$$

The region W that minimizes the probability of wrongly accepting the H_0 is just a contour of the Likelihood Ratio:

$$\frac{P(x|H_0)}{P(x|H_1)} > k_\alpha \tag{4}$$

In principle, all that is left to do is estimate the probability density functions $P(x|H_0)$ & $P(x|H_1)$.

But this is not the end of the story...

The Artificial Intelligence Point of View

Learning Machines / Pattern Recognition Algorithms / A.I.
are essentially Black Boxes with some parameters.

Formally, a learning machine looks like a family of functions
from an input space I to an output space O,
each specified by some parameters α.

$$f_\alpha(x \in I) \equiv f(x;\alpha) = y \in O \tag{5}$$

Training Data is a set of pairs $\{x_i, y_i\}$

The way in which the function's parameters are determined from
training data is associated *learning*.

Formally, learning is a variational problem:
find the function f that maximizes the performance $\mathcal{P}$.

$$\frac{\delta \mathcal{P}[f]}{\delta f} = 0 \tag{6}$$

The most common choice of $\mathcal{P}$ is the Error Functional or Risk:

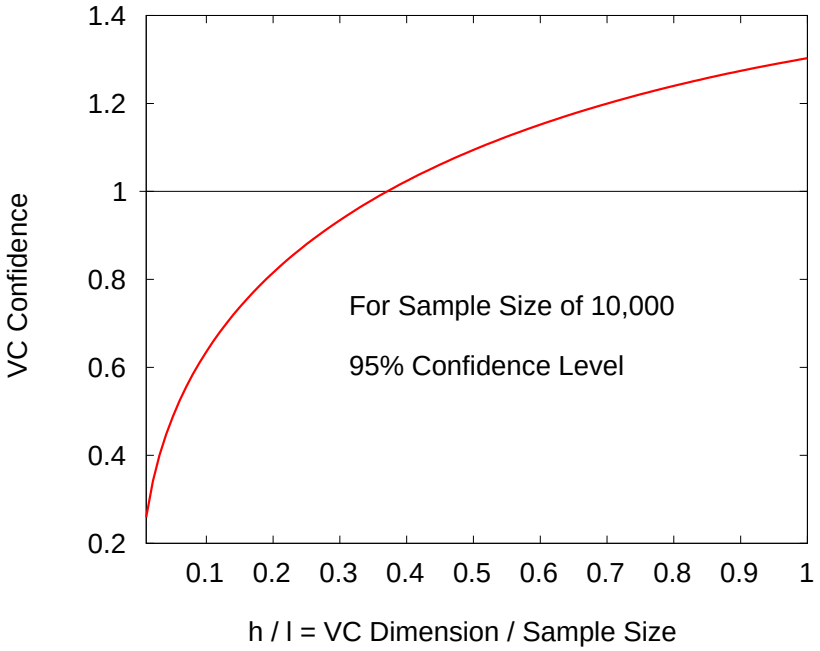
$$R_{\text{emp}}(\alpha) = \frac{1}{2l} \sum_{i=1}^{l} |y_i - f(x_i; \alpha)|.$$

For instance, a Neural Network with Back Propagation adjusts α so as to minimize R_{emp}.

Suprisingly, there are general bounds on the generalization performance of a Pattern Recognition Algorithm, given by:

$$R(\alpha) = \int \frac{1}{2}|y - f(x;\alpha)|\, p(x,y)dxdy$$

$$\leq R_{\text{emp}}(\alpha) + \sqrt{\left(\frac{h(\log(2l/h) - \log(\eta/4)}{l}\right)}$$



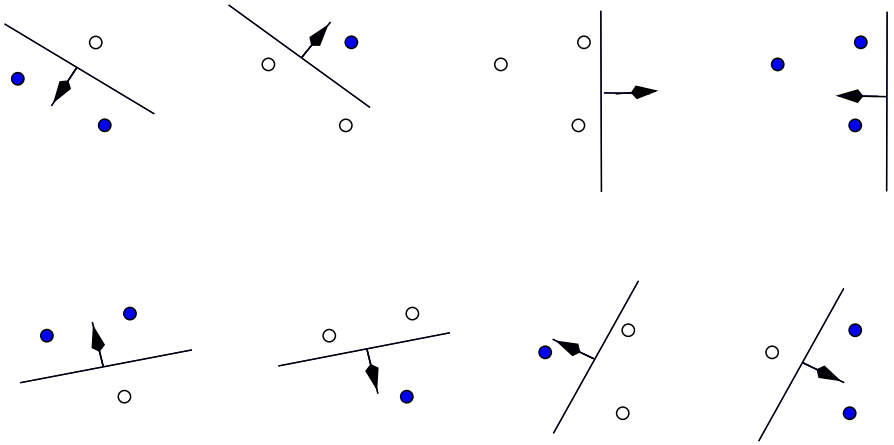
$h \rightarrow$ the Vapnik Chervonenkis (VC) dimension

$l \rightarrow$ the sample size

$1 - \eta \rightarrow$ the confidence the bound holds.

The VC dimension h is equal to the maximal number of points that can be *shattered* by the learning machine $f(x;\alpha)$.

"A set $\{x_i\}$ is shattered by $f(x;\alpha)$" means that for every permutation of classifications $\{x_i, y_i\}$, there is an α such that $f(x_i;\alpha) = y_i$.



Examples:

An oriented line can shatter 3 points in $\mathbb{R}^2$

A Hyperplane can shatter $d+1$ points in $\mathbb{R}^d$

Note: Not every set of h elements must be shattered by $f(x;\alpha)$, but just one.

Examples of VC Dimension

The somewhat counter-intuitive point is that to minimize the risk, one should find the learning machine with the lowest VC dimension.

Higher VC dimension $\rightarrow$ Higher Generalization Capacity $\rightarrow$ Higher Risk

These idea of minimizing the risk is the motivation for Support Vector Machines

In General Neural Networks have a very high or even infinite VC dimension

For Learning Machines which form a Vector Space:
VC dimension = dimensionality the parameters span in the Vector Space.

Neyman-Pearson not symmetric under $H_0 \leftrightarrow H_1$, but Error Functional is symmetric.

For a given learning machine, there may not be any parameters α that can reproduce the Likelihood Ratio $P(x|H_0)/P(x|H_1)$.

The Classification Problem is distinct from Hypothesis Testing.

The Signal Processing Point of View

How much information is in the this message?

$$\underbrace{01100111}_{\text{len}=8}$$

a) 8 bits

b) No information

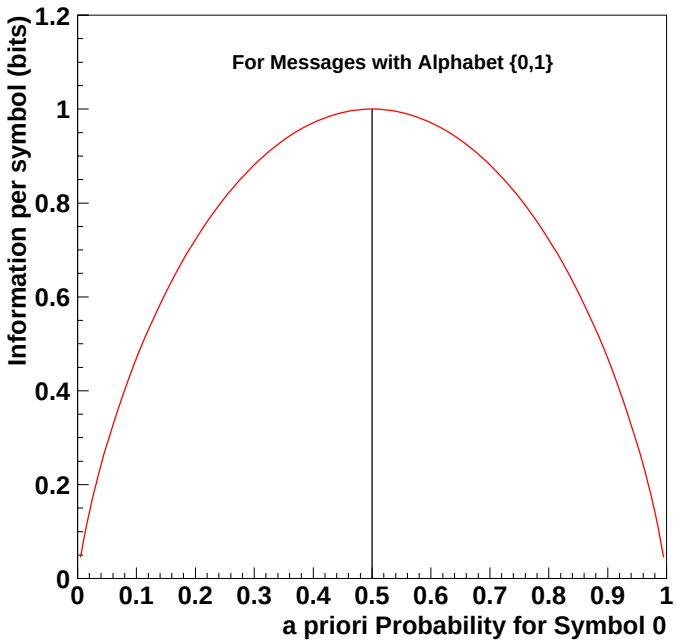
c) < 8 bits

d) Not well defined

The information contained in a "message" of length 1 with an "alphabet" $\{a_i\}$ is given by:

$$H = -\sum_i p_i \log_2(p_i) \ \text{bits} \tag{7}$$

where p_i is the *a priori* probability of the i^{th} "letter" to occur.



Examples:

If you know that the sender will send letter a_i with probability 1 $\Rightarrow$ no information in the message.

Expect Information to be symmetric under $a_i \leftrightarrow a_j$
$\Rightarrow$ Information Maximized when $p_i = p_j \quad \forall i, j$

Now consider the messages which answer the following questions:

- *Will this Monte Carlo Algorithm accept or reject this event?* $\in \{\text{accept}, \text{reject}\}$
  $\Rightarrow$ Find Phase Space generator that gives narrow weight distribution.

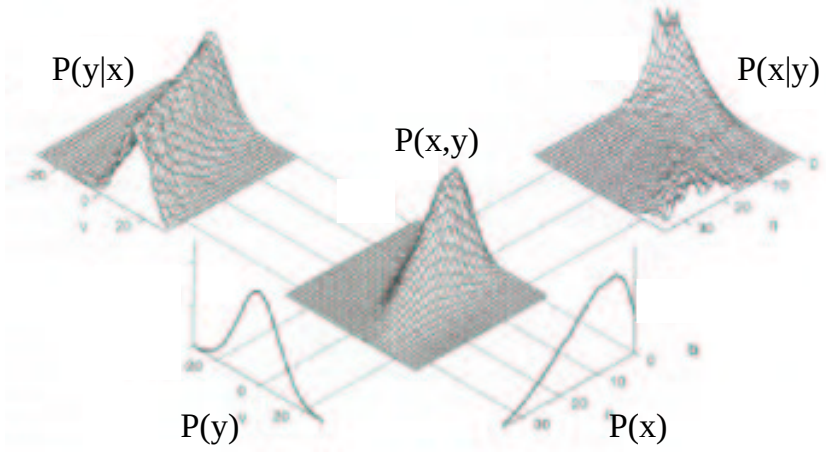
- *Did this event come from signal or background?* $\in \{\text{signal}, \text{background}\}$
  $\Rightarrow$ Boundry of Critical Region is *dense* in information.

The answers to these questions carry information in a formal way.

When the "messages" are longer, we must look at joint probabilities

Define the Mutual Information between X and Y as follows:

$$I(X,Y) \equiv H(X) - H(X|Y) = H(Y) - H(Y|X) \equiv I(Y,X) \tag{8}$$



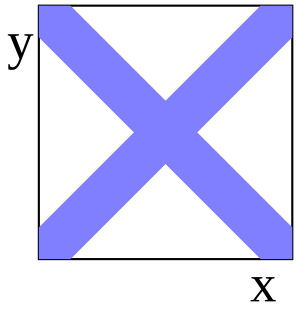
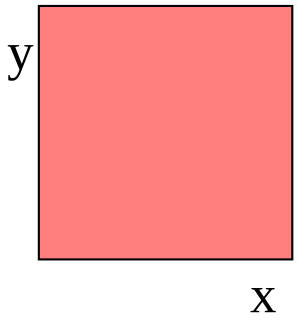
Correlations between variables:

- Reduce the Joint Information $H(X,Y)$

- Increase the mutual information $I(X,Y)$

Mutual Information tells us how two variables are correlated, but unlike the covariance matrix it sees all orders.

Consider two situations:

	Probability Density	Covariance Matrix	Mutual Information
x is uniformly distributed $y = x$ or $y = -x$		$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	H(x)−1
x and y are uniformly distributed & independent		$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	0

Mutual Information offers yet another Performance Functional

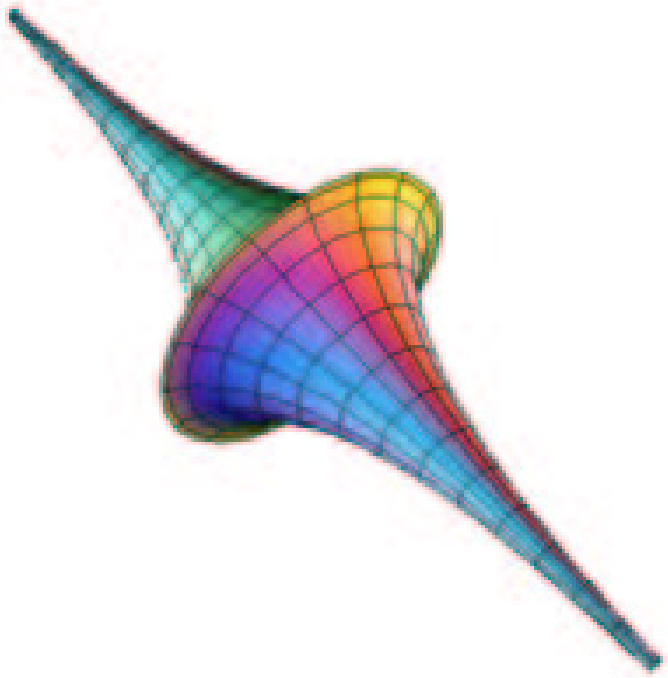
In different words, we want to reduce the information loss $\Delta I(X,Y) = I(X,\Omega) - I(Y,\Omega)$ (where now X is thought to be the input after noise is added to the true source Ω.

It is fairly intuitive that the backpropogation algorithm is aiming to reduce information loss, by minimizing the error R_{emp}.

Quite significantly, this learning rule makes sense even when there is no "training data". The InfoMax technique is called an unsupervised learning rule.

Amari considered the *Fisher Information Matrix* g_{ij} *as a metric on a Manifold* M *parmetrized by* α*:*

$$g_{ij}(\alpha) = \int dx f_\alpha(x) \left[\frac{\partial \log f_\alpha(x)}{\partial \alpha_i}\right] \left[\frac{\partial \log f_\alpha(x)}{\partial \alpha_j}\right] \tag{9}$$



Example:

Consider Gaussians $G(x;\mu,\sigma)$ as a Manifold parametrized by $\alpha = (\mu, \sigma)$

the geometry is isotropic and negatively curved

Natural Learning Rules correspond to *geodesics* on the Manifold M.

Projection Pursuit

Boundary Magnification

Choosing Variables

Intuitively we want as much information as possible in our variables x_i
$\Rightarrow$ maximize $I(x, \mathcal{C})$

But we need exponentially more samples as the dimensionality d grows
"The Curse of Dimensionality" viz. $N \propto e^d$

This tradeoff means we want the variables to have little redundancy
$\Rightarrow$ minimize $I(x_i, x_j)$

Boos, Dudko, & Ohl (EPJ C 11) have suggested that angular and singular variables from Feynman graphs as 'optimal' variables.

Including detector effects, these variables may not be optimal even in theory.
Practicalities of the Multivariate Algorithm may in fact dominate.

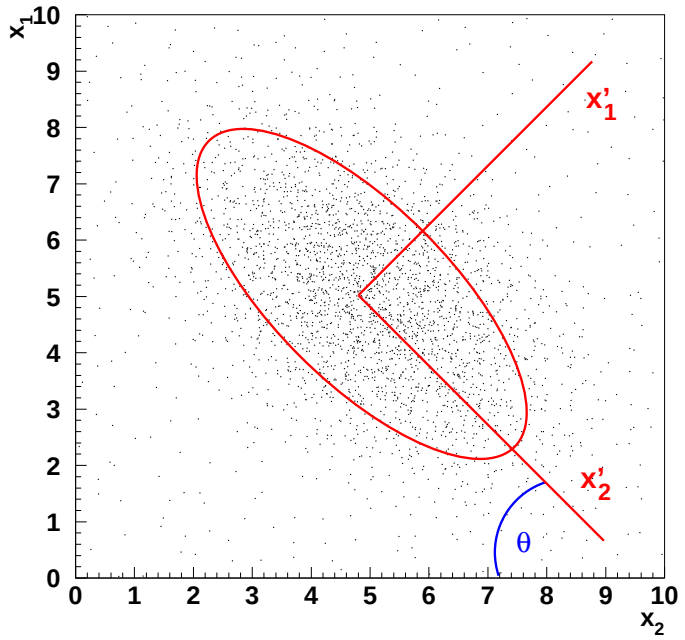
Projection Pursuit: Principal Component Analysis

Typically, for HEP data sets (which are very large) dimensionality of greater than 5 or 6 is the "histogram limit".

To reduce the dimensionality, statisticians have searched for novel types of projections which preserve as much *information* in the original data as possible. This search is called "Projection Pursuit"

Principal Component Analysis is a very common technique which rests on the eigenvalue decomposition of the Covariance Matrix.

Principle Component Analysis (PCA) diagonalizes the covariance matrix, and projects the data to the x'_i with the largest variance.



PCA is constructive, not an optimization.

Consider:

$$\frac{\partial I(x, x'_2)}{\partial \theta} \tag{10}$$

Can generalize PCA by considering projections paremtrized by β and maximizing Mutual Information $I(x, x'_i)$ w.r.t. β

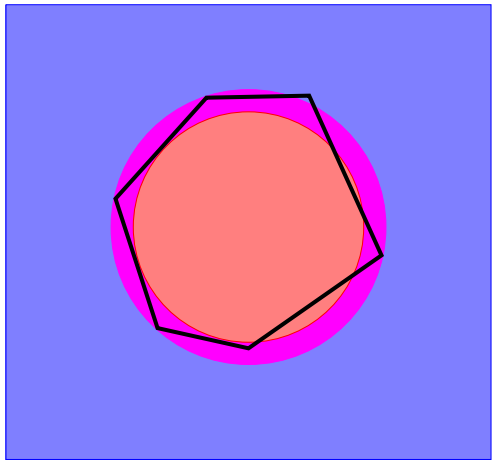
In regions dominated by a single class, classification of x is obvious.
It is near the boundary of critical regions that it is difficult to classify.

From an information theory point of view, we ask the question:
"To which class does this event belong?"
The answer to that question carries information $H(x)$.

It seems natural to provide bandwidth to the pattern recognition algorithm in proportion to the *Information Density* $H(x)[P(x|H_0) + P(x|H_1)]$.

The Equi-partition of Information Density (EpID) is an algorithm I developed to do just that.

Boundary Magnification



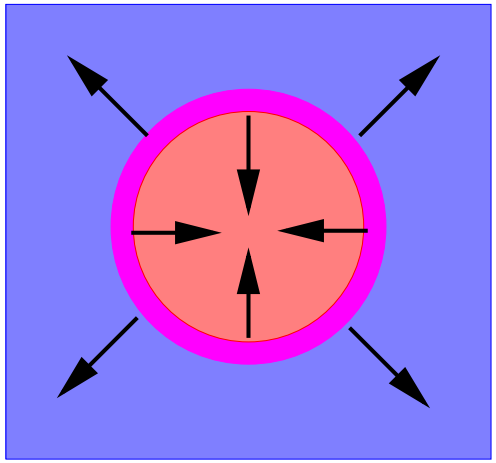
Before

Background

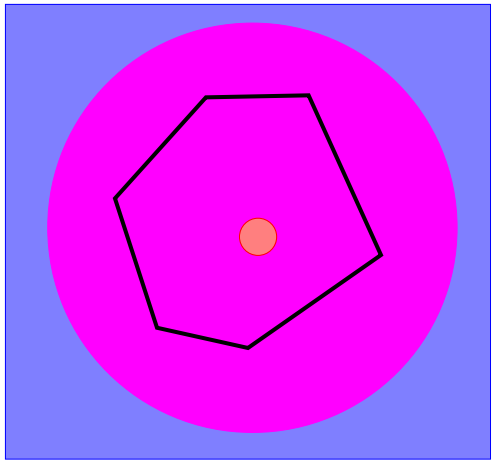
Overlap

Signal

Learning Machine



Boundary Magnification

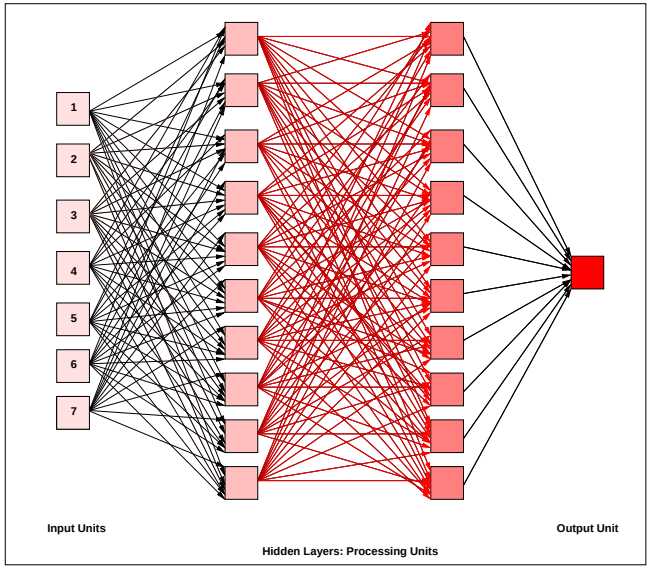


After

Examples

Neural Networks are a very popular multivariate algorithm.

As a correlarry of Kolmogorov's solution to Hilbert's 13^{th} problem, Hecht-Nielson showed that a Neural Network can approximate an *arbitrary function* with an arbitrary accuracy.



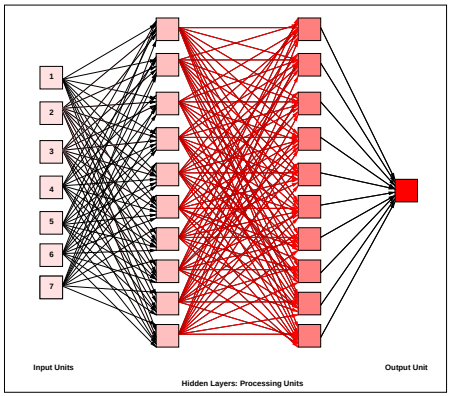
The Network is made of Nodes N_j which process inputs I_i

The node function is usually something simple like:

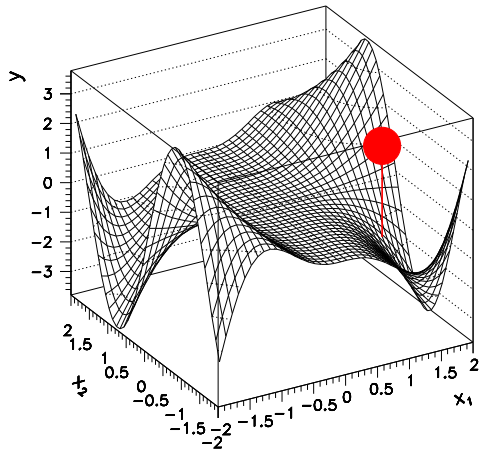
$$N_j = \arctan(\sum_i W_{ij} I_i + \delta_j) \quad (11)$$

Neural Networks gained prominance when the PDP Group introduced the back-propogation learning algorithm.

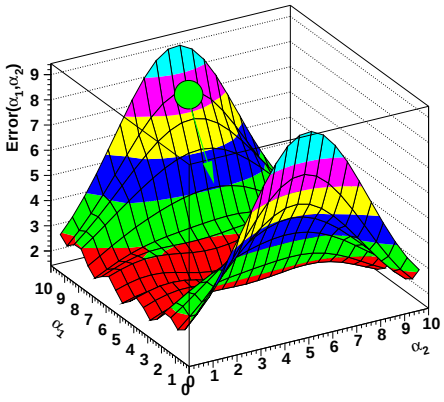
An Example: Neural Networks



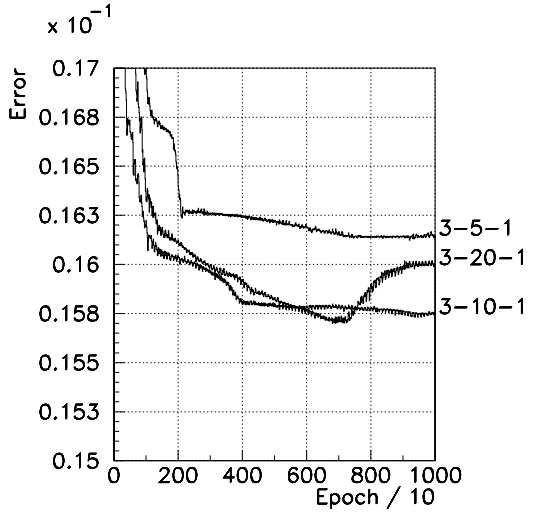
1) Start with some parameters α which correspond to some $f_\alpha(x)$



2) Calculate the Error $|y - f_\alpha(x)|$



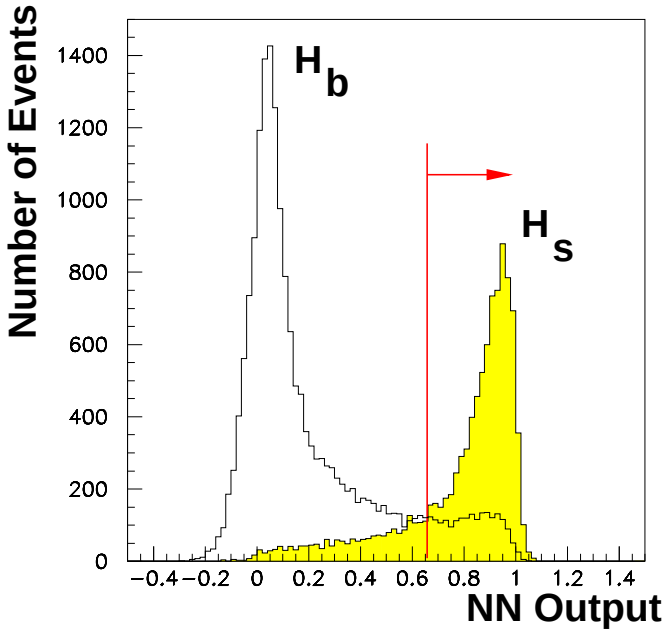
3) Apply the Chain rule to find $\Delta\alpha$ of steepest decent



4) Repeat Many Times until Error is minimized

Once the neural network has been trained, we *test* with an independent sample.

Typically, histograms of the `NN Output` are made which show the discrimination between signal and background.

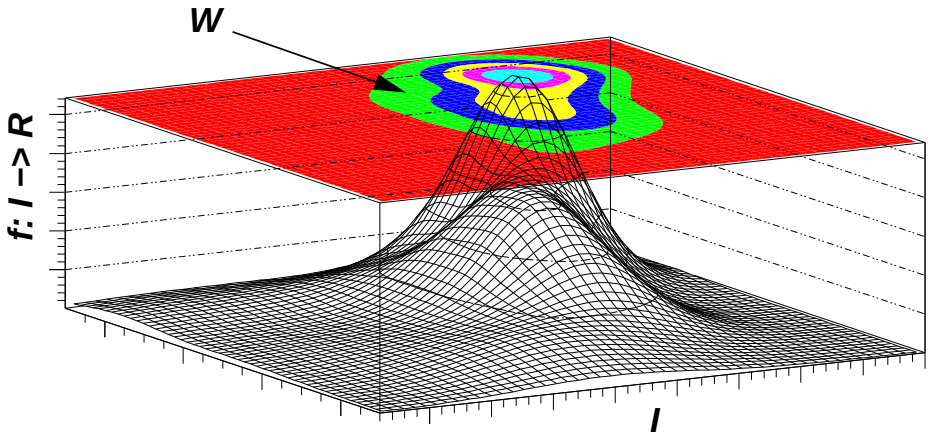


Candidate events are chosen by requiring that `NN Output` is greater than some NN_{cut}.

The choice of the NN_{cut} can be chosen to optimize:
- S/B - The Signal-to-Background Ratio
- $S/\sqrt{B}$ - The Gaussian Significance
- $P(S+B;B)$ - The Poisson Significance

It is also possible to use the shape in a statistical calculation based on the likelihood ratio. In that case you weight each event x_i by $\log(1+H_s(x_i)/H_b(x_i))$

It is possible to discriminate between classes of events if we have pdf's for all the classes.

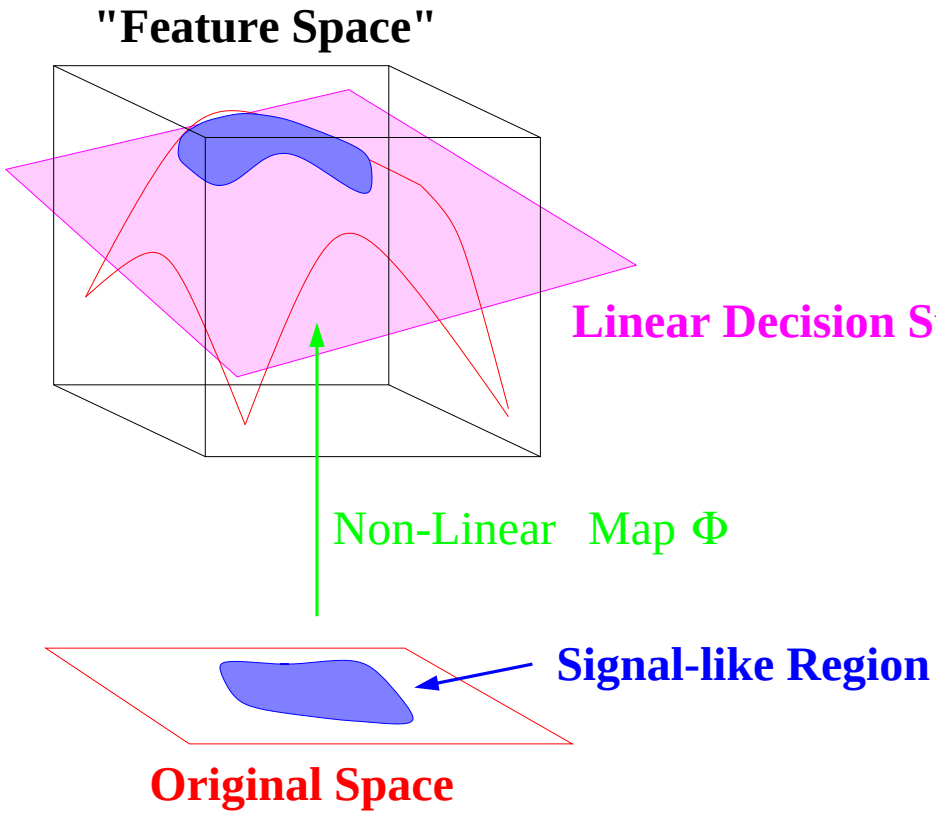


We can construct the pdf's from an emperical source, sort of like Monte Carlo in reverse: samples $\rightarrow$ pdf

$$D(x) = \frac{f_S(x)}{f_S(x) + f_b(x)} \tag{12}$$

Support Vector Machines find an "optimal" decision hyperplane in a high-dimensional "Feature Space".

A non-linear map from the original space to the "feature space" is what allows the signal region to be of arbitrary complexity.



The Decision Boundary is Unique!!

Genetic Programming Approach: The Metaphor

Genetic Programming is usually discussed within the context of a Metaphor...

Population	$\Longleftrightarrow$	Set of cuts
Individual	$\Longleftrightarrow$	A particular cut
Evolution	$\Longleftrightarrow$	Optimization of a set of cuts
Generation	$\Longleftrightarrow$	A training epoch
Mutation	$\Longleftrightarrow$	Stochastic search
Fitness	$\Longleftrightarrow$	Significance (in "sigma")
Competition	$\Longleftrightarrow$	Sampling a Fitness Distribution

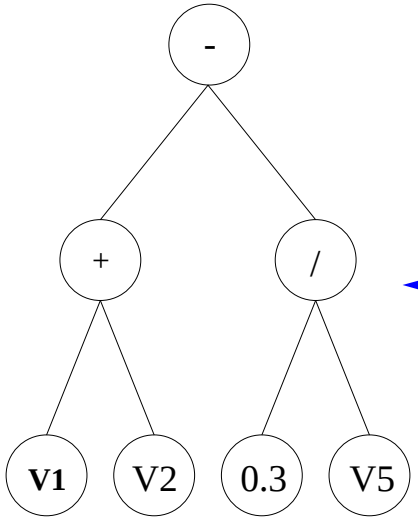
CAUTION! Genetic Programming $\neq$ Genetic Algorithms

Genetic Programming and Genetic Algorithms rest on a similar Metaphor, but the techniques are quite different.

While Genetic Algorithms have been used in HEP, Genetic Programming seems to be new technique for event selection!

Cuts take the form: $-1 < [\text{expression}] < 1$

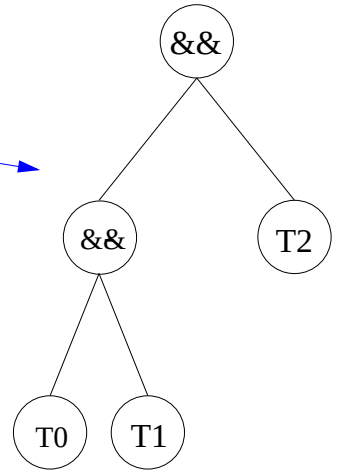
where [expression] is constructed as a tree



Example Expression:

$T0 = V1 + V2 - 0.3/V5$

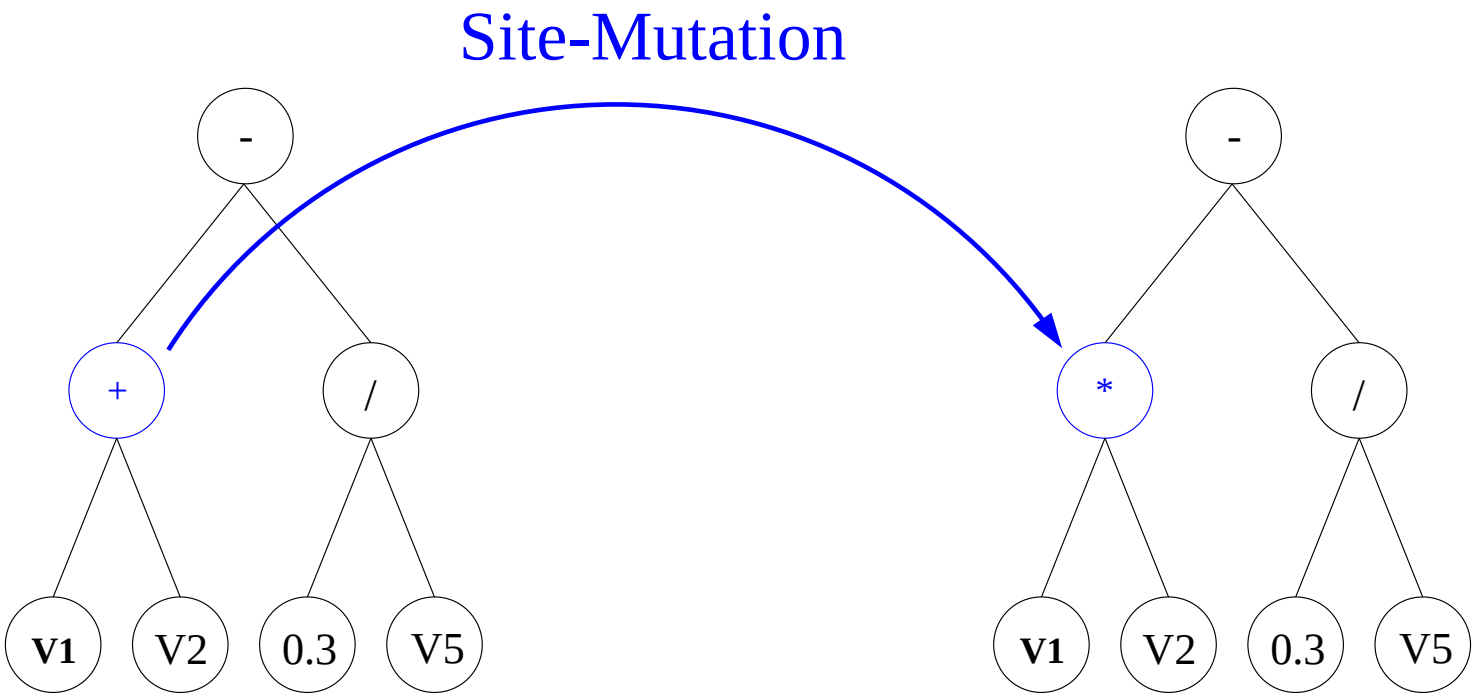
Boolean Conjunctions of Cuts form another tree



Example Conjunction:

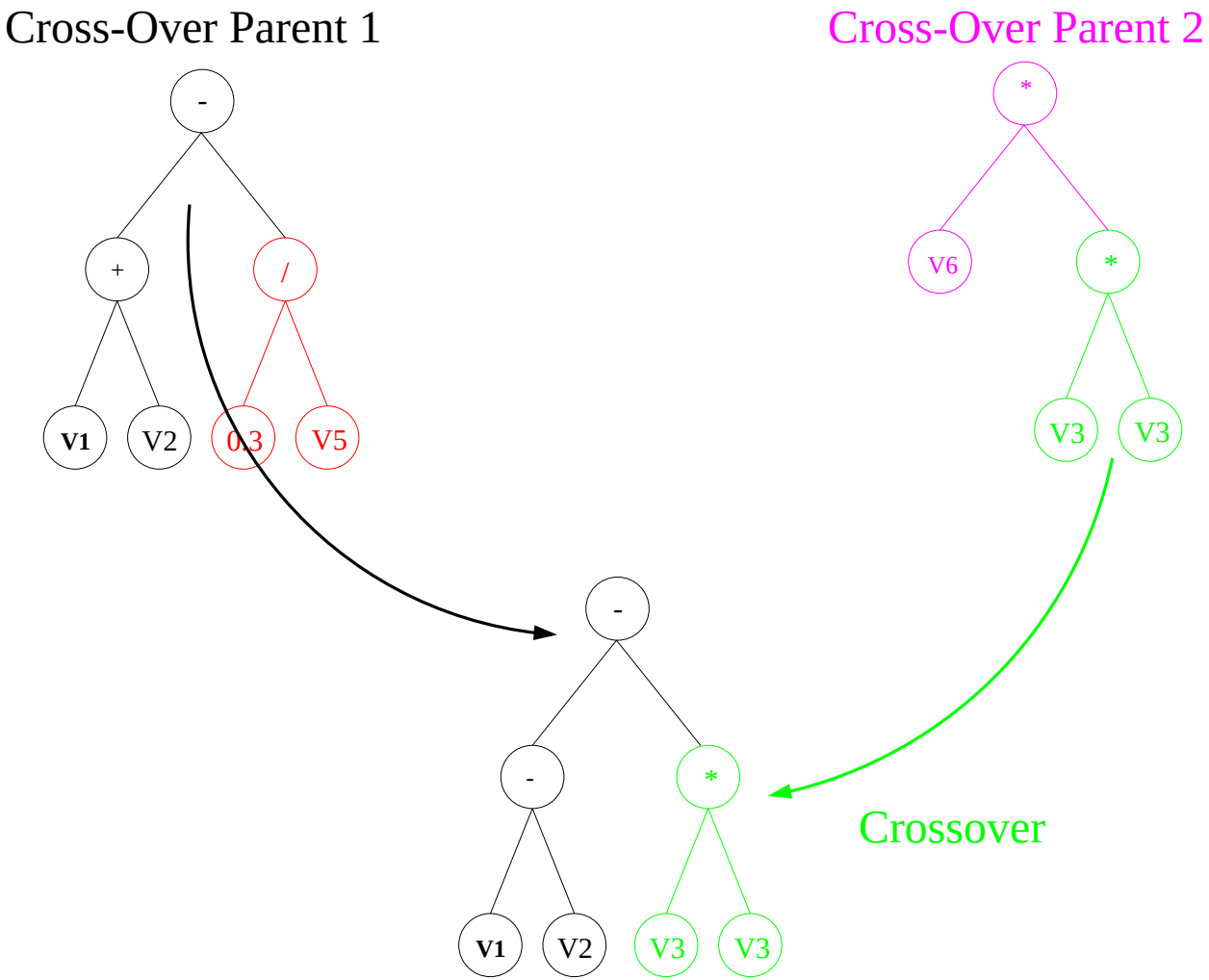
$Test = (T0 \&\& T1) \&\& T2$

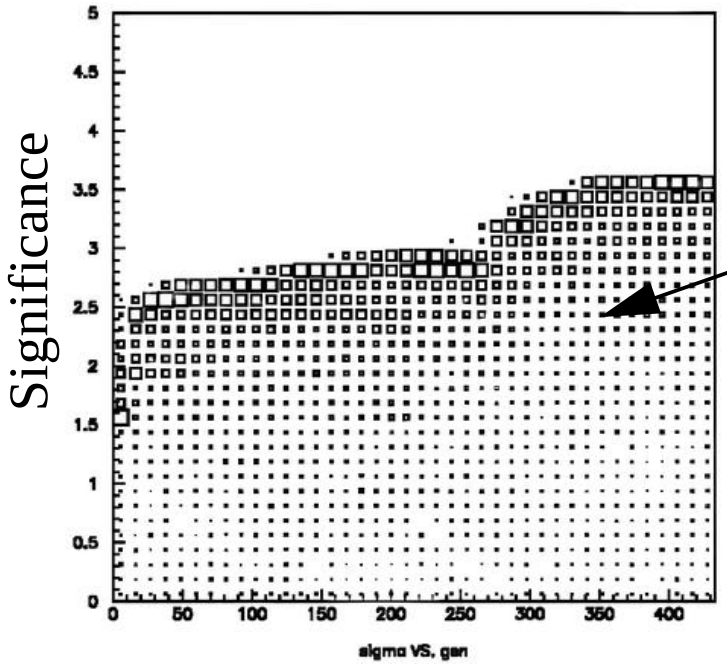
Mutation is a random process which results in a stochastic search in the space of all possible cuts.



Because the mutations are based on the previous generation, the search is a Markov process $\rightarrow$ it's much faster than an ergodic search would be.

Cross-over is a mutation which takes two parents and swaps a random sub-tree.

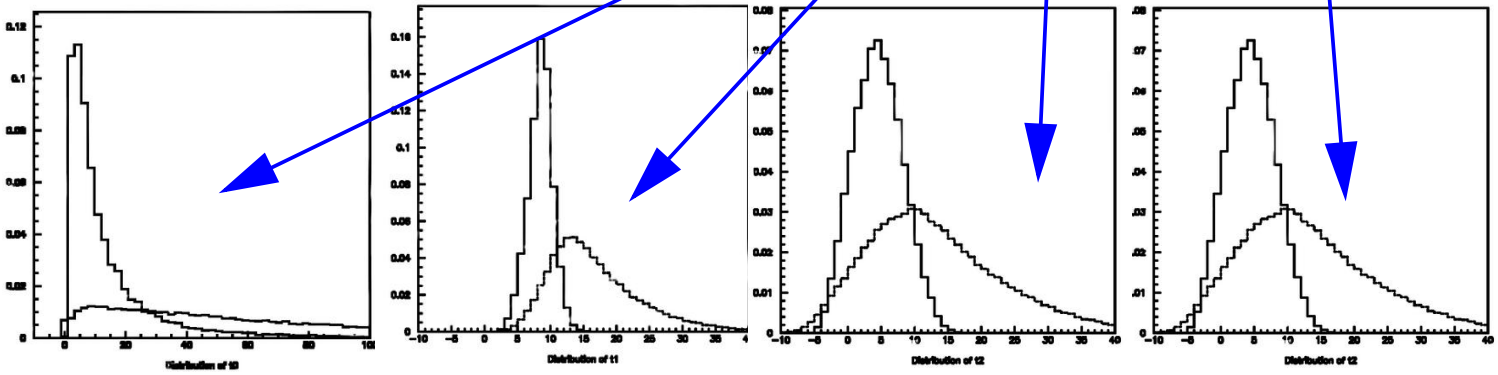




Here we see the significance improving as the cuts evolve.

After training, we select the most fit individual (cut) and apply it to the data.

In this case, the individual has constructed 4 variables each with discriminating power.



Multivariate Algorithms have a wide range of applications

Each application is interested in optimizing
a specialized notion of performance

As physicist we must be clear about what we want to optimize

In addition, we must take care that we use these
techniques in a sensible way