Multivariate Analysis
A Unified Perspective

Harrison B. Prosper

Florida State University

Advanced Statistical Techniques in Particle Physics

Durham, UK, 20 March 2002

Outline

- Introduction
- Some Multivariate Methods
  - Fisher Linear Discriminant (FLD)
  - Principal Component Analysis (PCA)
  - Independent Component Analysis (ICA)
  - Self Organizing Map (SOM)
  - Random Grid Search (RGS)
  - Probability Density Estimation (PDE)
  - Artificial Neural Network (ANN)
  - Support Vector Machine (SVM)
- Comments
- Summary

Introduction – i

- Multivariate analysis is hard!
  - Our mathematical intuition based on analysis in one dimension often fails rather badly for spaces of very high dimension.

- One should distinguish the problem to be solved from the algorithm to solve it.

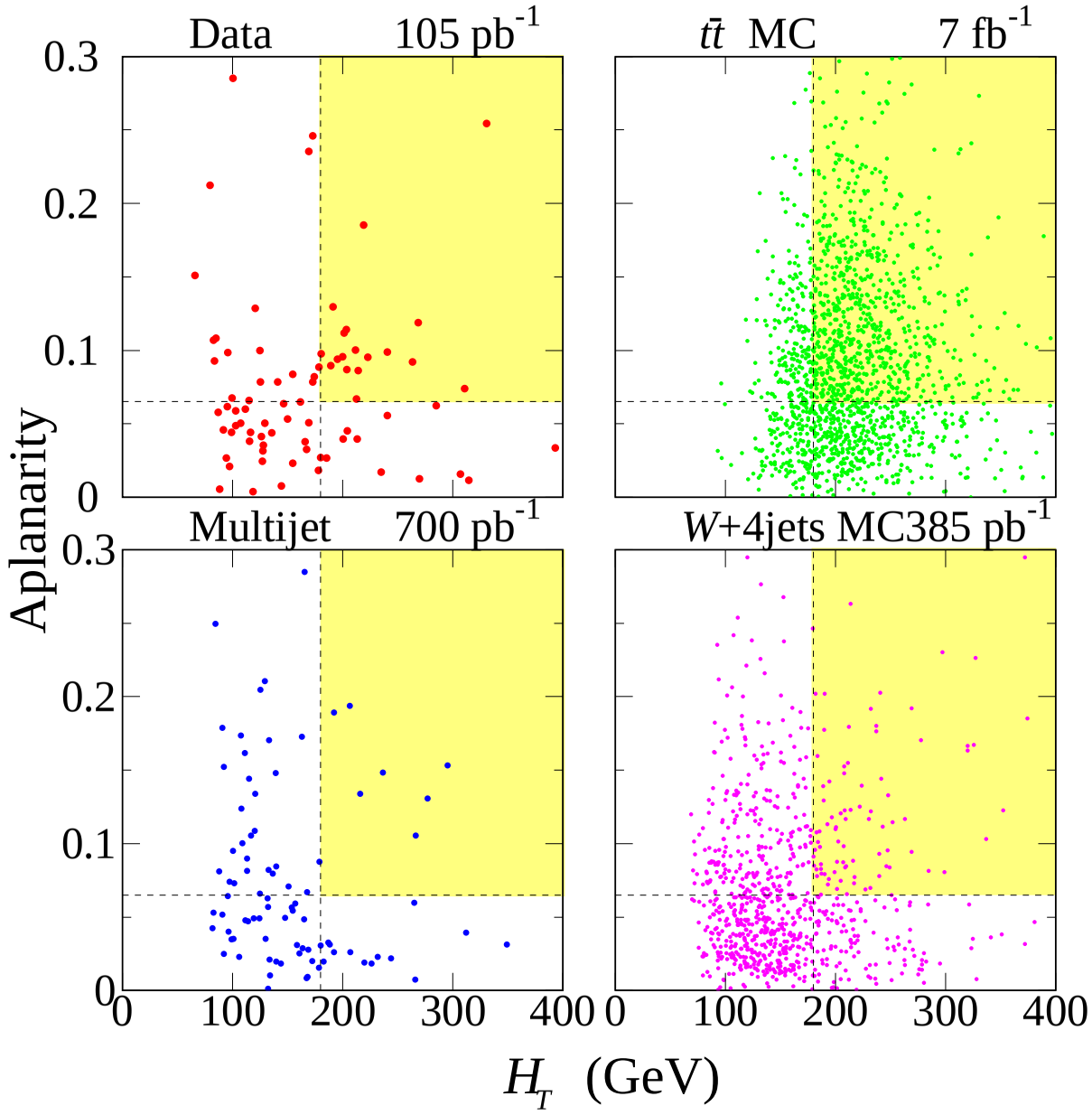
- Typically, the problems to be solved, when viewed with sufficient detachment, are relatively few in number whereas algorithms to solve them are invented every day.

Introduction – ii

- So why bother with multivariate analysis?
- Because:
  - The variables we use to describe events are usually *statistically dependent*.
  - Therefore, the N-d density of the variables contains more information than is contained in the set of 1-d marginal densities $f_i(x_i)$.
  - This extra information may be useful

$p\bar{p} \rightarrow t\bar{t} \rightarrow l + jets$

Dzero 1995
Top
Discovery

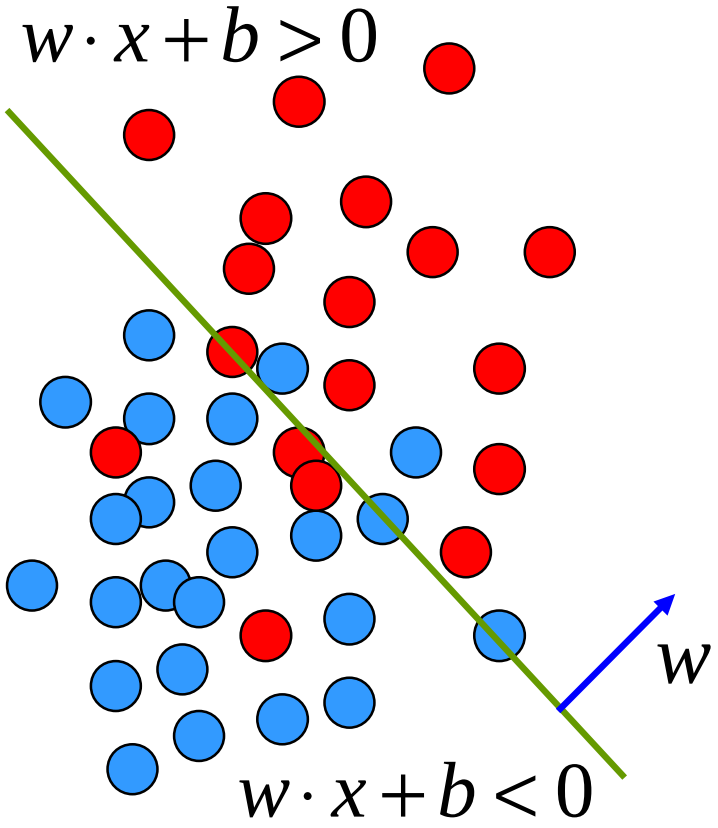


Introduction - iii

- Problems that may benefit from multivariate analysis:
  - Signal to background discrimination
  - Variable selection (e.g., to give maximum signal/background discrimination)
  - Dimensionality reduction of the *feature* space
  - Finding *regions of interest* in the data
  - Simplifying optimization (by $f : \Re^N \rightarrow U^1$)
  - Model comparison
  - Measuring stuff (e.g., tanβ in SUSY)

Fisher Linear Discriminant

- Purpose
  - Signal/background discrimination



$$\log \frac{g(x \mid \mu_1, \Sigma)}{g(x \mid \mu_2, \Sigma)}$$

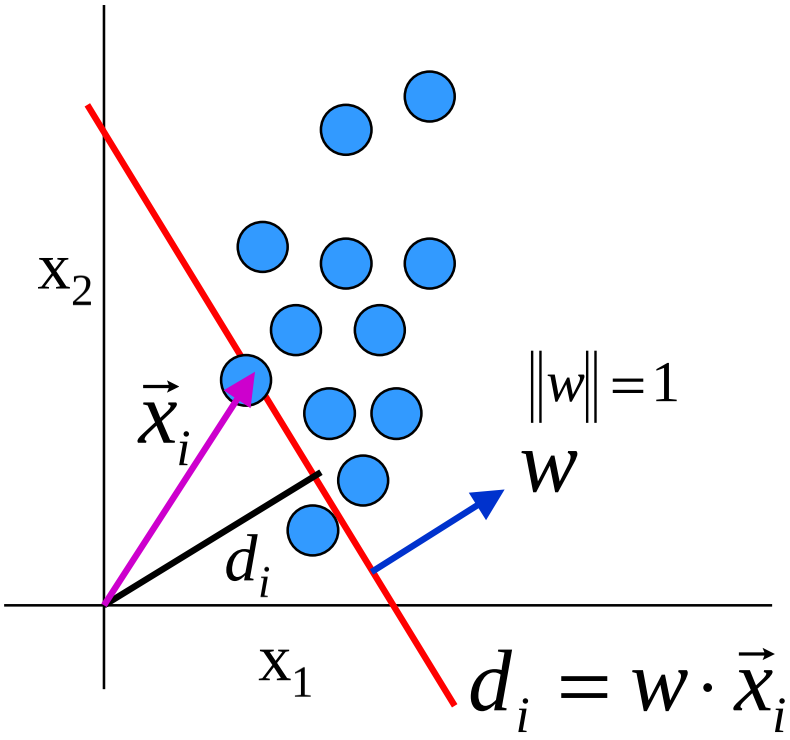
$$\rightarrow \chi^2(\mu_2) - \chi^2(\mu_1)$$

$$\rightarrow w \cdot x + b$$

g is a Gaussian

Principal Component Analysis

- Purpose
  - Reduce dimensionality of data



1st principal axis

$$w_1 = \arg\max \sum_{i=1}^{K} d_i^2(w)$$

2nd principal axis

$$w_2 = \arg\max \sum_{i=1}^{K} [w \cdot (\vec{x}_i - w_1 d_i(w_1))]^2$$

PCA algorithm in practice

- Transform from $\mathbf{X = (x_1,..x_N)^T}$ to $\mathbf{U = (u_1,..u_N)^T}$ in which lowest order correlations are absent.
  - Compute **Cov(X)**
  - Compute its eigenvalues λ_i and eigenvectors $\mathbf{v_i}$
  - Construct matrix $\mathbf{T = Col(v_i)^T}$
  - **U = TX**
- Typically, one eliminates u_i with smallest amount of variation

Independent Component Analysis

- Purpose
  - Find statistically independent variables.
  - Dimensionality reduction
- Basic Idea
  - Assume $\mathbf{X} = (x_1,..,x_N)^T$ is a linear sum $\mathbf{X = AS}$ of independent sources $\mathbf{S} = (s_1,..,s_N)^T$. Both **A**, the *mixing* matrix, and **S** are *unknown*.
  - Find a ***de-mixing*** matrix **T** such that the components of $\mathbf{U = TX}$ are *statistically independent*

ICA-Algorithm

Given two densities ***f(U)*** and ***g(U)*** one measure of their "closeness" is the Kullback-Leibler divergence

$$K(f \mid g) \equiv \int f(U) \log\left(\frac{f(U)}{g(U)}\right) dU \quad \geq 0$$

which is *zero* if, and only if, ***f(U) = g(U)***.

We set

$$g(U) = \prod_i f_i(u_i)$$

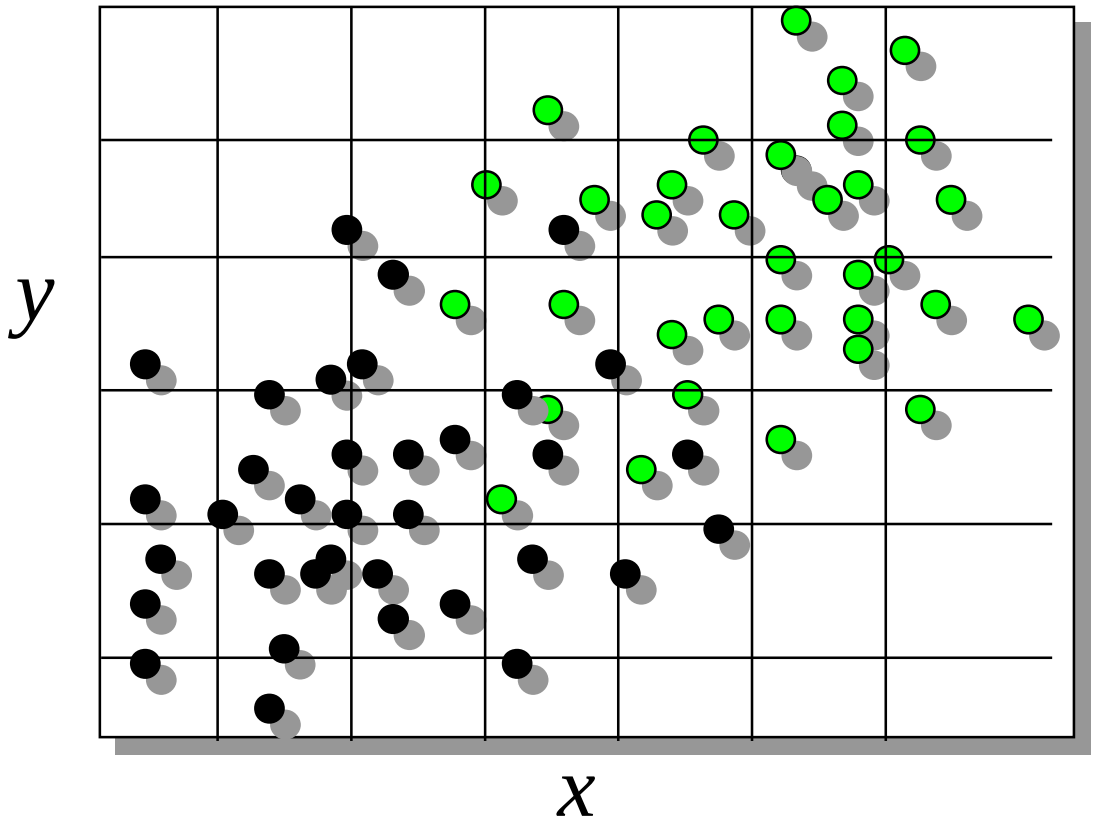
and minimize ***K(f | g)*** (now called the *mutual information*) with respect to the de-mixing matrix **T**.

Self Organizing Map

- Purpose
  - Find regions of interest in data; that is, clusters.
  - Summarize data
- Basic Idea (Kohonen, 1988)
  - Map each of **K** feature vectors $\mathbf{X} = (x_1,..,x_N)^T$ into one of **M** regions of interest defined by the vector $\mathbf{w_m}$ so that all **X** mapped to a given $\mathbf{w_m}$ are closer to it than to all remaining $\mathbf{w_m}$.
  - Basically, perform a coarse-graining of the feature space.

Grid Search

Purpose: Signal/Background discrimination



Apply cuts at each grid point

$$x > x_i$$

$$y > y_i$$

We refer to (x_i, y_i) as a *cut-point*

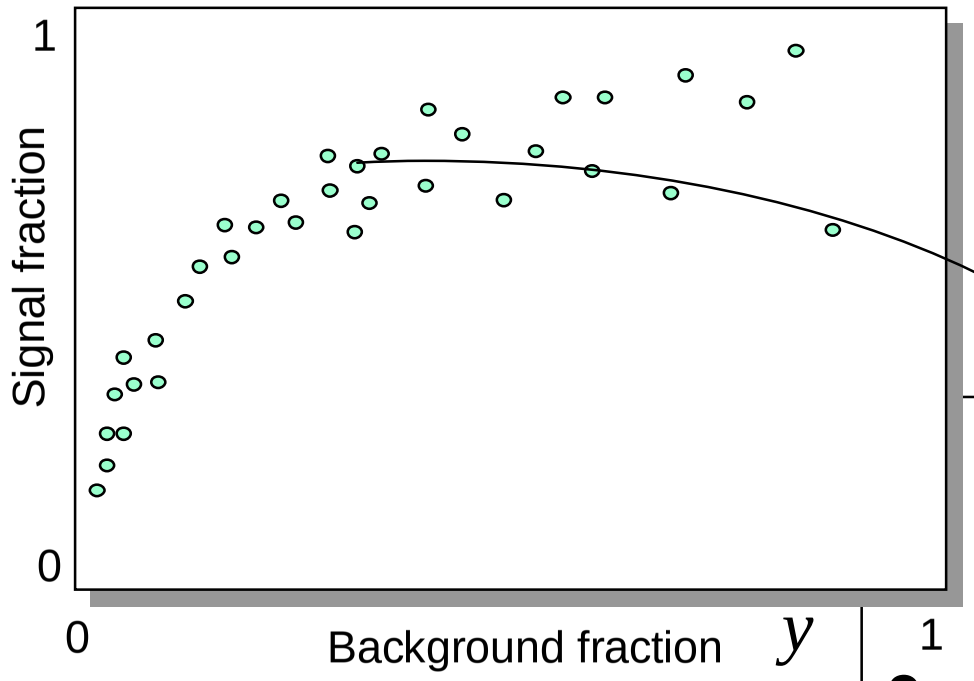
Number of cut-points ~ N_{bin}^{Ndim}

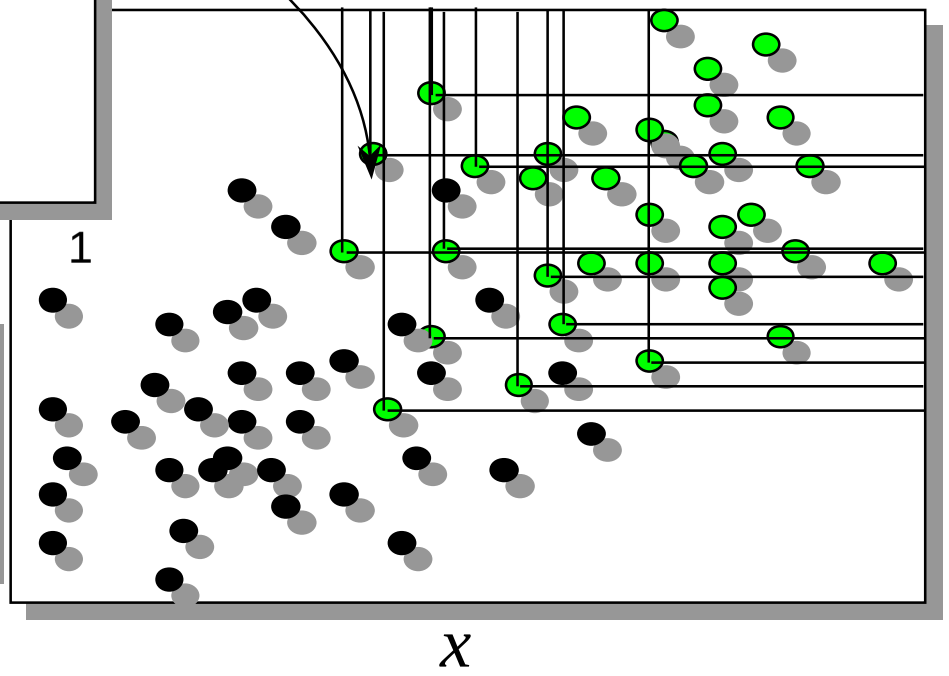
Random Grid Search

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

$x > x_i$

$y > y_i$





N_{tot} = # events before cuts

N_{cut} = # events after cuts

Fraction = N_{cut}/N_{tot}

H.B.P. et al, Proceedings, CHEP 1995

Probability Density Estimation

- Purpose
  - Signal/background discrimination
  - Parameter estimation
- Basic Idea
  - Parzen Estimation (1960s)

$$p(x) = \frac{1}{N}\sum_n \frac{1}{h^d}\varphi\left(\frac{x - x_n}{h}\right) \qquad 1 \le n \le N$$

  - Mixtures

$$p(x) = \sum_j \varphi(x \mid j) q(j) \qquad j << N$$

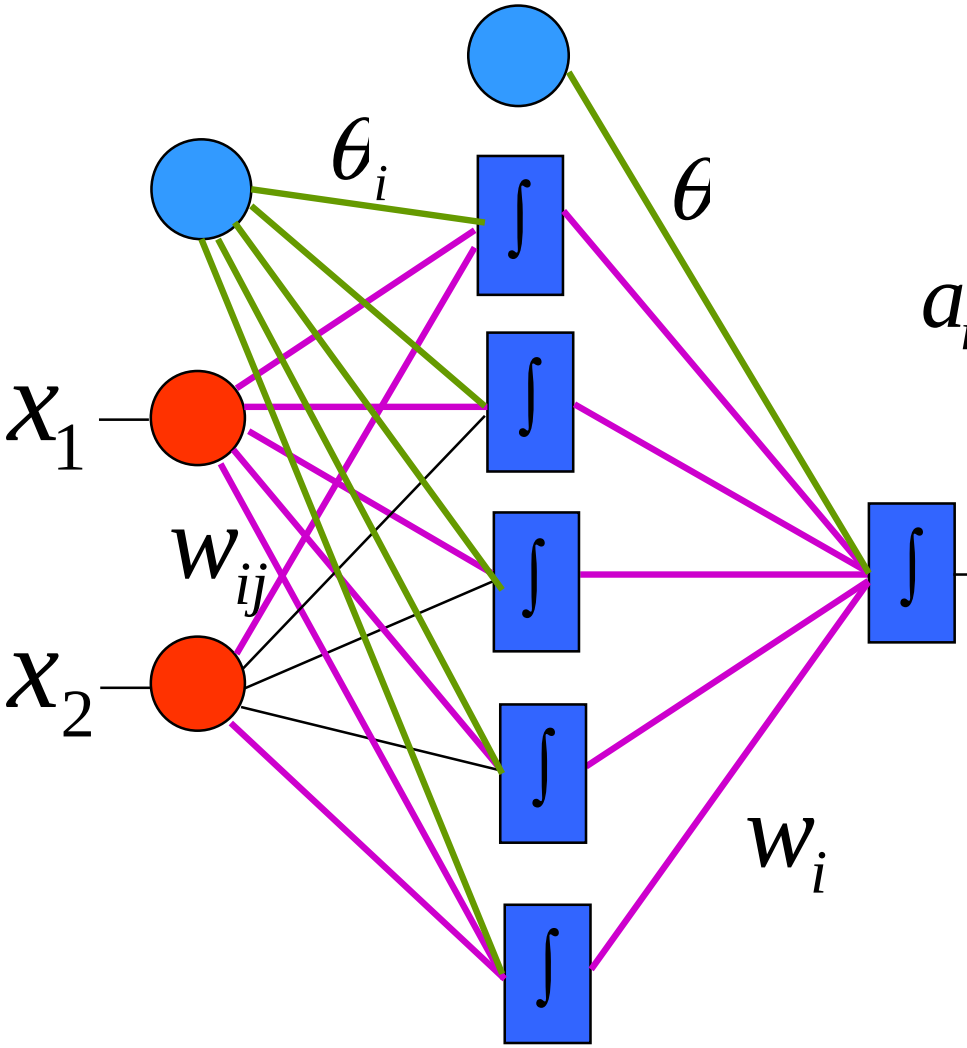
Artificial Neural Networks

- Purpose
  - Signal/background discrimination
  - Parameter estimation
  - Function estimation
  - Density estimation
- Basic Idea
  - Encode mapping (Kolmogorov, 1950s).

$$f : U^N \rightarrow U^M \qquad f(x) = F[\varphi_1, .., \varphi_K]$$

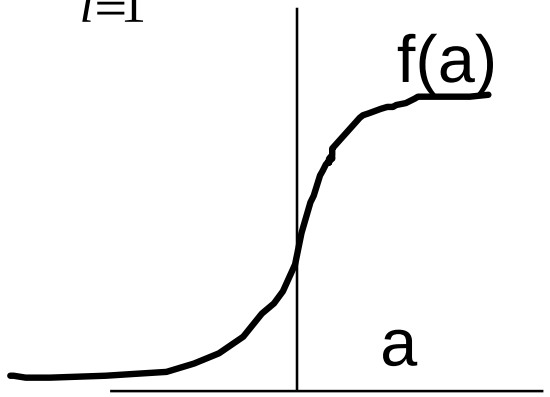
  - Using a set of 1-D functions.

Feedforward Networks



$$a_i = \sum_{j=1}^{2} w_{ij} x_j + \theta_i \quad \rightarrow f(a_i)$$

$$n(x,w) = f\left(\sum_{i=1}^{5} w_i f(a_i) + \theta\right)$$



Input nodes Hidden nodes Output node

ANN- Algorithm

Minimize the *empirical risk function* with respect to ω

$$R(\omega) = \frac{1}{N}\sum_i [t_i - n(x_i, \omega)]^2$$

Solution (for large N)

$$n(x, \omega) \to \int t(x)\, p(t \mid x) dt$$

If t(x) = kδ[1–I(x)], where I(x) = 1 if x is of class k, 0 otherwise

$$n(x, \omega) \to p(k \mid x) = p(x \mid k)\, p(k) \,/ \sum_k p(x \mid k)\, p(k)$$

D.W. Ruck *et al*., IEEE Trans. Neural Networks 1(4), 296-298 (1990)
E.A. Wan, IEEE Trans. Neural Networks 1(4), 303-305 (1990)

Support Vector Machines

- Purpose
  - Signal/background discrimination
- Basic Idea
  - Data that are non-separable in N-dimensions have a higher chance of being separable if mapped into a space of higher dimension

$$\varphi : \Re^{N} \rightarrow \Re^{Huge}$$

  - Use a linear discriminant to partition the high dimensional feature space.

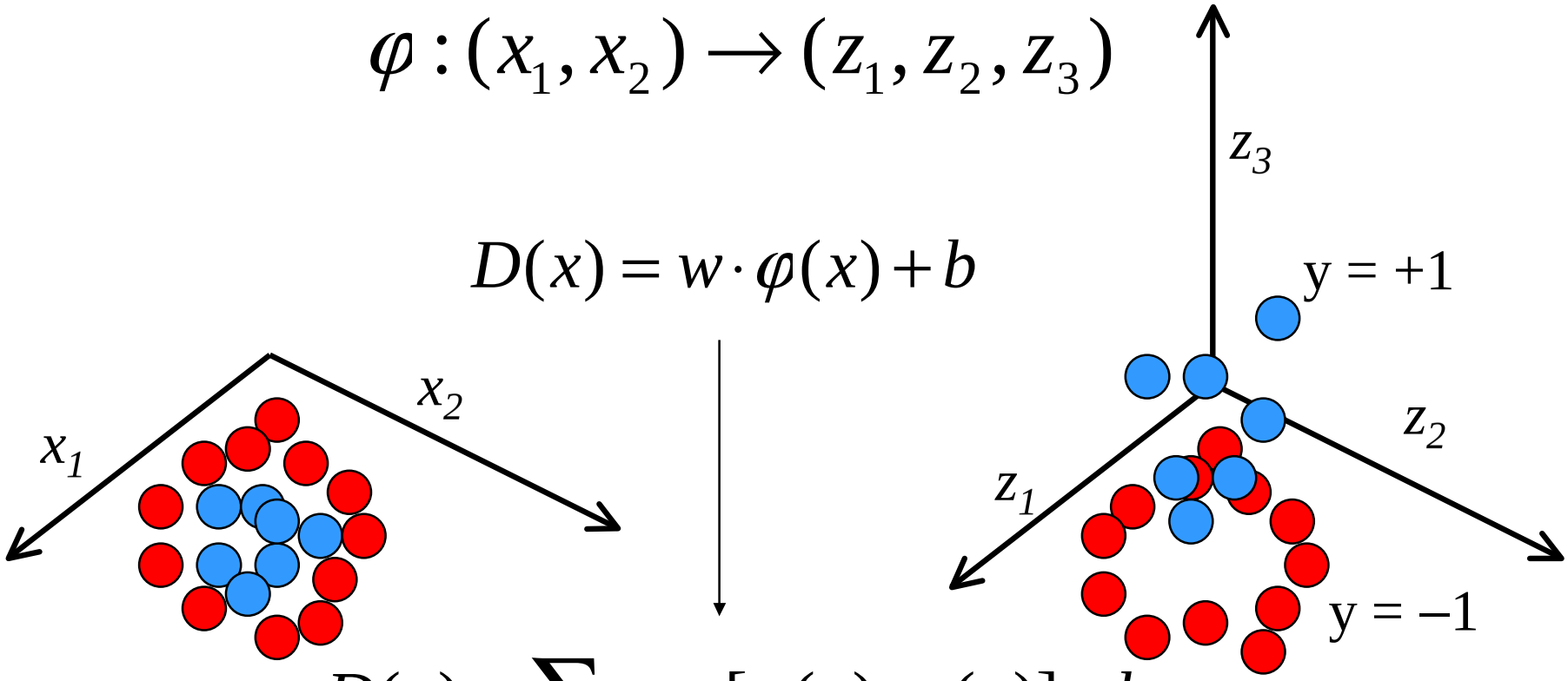
$$D(x) = w \cdot \varphi(x) + b$$

SVM – Kernel Trick

Or how to cope with a possibly infinite number of parameters!

$$\varphi : (x_1, x_2) \rightarrow (z_1, z_2, z_3)$$

$$D(x) = w \cdot \varphi(x) + b$$



$$D(x) = \sum_j \alpha_i y_i [\varphi(x) \cdot \varphi(x)] + b$$

Try different $K(x, x_j) \equiv \varphi(x) \cdot \varphi(x_j)$ because mapping unknown!

Comments – i

- Every classification task tries to solves the *same* fundamental problem, which is:
  - After adequately pre-processing the data
  - …find a ***good***, and ***practical***, approximation to the *Bayes decision rule*: Given **X**, if **P(S|X) > P(B|X)** , choose hypothesis **S** otherwise choose **B**.

- If we knew the densities p(X|S) and p(X|B) and the priors p(S) and p(B) we could compute the *Bayes Discriminant Function* (BDF):
  - D(X) = P(S|X)/P(B|X)

Comments – ii

- The Fisher discriminant (FLD), random grid search (RGS), probability density estimation (PDE), neural network (ANN) and support vector machine (SVM) are simply different algorithms to approximate the Bayes discriminant function D(X), or a function thereof.

- It follows, therefore, that if a method is already close to the Bayes limit, then *no* other method, however sophisticated, can be expected to yield dramatic improvements.

Summary

- Multivariate analysis is hard, but useful if it is important to extract as much information from the data as possible.
- For classification problems, the common methods provide different approximations to the Bayes discriminant.
- There is considerably empirical evidence that, as yet, no uniformly most powerful method exists. Therefore, be wary of claims to the contrary!