MULTIVARIATE METHODS
A Unified Perspective

Harrison B. Prosper
Department of Physics, Florida State University, Tallahassee, Florida 32306, USA

Abstract
I survey several multivariate methods of interest to physicists and explore to what extent each can be viewed as an algorithm to approximate the same mathematical object, namely, the Bayes discriminant function.

1 INTRODUCTION

High energy physicists have had considerable success in using multivariate methods of analysis[1]. It seems likely, therefore, that as analyses become more challenging these methods will be used routinely. We have already agreed that our field would benefit from an improvement in our knowledge and understanding of statistics. Perhaps we could agree on the need to do the same thing for multivariate methods. If a method is considered to be a "black box" this is surely an invitation for us to open it up and look inside.

When one peers in, one finds that each method is typically well characterized mathematically. Peering a bit deeper, they seem (at least to me) to be less different than they might appear at first glance. Indeed, I shall argue that many of the multivariate methods of interest to physicists are simply different algorithms to approximate the same mathematical object. Moreover, the problems they solve, when viewed with sufficient detachment, are relatively few in number whereas new methods are invented almost every day. A typical list of problems is

- Signal to background discrimination
- Variable selection (e.g., to give the maximum signal/background discrimination).
- Dimensionality reduction of the *feature* space
- Finding *regions of interest* in data
- Simplifying optimization (by $f : \mathbb{R}^N \to \mathbb{R}$)
- Model comparison
- Measuring parameters (e.g., $\tan\beta$ in SUSY).

Before I discuss their common theme, I survey some of these methods in a way that emphasizes their differences.

2 SURVEY OF METHODS

This section provides sketches of several multivariate methods. I shall restrict my attention to the problem of discriminating between signal (S) and background (B). The notation used is given in the glossary.

2.1 Fisher Linear Discriminant (FLD)

Purpose

Signal/background discrimination.

Mathematical Object

$\boldsymbol{w} \cdot \boldsymbol{x} + b$ — Best separating hyper-plane between the signal and background classes, that is, the plane that maximizes the between-class distance while minimizing the within-class distances. $\boldsymbol{w}$ is the unit vector normal to the plane and $|b|$ the distance of the plane from the origin.

Algorithm

Let $d(\boldsymbol{w}) = (\boldsymbol{\mu}_S - \boldsymbol{\mu}_B) \cdot \boldsymbol{w}$ be the projected distance between the means $\boldsymbol{\mu}_S$ and $\boldsymbol{\mu}_B$ of the signal and background classes, respectively, along the direction specified by $\boldsymbol{w}$ and let $d_{ij}(\boldsymbol{w}) = (\boldsymbol{x}_i - \boldsymbol{\mu}_j) \cdot \boldsymbol{w}$ be the projected distance between the point $\boldsymbol{x}_i$ and the mean $\boldsymbol{\mu}_j$ of the class j (S or B) to which it belongs. The best separating hyper-plane is found by maximizing Fisher's criterion[2]

$$f(\boldsymbol{w}) = \frac{d^2}{\sum_{i,j=S,B} d_{ij}^2}, \tag{1}$$

with respect to the direction $\boldsymbol{w}$.

2.2 Principal Component Analysis (PCA)

Purpose

- Find linearly uncorrelated variables.
- Reduce dimensionality of data.

Mathematical Object

n orthogonal unit vectors $\boldsymbol{w}_m$ along which the variances are maximal.

Algorithm

1st principal component — For each point $\boldsymbol{x}_i$, compute the distance $d_i(\boldsymbol{w}) = \boldsymbol{w} \cdot \boldsymbol{x}_i$ of the point from the origin along the direction $\boldsymbol{w}$ and maximize $\sum_i d_i^2(\boldsymbol{w})$, that is, the variance, with respect to $\boldsymbol{w}$. This yields the first principal component $\boldsymbol{w}_1$.

2nd principal component — Collapse the distribution onto the plane, passing through the origin, whose orientation is defined by the first principal component $\boldsymbol{w}_1$, that is, perform the operation $\boldsymbol{x}_i \to \boldsymbol{x}_i - d_i(\boldsymbol{w}_1)\boldsymbol{w}_1$. Then, using the transformed points, proceed as for the first principal component. This gives the second principal component $\boldsymbol{w}_2$. The algorithm proceeds recursively until all n principal components have been found.

In practice, one computes the eigenvalues λ_i and eigenvectors $\boldsymbol{v}_i$ of the covariance matrix $\mathrm{Cov}(\boldsymbol{x}) = \mathcal{E}(\boldsymbol{x}\boldsymbol{x}^T) - \mathcal{E}(\boldsymbol{x})\mathcal{E}(\boldsymbol{x}^T)$ of the points $\boldsymbol{x}$. Then, given the transformation matrix $\boldsymbol{T} = \mathrm{Col}(\boldsymbol{v}_i)^T$, whose columns are the eigenvectors, one transforms to the linearly uncorrelated variables $\boldsymbol{u} = T\boldsymbol{x}$.

2.3 Independent Component Analysis (ICA)

Purpose

- Find statistically independent variables.
- Reduce dimensionality of data.

Mathematical Object

n non-orthogonal unit vectors $\boldsymbol{w}_m$ such that the variables defined by displacements along these vectors are as statistically independent as possible.

Algorithm

Assume that $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a linear sum $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$ of unknown *independent* components $\boldsymbol{s} = (s_1, \ldots, s_n)^T$. The *mixing matrix* A is unknown. The aim is to estimate $\boldsymbol{s}$ by finding a *de-mixing matrix* $\boldsymbol{T}$ such that the variables $\boldsymbol{u} = \boldsymbol{T}\boldsymbol{x}$ are as statistically independent as possible. To do so, minimize, with respect to the matrix $\boldsymbol{T}$, the Kullback-Liebler divergence

$$D(f||g) = \boldsymbol{T} \int f(\boldsymbol{T}\boldsymbol{x}) \ln [f(\boldsymbol{T}\boldsymbol{x})/g(\boldsymbol{T}\boldsymbol{x})] \, d\boldsymbol{x}, \tag{2}$$

between $f(\boldsymbol{u})$, the probability density of $\boldsymbol{u}$, and $g(\boldsymbol{u}) = \prod_i f_i(u_i)$, the density formed from the product of the 1-dimensional marginal densities of $f(\boldsymbol{u})$. $D(f||g)$ is zero if, and only if, $f = g$, that is, if the variables $\boldsymbol{u} = (u_1, \ldots, u_n)$ are statistically independent. Note that since the density f is unknown, it too must be estimated.

2.4 Self Organizing Map (SOM)

Purpose

- Find regions of interest in data; that is, clusters.
- Summarize data.

Mathematical Object

n unit vectors $\boldsymbol{w}_m$ such that all vectors $\boldsymbol{x}$ mapped to a given $\boldsymbol{w}_m$ are closer to it than to all remaining $\boldsymbol{w}_m$.

Algorithm

The notion of a self-organizing map was first introduced by Kohenen in 1988. One begins with n randomly directed vectors, usually modeled as a neural network node whose weights are the components of the vector $\boldsymbol{w}_m$ and whose inputs are the components of the vector $\boldsymbol{x}$. Define a distance measure between vectors (either the Euclidean distance $||\boldsymbol{w}_m - \boldsymbol{x}||$ or the vector dot product $\boldsymbol{w}_m \cdot \boldsymbol{x}$.) For each vector $\boldsymbol{x}$ find the closest vector $\boldsymbol{w}_m$ to the vector $\boldsymbol{x}$ (that is, find which neural network node "wins"). One then adjusts the winning vector $\boldsymbol{w}_m$ to be closer to $\boldsymbol{x}$ and adjusts the other vectors by lesser amounts, depending on their distance from the winning vector. This is repeated for each vector $\boldsymbol{x}$ and one cycles through until the vectors $\boldsymbol{w}_m$ are sufficiently stable, that is, their directions change, per cycle, by less than a specified amount.

2.5 Random Grid Search (RGS)

Purpose

Rapid search for optimal cuts[3].

Mathematical Object

Estimates of $P(\boldsymbol{X} > \boldsymbol{x}|S) \equiv 1 - F(\boldsymbol{x}|S)$ and $P(\boldsymbol{X} > \boldsymbol{x}|B) \equiv 1 - F(\boldsymbol{x}|B)$ for the signal and background classes, respectively, where $F(\boldsymbol{X}|S)$ and $F(\boldsymbol{X}|B)$ are the corresponding $d-$dimensional signal and background distribution functions. Given these estimates (in practice, based on signal and background counts) one computes, for each set of cuts, an optimality measure and chooses the cut that is optimal.

Algorithm

For each signal or background event, each characterized by a point $\boldsymbol{x}$, compute $\sum_{\boldsymbol{X}>\boldsymbol{x}_S} w_i$ where $\boldsymbol{x}_S = (x_1, \ldots, x_d)$ are the feature vectors of the *signal* class and w_i are either the signal or background event-by-event weights. $\boldsymbol{X} > \boldsymbol{x}$ represents a set of cuts (called a *cut-point*) defined as follows: $X_1 > x_1, \ldots, X_d > x_d$. In the random grid search the set of cut-points is identical to the points of the signal distribution.

The random grid search is similar to the well-known search over a uniform grid, except that the grid, which is defined by the signal points, has random spacings. The key practical difference is that whereas the uniform grid search suffers the "curse of dimensionality"[1] the time for the random grid search scales like the number of signal points.

2.6 Probability Density Estimation (PDE)

Purpose

- Signal/background discrimination.
- Parameter estimation.

Mathematical Object

Estimates of $p(\boldsymbol{X}|S)$ and $p(\boldsymbol{X}|B)$ for the signal and background classes, respectively.

Algorithm

The idea, introduced by Parzen in the 1960s, is very simple: one approximates an unknown probability density as a sum of probability densities of known functional forms, usually with a few adjustable parameters. The Parzen method

$$p(\boldsymbol{X}) = \sum_n \phi\left(\boldsymbol{X}, \boldsymbol{x}_n\right), \tag{3}$$

estimates $p(\boldsymbol{X}|S)$ and $p(\boldsymbol{X}|B)$ by placing a density ϕ, called a *kernel*, at each point $\boldsymbol{x}_n$ of a sample of points; one sample from the signal class and one from the background class. There is considerable freedom in the choice of kernel function. See, for example, Ref. [4] and references therein. The main mathematical requirement is that $\phi(\boldsymbol{X}, \boldsymbol{x}_n) \to \delta(\boldsymbol{X} - \boldsymbol{x})$ as the sample sizes grow indefinitely. Note, if K is the sample size of each class this method requires evaluating the kernel function K times for each class separately. The computational burden can be reduced by using relatively few densities in the sums and locating them, and choosing their shapes, so as to provide the best estimate of the density $p(\boldsymbol{X})$

$$p(\boldsymbol{X}) = \sum_j \phi(\boldsymbol{X}|j)\omega(j), \tag{4}$$

where $\omega(j)$ are suitably chosen weights. One way to locate the densities is to use a self-organizing map. These PDE algorithms are called *mixture models.*

2.7 Artificial Neural Networks (ANN)

Purpose

- Signal/background discrimination.
- Parameter estimation.
- Function estimation.
- Density estimation.

[1] The time to consider all possible cut-points scales like k^d, where k is the number of bins (in each coordinate) and d is the dimensionality of the feature space

Mathematical Object

Estimate of $\int t\, p(t|\boldsymbol{X})\, dt$, that is, the first moment of the posterior probability density $p(t|\boldsymbol{X})$[5].

Algorithm

Given a sample of pairs of variables $(t_i, \boldsymbol{x}_i)$ minimize the empirical risk function

$$\mathcal{R}_{emp} = \frac{1}{N} \sum [t_i - n(\boldsymbol{x}_i, \boldsymbol{\omega})]^2, \tag{5}$$

with respect to the parameters $\boldsymbol{\omega}$, called *weights*, of a non-linear function $n(\boldsymbol{x}_i, \boldsymbol{\omega})$, called a neural network. If we choose $t = 1$ when $\boldsymbol{x} \in S$ and zero otherwise, the posterior mean $\int t\, p(t|\boldsymbol{X})\, dt$ collapses to the probability

$$P(S|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|S)P(S)}{p(\boldsymbol{x}|S)P(S) + p(\boldsymbol{x}|B)P(B)}, \tag{6}$$

of the signal S given data $\boldsymbol{x}$, where $P(S)$ and $P(B)$ are the prior probabilities of signal and background, respectively[6].

2.8 Support Vector Machines (SVX)

Purpose

- Signal/background discrimination.
- Parameter estimation.

Mathematical Object

$\boldsymbol{w} \cdot \boldsymbol{x} + b$ — Best separating hyper-plane, in a suitably high-dimensional space, between the signal and background classes. $\boldsymbol{w}$ is the unit vector normal to the plane and $|b|$ the distance of the plane from the origin.

Algorithm

Perform a non-linear map $f : \mathbb{R}^d \to \mathbb{R}^{\mathrm{Huge}}$ of data $\boldsymbol{x}$ into a space of sufficiently high dimension, such that the signal and background classes have the best separation that can be had using a hyper-plane. For details see the contribution by Tony Vaiciulis in these proceedings.

3 DISCUSSION

With the exception of PCA, ICA and SOM, each of the methods sketched above effects signal to background discrimination by trying to minimize the probability of mis-classification. Consequently, although in practice each method minimizes a different empirical risk function all such functions are equivalent to minimizing an approximation to the function

$$\epsilon(D) = c(S) \int_{D(\boldsymbol{X})<0} p(\boldsymbol{X}|S)P(S)d\boldsymbol{X} + c(B) \int_{D(\boldsymbol{X})\geq 0} p(\boldsymbol{X}|B)P(B)d\boldsymbol{X}, \tag{7}$$

with respect to the function $D(\boldsymbol{X})$, which is such that $D(\boldsymbol{X}) \geq 0$ leads to the assignment of $\boldsymbol{X}$ to the signal class S and $D(\boldsymbol{X}) < 0$ leads to the assignment of $\boldsymbol{X}$ to the background class B. $P(S)$ and $P(B)$ are the signal and background prior probabilities, respectively, and $c(S)$ and $c(B)$ quantify the cost of each type of mistake: classifying a signal event as background and vice versa. When ϵ is minimized with respect to D one finds that the condition $D(\boldsymbol{X}) \geq 0$ holds if and only if

$$c(S)p(\boldsymbol{X}|S)P(S) - c(B)p(\boldsymbol{X}|B)P(B) \geq 0 \tag{8}$$

while $D(\boldsymbol{X}) < 0$ holds if and only if

$$c(S)p(\boldsymbol{X}|S)P(S) - c(B)p(\boldsymbol{X}|B)P(B) < 0. \tag{9}$$

In other words,

$$\frac{p(\boldsymbol{X}|S)P(S)}{p(\boldsymbol{X}|B)P(B)} \geq \frac{c(B)}{c(S)} \text{ when } D(\boldsymbol{X}) \geq 0 \tag{10}$$

and

$$\frac{p(\boldsymbol{X}|S)P(S)}{p(\boldsymbol{X}|B)P(B)} < \frac{c(B)}{c(S)} \text{ when } D(\boldsymbol{X}) < 0. \tag{11}$$

The inequalities will be satisfied if $D(\boldsymbol{X})$ is a monotonic function of the left-hand side. Any such function will do. Normally one chooses the simplest

$$D(\boldsymbol{X}) = \frac{p(\boldsymbol{X}|S)}{p(\boldsymbol{X}|B)}\frac{P(S)}{P(B)}. \tag{12}$$

The ratio

$$\frac{p(\boldsymbol{X}|S)}{p(\boldsymbol{X}|B)}\frac{P(S)}{P(B)} = \frac{P(S|\boldsymbol{X})}{P(B|\boldsymbol{X})} \tag{13}$$

is called the *Bayes discriminant function* (BDF). The classification rule it determines is called the *Bayes rule*[5] and is closely related to the Neyman-Pearson test for simple hypotheses. (See the glossary for details.)

The point I wish to stress is that the methods described above are merely different algorithms to approximate $D(\boldsymbol{X})$ or some function (or functional) thereof. In some cases, such as the various probability density estimation methods, this is done by calculating estimates of the densities $p(X|S)$ and $p(X|B)$. In other cases, such as artificial neural networks (ANN), some simple monotonic function, e.g., $D/(1+D)$, of $D(\boldsymbol{X})$ is estimated. The random grid search (RGS) estimates integrals of the densities. The Fisher linear discriminant (FLD) can be construed as an estimate of $D(\boldsymbol{X})$ in which each density is approximated by a Gaussian whose covariance matrix is the quadrature sum of those of the signal and background classes. If one allows different covariance matrices for the signal and background classes, one obtains a quadratic discriminant, sometimes referred to as a Gaussian discriminant. The support vector machine method can be regarded as an algorithm to "Gaussianize" distributions by projecting them into a space of sufficiently high dimension wherein the Fisher criterion can be applied to good effect.

There is much debate about which method is superior. Experience suggests that none is superior in every circumstance. For a given problem, a reasonable definition of "best method" is that which yields the most accurate estimate of the physical quantity (or quantities) of interest for a given computational budget. Unfortunately, finding the best method can sometimes be as demanding a computational task as the problem to be solved! One way to reduce the size of this task is to apply each method to a d-dimensional problem with known densities that approximate the actual (but unknown) densities for the signal and background classes. Since the densities are known one can compute the Bayes discriminant function (BDF) and then assess to what degree each method approaches it. But, of course, there is always the pitfall that the densities used fail to approximate the unknown densities well enough to provide a reliable guide as to which method is best for the actual problem to be solved.

The good news is that for the problems addressed in our field the non-linear methods sketched above give approximately the same performance. A far more pressing issue than the choice of method is the absence of a well-founded (that is, non-heuristic) procedure to test whether a d-dimensional density is well-modeled. Typically, what one does is analyze 1-d and 2-d projections and apply standard "goodness of fit" tests to them. However, even if one finds that all such projections fit well this does not guarantee that all is well in d-dimensions. We need a procedure that indicates which sub-sets of the d-dimensional space are poorly modeled so that the sub-sets can be excluded from further consideration.

4 SUMMARY

I surveyed a few methods that either have been, or are likely to be, used by high energy physicists to analyze multi-dimensional data. There is a bewildering number of methods and a seemingly endless debate about which method is best. Rather than engage in abstract debate, it is more useful to follow Rudy Bock's[1] systematic approach: try a few methods and see how they perform. But, in the end, it should be borne in mind that each of these methods approximates the same mathematical object. Therefore, to the degree that the methods are properly applied (and sufficiently non-linear) they must give about the same results, modulo the usual uncertainties.

Links to web articles describing some of the methods discussed in this paper can be found at the site of the Fermilab Run II Advanced Analysis Group[7].

ACKNOWLEDGEMENTS

I thank all participants for their efforts in making this an interesting conference. I also thank my colleagues of the Run II Advanced Analysis Group at Fermilab, in particular, Pushpa Bhat. This work was supported in part by the U.S. Department of Energy.

References

[1] See the contributions by Bock, Hakl, Marcin, Towers and Vaiciulis in these proceedings.

[2] S. E. Fienberg and D. V. Hinkley, eds., *R. A. Fisher: An Appreciation (Lecture Notes on Statistics, Vol. 1)* (Springer Verlag, 1990).

[3] H. B. Prosper *et al.,* DØ Collaboration, in *Proc. Int. Conf. on Computing in High Energy Physics '95*, Rio de Janeiro, Brazil (World Scientific, River Edge, New Jersey, 1996).

[4] S. Towers, these proceedings.

[5] C. M. Bishop, *Neural Networks for Pattern Recognition* (Clarendon Press, Oxford, 1998).

[6] D. W. Ruck *et al.*, IEEE Trans. Neural Networks **1(4)** (1990) 296; E. A. Wan *et al.*, IEEE Trans. Neural Networks **1(4)** (1990) 303.

[7] `http://projects.fnal.gov/run2aag/`