Eur. Phys. J. C 46, s02, 1–11 (2006)
Digital Object Identifier (DOI) 10.1140/epjcd/s2006-03-001-7

EPJ C direct
electronic only

Young Scientist

Development of a method for fast, easy and optimized tuning of selection cuts

Modification of the Fisher Linear Discriminant Analysis for use in the low signal-to-noise environments

J. Faivre[a]

INFN sezione di Padova, Padova, Italy

Received: 30 June 2006 /
Published online: 3 August 2006 –

Abstract. In situations where the signal of the analysed particle is tangled up with orders of magnitude more background, its analysis may benefit from the use of a pattern classification method to discriminate the signal out of the background candidates. We present and explain the basic Linear Discriminant Analysis and the modifications brought – the use of cascaded cuts and of a locally optimized criterion – to adapt it to the conditions encountered in the field of heavy ion Physics. We show that this optimized multicut Linear Discriminant Analysis has a higher performance than classical selection cuts and provides a very fast and easy selection cut optimization.

1 Introduction

Particle search in a collision event consists in discriminating the signal (what is wanted) and the background, or noise (fake candidates). This *pattern classification* is achieved through the measurement of several characteristics for each candidate, herein called *cut variables* or *observables*, and the discrimination relies on the fact that the probability distributions of these characteristics are different for the signal and the background.

The simplest method consists in applying a "straight selection cut" on each of these variables separately, as shown in the top of Fig. 4. We will refer to this approach as *classical analysis* or *classical cuts*. Because these – numerous – cut values are considered as independent parameters while those variables are generally correlated, this method may be very long to tune and usually provides an improvable discrimination. In this article, we will describe the adaptation of Fisher Linear Discriminant Analysis (Fisher-LDA), a pattern classification method widely used in data processing, to the extreme signal-to-noise conditions of the relativistic heavy ion collisions, and show its advantages over the "classical analysis"

The first section explains why such methods are needed for heavy ion Physics, and gives examples. The second section is a short introduction to pattern classification. Fisher-LDA and its modifications are presented in the third and fourth sections. Finally, the last section explains how the final multi-variable cut is tuned and used in practice, and shows some results. Duda et al. and Faisan [1, 2] have helped writing Sects. 3.1 and 4.2.

The method described in this paper has been implemented as a plug-and-play C^{++} class. Its source code and documentation are available upon request to the author.

2 Low signal-to-noise environments

As an example of low signal-to-noise environment, we introduce here briefly the context which led us to use the LDA method. More information about (ultra-)relativistic heavy-ion collisions can be found elsewhere, e.g. [3–5].

2.1 Relativistic heavy-ion collisions

Lattice-QCD predicts that when the energy density of a strongly interacting system of hadrons is large enough, matter should undergo a phase transition from a hadronic state to a quark-gluon plasma (QGP), in which the degrees of freedom are partonic. Parton deconfinement is made possible in such a medium by the strong force screening resulting from the high density of colour charges, similarly to the Debye screening in an electromagnetic plasma.

[a] e-mail: julien.faivre@pd.infn.it

In this purpose, (ultra-)relativistic heavy ion (Pb–Pb, Au–Au) collisions are made, with $\sqrt{S_{NN}}$ currently ranging from a few GeV (AGS) to 200 GeV (RHIC), and up to 5.5 TeV when LHC starts to run.

Some of the probes of the QGP are based on strangeness and charm, absent in the initial state of the collision. Hadrons containing those quarks decay weakly, which enables an identification up to transverse momenta in the pQCD domain.

Those particles are best studied by reconstructing topologically their secondary decay vertex. This can be achieved in the detectors STAR at RHIC [6], and ALICE at LHC [7], thanks to the tracking subdetectors: in both cases, a time projection chamber (TPC) surrounding several layers of silicon detectors makes possible the reconstruction of the trajectory of the charged particles and their extrapolation towards the primary vertex.

2.2 Cut variables

This paragraph gives examples of cut variables which can be used for discriminating between the signal and the background (fortuitous associations of tracks) in the case of a $\Lambda^0 = uds \rightarrow p\pi^-$ analysis by topological reconstruction.

A weak decay is characterized by a sizeable decay length ($c\tau$ of a hundred microns for charm decays, of a couple of centimeters for strange decays). The reconstruction of a neutral particle decay (V0 vertex) is made by examining all combinations of pairs of opposite charge tracks, and filtering out those (background) which have a geometry incompatible with that of a real particle (signal).

In reality, because of the finite resolution of the detectors, the reconstruction is not perfect and real particles and a significant fraction of the background have a similar geometry. This makes the discrimination challenging, and achievable only statistically: the candidates selected as signal are *mostly* signal, those which are filtered out are *mostly* background. The proportion of signal kept or rejected by the selection process can be estimated by simulation studies for instance.

The projection in two dimensions of the geometry of a V0 vertex is shown in Fig. 1. The charged tracks are bent by the magnetic field (here perpendicular to the figure plane), and because the reconstruction is imperfect, the tracks of the two decay daughters do not cross and the trajectory of the reconstructed parent particle does not meet the primary vertex.

The decay length, the distances of closest approach between the tracks, or between a track and the primary vertex, constitute geometrical variables which can be used to discriminate the background and the signal. Most of these variables are correlated, e.g. the distance of closest approach between a daughter track and the primary vertex is correlated with the Λ decay length.

The cosine of the decay angle ($\cos\theta^*$) is also often used to eliminate the background: the distribution of this variable shows strong peaks at -1 and $+1$ for the background. Examples of other cut variables may be found in [8], and in [9, 10] for other analyses.

2.3 Examples of signal-to-noise ratios

Heavy-ion collisions make topological reconstruction of the weak decays a challenging task, because of the high charged track density (multiplicity) in the detectors. The amount of background for a 2-particle decay scales with the square of this multiplicity, while for of a 3-particle decay it scales with its cube.

For the case of the $\Omega^- = sss \rightarrow \Lambda^0 K^-$ in STAR's central collisions, the yield of about 0.6 $\Omega + \overline{\Omega}$ per event [11] and the multiplicity of more than 3000 tracks give an initial signal-to-noise ratio[1] only slightly above 10^{-10}. At the reconstruction stage[2], loose cuts are applied to reduce the computing time and the disk space taken by the storage. While these cuts remove 99.99% of the combinatorics, the signal-to-noise ratio is still as low as 10^{-6}.

For the $D^0 = c\overline{u} \rightarrow K^-\pi^+$ in ALICE, the initial signal-to-noise ratio is of the order of 10^{-8} [12]. Although this is higher in value than for the Ω, the fact that the signal and background distributions of the geometrical variables differ more in the case of the Ω than in that of the D^0 makes the latter more difficult to reconstruct than the Ω.

Analyses in such extreme conditions, also encountered in the fields of top quark analysis or Higgs search, benefit from the advantages brought by the pattern classification methods. Other fields – industry, health, image processing in general – do not deal with such situations, but rather with poor training statistics and high numbers of observables and/or classes. The methods created for their needs therefore do not meet ours, which made necessary the development of a method adapted to our conditions.

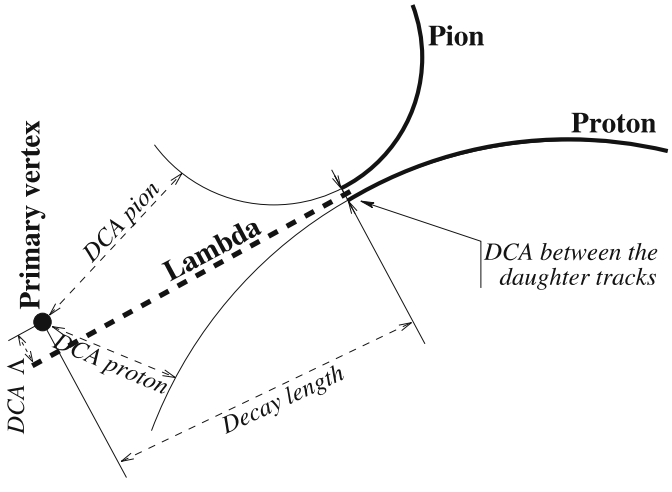

Fig. 1. 2-dimensional geometry of a V0 vertex. The trajectory of each of the daughter particles is a *thick solid line*, the extrapolations towards the primary vertex are *thin solid lines*. The trajectory of the reconstructed parent particle is the *thick dashed line*. DCA stands for "distance of closest approach"

[1] Here, not calculated in an invariant mass window selecting the signal peak. It can therefore not be compared with the numbers given in Fig. 6.

[2] Reconstruction of the secondary decay candidates from the tracks, themselves reconstructed from the hits in the detectors.

3 Pattern classification

3.1 Short introduction and general procedure

Pattern classification consists in classifying an object (a candidate) in a category (class). The input data are generally: p classes of objects (e.g. signal, noise), n observables defined for all the classes (n is thus the dimension of the space with which we will work), and, for each of the p classes, a sample of N_k objects for training and test, k being the class index. These notations will be kept in the rest of the article. The observables used may be chosen amongst the parameters which are directly measured (see Sect. 2.2 for an example), or may be calculated from these.

The aim is the creation of an algorithm which is able to classify an object into one of the classes defined. A training (or *learning*) phase first optimizes (tunes) the method's parameters until a maximum of candidates whose class is known are classified correctly. A new object, the class membership of which is unknown, can then be presented to the algorithm for classification.

Figure 2 describes the data classification process. The phase involving the detectors is the data collection. In our case, it is the collection of the hits in the subdetectors for example. The phases of segmentation and feature extraction transform this low-level information into mid-level information, smaller in size but more suited to distinguish various classes. For us, segmentation corresponds for example to the track and vertex reconstruction, and feature extraction is the calculation of the various cut variables of the candidates.

Sorting is the phase in which pattern classification methods are involved. It consists in calculating, from the previously mentioned mid-level information, high-level information: only a handful of variables – or even just one – but which contain the relevant information to distinguish signal and background. They are those used for the discrimination between the classes. At this stage, two objects can be compared. Yet, the final decision – classifying the candidate into one of the classes – can be taken only after the post-treatment phase, which takes into account an efficiency and a background rejection, via the minimization of a cost, in the calculation of the decision.

In our case, the number p of classes is two, hereafter called *signal* and *background* (or *noise*) and indexed by $k \in \{1; 2\}$. The signal is made of the real particle, while the background is made of all the other candidates (e.g. combinatorial association of tracks, in the case of particles which decay into two or three daughters).

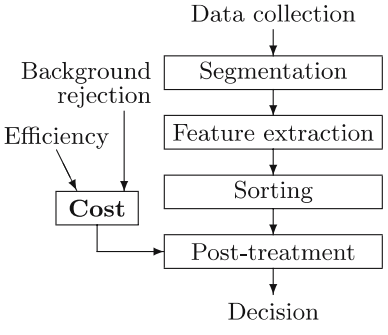

Fig. 2. Data classification process

3.2 Comparing the performance of various methods

In this paragraph, S and N will refer respectively to an amount of signal and of noise left when cuts are applied.

Here are some variables that can be used as indicators of cuts' performance:

- *Amount of signal* S
- *Efficiency* or *sensitivity* or *detection probability* $\varepsilon_S = \frac{S_{\text{post-cuts}}}{S_{\text{pre-cuts}}}$: proportion of signal that is kept by the cuts
- *Background rejection* $1-\varepsilon_N = \frac{N_{\text{removed by cuts}}}{N_{\text{pre-cuts}}}$: proportion of background that is rejected by the cuts
- *Signal-to-noise ratio* S/N
- *Purity* or *specificity* $\pi_S = \frac{S}{S+N}$: proportion of kept candidates that actually *are* signal
- *False alarms rate* $\frac{N}{S+N}$: proportion of kept candidates which are actually background
- *Significance* $S/\sqrt{N}$ or $S/\sqrt{S+N}$
- *Relative uncertainty* σ_S/S, where σ_S is the error on S

Our cost function will be the relative uncertainty, as it is the indicator that directly guarantees the smallest possible statistical error on the result[3]. To determine the performance of a method or to compare various methods, it is common to show other indicators as well, usually by pairs:

- Signal with respect to the signal-to-noise ratio
- Signal with respect to purity: this diagram is strictly equivalent to an efficiency-purity diagram, and also strictly equivalent to the diagram mentioned below
- Efficiency with respect to the false alarms rate: this diagram is called "ROC curve" (Receiver Operating Characteristic)
- Relative uncertainty with respect to signal

In such diagrams, all the points that are reachable with a given method, by changing the cuts, define a region, which may be a surface (case of the classical cuts) or a curve (case of most of the other methods). In a signal-S/N or an efficiency-purity diagram, a movement along the curve (or along the border of the surface) inducing an improvement of one of the variables results in a deterioration of the other one. In a diagram showing the relative uncertainty versus the efficiency, the curve is a decreasing, then increasing function which has a global minimum. The latter corresponds to the searched optimal cut.

The performance of a method can then be defined as the minimal relative uncertainty achievable. In other fields though, using the relative uncertainty as the cost function may be irrelevant. In the case of tumor detections for example, one certainly wants to use the tumor detection probability, and set the cut value to a high probability: this selects more background but is safer.

[3] Thus no discrimination criterion will be defined here to compare various pattern classification methods.

3.3 Some pattern classification methods

Many types of pattern classification methods exist, each of them having several subtypes. We can e.g. mention the Markov fields, the nearest neighbours methods, the trees, the Parzen windows, the discriminant analyses or the neural networks, on top of unsupervised learning methods which are able to determine themselves how many classes are dealt with. Details can be found in [1]. Some methods have already been applied to particle Physics [9, 13–17]. Discriminant analyses themselves can be subdivided into Linear, Quadratic and higher orders Discriminant Analyses, each with several criteria usable for the training, and each with additional possibilities of non-linear data transformations, space dimensionality expansion or reduction, entangled with the basic method.

A common property of all of them is their ability to better discriminate between the signal and the background with respect to the classical method, and therefore to provide better results.

Many of the methods share a second essential advantage over the classical cuts: they provide a transformation of the n-dimensional space of the cut variables to a single output value, and can be seen as functions defined from $\mathbb{R}^n$ to $\mathbb{R}$. While using the classical cuts consists in minimizing a function (the relative uncertainty) of n variables, most pattern classification methods reduce the problem of cut-tuning to the minimization of an only 1-dimensional function.

We decided to give a try to linear discriminant analysis (LDA). Table 1 compares several characteristics of the classical cuts, of the LDA method developed in this article (multicut-LDA, cf. Sect. 5), and of the artificial neural networks (ANN), the latter being probably the most used pattern classification method in particle Physics. The positive characteristics are emphasized in bold.

The choice of LDA over a higher order discriminant analysis or over a pattern classification method like the neural networks is justified by its extreme simplicity, which has direct consequences like a better control of how data are handled and selected, as well as a large gain in the amount of time spent on setting up and tuning the method [17–19]. Moreover, although the ANN should reach a higher performance than LDA in theory, choosing the right configuration, the right values of the free parameters and the training method is far from trivial, and in fact may rapidly result in lower performances than what could be expected.

Neural networks also suffer from the huge background statistics: they focus on removing its overwhelming part and leave untouched the comparatively small amount of background which is close to the signal area. This problem can be avoided by cascading several neural networks (it can be seen as an equivalent of multicut-LDA, which naturally solves the problem), each stepping up the S/N ratio by an order of magnitude, but at the cost of an exploding number of parameters of the method: it scales with n^2, while that of LDA scales with n.

The performances of cascaded neural networks, multicut-LDA and classical cuts have been compared for $\Lambda^0 \to p\pi^-$ topological decays in ALICE. Although almost an order of magnitude more time had been dedicated to the ANN, LDA achieved a better performance [18, 19].

For all methods, a test phase is essential to obtain an algorithm which works well, as its performance is not the same if calculated on the training sample or on an independent test sample. The latter performance is always worse than the former, which is biased since the cuts have been optimized for the training sample.

This is illustrated by Fig. 3, in which the distributions of two classes are shown for the *training* sample, as well as three examples of border: a straight line $\mathcal{L}$, a simple curve $\mathcal{C}$ which describes a bit better the boundary between both classes, and a complex parametrization $\mathcal{P}$ that describes the samples almost candidate-by-candidate.

The result of those boundaries on a test sample will be very different: the line will have a fair performance and the simple curve will have a better one, but the performance of the complex curve will be bad, because, while two sam-

Table 1. Comparison of the characteristics of classical cuts, ANN, and the multicut-LDA method presented in this article

	Classical cuts	**Multicut-LDA**	Neural networks
Setting up of the method	**Trivial, fast**	**Easy, fast**	Complex, long
Nb. of free parameters to be chosen	Several (or **none**[a])	**One**	Several
Training	**None** (or long and complex)	**Simple**	Complex Overtraining
Nb. of parameters tuned during the training	**None** (**or few**[a])	**Few**	Many
Clarity of the analysis and data treatment	**Under control**	**Under control**	"Black box"
Linear treatment of the observables	**Yes**	**Yes**	No
Gives optimized cuts (in the method's scope)	No[a]	**Yes**	**Yes**
Final tuning to minimize the cost function	Complex, long	**Simple, fast**	**Simple, fast**
Shape of the boundary signal/background	Linear	**Linear, but multicut ⇒ OK**	**Non linear**
Volume selected as being signal	Convex	Convex	**Non connex**

[a] A maximization algorithm of the n-dimensional function can always be set up, but the complexity and processing time is prohibitive as soon as more than half a dozen variables are used [18]

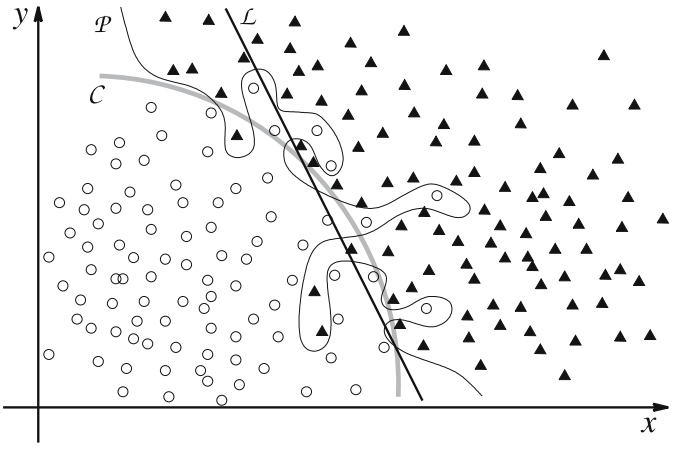

Fig. 3. *Training* samples and shape of various boundaries

ples are globally close to identical, they have yet significant local differences. This phenomenon is called *overtraining* when $\mathcal{L}$, $\mathcal{C}$ and $\mathcal{P}$ are actually obtained with the same pattern classification method.

The performance therefore always has to be evaluated on a sample that is independent of the training sample. Because they are able to select very sophisticated shapes in the variables space, fancier, non-linear methods are intrinsically prone to overtraining, while basic LDA is not. We will show that the LDA method developed here is also not, provided a simple condition is respected.

4 Fisher Linear Discriminant Analysis

4.1 Basic principle of LDA

The principle of the LDA method is illustrated by the two drawings of Fig. 4. It has been supposed that two observables, x and y, were accessible to the observer, and the signal and background candidates have respectively been attributed opened circles and closed triangles.

The first drawing shows the behavior of the classical cuts, i.e. straight cuts on one or several of the observables. It has to be kept in mind that we are interested in applications where the number of background candidates is much higher than that of signal candidates. When the cuts are chosen loose for the efficiency to be high (solid thick lines, the eliminated region is in grey), the contamination of the signal by the background is high. Tighter cuts (dashed thick lines, the additional region cut is hatched in black) drastically reduce the background, but the price to pay is a small efficiency.

LDA consists in cutting along a linear combination of all the observables, rather than along each of the observables. This linear combination is defined by an LDA direction (or axis). The result, shown in the bottom plot, is a better discrimination between the classes. In an efficiency-purity diagram for example, this translates into a more interesting position than any of the positions accessible with classical cuts.

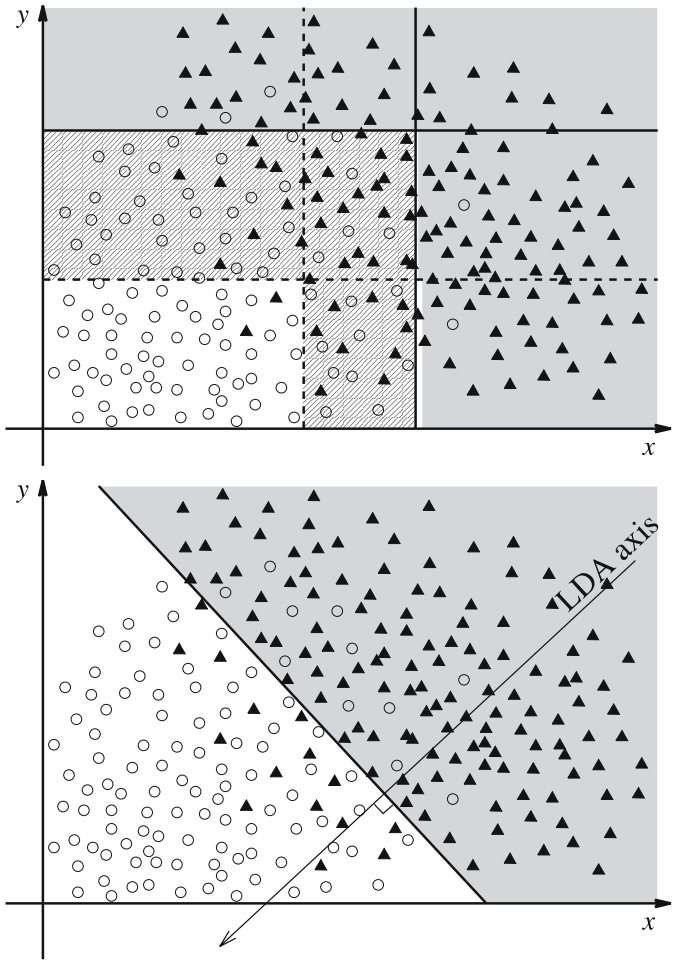

Fig. 4. Basic principle of LDA: example with two variables. *Top plot*: loose (*solid line* and *grey area*) and tight (*dashed line* and *hatched area*) classical cuts. *Bottom plot*: LDA cut

The algorithm consists in calculating the direction of the axis that gives an optimal discrimination between the classes according to a given criterion. After this training phase, a cut on the axis's coordinate minimizing the cost function is then chosen, which defines as border between both classes a hyperplane perpendicular to the axis.

4.2 Fisher criterion

The most frequently used criterion for the calculation of the axis's direction is the Fisher criterion, which results in what is called Fisher-LDA, introduced by Fisher in 1936 [20]. The advantage of the Fisher criterion is that, on top of being easy to settle, it gives the exact expression of the direction of the LDA vector, without need for a numerical optimization. There is indeed maximization, but the solution is analytical.

Let's call $\mathcal{D}_k$ the training samples (with e.g. $k = 1$ for the signal and $k = 2$ for the background). The Fisher criterion consists in requiring the best separation of the projections of the classes on an axis Δ defined by $\mathbf{u}$, i.e. the

averages μ_k of those projected distributions should be as far as possible from one another relatively to their squared widths $\sigma_k^2 = \sum_{\mathbf{x}\in\mathcal{D}_k}(\mathbf{u}\cdot\mathbf{x}-\mu_k)^2$, for the overlap between the distributions to be minimal. The criterion to be maximized is:

$$\lambda(\Delta) = \frac{|\mu_1(\Delta)-\mu_2(\Delta)|^2}{\sigma_1^2(\Delta)+\sigma_2^2(\Delta)}\,. \tag{1}$$

Let $\mathbf{x}$ be an observation (so an n-coordinate vector). The normalized n-coordinate vector $\mathbf{u}$ which drives the line Δ characterizes, together with the cut value defined on the axis's coordinate – that is to say, on the scalar product $\mathbf{x}\cdot\mathbf{u}$ –, the hyperplane which plays the role of a border between both classes.

Let's call $\mathbf{m}_k$ the n-dimensional averages of the distributions and write ${}^tM$ for the transposed matrix of a generic matrix M. With $S_B = (\mathbf{m_1}-\mathbf{m_2})\cdot{}^t(\mathbf{m_1}-\mathbf{m_2})$ the between-class scatter matrix, $S_k = \sum_{\mathbf{x}\in\mathcal{D}_k}(\mathbf{x}-\mathbf{m}_k)\cdot{}^t(\mathbf{x}-\mathbf{m}_k)$ and $S_\mathrm{W} = S_1+S_2$ the within-class scatter matrix, we can write the Fisher criterion matricially:

$$\lambda(\Delta) = \lambda(\mathbf{u}) = \frac{|\mu_1-\mu_2|^2}{\sigma_1^2+\sigma_2^2} = \frac{{}^t\mathbf{u}S_B\mathbf{u}}{{}^t\mathbf{u}S_\mathrm{W}\mathbf{u}}\,. \tag{2}$$

Maximizing this expression can be done analytically by using the Lagrange multiplier method. A vector $\mathbf{u}$ maximizing (2) obeys: $\exists\,\omega\in\mathbb{R}/S_\mathrm{W}^{-1}S_B\mathbf{u} = \omega\mathbf{u}$. ${}^t(\mathbf{m_1}-\mathbf{m_2})\cdot\mathbf{u}$ being a scalar, $S_B\mathbf{u}$ is always collinear to $\mathbf{m_1}-\mathbf{m_2}$, and the expression giving $\mathbf{u}$ becomes:

$$\exists\,\xi\in\mathbb{R}/S_\mathrm{W}^{-1}(\mathbf{m_1}-\mathbf{m_2}) = \xi\mathbf{u}\,. \tag{3}$$

The problem is therefore reduced to a matrix inversion.

4.3 Problems with Fisher-LDA

Using the Fisher criterion, even though it is satisfactory for many applications, raises problems in our case. The fact that Fisher relies only on the mean and width of the distributions makes it a "global" criterion, hardly sensitive to the local features of the distributions. The Fisher approach can not succeed in our situations, where the initial S/N is extremely small and the background populates the whole space, including all the signal area. A better discrimination between signal and background requires a local description of the zones where the signal lies and where the background has to be cut. The next section describes how LDA can be improved to meet this requirement without resorting to non-linearity.

5 Optimized multicut-LDA

5.1 Multicut-LDA

A first modification brought to LDA to better cope with the low S/N environments is the application of several successive LDA cuts. This also allows for a finer description of the non-linear boundary between the classes, the same way as a circle can be approximated by an n-sided polygon, all the better as n is high, and yet keeping linear properties to some extent.

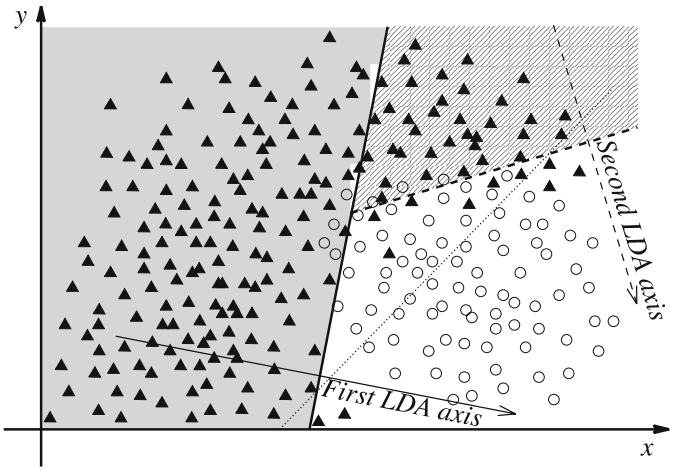

Fig. 5. Mechanism of the multicut-LDA method. The first LDA cut is the *thick solid line* and eliminates the *grey area*. The second LDA cut is the *thick dashed line* and filters out the *hatched area*. If only one cut had to be used to reach a similar background rejection, its efficiency would be much lower, as shown by the *dotted line*

The mechanism of this method, which will be called *multicut-LDA*, is depicted in Fig. 5. The first LDA direction is determined by a learning phase using all the candidates of both samples. A cut value is then determined according to a criterion which will be described in Sect. 5.2, with an efficiency on the signal close to 100%. This first cut is applied to the learning samples, and a second LDA direction is calculated with the remaining candidates. The process is then repeated until not enough candidates remain in the training samples to calculate more LDA axis.

Multicut-LDA therefore provides a set of LDA directions, each being a vector $\mathbf{u}_i$ of the space of the observables (n coordinates). It also provides a cut value c_i associated to the direction $\mathbf{u}_i$. The value of c_i depends on $\mathbf{u}_i$, and the direction $\mathbf{u}_i$ is a function of $\mathbf{u}_{i-1}$ and c_{i-1}. Each pair (direction, cut) defines a hyperplane, and this set of hyperplanes demarcates a connex and even convex shape, by construction, in which the candidates are considered as being signal by the algorithm. Further studies can estimate the amount of background selected as signal.

5.2 Optimized criterion

Multicut-LDA can be improved by replacing the Fisher criterion by another one, which takes the local – and not global – behaviour of the distributions into account, therefore more adapted to the multicut method.

Here are two such criteria:

– Optimized criterion I: given an efficiency of the LDA cut on the signal, maximization of the amount of background removed;

– Optimized criterion II: given a background rejection of the LDA cut, minimization of the amount of signal removed.

Their formulation is antisymmetric for the signal and the background, but, as the criterion II requires a prohibitive computing time to be run, we have not tested if these two criteria give a similar performance. We therefore used only the criterion I.

5.3 Function to maximize

Contrarily to the Fisher criterion, using an optimized criterion requires the implementation of a maximization algorithm, here expressed for the criterion I.

Let ε_{S_i} be the given efficiency (chosen) of the i-th cut on the signal and $\mathcal{D}_{S_i}$ the set of candidates of the signal sample after the $i-1$ first cuts. If 1 is assigned to *true* and 0 to *false* in the sum in (4), the number of signal candidates removed by cutting at value c_i along the axis $\mathbf{u}_i$ is:

$$S_i - S_{i+1} = (1-\varepsilon_{S_i})S_i = \sum_{\mathbf{x}\in\mathcal{D}_{S_i}} (\mathbf{x}\cdot\mathbf{u}_i < c_i)\,, \qquad (4)$$

with S_i the number of signal candidates (in the training sample) used to determine the i-th direction. The value of c_i can also be determined so as to obey the following equality:

$$1 - \frac{\sum_{\mathbf{x}\in\mathcal{D}_{S_i}} (\mathbf{x}\cdot\mathbf{u}_i < c_i)}{S_i} = \varepsilon_{S_i}\,. \qquad (5)$$

Using directly the number $S_i - S_{i+1}$ to calculate c_i is however more judicious than using the efficiency, because it allows to control the "locality degree" of the optimized criterion. This "locality degree" is determined by a comparison of the numbers of candidates which are removed with two numbers. 1°) The number of signal candidates removed has to be larger than the typical size of the statistical fluctuations for a sample of size S_i, for the algorithm not to trigger on one of those fluctuations. The number of candidates removed should not be too much above this threshold though, as the efficiency of the cut should be kept close to 100% to take all advantage of the multicut method. 2°) The numbers of signal and background candidates which are removed have to be higher than a fixed absolute number which ensures that the candidates that are removed are numerous enough to be really representative of the actual shape of the distributions in the area that is cut.

In our studies, in 25 dimensions, values of 500 signal candidates removed and above were satisfactory, for samples of 10 000 to 100 000 signal candidates.

Respecting this simple condition guarantees that there is no overtraining, so this problem can basically be considered as absent from the optimized multicut-LDAmethod.

The function f that is maximized is of course the number of background candidates that are removed by the cut; hence we can write:

$$\begin{aligned} f:\quad & \mathbb{R}^n \longrightarrow \mathbb{N} \\ & \mathbf{u}_i \longmapsto \sum_{\mathbf{x}\in\mathcal{D}_{B_i}} (\mathbf{x}\cdot\mathbf{u}_i < c_i)\,, \end{aligned} \qquad (6)$$

where $\mathcal{D}_{B_i}$ is the set of candidates of the background sample after the $i-1$ first cuts.

The efficiency ε_{S_i} being fixed (it is a chosen parameter), the optimization consists in maximizing f as a function of $\mathbf{u}_i$. As the value of c_i depends on $\mathbf{u}_i$, it needs to be recalculated at each step.

For the results presented in the next section, the algorithm chosen to maximize f consists in varying each coordinate of the vector $\mathbf{u}$ at a time, moving the vector by a given angle α (for example 8°). The first coordinate is changed until f has reached a maximum, then the second coordinate is changed, etc. ... When all n coordinates have been changed, the process is repeated until the vector does not move anymore. Then α is divided by two and the whole algorithm is repeated. One keeps dividing α by two until the gain in number of background candidates removed becomes null or unsignificant.

A common problem to many maximization algorithms is the possibility of being trapped in a local maximum. In our case, the problem is partially resolved by the initial condition: the natural start vector for this algorithm is the direction found with the Fisher criterion, which guarantees that the final result will necessarily be better than with the Fisher criterion. We observed an average improvement with respect to it of around 50% more background candidates cut per LDA cut, although with strong variations. One can also implement a genetic algorithm, which in principle converges to the global maximum [21].

5.4 Size of the training samples

Determining the minimal statistics needed for the calculation of the LDA directions can be done by calculating the performance for various sizes of the training samples. The performance should rise, with possible oscillations, when the size of the training samples is increased, and saturate when the latter reaches the minimal size necessary for a good determination of the LDA directions.

A lazier way to do, but a priori as reliable, is to check that the performance of the i-th LDA cut is better than that of the $i-1$-th cut tightened beyond the cut value at which the i-th cut should begin to be applied. If it is not the case, it is a strong indication that the statistics used to determine the i-th direction was not sufficient.

Preliminary studies done with the "lazy method" indicate that 2000 candidates in each sample, possibly even less, are already enough. This was done with 10 dimensions and is not expected to change with the number of dimensions.

Unlike for Fisher-LDA (see (1)), when the optimized criterium is used the relative proportion of signal and background candidates in the training samples has no importance.

5.5 Variables used

Because multicut-LDA selects a convex (and therefore connex) signal area, the signal's distributions have to show only one main peak whenever possible, for the method to be efficient.

It is also preferable to use reasonably well shaped distributions, e.g. an angle value is better than using its cosine, as the cosine function will flatten everything towards 1, which may cause the algorithm to fail using that variable in the optimization (the peak would be extremely narrow). Normalizing the distributions' shapes eliminates this constraint, but is a non trivial issue.

The number of variables to use is a study by itself, although not very time-consuming because it should simply be as high as possible, so as to reach the highest possible discrimination. Non-linear combinations of the initial cut variables can be added. The number of variables to use can yet be limited by statistics and processing time reasons – numbers of a few tens are not critical. Methods exist to reduce this number of variables while avoiding a drop in discrimination (see [8]): under-optimal LDA, Principal Component Analysis (much faster, but unsupervised, matrix-based analysis which reduces the dimension of the working space by taking out the least informative dimensions), or fractional-step LDA [22] (performs dimensionality reduction while keeping a better discrimination than Principal Component Analysis). Principal Component Analysis provides a partial solution to the normalization problem by giving a new base of the space.

6 LDA practical guide and results

6.1 Results on real data

Optimized multicut-LDA has been tested with 25 cut variables on the Ξ and Ω multi-strange baryons in the 200 GeV Au–Au collisions in STAR and has shown to provide more precise results than the classical cuts [11]: the statistical error on the production yields was 25% (Ξ) to 40% (Ω) larger with the classical cuts than with LDA. Furthermore, the transverse mass spectra obtained with LDA had small enough error bars to rule out the formula that was then commonly used to fit it.

The method has then succesfully been applied to STAR's 62 GeV Au–Au collisions [23], and has proven one of its advantages: while classical cuts had to be tediously re-tuned to be adapted to the new – lower than at 200 GeV – track multiplicity, a simple loosening of the LDA cuts calculated in [11] provided, with respect to the re-tuned classical set of cuts, 75% more signal and a relative uncertainty on the raw amount of signal 1.6 times lower, as shown in Fig. 6. This loosening of the LDA cuts is made by plotting the new valley-shaped curve (relative uncertainty versus efficiency) and determining its minimum. One could also have re-calculated LDA directions with 62 GeV simulation to have possibly more optimized LDA cuts.

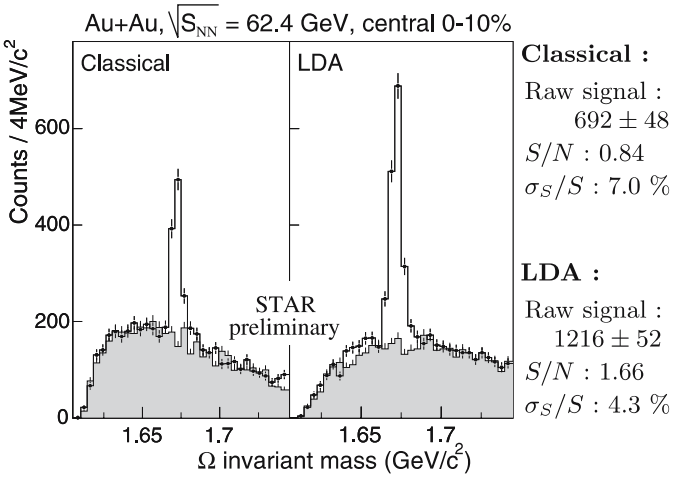

Fig. 6. STAR preliminary $\Omega + \overline{\Omega}$ invariant mass distributions for central Au–Au collisions at $\sqrt{S_{NN}} = 62\,\mathrm{GeV}$, with classical cuts (*left*) and LDA cuts (*right*) [24]. The *grey histograms* show the amount of background estimated by rotating

6.2 How to do the final tuning

The final tuning consists in finding how many LDA directions have to be applied, and how much is the cut value of the last one, to reach the minimum of the relative uncertainty. In this process, those of the last directions which have not been calculated with enough candidates in the training samples should be discarded. The way the final tuning is done can be illustrated by the example of the $D^0 \to K^-\pi^+$ study in ALICE, in central Pb–Pb collisions at $\sqrt{S_{NN}} = 5.5\,\mathrm{TeV}$. As no data have been taken yet, the results shown come from a simulation sample independent of the training sample, and the relative and absolute amounts of signal and background have been rescaled to match the amounts that are expected to be found in 10^7 real events. The number of cut variables used is 10, and the number of signal candidates of the training sample to be removed by each LDA cut has been set to 499 (the candidate with $\mathbf{x} \cdot \mathbf{u}_i = c_i$ is not removed), for an initial sample size of 15 967. One may also choose to remove 500 D^0 for the two first cuts, and then 1000 or more for the following ones, which have a lower background rejection factor. This strategy was used in [11], as the initial signal training sample was larger.

Figure 7 shows our cost function: with respect to the amount of signal that passes the cuts, the relative uncertainty on that variable. The valley is clearly visible, and finding its minimum is a trivial task. In the zoom presented in the inset, black squares have been put when an additional LDA cut was applied: the rise in performance is very visible, in the form of a steeper slope with an additional direction (left of a point) than without (right).

The corresponding efficiency-purity plot is shown in Fig. 8. Such plots can be used when a different cost function is wanted. One may for example wish to have a higher background rejection to avoid problems due to the background estimation: here, the proportion of background falls down by a factor of more than 3.5 for the same effi-

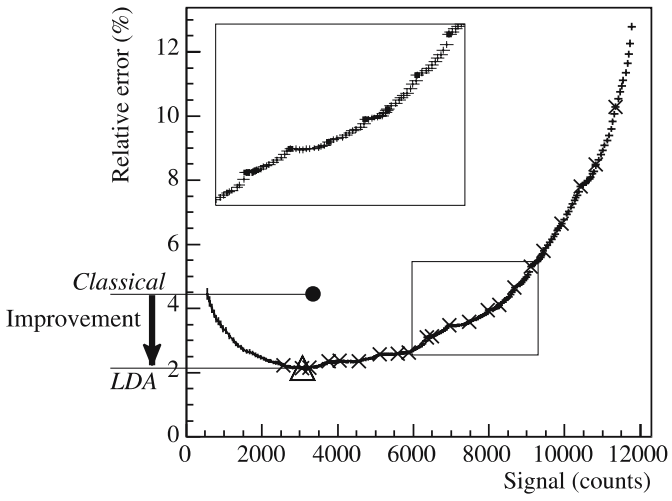

Fig. 7. σ_S/S of the D^0 as a function of the amount of signal, for central Pb–Pb collisions at $\sqrt{S_{NN}} = 5.5$ TeV in ALICE. The *closed circle* shows the performance reached with the current classical cuts [25], and the *valley-shaped curve* is the locus of the points described when the LDA cut is gradually tightened (right to left). The *crosses* (×) mark each addition of a new LDA direction; the *open triangle* shows the minimal value reached. A zoom on the *boxed area* is shown in the *inset*

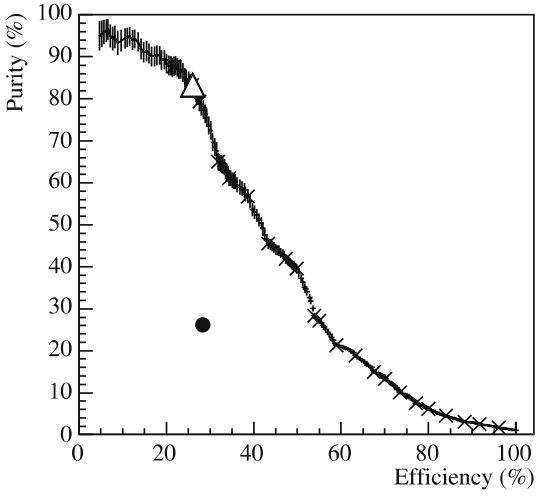

Fig. 8. D^0 efficiency-purity plot corresponding to Fig. 7. The efficiency is that of the cuts only

ciency as the classical cuts. Conversely, a higher efficiency may be desired to reduce possible bias due to tight cuts: here, LDA multiplies the efficiency by 2 for a purity equal to that given by the classical cuts. It can then be checked on the relative uncertainty curve that the statistical error obtained is not too much higher than the minimal one.

A zoom on Fig. 7 is presented in Fig. 9 to illustrate the multicut process. The black curves (LDA) and the black point (classical) have been obtained by assuming a background estimation via rotating, i.e. the V0 vertices are reconstructed with one of the two tracks rotated by 180°. This preserves the combinatorial background and destroys the signal (it can not be reconstructed). The grey curve and

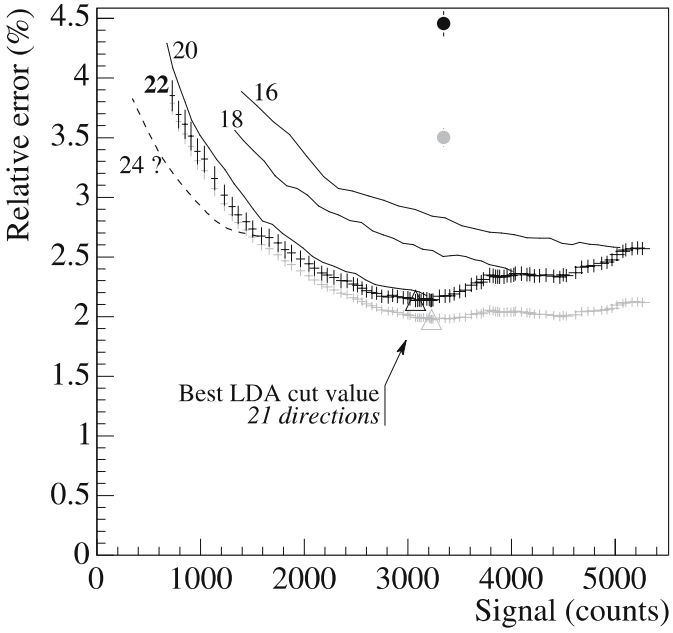

Fig. 9. Zoom on Fig. 7. *Black* and *grey curves and symbols* are for background estimation respectively with rotating and event-mixing. The *lines* show the curves that are obtained with 16, 18 and 20 LDA cuts instead of 22 (with rotating), the *dashed line* shows how could be this curve when 24 cuts are applied

point will be explained below. In this relative uncertainty versus efficiency diagram, as already said, each LDA cut results in a valley-shaped curve. A cut along the first direction is applied and progressively tightened until the cut value c_1 is reached (cf. Sect. 5.3). At this point, the algorithm has calculated a second LDA direction, which has a better performance than the first one. The second LDA cut is therefore applied and progressively tightened, till the cut value c_2, and so on. The plot shows the end of the process, and the envelope of all the valley-shaped curves is the locus drawn when the LDA cut is gradually tightened (and directions progressively added).

The minimum of this locus, indicated by the opened triangle, is obtained with the searched number of directions, here 21. The optimal number of cuts to use in this case is therefore 21, and the 21st cut value corresponding to the minimal relative uncertainty can easily be determined by the program. It obviously belongs to $]-\infty; c_{21}]$.

The curve obtained with a 22nd direction is rather similar to that obtained by keeping tightening the 21st cut beyond c_{21}. The 22nd direction has been determined with 5488 signal and 1120 background D^0 in the training samples, and was the last one calculated for low statistics reasons. If the statistics were enough, further calculated directions should behave like the dashed curve labelled "24 ?": an improvement is brought, but the relative uncertainty does not fall below the previously found minimal point, as this point is the absolute minimum of the envelope.

Yet, the final value of the LDA cut is not determined at this stage, because the final error bar depends on other factors. There is usually further "manipulation" of the data, such as an efficiency correction and a fit to some distri-

bution. The minimal final statistical error may then be obtained for another value of the LDA cut. The process is identical though, therefore simple and instantaneous, and one may wish to use various LDA cut values, depending on the physics observable looked at.

The grey curve of Fig. 9 illustrates how fast is a retuning: this curve results from the assumption that the estimated background is obtained from event-mixing rather than rotating. Event-mixing consists in reconstructing V0 vertices with one track taken in an event, and the second track taken in another event of similar multiplicity and primary vertex location. Like rotating, this destroys the signal candidates, but preserves the combinatorial background. As many events can be mixed together, event-mixing provides an estimation of the amount of background with a much smaller error bar than rotating (other characteristics have to be taken into account though, but a comparison of the background estimation techniques will not be addressed in this article). This results in a different cost function curve. Again, the new minimum is easily found, as well as the corresponding LDA cut.

As a summary, these preliminary results show that the relative uncertainty on the raw number of D^0 is divided by 2 for a similar efficiency when LDA is used, compared with the classical cuts. When rotating is replaced by event-mixing to estimate the amount of background, the bottom of the valley is not much lower, but is much flatter and extends up to 5500 signal counts. The cut efficiency can thus be multiplied by 1.8 while the relative uncertainty is left untouched.

7 Conclusions

Because it uses linear combinations of the n observables, LDA transforms $\mathbb{R}^n$ into a set of segments that is equivalent to $\mathbb{R}$. Cut-tuning is therefore obvious, as it consists in a simple minimization of a one-dimensional function (e.g. the relative uncertainty). Moreover, as the n-dimensional information of the distributions is taken into account, rather than the n projections as the classical cuts do, LDA also provides an improvement of the statistical error bars. While Fisher-LDA can not deal with low initial signal-to-noise ratios such as those considered above, optimized multicut-LDA significantly reduces the statistical uncertainty in the analyses presented. A drawback of this method is that it can be used only with two classes: it is not able to distinguish e.g. several sorts of backgrounds, and removes all of them as if it was a single contribution. Moreover, the selected area is always convex.

The method has fewer parameters to be tuned during the training phase than pattern classification methods like the neural networks. The minimal size of the samples needed to train the method correctly is therefore lower for LDA. Furthermore, LDA has only one parameter to be set by the user – namely, the efficiency of each cut –, which makes the method fast to set up.

LDA can also be used to calculate a systematic error due to the cuts: the LDA cut can be tightened and loosened while permanently staying on the optimized curve shown in Fig. 7. As a set of LDA cuts is easy and fast to determine, results can be obtained with several of them and compared together. Finally, classical cuts can be rapidly derived from the projected distributions of the candidates cut with LDA, and they can for instance also be used to estimate a systematic error.

When an analysis is done under other conditions (e.g. other collision centrality, other range of transverse momentum $p_\perp$), the relative proportion of background may be different, and hence tighter or looser cuts may be needed. While classical cuts require another n-dimensional minimization, previously calculated LDA cuts can simply be adapted to the new environment by a tightening or a loosening, until the new minimum in relative uncertainty is reached. LDA can also be trained with specific candidates (e.g. low-$p_\perp$, mid-multiplicity events, ...) and therefore be optimized for such characteristics.

Acknowledgements. I thank the organizers and lecturers of the Erice school of subnuclear Physics, as well as the juniors for the nice discussions and lively atmosphere. All junior's talks really deserved attention, the proceedings of some of them can be read in [26].

I also warmly thank S. Faisan for the numerous fruitful discussions about pattern classification and related topics.

References

1. R. Duda, P. Hart, D. Stork, *Pattern classification* (John Wiley & Sons, 2001)
2. S. Faisan, Private communication
3. J. Collins, M. Perry, Phys. Rev. Lett. **34**, 1353 (1975)
4. N. Cabibbo, G. Parisi, Phys. Lett. B **59**, 67 (1975)
5. J. Bjorken, Phys. Rev. D **27**, 140 (1983)
6. STAR Collaboration, K.H. Ackermann et al., Nucl. Instrum. Methods A **499**, 624 (2003)
7. ALICE Collaboration, F. Carminati et al., J. Phys. G: Nucl. Partic. **30**, 1517 (2004)
8. J. Faivre, Development of a pattern classification method (LDA) to improve signal selection and cuts optimization. STAR note 04xx (2005); To be released at the following URL: www.star.bnl.gov/STAR/sno/sno.html
9. T. Carli, B. Koblitz, Nucl. Instrum. Methods **A501**, 576 (2003)
10. D0 Collaboration, V.M. Abazov et al., Phys. Lett. **B606**, 25 (2005)
11. J. Faivre, Ph.D. thesis, Université Louis Pasteur, Strasbourg (2004) (in French)
12. A. Dainese, Ph.D. thesis, Università degli studi di Padova (2003)
13. B. Knuteson, H. Miettinen, L. Holmström, Comput. Phys. Commun. **145**, 351 (2002)
14. M. Mjahed, Nucl. Instrum. Methods A **481**, 601 (2002)
15. B. Roe, H.J. Yang, J. Zhu, Y. Liu, I. Stancu, G. McGregor, Nucl. Instrum. Methods A **543**, 577 (2005)
16. D0 Collaboration, V.M. Abazov et al., Phys. Lett. B **517**, 282 (2001)
17. M. Mjahed, Nucl. Instrum. Methods A **449**, 602 (2000)

18. L. Gaudichet, Proc. Hadron Collider Physics Symposium, Les Diablerets, Switzerland, July 2005. To be published in Eur. Phys. J
19. L. Gaudichet, Private communication
20. R. Fisher, Ann. Eugenic. **7**, 179 (1936)
21. D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Kluwer Academic Publishers, 1989)
22. R. Lotlikar, R. Kothari, IEEE Trans. Pattern Anal. **22**, 623 (2000)
23. J. Speltz, STAR Collaboration, J. Phys. G: Nucl. Partic. **31**, S1025 (2005)
24. J. Speltz, Private communication
25. ALICE collaboration, ALICE physics performance report, Vol. 2, CERN/LHCC 2005-030, to be published in J. Phys. G
26. A. Zichichi (2005) Towards new milestones in our quest to go beyond the standard model. In: Proc. Erice international school of subnuclear Physics. The Subnuclear Series vol. 43