EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
European Laboratory for Particle Physics

Internal Note/Physics General
ALICE reference number
ALICE-INT-2007-002 version 1.0

Date of last change
11 December 2006

Development of a pattern classification method adapted to low signal-to-noise environments for fast, easy and optimized tuning of selection cuts

Authors:

Julien Faivre [1] and Ludovic Gaudichet [2]

In situations where the signal of the analysed particle is tangled up with orders of magnitude more background, its extraction 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 relativistic heavy ion collisions. We show, on the examples of the Λ, Ξ and D^0 particles in ALICE, 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. infn-Padova, julien.faivre@pd.infn.it
2. infn-Torino, gaudichet@to.infn.it

Introduction

Extracting a particle signal 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 use 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, described in § 2.4.1, will be refered to as *classical analysis* or *classical cuts*. The values of the applied cuts, sometimes numerous, are often considered as independent parameters while the variables are generally correlated. This method may therefore be very long to tune and usually provides an improvable discrimination. This note describes the adaptation of the Fisher Linear Discriminant Analysis (Fisher-LDA), a pattern classification method widely used in data processing, to the extreme signal-to-noise conditions of central heavy ion collisions in ALICE, and shows its advantages over the "classical analysis".

The first section explains why such methods are needed in 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 obtained in the ALICE framework for the Λ and Ξ hyperons and the D^0 charmed meson.

Paragraphs 2.1 and 3.2 have been written with the help of [1, 2]. A part of the material of this note has been published in [3].

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 authors, or currently at this URL : **www.pd.infn.it/~faivre/lda.html**.

1 Low signal-to-noise environments

1.1 Examples of signal-to-noise ratios

We will focus on the weak decays, studied by reconstructing topologically their secondary decay vertex [4], as this is the analysis type for which the method presented in this note has been tried.

Heavy-ion collisions however 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 a 3-particle decay it scales with its cube.

For the case of the $\Omega^- = sss \to \Lambda^0 K^-$ in STAR's central collisions, the yield of about 0.6 $\Omega + \overline{\Omega}$ per event [5] and the multiplicity of more than 3 000 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 $D^0 = c\overline{u} \to K^-\pi^+$ in ALICE, the initial signal-to-noise ratio [1] is of the order of 10^{-8} [6]. 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 (because of the larger $c\tau$ of the Ω) 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. Yet, other fields – industry, health, image processing in general – do not deal with such situations, but rather with poor training statistics and large numbers of observables and/or of classes. The methods created for their needs therefore do not meet ours, which made necessary the development of a method adapted to our conditions.

1.2 Cut variables

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

A weak decay is characterized by a sizeable decay length ($c\tau$ of a hundred microns for charm decays, more than a centimeter 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, 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 the transverse plane of the geometry of a V0 vertex is shown in Fig. 1. 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,

1. Here, not calculated in an invariant mass window selecting the signal peak.

2. Reconstruction of the secondary decay candidates from the tracks.

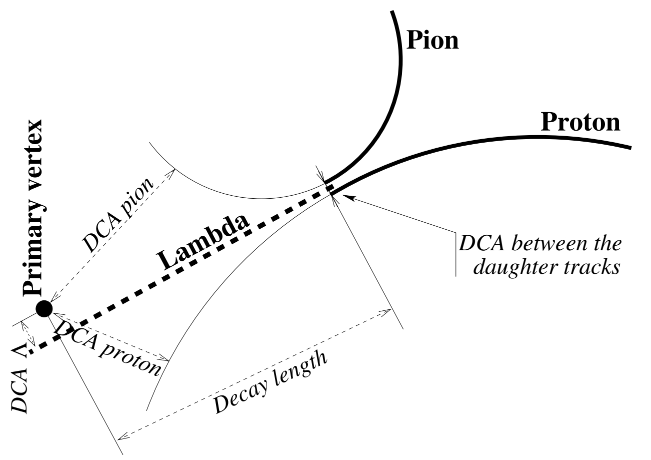

FIG. 1 – *2-dimensional geometry of a V0 vertex. The trajectory of each of the daughter particles is a thick solid line, the extrapolations towards the primary vertex are thin solid lines. The trajectory of the reconstructed parent particle is the thick dashed line.* DCA *stands for "distance of closest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

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.

Other interesting geometrical variables are the pointing angles, which are, in the general case, the angle between the reconstructed momentum of a particle and its flight line given by the vector joining the primary vertex to the decay vertex of the considered particle.

The cosine of the decay angle ($\cos\theta^*$, θ^* is the angle between the momentum of the mother particle and that of either daughter in the center-of-mass frame) is also often used for 2-body decays to eliminate the background : the distribution of this variable shows strong peaks at -1 and $+1$ for the background.

The track quality can also be of some help, for example by cutting tracks having a too small number of hits in the TPC. Their PID, obtained from detectors such as the TOF and the TPC, also allows to achieve large background rejections.

Finally, variables can be combined so as to obtain a more discriminant variable, such as the product of the signed distances of closest approach between each daughter track and the primary vertex.

Examples of usage of these variables may be found in [5,7] and [6] respectively for the multi-strange hyperons and for the D^0 analyses, and in [8,9] for other analyses.

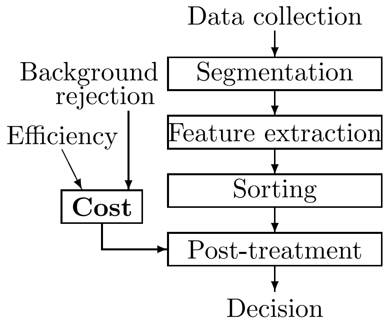

FIG. 2 – *Data classification process.*

2 Pattern classification

2.1 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 note. The observables used may be chosen amongst the parameters which are directly measured (see § 1.2 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).

2.2 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* $\varepsilon_S = \frac{S_{\text{post-cuts}}}{S_{\text{pre-cuts}}}$: 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* $\pi_S = \frac{S}{S+N}$: 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 [1]. To determine the performance of a method or to compare various methods, it is common to show these indicators by pair :

- Signal with respect to the signal-to-noise ratio,
- Signal with respect to purity (strictly equivalent to an efficiency-purity diagram, as well as to the diagram mentioned below),
- Efficiency with respect to the false alarms rate,
- 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 or a curve [2]. 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.

1. Thus no discrimination criterion will be defined here to compare various pattern classification methods.

2. For an optimized pattern classification method, this region is most often a curve, as such methods usually tranform the n-dimensional space of the observables to a one-dimensional cut variable. For classical cuts though, it is possible to obtain e.g. various purities for a given efficiency ; the region is therefore a surface.

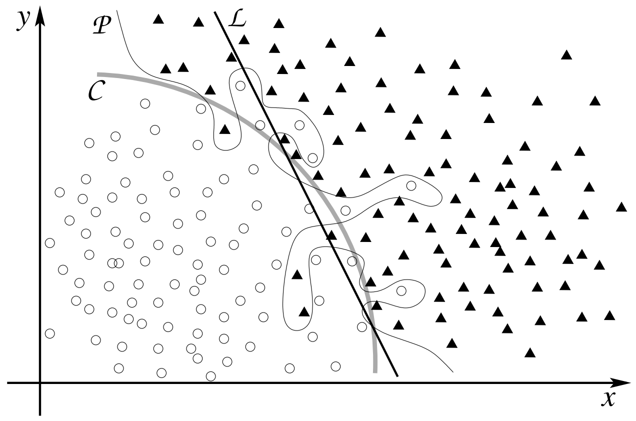

FIG. 3 – Training *samples and shape of various boundaries.*

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 base his decision on the tumor detection probability and set the cut value to a high probability : this selects more background but is safer.

2.3 Performance tuning and evaluation

For all trained 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 another sample (a 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* samples, 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 would be very different : the line would have a fair performance and the simple curve would have a better one, but the performance of the complex curve would be bad, because, while two samples are globally close to identical, they have yet significant local differences. The performance therefore always has to be evaluated on a sample that is different from the training sample.

This phenomenon is called *overtraining* when $\mathcal{L}$, $\mathcal{C}$ and $\mathcal{P}$ are actually obtained with the same pattern classification method.

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.

Overtraining should not be mixed up with the lack of statistics in the training sample : they have similar effects, but the latter is bound to happen whenever the training sample is not large enough, whatever the method, while the former may also occur when large samples are used, but only for methods which give too much importance to local parts of the distributions.

2.4 Some pattern classification methods

2.4.1 Classical cuts

In this note, we chose the expression "classical cuts" to name the simplest pattern classification method. It is illustrated by the top sketch of figure 5 (p. 6), and consists in applying a straight selection cut on the variables. The zone selected as being signal is thus delimited by hyperplanes all perpendicular (or possibly parallel) one with another.

Correlated cuts can occasionally be used, when the background is found to be well demarcated in a two-dimensional plane. While uncorrelated classical cuts are expressed as e.g. "$x_i > c_i$", correlated classical cuts take the following form : "$x_i > f_i(x_j)$", where f_i can be any non-constant function. Uncorrelated and correlated classical cuts can, of course, be used jointly. Examples of such cuts can be found in [10] for an Ω (triply strange baryon) analysis.

Optimizing classical cuts is most often done "by hand", plotting the 1- and sometimes 2-dimensional distributions of the signal and of the background, and adjusting step by step the cut values. Even when automatized, this procedure is very long and, due to the correlation between the variables, complex, since the distribution shape of variable x_i depends also on the cut values $c_{j,\, j \neq i}$.

Attempts to optimize directly letting free all the parameters (the cut values) have been made with MINUIT [11], with 6 cut variables : the procedure does converge and the processing time is kept at a reasonable level by applying (classical) pre-cuts before the optimization step. Yet, if these cuts are not tight enough, the processing time may rapidly become a limiting factor. Care should be taken that none of the variables chosen be too strongly correlated, otherwise the convergence is made more difficult.

A probably faster method, which could be called the "matrix method", consists in attributing a range to the possible cut values c_i and dividing each of these ranges into m_i values $\{c_i^1, c_i^2, \cdots, c_i^{m_i}\}$. For each set of cut values $(c_1^{j_1}, c_2^{j_2}, \cdots, c_n^{j_n})$ $(j_i \in < 1; m_i >)$, the signal significance is computed and stored in the corresponding matrix element $(j_1, j_2, \cdots, j_n)$. Searching for the maximum of the matrix then directly provides the optimal cut values. This method has successfully been applied to the D^0 [12] and D^+ [13–15] topological reconstruction in ALICE. When the number n of cuts is large, the number of matrix cells $m_1 \times m_2 \times \cdots \times m_n$ becomes very large and the calculation time may be prohibitive. It may then be advantageous to read twice the data, with smaller values m_i but with restricted variable ranges on the second pass according to the position of the maximum found after the first pass.

However, even with fancy optimization methods, classical cut tuning remains a tedious task and leads anyway to perfectible cuts since, even when optimized and apart from the correlated cuts, it always consists in a set of perpendicular hyperplanes.

2.4.2 Neural networks

The purpose of this section is to give a very brief description of the mechanism of the neural networks. More complete information can be found in virtually any book dealing with pattern classification (e.g. [1]), or with pattern recognition as they can also be used for that purpose. Some information can also be found in e.g. [16, 17], and particularly [18, and references therein], from which the information given in this paragraph is taken.

Artificial neural networks (ANN) are not easily defined since their common feature is a concept rather than a technique. They involve computing elements (the neurons) which are interconnected and often operate in parallel. The network structure and the links between the elements define the answer to an input. The knowledge of the network is acquired during a learning process, during which the data relative to each element are modified.

A well known category of ANN is the feed-forward ANN, which consists of networks organized in layers of neurons. The information flows towards only one direction, such that the input of each neuron depends only on the outputs of the neurons of the previous layer.

We will take the example of the multilayer perceptrons, which we have chosen for the comparison between neural networks and multicut-LDA in section 5.6. It is the most widely used and simplest type of neural networks. They are feed-forward ANN which have one input layer, one output layer, and for which any neuron receives input from all the neurons of the previous layer and from no other neuron.

An example structure of a multilayer perceptron is represented on FIG. 4. The first layer of neurons – the input layer – collects the entries, so there is one per variable used. The output layer has only one neuron in the case of a discrimination between two classes, and the output of that neuron is the discriminating variable. In between, there is at least one hidden layer.

The connection between two neurons is weighted, that is : a weight is applied to the output of a neuron before it is sent to the input of another neuron. There is one weight per connection, and their values are tuned during the training process. They can be called synaptic weights, from the analogy between the neural networks and the brain.

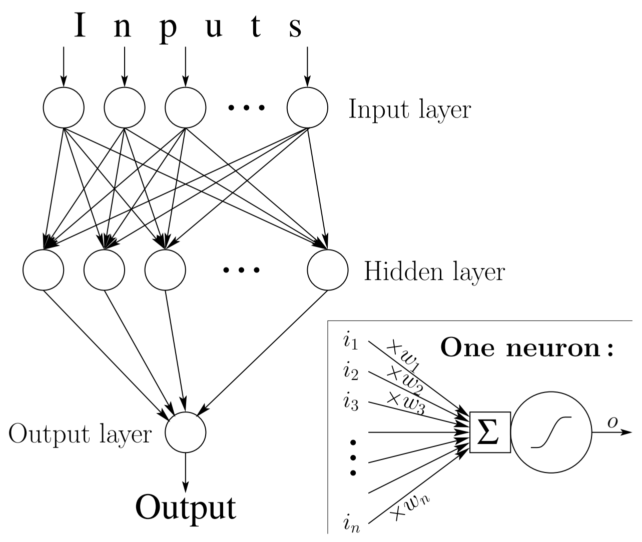

FIG. 4 – *Example structure of a multilayer perceptron.*

Every neuron of the network normalizes its input – a linear combination of the (weighted) connections – by applying it a sigmoid function, and sends this output to each of the neurons of the next layer. The sigmoid function usually normalizes into $[0; 1]$ or $[-1; +1]$. Its shape and steepness choice are left to the user, while the position in abscissa of its maximal steepness (the so called "offset") is a parameter tuned during the training.

The good performance of such a structure is due to the fact that any continuous function can be approximated by a linear combination of sigmoids. Moreover, if the system is trained to give an output equal to one for the signal and to zero for the background, the answer of the network to an input is equal to the probability that this input is signal.

The network learns by modifying the weights and the offsets during a training stage, for which a set of examples with inputs and correct outputs are given. There are several learning methods, most of them implementing what is called a back-propagation of the errors. The back-propagation of the errors minimizes an error function calculated by comparing the calculated output with what it should have been. The first derivative of that error with respect to the weights allows to calculate a small correction to apply to the weights. A network needs to process all the examples several times in order to minimize the error and converge to an optimized state. Processing the full example set is called an epoch.

The fundamental parameters of the multilayer perceptrons which have to be chosen by the user are thus the number of hidden layers, the number of neurons in those hidden layers, and the number of epochs for the training stage. The choice of the learning method as well as the associated parameters, to which belong the sigmoid steepness, are also free. Unfortunately, there are no general rules to set those free parameters and it is clear that a systematic study of the performance of the neural networks as a function of those parameters is long and difficult.

One can though limit a bit the range of possibilities considering that for applications like ours, the improvement brought by a second hidden layer is often small – and by a third one negligible –, and that too many hidden neurons make the network have a lower generalization power (overtraining), while too few ruin its performance. A good range generally found in the literature is n to $2n$ neurons in the hidden layer.

We can eventually mention that a neural network with no hidden layer is strictly equivalent to a Linear Discriminant Analysis, it just has different optimization criteria (they change with respect to the training method). We did not try to compare the performance of such criteria with the Fisher criterion and with the "optimized criterion" described respectively in sections 3.2 and 4.2.

2.4.3 Other methods

Many other types of pattern classification methods exist, each of them having several subtypes. We can e.g. mention the Markov fields, the nearest neighbors methods, the trees, the Parzen windows or the discriminant analyses, 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 [8, 19–23].

Discriminant analyses themselves can be subdivided into Linear (LDA), Quadratic (QDA) and higher degree Discriminant Analyses as a function of the shape of the separating surface. We can also mention the existence of a second-order Linear Discriminant Analysis [24, 25].

For each (trained) method coexist several criteria usable for the training, and often several training and/or optimization methods. Most of the time, there does not exist any general mathematic way to chose between them, and one has to find the best solution basing himself on personal judgment and tries on the data.

Let's finally add that, independently of the method chosen, the input data can be transformed non-linearly and/or combined before the pattern classification method is used, so as to provide more discriminant input variables. Space dimensionality expansion or reduction methods can also be applied, upstream as well, again to create more discriminant variables, or to be able to cope with too low statistics training samples or too large computing times (see also § 5.4).

2.4.4 Advantages over classical cuts

With respect to the classical method, a common property of all the other pattern classification methods is their ability to better discriminate between the signal and the background, 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.

3 Fisher Linear Discriminant Analysis

3.1 Basic principle of LDA

The principle of the LDA method is illustrated by the two drawings of Fig. 5. 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 large. 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.

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 chosen. This cut defines as border between both classes a hyperplane perpendicular to the axis.

3.2 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 Ronald Fisher in 1936 [26]. 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 a maximization, but the solution is analytical.

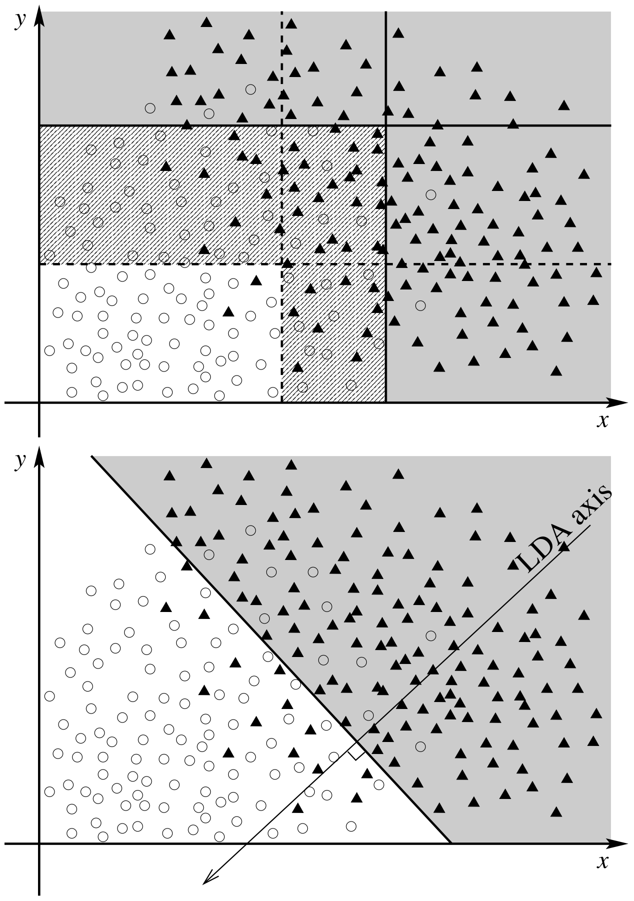

Fig. 5 – *Basic principle of classical selections and of* LDA *: example with two variables. Top plot : loose (solid line and grey area) and tight (dashed line and hatched area) classical cuts. Bottom plot :* LDA *cut.*

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 $\overrightarrow{u}$, i.e. the averages μ_k of those projected distributions should be as far as possible from one another relatively to their squared widths $\sigma_k^2 = \sum_{\overrightarrow{x} \in \mathcal{D}_k} (\overrightarrow{u} . \overrightarrow{x} - \mu_k)^2$, 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 $\overrightarrow{x}$ be an observation (thus an n-coordinate vector). The normalized n-coordinate vector $\overrightarrow{u}$ 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 $\overrightarrow{x} . \overrightarrow{u}$ –, the hyperplane which plays the role of a border between both classes.

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

$$\lambda(\Delta) = \lambda(\overrightarrow{u}) = \frac{|\mu_1 - \mu_2|^2}{\sigma_1^2 + \sigma_2^2} = \frac{{}^t\overrightarrow{u} S_B \overrightarrow{u}}{{}^t\overrightarrow{u} S_W \overrightarrow{u}} \qquad (2)$$

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

$$\exists\ \xi \in \mathbb{R} \quad / \quad S_W^{-1}(\overrightarrow{m_1} - \overrightarrow{m_2}) = \xi \overrightarrow{u} \qquad (3)$$

The problem is therefore reduced to a matrix inversion.

3.3 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.

4 Optimized multicut-LDA

4.1 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.

The mechanism of this method, which will be called *multicut*-LDA, is depicted in FIG. 6. 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 paragraph 4.2, with an efficiency on the signal close to 100 %. This first cut is applied to the learning samples, and a second LDA direction

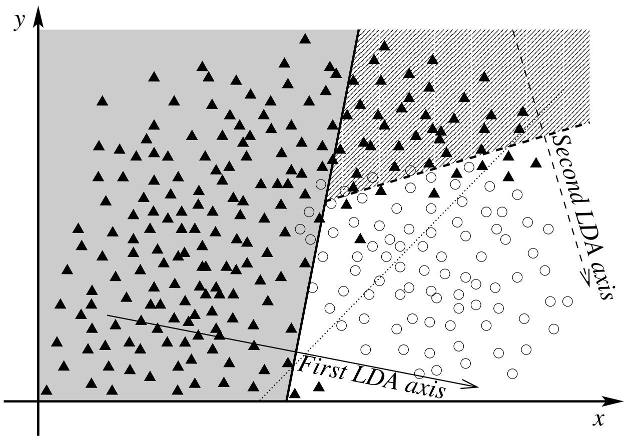

FIG. 6 – *Mechanism of the multicut-*LDA *method. The first* LDA *cut is the thick solid line and eliminates the grey area. The second* LDA *cut is the thick dashed line and filters out the hatched area. If only one cut had to be used to reach a similar background rejection, its efficiency would be much lower, as shown by the dotted line.*

is calculated with the remaining candidates. The process is then repeated until not enough candidates remain in the training samples to calculate more LDA axes.

Multicut-LDA therefore provides a set of LDA directions, each being a vector $\overrightarrow{u_i}$ of the space of the observables (n coordinates). It also provides a cut value c_i associated to the direction $\overrightarrow{u_i}$. The value of c_i depends on $\overrightarrow{u_i}$, and the direction $\overrightarrow{u_i}$ is a function of $\overrightarrow{u_{i-1}}$ 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.

4.2 Optimized criterion

Multicut-LDA can be improved by replacing the Fisher criterion by another one, which takes the local – and not global – behavior 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.

4.3 Optimization

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

4.3.1 Mathematical expressions

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 term in (4), the number of signal candidates removed by cutting at value c_i along the axis $\overrightarrow{u_i}$ is :

$$S_i - S_{i+1} = (1-\varepsilon_{S_i})S_i = \sum_{\overrightarrow{x}\in\mathcal{D}_{S_i}} (\overrightarrow{x}.\overrightarrow{u_i} < c_i), \qquad (4)$$

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 :

$$1 - \frac{\displaystyle\sum_{\overrightarrow{x}\in\mathcal{D}_{S_i}} (\overrightarrow{x}.\overrightarrow{u_i} < c_i)}{S_i} = \varepsilon_{S_i}. \qquad (5)$$

4.3.2 Limits set by the no-overtraining condition

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 :

- 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.
- The numbers of signal and background candidates which are removed have to be larger 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-LDA method.

4.3.3 Function to maximize

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

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

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 $\overrightarrow{u_i}$. As the value of c_i depends on $\overrightarrow{u_i}$, it needs to be recalculated at each step.

4.3.4 Maximization algorithm

For the results presented in the next section, the algorithm chosen to maximize f consists in varying each coordinate of the vector $\overrightarrow{u}$ at a time, moving the vector by a given angle α. 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 insignificant.

We used 8° as starting value for α, and final values depending on the analysis, but ranging from 0.5° down to $(\frac{1}{256})^{\circ}$, with basically no change in the performance.

The expression of the coordinate change for the axis vector $\overrightarrow{u}$ as a function of α is given by the following expression :

$$\begin{cases} \overrightarrow{u_\ell} &= (u_1, u_2, \cdots, u_j, \cdots, u_n) \\ \overrightarrow{u_{\ell+1}} &= \overrightarrow{v_{\ell+1}} / \|\overrightarrow{v_{\ell+1}}\| \\ \overrightarrow{v_{\ell+1}} &= (u_1, u_2, \cdots, u_j + \delta_j, \cdots, u_n) \end{cases}$$
$$\delta_j = \frac{-2u_j \sin^2\alpha \pm \sqrt{1-u_j^2}\,\sin(2\alpha)}{2(u_j^2 - \cos^2\alpha)} \qquad (7)$$

It is worth mentioning that when the j^{th} coordinate is varied (e.g. increased) to move $\overrightarrow{u}$ by α, this loop is exited when $(\widehat{\overrightarrow{u},\overrightarrow{x_j}})$, $\overrightarrow{x_j}$ being the unitary vector driving the j^{th} coordinate axis, reaches a value smaller than α. This is due to the fact that to move by another step of angular width α, $\overrightarrow{u}$ has to "pass on the other side" of $\overrightarrow{x_j}$, and therefore one of its other coordinates has to change sign. This is not possible, since the algorithm changes only the j^{th} coordinate. The loop is therefore exited, and the algorithm starts another one, changing the $j+1^{\text{th}}$ coordinate.

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

TAB. 1 – *Comparison of the characteristics of classical cuts,* ANN*, and the multicut-*LDA *method presented in this note.*

	Classical cuts	**Multicut-LDA**	Neural networks
Setting up of the method	**Trivial, fast**	**Easy, fast**	Complex, long
Nb. of parameters to be chosen	Several (or **none**[a])	**One**	Several
Training	**None** (or long and complex [a])	**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 (or **Yes**[a])	**Yes**	**Yes**
Gives optimized cuts (in $n \times n$)	No	**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] When a maximization algorithm of the n-dimensional function is used (see text for details).

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 [27].

4.3.5 Comparison between ANN and LDA

Table 1 compares several characteristics of the classical cuts, of multicut-LDA 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, as described in § 5.6 and e.g. [23].

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. An example of this effect can be seen for example in FIG. 8 (p. 13).

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 multicut-LDA scales with n.

The exact number of degrees of freedom can be calculated by the following formulas : with n cut variables and N cascaded cuts (either multicut-LDA or neural networks), and one hidden layer with m neurons in the neural networks, the numbers of degrees of freedom write :

$$\begin{aligned} \mathfrak{N}_{LDA} &= N(n-1) \\ \mathfrak{N}_{NN} &= mN(n+2)\,, \quad m \simeq n \end{aligned} \tag{8}$$

As the number of candidates needed in the training samples is roughly proportional to the number of parameters set during the training (the degrees of freedom), multicut-LDA has the advantage of being less statistics demanding. As a numerical example, we can take the analysis presented in section 5.7, for which $n = 10$ and $N_{LDA} = 21$. For the neural network hypothesis, we can choose e.g. $N_{NN} = 3$ according to the study presented in section 5.6, and, conservatively, $m = n + 1$. We obtain :

$$\mathfrak{N}_{LDA} = 189, \qquad \mathfrak{N}_{NN} = 396 \ \simeq \ 2 \times \mathfrak{N}_{LDA}$$

5 LDA practical guide and results

5.1 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 probably as reliable, is to check that the performance of the i^{th} LDA cut is better than that of the $i-1^{\text{th}}$ 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 2 000 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. This number is, of course, to be considered for the last direction calculated, so the size of the initial samples is larger and depends on the efficiency on the signal and background of the previous calculated directions.

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

5.2 Composition of the training samples

The LDA cuts calculated are optimized for signal and background populations which are similar to those of the training samples. As a consequence, 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, as well as the signal and background distributions in the phase space.

Therefore tighter or looser cuts may be needed (case of centrality change, for which the distributions are not expected to change so much), or even a recalculation of the LDA cuts (case of a change in the $p_\perp$-range of the analysis).

While classical cuts require in both cases another n-dimensional minimization, multicut-LDA provides rapidly another set of cuts optimized for the new situation, simply by using in the training samples only candidates belonging to the desired part of the whole sample, for example the low-$p_\perp$ part of the mother particle $p_\perp$-spectrum.

Multicut-LDA also enables the use of previously calculated LDA cuts, which can simply be adapted to the new environment by a tightening or a loosening, until the new minimum of the cost function is reached. This solution is extremely fast, as it only consists in minimizing a 1-dimensional function ; it has been used in the first analysis presented in section 5.5.

5.3 Setting the LDA parameter

The only parameter of the multicut-LDA method that has to be chosen by the user is the number of signal candidates of the training sample removed at each step.

Section 4.3.2 gives a reasonable range for this value. We did not make a full study, but the results obtained never seemed to depend much on the choice of that parameter, as soon as it was in the range.

We can though mention that one strategy, that was used for example in the second analysis presented in section 5.5, consists in using the minimal value (e.g. 500 candidates) for the first 2 or 3 cuts so as to remove the dominant parts of the background with an efficiency on the signal as close as possible to 100 %, and then to increase that value (possibly progressively, up to e.g. 1 000 or 2 000, depending on the size of the training samples) when the background rejection factor becomes smaller, so as to reach the minimum of the cost function with a much lower number of LDA directions. A sufficiently large number of candidates in the training samples is yet needed to use this trick.

5.4 Variables used

5.4.1 Distributions shapes

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 may be 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 very narrow).

An identical consideration leads to the conclusion that the various variables should have values of the same order of magnitude.

Normalizing the distributions' shapes eliminates this constraint, but is a non trivial issue, above all because the real data are a mixture of signal and background, whose distributions' shapes are different.

Two studies have been made on the data presented in section 5.7.

First, we modified the convergence criterion of the maximization algorithm to investigate the possibility to make it independent of the variables distributions : instead of having the algorithm stop when the angle α (see § 4.3.4) reaches a minimal value, or when the variation in the number of background candidates removed becomes smaller than a threshold, we had it stop only when all the variables were used at least x times in the optimization process [1]. This attempt did not lead to any satisfactory result, mostly because some variables were actually never used, even for values of α as small as $(\frac{1}{1024})^{\circ}$.

On the contrary, an unsophisticated normalization had some effect on the same data. The variables distributions have been normalized using the mean and the variance either of the signal candidates only, or of the background candidates only, or of a mixture of both, either in equal propor-

1. By "variable j used in the optimization process", we intend that the j^{th} coordinate of the axis' vector has been changed.

tions or in the proportions of the training samples. An immediate consequence that appeared is that fewer variables were not used during the optimization, and those which were "left out" were no longer the same for all the LDA directions. The result on the performance was very slight though. All four normalizations provided a similar, tiny improvement. None provided better results than the others, but a normalization by the mean and variance of the background candidates provided slightly worse results than the three other methods.

5.4.2 Number of variables

The number of variables to use should in principle 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[1].

The number of variables to use can yet be limited by statistics or processing time reasons – numbers of a few tens are in principle not critical. When the training statistics is very limited, it may be desirable to give a try in a space of lower dimensionality : a better performance may be reached. Methods exist to reduce this number of variables while avoiding a drop in discrimination : we can cite for example under-optimal LDA, Principal Component Analysis, or fractional-step LDA.

Under-optimal LDA (see [5, 7]) is the outcome of the fact that due to the large number of combinations, it is virtually impossible to examine the performance of all the 2^n–2 m-uplets, $m < n$, to find those which perform better than the full n-uplet, as illustrates FIG. 7. The method is therefore to search for the most performant j-uplet, with j a small integer, and to iterate in adding variables, examining only the j+1-uplets containing the most performant j-uplet. The limitations of that method are firstly that there still is a total number of combinations to examine of $\frac{n(n+1)}{2}$, and secondly that the most performant $j+1$-uplet does not necessarily contain the most performant j-uplet. Yet, it may be worth a try if the processing time makes it possible : if a better performance is reached, the method can be considered the lesser of two evils.

Principal Component Analysis (PCA) [5, 7, and described in most data processing books] is a much faster – because based on a simple $n \times n$ matrix diagonalization –, but unsupervised, matrix-based analysis which reduces the dimension of the working space by taking out the least informative dimensions. PCA provides an ordered base ($\overrightarrow{v_1}$,$\overrightarrow{v_2}$, $\cdots$, $\overrightarrow{v_n}$) of the n-dimensional space such that $\overrightarrow{v_1}$ drives the direction along which the distribution (signal + background) has most information (i.e. its dispersion is largest), $\overrightarrow{v_2}$ the second best one, and so forth. The base calculation and dimensionality reduction has to be performed in the full n-dimension space before each LDA cut. The results are not guaranteed to improve when PCA is used, because the least informative directions may also be the most discriminating ones. Here again, one has to try and compare.

1. Such as the product of the signed distances of closest approach of the daughter tracks to the primary vertex, as mentioned in § 1.2. Kinematic variables should be added with care though, as too clever algorithms may well be able to calculate and cut on the invariant mass themselves (if in doubt, one might feed the method for training with simulated background candidates very close to the PDG mass of the analysed particle, and see whether a smaller bias is obtained).

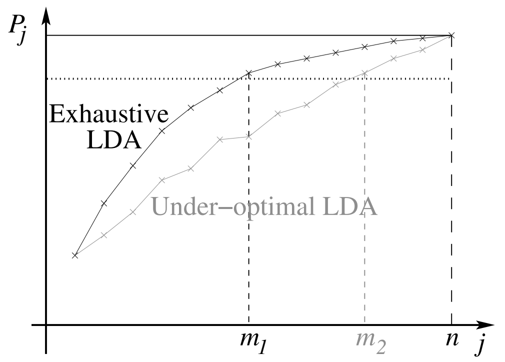

FIG. 7 – LDA *performance P_j on a sample with infinite statistics as a function of the number j of variables used. A satisfactory performance (dotted line) is reached faster by an exhaustive test of all the combinations (m_1 variables used, in black) than by under-optimal* LDA *($m_2 > m_1$ variables used, in grey). When calculated with low statistics samples, these curves may saturate earlier and to a lower value, and decrease when j approaches n.*

It is worth to note that PCA also provides a partial solution to the normalization problem : a whitening of the data (with the usual limitation that signal and background have to be considered as a unique distribution) can be performed by normalizing the distributions along the directions of the PCA base by a factor of $\frac{1}{\lambda_j}$, λ_j being the eigenvalue associated to the j^{th} vector of the PCA base.

Fractional-step LDA, better described in [28], performs dimensionality reduction while keeping a better discrimination than PCA. Instead of removing abruptly the least informative direction $\overrightarrow{v_n}$, data along that direction are squeezed by some factor[2]. The new v_j vectors are calculated and the process is repeated until the squeezing is judged sufficient to remove completely one dimension. Although initially conceived to avoid class overlap after dimension reduction when many classes are used, this method should also perform well in our case. Its drawback however is a large increase in the CPU-time needed, as it multiplies the multicut-LDA-time by the number of steps in which a dimension is removed. It remains yet faster than under-optimal LDA.

2. Technically, this is made possible by the scaling of each row i of the between-class scatter matrix S_B (see § 3.2) by a weight w_i which is a decreasing function of the distance between the means $\overrightarrow{m_k}$ of the classes projected on the i^{th} vector.

TAB. 2 – STAR *preliminary results for the* $\Omega + \overline{\Omega}$ *in the top 10 %* `Au-Au` *central data at two different* $\sqrt{S_{NN}}$ *energies : gain brought by* LDA *over classical cuts for various variables relative to the amount of signal counts integrated in* $p_\perp$ *[3, 5, 31].*

	200 *GeV*	62 *GeV*
$S\ (\varepsilon_S)$	$\times 1.3$	$\times 1.8$
N	$\div 1.1$	$\div 1.1$
π_S	$\times 1.2$	$\times 1.4$
S/N	$\times 1.4$	$\times 2.0$
σ_S/S	$\div 1.2$	$\div 1.6$

5.5 Results on real data

Optimized multicut-LDA has been tested with 25 cut variables on the Ξ and Ω multi-strange baryons in 200 *GeV* `Au-Au` collisions in STAR and has shown to provide more precise results than the classical cuts [5] : 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 successfully been applied to STAR's 62 *GeV* `Au-Au` collisions [29, 30], and has proven another 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 [5] 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. 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 recalculated LDA directions with 62 *GeV* simulation to have possibly more optimized LDA cuts.

These results are summed up in table 2. The notations adopted are those introduced in § 2.2.

5.6 Comparison with the neural networks

In this section, we show on simulated events that the reconstruction of the hyperons in ALICE can be optimized by using the Linear Discriminant Analysis. The examples of Λ and Ξ in central `Pb-Pb` collisions at $\sqrt{S_{NN}} = 5.5\ TeV$ are shown, in comparison with results obtained with neural networks and classical analyses.

The topological variables used for the V0 are described in [32]. As already mentioned in the previous sections, investigating the best cut values for classical cuts is a non-trivial and time-consuming task, since the parameters are numerous and correlated. Moreover, these classical cuts clearly do not give the best signal to noise separation, it is therefore worth using a better selection method. Neural networks are usually considered in this case ; however, they can have major drawbacks. Indeed the definition of the suitable parameters of a neural network is not trivial, as one has to decide the numbers of hidden layers and of neurons in these layers, the calculation method of the weights and the number of epochs, and there are also risks of overtraining. The results obtained with multicut-LDA on STAR's real data, presented in the previous subsection, encouraged us to test it on our data.

For this purpose, a sample of 198 `Pb-Pb` events has been generated by the HIJING event generator with a multiplicity of $dN_{ch}/d\eta \simeq 4\,000$ at mid-rapidity. In addition, 100 Λ, 100 $\overline{\Lambda}$, 15 Ξ^- and 15 $\overline{\Xi}^+$ have been simulated separately and merged in each HIJING event. The resulting events have been reconstructed by the ALICE offline software, after which the V0 and cascade vertices have been reconstructed with very loose cuts. From these secondary vertices, we have extracted the samples of signal and background for Λ and Ξ, which have been used to test multicut-LDA and to confront it to other methods.

The initial statistics in the training samples, obtained after propagation and reconstruction of the generated particles and application of very loose cuts at the reconstruction level, was 11 376 signal and 964 281 background for the Λ, and 2 545 signal and 593 099 background for the Ξ.

To discriminate the true Λ and $\overline{\Lambda}$ hyperons against their background, we have tested multicut-LDA, a neural network and the classical cuts. 11 variables were used for the multicut-LDA and neural network analyses : the decay length, the DCA (distance of closest approach) between the daughters, the pseudo-rapidity difference between both daughters, the smaller of the DCA to the primary vertex among both daughters, the pointing angle of the V0, the opening angle of the daughters in the lab frame, the cosine of the decay angle, and the number of hits of both daughters in the ITS and in the TPC. The number of neurons in the unique hidden layer of the neural network was set to 11. It was found that a second hidden layer did not bring improvement, and that the performance of the neural network saturated beyond about 11 hidden neurons. For the classical cuts, only 4 variables were used – the most efficient ones as classical cuts have been chosen –, because of the increasing difficulty to optimize them when the number of variables rises : the decay length, the DCA between the daughters, the smaller of the DCA to the primary vertex among both daughters, and the pointing angle.

For the selection of the Ξ and $\overline{\Xi}$ hyperons, we have tested multicut-LDA with 9 variables (8 plus the output from the LDA applied to the V0) and the classical cuts with 5 variables – here too, the variables discarded where the least efficient ones as classical cuts. Those used for LDA were the cascade and V0 decay lengths, the DCA of the bachelor and of the V0 to the primary vertex, the cascade pointing angle,

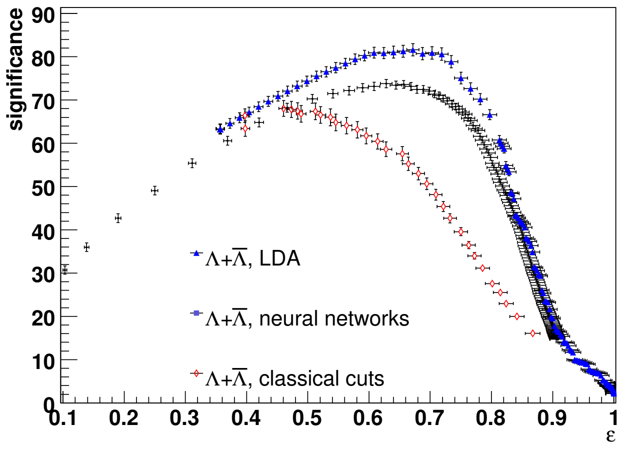

FIG. 8 – *Significance as a function of the intrinsic efficiency ε for $\Lambda+\overline{\Lambda}$ for central* Pb-Pb *collisions at $\sqrt{S_{NN}} = 5.5$ TeV in* ALICE. *The significance is calculated as $S/\sqrt{S+N}$, with S the amount of signal and N that of noise.*

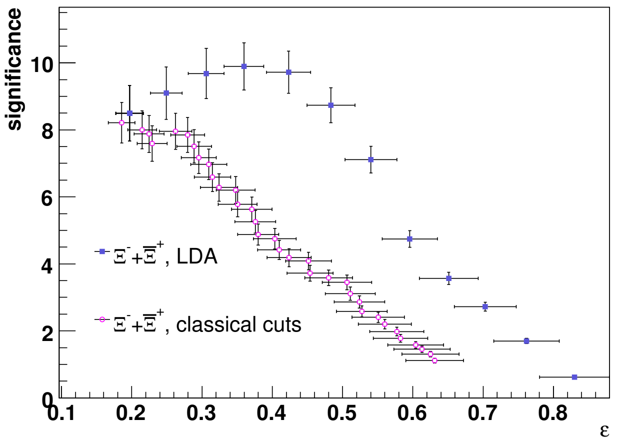

FIG. 9 – *Equivalent of figure 8 for $\Xi + \overline{\Xi}$.*

the mass difference between the reconstructed V0 and the PDG Λ mass, the number of hits of the bachelor in the ITS and in the TPC, and the LDA output value for the V0. The variables used for the classical cuts were the cascade decay length, the DCA of the bachelor and of the V0 to the primary vertex, the DCA between the reconstructed V0 and the bachelor, and the cascade pointing angle.

We have also developed an automatic procedure to calculate the cut values for the classical cuts. It consists in generating randomly some sets of cut values, calculating the corresponding efficiency and background rejection and keeping only the cut values which give the best efficiency for a given noise rejection. This procedure allows to obtain the best values of the classical cuts as a function of the efficiency, which gives optimal results and is used for the comparison with multicut-LDA. However, we estimate that the processing time may become prohibitive as soon as more than half a dozen variables are used.

Figures 8 and 9 show the significance obtained with the various methods which have been tried, as a function of the intrinsic efficiency ε, i.e. the quantity of signal kept over the quantity of signal that can be reconstructed (when all the tracks produced by the weak decay have been reconstructed in the central tracker). From right to left, each LDA marker corresponds to an additional LDA cut removing a small amount of signal. Each marker of the classical selections represents a set of fixed cuts for a given efficiency. The Λ neural network analysis uses 3 cascaded networks. 3 was found to be the minimal number required for the neural network method to be able to deal with the large, complex-shaped background and to reach reasonable significance values. The variations of the significance and of the efficiency are obtained by moving the selection threshold on the output of the third neural network. A loose threshold is fixed on the outputs of the previous networks.

The comparison between the results of the various methods is done by looking at the best achievable significance and the corresponding efficiency. The best significance obtained with multicut-LDA for both hyperons is 20 % better than that obtained with the classical cuts, and, in addition, the efficiencies have been increased by more than 60 %. The neural networks give an intermediate result, which illustrates the difficulty to tune them.

The time spent on the analyses can also be considered as a valid criterion for the comparison. Multicut-LDA stays in the lead, since it runs just once and only one parameter has to be set. On the contrary, the analysis with a neural network needs a lot of parameter tuning and was effectively the most time-demanding analysis, as about an order of magnitude more time has been dedicated to it than to multicut-LDA.

In conclusion, the multicut-LDA method has proved to be able to improve the results of the hyperon reconstruction in comparison with classical analysis, and is moreover fast and simple to apply. In our case, multicut-LDA appears also to have advantages over neural networks, not only for the performances reached, but also because the setting is faster, the training as well, and the reliability is higher.

5.7 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 $\sqrt{S_{NN}} = 5.5\ TeV$. The results shown come from a simulation sample different from 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 was 10 : the decay length, the DCA (distance of closest approach) between the daughters, the DCA to the primary vertex of the V0 and of both daughters, the product of the signed DCA to the primary vertex of both daughters, the pointing angle of the V0 and of both daughters, and the cosine of the decay angle. The number of signal candidates of the training sample to be removed by each LDA cut has been set to 499 (the candidate with $\overrightarrow{x}.\overrightarrow{u_i} = c_i$ is not removed), for an initial sample size of 15 967.

5.7.1 The valley-shaped curve

Figure 10 shows our cost function : with respect to the amount of signal that passes the cuts, the relative statistical 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. 11. 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 efficiency 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. 10 is presented in FIG. 12 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 point will be explained in the next section. 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.* § 4.3). 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

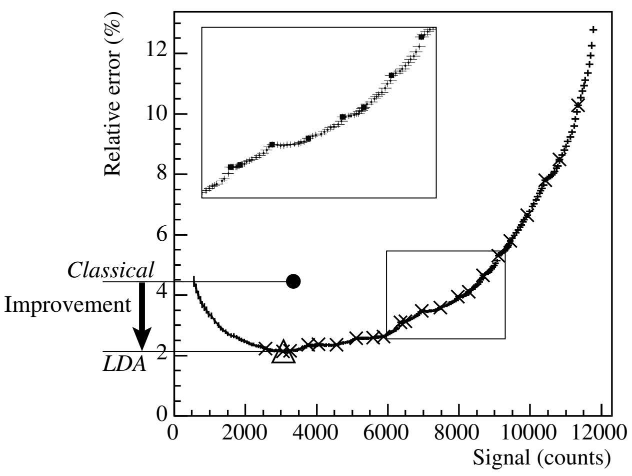

FIG. 10 – *σ_S/S of the D^0 as a function of the amount of signal, for central* Pb-Pb *collisions at $\sqrt{S_{NN}} = 5.5\ TeV$ in* ALICE. *The closed circle shows the performance reached with the current classical cuts [32], and the valley-shaped curve is the locus of the points described when the* LDA *cut is gradually tightened (right to left). The crosses ($\times$) 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.*

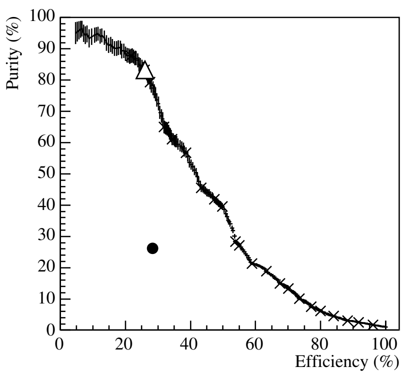

FIG. 11 – *D^0 efficiency-purity plot corresponding to* FIG. *10. The efficiency is that of the cuts only.*

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 5 488 signal and 1 120 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 labeled "24 ?" : an im-

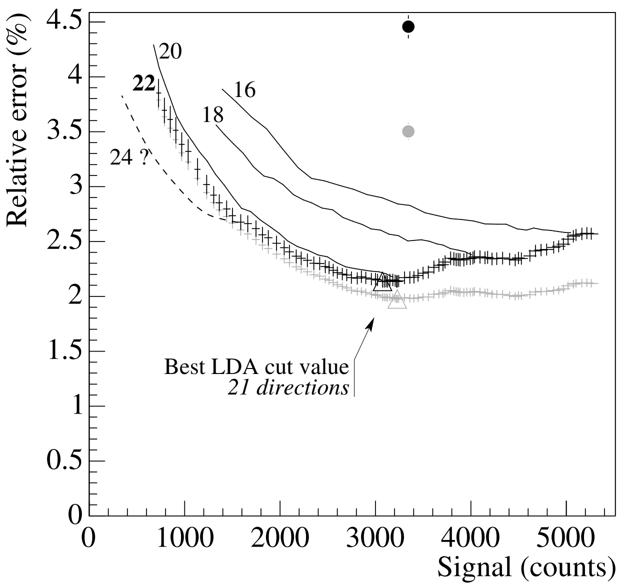

FIG. 12 – *Zoom on* FIG. *10. Black and grey curves and symbols are for background estimation respectively with rotating and event-mixing. The lines show the curves that are obtained with 16, 18 and 20* LDA *cuts instead of 22 (with rotating), the dashed line shows how could be this curve when 24 cuts are applied.*

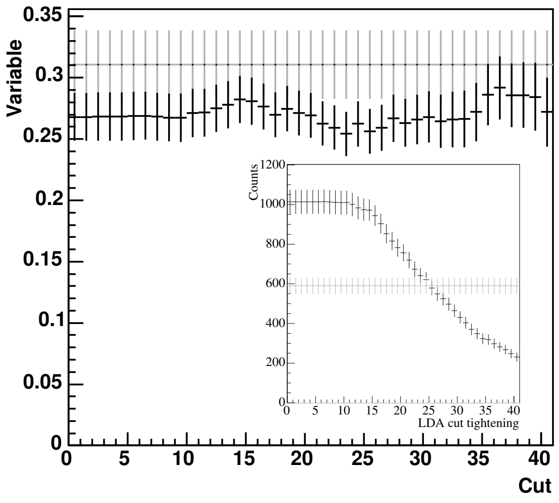

FIG. 13 – *Example of systematics study taken from [5] : the variation of a variable is plotted as a function of the tightening of the* LDA *cuts (black). The value obtained with the classical cuts is shown in grey (identical value duplicated all along the axis). The inset shows the variation in raw number of candidates : there is a factor of 4 between the tightest and loosest* LDA *points.*

provement 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.

5.7.2 The minimum

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 distribution. 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. 12 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 note). 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 5 500 signal counts. The cut efficiency can thus be multiplied by 1.8 while the relative uncertainty is left untouched.

5.8 Extended usage of LDA

5.8.1 Systematic error estimation

As we have seen, LDA, even multicut, provides a transformation from $\mathbb{R}^n$ into a set of segments that is equivalent to $\mathbb{R}$. Changing the cut efficiency and obtaining the corresponding best purity is therefore an obvious task, as it simply consists in tightening or loosening the LDA cut, possibly adding or removing directions if needed. Doing so describes the valley-shaped curve.

Figure 13 illustrates on a real example how this helps estimating a systematic error due to the cuts : as a function of an arbitrary quantification of the tightening of the LDA cuts, one plots the evolution of the value of the observables he is studying.

Moreover, as a set of LDA cuts is rapidly calculated, studying the variation of the Physics results as a function of the cuts used is straightforward. Various LDA cuts can be calculated for example leaving out or inserting one of the variables, applying kinematical cuts or reducing the space with classical pre-cuts.

5.8.2 Deriving classical cuts

Classical cuts also reinforce a systematic study. Yet, as we mentioned earlier, obtaining optimized classical cuts is a tedious and time-consuming task.

In some cases, it is possible to use LDA to find classical cuts. The principle is simple : naming f_j and g_j the distributions of the j^{th} cut variable respectively before and after the LDA cuts, one only has to examine the ratios $R_j = \frac{g_j}{f_j}$ for the signal and background distributions. If R_j falls off or rises steeply for both distributions, an equivalent classical cut can be found on that variable. If R_j is constant, the variable should not be cut.

Unfortunately, it seems that most of the time, the steepness is far from large. References [5, 7] can be consulted for results and for details about how to quantify the steepness.

Yet, a try on the analysis shown in section 5.7, with LDA cuts giving a similarly low steepness of the R_j functions, lead to classical cuts more performant than those which were used before [33, 34]. This encourages tries to derivate classical cuts out of LDA cuts even when the shape of the R_j functions does not look promising.

Conclusions

Because LDA provides an optimized transformation from $\mathbb{R}^n$ to $\mathbb{R}$, cut-tuning is made 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 succeeds in reducing significantly 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.

On a more technical side, additional advantages are that 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 is not subject to overtraining, and 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. 10. As a set of LDA cuts is easy and fast to determine, results can be obtained with several of them and compared together, and sets specific to conditions which require a maximal improvement (e.g. collision centrality, particular range in transverse momentum) can be determined quickly. Finally, classical cuts can sometimes be rapidly derived from the projected distributions of the candidates cut with LDA, and be used to estimate a systematic error.

Multicut-LDA has successfully been applied to the Λ, Ξ and D^0 analyses in `Pb-Pb` collisions in ALICE. It should work equally well for other topologically reconstructed particles like the Ω, the D^+, the D_s^+, the Λ_c^+, etc..., and could possibly lead to dramatic improvements for resonance analysis, for which the statistical error bar is about proportional to the background level. The method can more generally be applied to any 2-class discrimination problem for which the zone to be selected is thought to be convex. It can also be seen as a tool to explore the n-dimensional space more easily than with the classical method, and to investigate correlations and biases, with the possible addition of variables which are not used in the final analysis, but which play a role in the estimation of the bias and systematic errors.

Bibliography

[1] R. Duda, P. Hart, D. Stork. *Pattern classification.* John Wiley & Sons (2001)

[2] S. Faisan. Private communication

[3] J. Faivre. Eur. Phys. J. **C46, s02** (2006), 1–11. DOI : 10.1140/epjcd/s2006-03-001-7

[4] ALICE collaboration. J. Phys. G: Nucl. Partic. **30** (2004), 1517–1763

[5] J. Faivre. *Reconstruction and study of the multi-strange baryons in ultra-relativistic heavy-ion collisions at $\sqrt{S_{NN}} = 200$ GeV with the* STAR *experiment at* RHIC. Ph.D. thesis, Université Louis Pasteur, Strasbourg (2004). In French

[6] A. Dainese. *Charm production and in-medium* QCD *energy loss in nucleus-nucleus collisions with* ALICE. Ph.D. thesis, Università degli studi di Padova (2003)

[7] 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`

[8] T. Carli, B. Koblitz. Nucl. Instrum. Methods **A501** (2003), 576–588

[9] D0 collaboration. Phys. Lett. **B606** (2005), 25–33

[10] B. Hippolyte. *Study of the strangeness production in the ultra-relativistic heavy-ion collisions at $\sqrt{S_{NN}} = 130$ GeV with the* STAR *experiment at* RHIC. Ph.D. thesis, Université Louis Pasteur, Strasbourg (2002). In French

[11] A. Shabetaï. Private communication ; PhD thesis in preparation

[12] A. Rossi. *Study of the strategy for measuring the charm production in* p-p *interactions in the* ALICE *experiment*. Master's thesis, Università degli studi di Padova (2006). In Italian

[13] E. Bruna. *D^+ reconstruction in* p-p *events*. Talk at the Alice Week, Oct 10th 2006. URL : `indico.cern.ch/conferenceDisplay.py?confId=a057582`

[14] E. Bruna. *Reconstruction of $D^+ \to K^-\pi^+\pi^+$ in the* ALICE *central detector*. Talk at the Trento ECT* International Workshop on Heavy Flavor Physics in Heavy Ion Collisions at the LHC. URL : `indico.cern.ch/conferenceDisplay.py?confId=a055772`

[15] E. Bruna. Ph.D. thesis, Università degli studi di Torino (2007). In preparation

[16] L. Holmström, S. Sain, H. Miettinen. Comput. Phys. Commun. **88** (1995), 195–210

[17] K. Hultqvist, R. Jacobsson, K. Johansson. Nucl. Instrum. Methods **A364** (1995), 193–200

[18] G. Mavromanolakis. *Neural networks technique based signal from background separation and design optimization for a W/quartz fiber calorimeter* (2003). Hep-ex/0303021

[19] B. Knuteson, H. Miettinen, L. Holmström. Comput. Phys. Commun. **145** (2002), 351–356

[20] M. Mjahed. Nucl. Instrum. Methods **A481** (2002), 601–614

[21] B. Roe, H.-J. Yang, J. Zhu, Y. Liu, I. Stancu, G. McGregor. Nucl. Instrum. Methods **A543** (2005), 577–584

[22] D0 collaboration. Phys. Lett. **B517** (2001), 282–294

[23] M. Mjahed. Nucl. Instrum. Methods **A449** (2000), 602–608

[24] D. Jeans. *A discussion of discriminant techniques*. DELPHI note 2001-135, TRACK-97 (2001). Available at the following URL : `delphiwww.cern.ch/pubxx/delnote/`

[25] T. Malmgren, K. Johansson. Nucl. Instrum. Methods **A403** (1998), 481–489

[26] R. Fisher. Ann. Eugenic. **7** (1936), 179–188

[27] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Kluwer Academic Publishers (1989)

[28] R. Lotlikar, R. Kothari. IEEE Trans. Pattern Anal. **22** (2000), 623–627

[29] J. Speltz, for the STAR collaboration. J. Phys. G: Nucl. Partic. **31** (2005), S1025–S1028

[30] J. Speltz. *Excitation functions of the multi-strange baryons production in ultra-relativistic heavy-ion collisions*. Ph.D. thesis, Université Louis Pasteur, Strasbourg (2006). In French

[31] J. Speltz. Private communication

[32] ALICE collaboration. J. Phys. G: Nucl. Partic. **32** (2006), 1295–2040

[33] J. Faivre. *Linear Discriminant Analysis (*LDA*) for selection cuts*. Talk at the Alice Week of Utrecht, June 14th 2005. URL : `indico.cern.ch/conferenceDisplay.py?confId=a051381`

[34] J. Faivre. *Linear Discriminant Analysis (*LDA*) for charm detection*. Talk at the Alice Physics Week of Erice, Dec 6th 2005. URL : `indico.cern.ch/conferenceDisplay.py?confId=a057060`