Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope

R.K.Bock [a,*], A.Chilingarian [b], M.Gaug [c], F.Hakl [d], T.Hengstebeck [e], M.Jiřina [d], J.Klaschka [d], E.Kotrč [d], P.Savický [d], S.Towers [f], A.Vaiciulis [g], W.Wittek [a]

[a] *Max-Planck Institut für Physik, München*

[b] *Cosmic Ray Division, Physics Institute, Yerevan*

[c] *Institut de Fisica de Altes Energies, Barcelona*

[d] *Institute of Computer Science, Academy of Sciences of the Czech Republic*

[e] *Fachbereich Physik, Universität-GH Siegen*

[f] *State University of New York at Stony Brook*

[g] *University of Rochester*

Abstract

We present results from a case study comparing different multivariate classification methods. The input is a set of Monte Carlo data, generated and approximately triggered and pre-processed for an imaging gamma-ray Cherenkov telescope. Such data belong to two classes, originating either from incident gamma rays or caused by hadronic showers. There is only a weak discrimination between signal (gamma) and background (hadrons), making the data an excellent proving ground for classification techniques.

The data and methods are described, and a comparison of the results is made. Several methods give results comparable in quality within small fluctuations, suggesting that they perform at or close to the Bayesian limit of achievable separation. Other methods give clearly inferior or inconclusive results. Some problems that this study can not address are also discussed.

Key words: Classification, discrimination, multivariate, neural networks, kernel methods, nearest-neighbour, regression trees

* Corresponding author.
Email address: rkb@mail.cern.ch (R.K.Bock).

1 Introduction

Astronomy, astrophysics, and particle physics all have made rapid progress as observational sciences in recent years. Much of this progress is due to the development of detector technology, coupled with a parallel development of analysis methods. Confronted with a daunting challenge of extracting a small number of interesting events from an overwhelming sea of background, both having very similar characteristics, physicists have become familiar with quite sophisticated multivariate techniques such as probability density estimators, regression (or classification) trees, kernel functions, neural networks, etc. Dedicated conferences like [1] have tried to create a close link between the observational sciences and recent developments in statistics.

Ground-based gamma-ray telescopes are an example of experiments exploring a new research frontier, that are likely to benefit from the many multivariate data analysis techniques developed in recent years. The following sections describe very briefly some problems arising in the analysis of their data, and the general characteristics of signal and background events in such devices. In this case study, various multivariate techniques are applied to the same set of data, in order to determine which techniques appear to afford the best discrimination between signal and background.

2 Reference data

2.1 Ground-based imaging Cherenkov telescopes

Ground-based atmospheric Cherenkov telescopes using the imaging technique are a comparatively recent addition to the panoply of instruments used by astrophysicists. The first results were demonstrated in 1989. They observe high-energy gamma rays, taking advantage of the radiation emitted by charged particles as they are produced abundantly inside the electromagnetic showers initiated by the gammas, and developing in the atmosphere. This Cherenkov radiation (of visible to UV wavelengths) leaks through the atmosphere and gets recorded in the detector, allowing reconstruction of shower parameters.

Cherenkov telescopes on the ground can be built with a much larger effective collection area than detectors sent into orbit, and hence make them respond better to the low fluxes of high-energy primary gamma-rays. On the other hand, the number of observable Cherenkov photons for primary gammas of lower energy (below 100 GeV) becomes comparatively small, and correspondingly the problems of discrimination against background get enhanced. Opti-

mal light collection and the best possible use of information thus are critical for the success of this technique. The available information consists of pulses left by the incoming Cherenkov photons on the photomultiplier tubes, arranged in a plane, the *camera*. Depending on the energy of the primary gamma, a total of few hundreds to some 10000 Cherenkov photons get collected, in patterns (called the *shower image*), allowing to discriminate statistically those caused by primary gammas (signal) from the images of hadronic showers initiated by cosmic rays in the upper atmosphere (background). An early review on this subject can be found in [2].

Hadron showers often contain an important, and sometimes even dominant electromagnetic component. Also, both types of showers are subject to fluctuations. Partly, they are due to the showering process itself; more fluctuations are caused by the atmosphere, which acts like a permanently changing calorimeter. Signal and background images thus are usually distinguishable only with a non-zero probability of misclassification.

Typically, the image of a shower after some pre-processing (not the subject of this note) is an elongated cluster; its long axis is oriented towards the camera center if the shower axis is parallel to the telescope's optical axis, i.e. if the telescope axis is directed towards a point source. Hence, a principal component analysis [3] is performed in the camera plane, which results in a correlation axis and defines an ellipse (see figure 1). If the depositions were distributed as a bivariate Gaussian, this would be an equidensity ellipse. The characteristic parameters of this ellipse (often called Hillas parameters [2]) are among the image parameters that can be used for discrimination. The energy depositions are typically asymmetric along the major axis, and this asymmetry can also be used in discrimination. There are, in addition, further discriminating characteristics, like the extent of the cluster in the image plane, or the total sum of depositions.

2.2 Data used in this comparative study

For our case study, we used data sets generated by a Monte Carlo program, Corsika, described in [4]. The program was run with parameters allowing to observe events with energies down to well below 50 GeV. The subsequent analysis, nevertheless, follows the known patterns proven for much higher energies: after applying calibration constants to the deposited energies, a trigger algorithm is used, and some image cleaning on pixel clusters is applied. Image cleaning eliminates local background, e.g. that caused by the night sky around the source to be measured, or outliers. Subsequently, the analysis is simplified, with hopefully little or no loss of information, by converting the pixel image of a shower into few image parameters as described earlier [2]. These parameters

constitute the only image characteristics to be used.

In the work presented here we perform a general study, applying different multivariate classification methods to events described by 10 chosen image parameters, without claiming the parameters to be optimal for these events. Neither in triggering nor in the processing up to image parameters, has any care been taken to optimize these Monte Carlo data. They are used only for a relative comparison of discrimination methods. Also, no particular care was taken to choose independent parameters since a robust discrimination method should itself take care of correlations between parameters.

The data consist of two classes: gammas (signal) and hadrons (background). Events were generated at shower energies from 10 GeV up to about 30 TeV, and for zenith angles from zero to 20 degrees. The samples used by all methods are identical, and consist of 12332 gamma events and 6688 hadron events [1].

Each event is characterized by the following ten parameters (see fig. 1):

1	*length* :	major half axis of ellipse [mm]
2	*width*:	minor half axis of ellipse [mm]
3	*size*:	10-log of sum of content of all pixels [photon count]
4	*conc2*:	ratio of sum of two highest pixels over *size* [ratio]
5	*conc1*:	ratio of brightest pixel over *size* [ratio]
6	*pdist*:	distance from brightest pixel to center, along major axis [mm]
7	*m3long*:	3rd root of third moment along major axis [mm]
8	*m3trans*:	3rd root of third moment along minor axis [mm]
9	*alpha*:	angle of major axis with vector to origin [deg]
10	*dist*:	distance from origin to center of ellipse [mm]

All multivariate methods studied here use identical disjoint training (learning) and control (test) samples. The advantage of working with Monte Carlo events is, of course, that results can be compared to the known classification of events. The figures of merit used to compare methods are estimators derived from the numerical relation between the hadron and gamma acceptances.

[1] All data and their description are available from the authors on request (email to rkb@mail.cern.ch)

3 Multivariate classification

If confronted with a single test statistic, resulting in a one-dimensional probability density different for signal and background events (in our case gammas and hadrons, respectively), the discrimination is simple: by applying a *cut*, i.e. a selection that retains all events with this parameter larger (or smaller) than a fixed *cut value*. For a small enough cut value one obtains zero acceptance for both signal and background, for a large enough cut value an acceptance equal to one for both samples. Cut values chosen in between will make the two acceptances lie on a curve typically deviating from equal acceptance for the two samples. The deviation is the larger the better the variable was chosen, i.e. acceptances are higher for signal than for background events. This diagram of signal vs. background acceptance (we use the terms ϵ_γ and ϵ_p) is known under the name *Neyman-Pearson diagram*, *decision quality diagram* or *Receiver Operator Characteristic* (ROC) curve [5]. In statistical terms, the ROC curve shows the probability of false alarm on the x-axis and the probability of detection on the y-axis.

Note that these acceptances are directly related to statistical quantities like power, purity, cost, contamination etc. Once the sample sizes are fixed, the acceptances also define the significance, defined in section 5, a very important measure for establishing the existence of an observed source.

The classification problem becomes considerably more involved when faced with multiple variables, the *multivariate discrimination* problem. Mathematically, the problem can be formulated like this: measurements $x = (x_1, x_2, ..x_p)$ are made and are known to belong to one of two groups, signal or background. The task is to construct a classifier, which associates every x to one of the groups such that misclassifications are minimized, on the basis of a training sample of events with known classification.

Most of the multivariate methods provide a model, which assigns a real-valued variable between 0 and 1 to each of the events, which may be understood as a confidence that the event is signal. For these methods, a good model of finding the new variable is the key issue, whereas the analysis may be done in exactly the same way as for univariate methods, i.e. the computed confidence may be considered a new, derived, variable fully describing the event. For these methods, the decision quality may be represented by the ROC curve defined above, i.e. by the relationship between $\Pr(accepted|background)$ and $\Pr(accepted|signal)$.

For multivariate methods, which do not provide a real-valued confidence variable, at least not by default, we have to construct several models using different values of a learning parameter. For each of these models, we obtain a pair

Pr(*accepted*|*background*), Pr(*accepted*|*signal*). These pairs represent points, which may be used instead of an ROC curve.

Since the exact description of the distribution is not known, we cannot construct the exact ROC curves even for methods, which, theoretically, allow this. Hence, a more pragmatic approach has to be taken, applying the tested methods to a standard benchmark sample. All methods used the first 2/3 of the simulated (Monte Carlo) background and signal events to construct the model (*training sample*) and the remaining 1/3 of the data to calculate the ROC curve (*control sample*).

A numerical comparison of the methods in tabular form, which is based on selected points of the ROC curve, is described in section 5.

4 Different classification methods

General multivariate classification methods are advertised in large numbers, with little or no guidance about their virtues and shortcomings. Some of them are available on the commercial market. This has motivated our case study: we compare several methods below; some we have not tried are several variants of discriminant analysis (see below) or variants of the kernel method, like adaptive kernels. Also more ANN (artificial neural network) methods than investigated here, are in widespread use.

4.1 Direct selection in the image parameters

Selecting by parameter cuts as in the one-dimensional case can, of course, also be applied in the n-space of features (in our case image parameters), one variable at a time or logically related like with AND or OR. The problem, however, gets unwieldy even at low n; nevertheless, this is a commonly used method amongst physicists. Wide experience with such cuts exists for all operating Cherenkov telescopes (e.g. [2,6,7]); any method claiming to be superior must use results from these as yardstick. Working in one projection at a time, correlations between the parameters, in particular non-linear correlations, are not easy to take into account, although cuts in a variable are often made dependent on the value of other parameters (*supercuts* or *dynamic cuts*), or the n-dimensional space gets partitioned. Decorrelation by standard methods (Principal Component Analysis, [3]) does not solve the problem in general, being itself a linear operation. Defining new variables may also overcome some difficulties. Usually, optimization leads to separate studies and approximations for each new data set (past experience), which makes results

sometimes difficult to reproduce. The direct selection method also does need an optimization criterion, and will not normally result in a relation between gamma acceptance and hadron acceptance, i.e. no single test statistic is defined, although the acceptance plane can, of course, be populated by varying the selection criteria.

In order to provide such a relation, we chose the cuteval method [8]. This method finds the optimum combination of selection (cut) parameters (together with the optimum cut values) on a test sample by numerical maximization of the quality factor Q defined $Q = \epsilon_\gamma \ / \ \sqrt{\epsilon_p}$, with ϵ_γ and ϵ_p corresponding acceptances (efficiencies) for gammas (signal) and protons (hadron background), respectively (equivalent to Pr($accepted|signal$) and Pr($accepted|background$)). Q could be replaced as a quality factor by the more adequate statistical significance $\sigma = S/\sqrt{2B+S}$ where S and B are the unmber of signal and background events in the sample retained after cuts; see also chapter 5. The significance, however, depends on the size of the chosen samples.

Out of the group of $n_p = 10$ parameters, 20 possible cuts $p_i(i = 1 : 2 * n_p)$ were defined (each being allowed to cut in one or the other direction of the one-dimensional projection of each parameter). In the spirit of [7], all parameters except *alpha* and *size* were corrected for their energy (*size*) dependence, yielding cuts on combinations with the parameter *size* (see figures 2 and 3).

In a first step, the most efficient (the one achieving the highest Q) selection parameter s_1 is determined (in this case: $alpha < 11.8$). Out of the pool of now $(2 * n_p - 1)$ available cuts, a second parameter s_2 is determined which achieves the highest value of Q together with s_1 and freely varying cut values [2]. Successively, more parameters are determined until the addition of new parameters cannot improve Q any more. The thus obtained m parameters $s_j(j = 1 : m)$ are called "independent". To each parameter s_j belongs a cut value $c_j(j = 1 : m)$. By this procedure correlations between cuts are taken into account and the final number of cut parameters is reduced to a minimum while at the same time maximizing the cut efficiency. Surprisingly, only four "independent" parameters were obtained and 16 rejected. These four cuts yield a quality factor of $Q = 4.0 \pm 0.15$ on the control sample and are shown in the following table:

[2] The search for the global maximum used simulated annealing methods.

Nr.	Parameter and selection
1	$alpha < 7.8$
2	$length/(-67. + 40. * size) < 1.35$
3	$width/(27.9 - 22.5 * size + 6.7 * size^2) < 1.12$
4	$size > 3.2$

In a second step, the optimum cut values of the "independent" parameters are calculated again, this time maximizing Q with imposed boundary conditions, e.g. a minimum hadron acceptance ϵ_{min}. The obtained cut values are then fitted and parameterized as a function of ϵ_{min} (see figure 4), a procedure which defines a single event quality parameter comparable to the "purity" of the sample. The obtained parameterized path is given in the following table:

Parameter	Path
$alpha <$	$7.3 + 94.1 * \epsilon_{min}$
$length/(-67. + 40. * size) <$	$1.37 + 6.39 * \epsilon_{min} - 12.42 * \epsilon_{min}^2$
$width/(27.9 - 22.5 * size + 6.7 * size^2) <$	$1.16 + 1.24 * \epsilon_{min}$
$size >$	$3.1 - 13.6 * \epsilon_{min} + 44.2 * \epsilon_{min}^2$

Note that the two parameters *size* and (corrected) *length* "saturate" at high values of ϵ_{min}, i.e. they are set to values with no rejection to signal or background at all.

4.2 Classification trees and forests

The basic classifier has the form of a binary decision tree. Its internal nodes represent tests, each of which compares a single predictor to a fixed threshold. Each of the leaves is labeled by one of the two classes: signal and background. The tree may be constructed from the training set using different strategies [9,10], which we briefly describe later. Classification of a new case starts at

the root node of the tree. The test assigned to this node is evaluated and the computation continues to the left or right subtree according to the answer. This is performed repeatedly until a leaf is reached. Its label represents the resulting prediction.

A significant improvement over the accuracy of a single classification tree may be achieved by constructing a collection of several different trees (a *forest* or an *ensemble* of predictors) and using a voting or an averaging procedure for the classification of new cases [11,12]. Constructing an ensemble of predictors may improve the accuracy of not only decision trees, but also other types of predictors, when dealing with single predictors of low bias but large variance [15].

Our experiments used three different strategies to construct forests. The first two used the two classical techniques CART [9] and C4.5/C5.0 [10] to construct several trees, which were further combined into a forest. In order to obtain different trees using C5.0 and CART, we used *bagging* (bootstrap aggregation), where each tree is grown on a random subsample of the training data, drawn with or without replacement. The last experiment used *Random Forest* technique, which is directly designed to construct a forest [13].

Let us briefly describe the construction of a single tree. The construction begins with all cases being contained in the root node. The root node is then split by a cut using one of the image parameters, into two successive nodes and this process is repeated recursively. In each step, the split is selected in order to achieve maximum decrease of *average impurity*. This partitioning is performed until either all the obtained regions are approximately pure, i.e. contain a clear majority of one class in the training data, or further splitting is not possible. In order to avoid overfitting, the resulting tree is finally simplified by *pruning*. The two tree-induction methodologies, namely CART [9] and C4.5 [10], use similar or the same definition of average impurity, but differ significantly in the strategy for pruning.

4.2.1 C5.0

We used C5.0 Release 1.15 (Linux PC) by RuleQuest Research [3], which uses the C4.5 algorithm for induction of individual trees, but extends C4.5, in particular, by the option of using forests. Bagging using C5.0 was performed by an external script, which called C5.0 for a sequence of different settings of the cost for misclassification of a background event (i.e. as signal). For each of these settings, C5.0 was called for 80 randomly selected subsamples, constructing one tree for each subsample. The resulting forest of 80 trees was then used to classify the control sample using *see5-public*, an open source

[3] http://www.rulequest.com/

utility provided together with C5.0. Each setting of the misclassification cost gives one point in the ROC curve. The best results were obtained, if each of the random subsamples contained 80% of the training data. This size of subsamples was used to obtain the results given below.

4.2.2 CART

We used CART 4.0 (Windows) marketed by Salford Systems [4] which uses a boostrap sample for growing the tree and the whole data set for pruning (technically, the "test=nosample" option for pruning was used). A sequence of settings of priors (prior probabilities of the two classes [9]) was selected and for each of these settings an independent construction of a forest of 80 trees was performed. Hence, we obtained a sequence of forest classifiers, each of which was tested on the control sample and the result of each forest is represented by one point in the graph. Since different settings of priors imply different amounts of accepted background and signal events, a curve is obtained.

4.2.3 Random Forest

In the random forest method, again a large number of classification trees is grown and combined. Two random elements serve to obtain a random forest, bagging and random split selection.

Bagging is done here by sampling multiple times with replacement from the original training data set. Thus in the resulting samples, a certain event may appear several times, and other events not at all. About 2/3 of the data in the training sample are taken for each bootstrap sample.

Random split selection is used in each tree's growing process. The tree growing begins with all cases being contained in the root node. The root node is then split by a cut using one of the image parameters, into two successive nodes to achieve a classification by separation of the classes. When using a total of 10 image parameters, three (square root of total, an experimental best value [16]) are chosen randomly (uniformly distributed) from the total 10. The image parameter yielding the smallest Gini-index among these three is used for splitting; [9] gives more information on the Gini-index. Using only one random variable for selection was not sufficient, but using 2 or 4 yield very similar result as 3 variables.

Subsequently, the same procedure is applied to each branch in turn. The tree growing stops only when all nodes contain pure data, i.e. from one class only.

[4] http://www.salford-systems.com/

These nodes are then called terminal nodes and receive their class label from the training data.

No pruning is performed. Using unpruned trees (in general poor classifiers) requires a reasonably large number of them to be combined. For our data, growing 50 trees turns out to be sufficient: the quality of the classification in terms of the Neyman-Pearson plot remains unchanged after a certain number of trees has been reached (see figure 5). Combining classifications from the trees is simply done by calculating the arithmetic mean from the 50 classifications of all trees (considered as 0 and 1).

We used the original Breiman's fortran sources for Random Forests technique [14] and verified and extended the results using the package randomForest 3.3-2 in R [5].

The simple arithmetic mean for combination seems to be adequate when dealing with a reasonably large number of unpruned trees. We also tried weighted combinations of trees using several different estimates of confidences of the prediction in individual leaves, but this gave better results only for a very small number of trees ≤ 10, which is too small a number to assure convergence in classification error. This fits well into the assumption that a large enough forest is less sensitive to variance than to bias.

4.3 Kernel methods

Kernel Probability Density Estimation (PDE) methods can be used as a non-parametric multivariate classification technique, and are based on the premise that the density function of the unknown distribution can be approximated by a sum of some other, appropriately chosen, 'kernel' functions [17]. 'Non-parametric', in this context, simply means that no other assumptions are made about the form of the probability density functions (PDF's) from which the samples are drawn.

We restrict these studies to PDE methods based on a Gaussian kernel, which is a natural choice for most physics analysis applications since nearly all variables used in an analysis have usually been Gaussian smeared by detector resolution, or other effects.

A typical application of Gaussian kernel PDE's begins with a sample of N Monte Carlo events generated in a k-dimensional parameter space. The Monte Carlo events are distributed according to some (unknown) probability density function (PDF). A Gaussian kernel PDE method estimates the value of the

[5] http://www.r-project.org/

PDF at a point $\vec{x}$ by the sum of Gaussians centered at the Monte Carlo generated points $y_1, \ldots, y_N$:

$$f(\vec{x}) = \frac{1}{N|V|^{1/2}(2\pi)^{k/2}h^k} \sum_{i=1}^{N} \exp\left[-\frac{\vec{d_i}^T V^{-1} \vec{d_i}}{2h^2}\right], \tag{1}$$

where $\vec{d_i} = (\vec{x} - \vec{y_i})$, V is a covariance matrix, and h is an additional scaling factor. The optimal forms of V and h are a matter of debate, but a minimally biased estimate of the original PDE can be obtained if one determines V from the covariance matrix of the overall sample and the scale factor h is set to $N^{-1/(k+4)}$[18]. An attempt to optimize h showed that the result is robust with respect to choices of h within a factor of 2. Because the parameters of the resulting Gaussian kernel are the same for all points, this is known as a *static kernel* method.

Given a data point in the k-dimensional parameter space, one can use Equation 1 to estimate, based on the Monte Carlo generated signal events, the signal PDF at that point, P_s. One can similarly estimate, based on the Monte Carlo generated background events, the background PDF at that point, P_b. A one-dimensional discriminator can then be formed from $P_s/(P_s + P_b)$.

Kernel PDE methods, in general, do an excellent job of modelling complex inter-correlations in high-dimensional parameter spaces, as long as the PDF's are smoothly varying within that parameter space. They thus often perform as well or even better than more sophisticated techniques such as classification trees or artificial neural networks. Static kernel PDE methods tend to suffer from the disadvantage, however, that the discriminant computational time per data event grows linearly with the size of the Monte Carlo sample used as reference. Techniques such as classification trees or artificial neural networks, get 'trained' only once, and the discriminant computational time depends only on the topology of tree or neural network, not the size of the Monte Carlo samples.

4.4 Artificial neural networks (ANN-s)

This class of methods has been presented frequently in the past; ANN-s resemble the tree-based methods in that they define simultaneous selections in variables, but instead of the original variables they work in locally linearly transformed data, and the transformation itself is part of the optimization (learning) process. Usually, the results match but are not superior in quality to whatever reference results exist for comparison. So far, no convincing case has been made for the use of ANN-s on Cherenkov telescope data, although

the Whipple collaboration has tried the method (and remained with their supercuts). There is a substantial randomness in choosing the topology of the net, in particular the depth of the tree, the number of nodes, the training method, transfer function, etc.

4.4.1 The NeuNet package

As one neural network implementation the NeuNet-package for ROOT [6] by J.P. Ernenwein was chosen [19]. This package allows for building multilayer feed forward nets. For our study a 3-layer architecture with 10 input nodes (for the 10 image parameters), 10 nodes in the hidden layer and 1 output node (giving the classification) was adopted. Further features of this neural network package are random initialization of weights and biases (boundaries for initialization of weights are user-defined). The transfer function is implemented as sigmoid, i.e. $\sigma(x) = 1/(1 + \exp(-x))$. For teaching the neural net, error back propagation with a user-defined learning rate is used.

The neural network output denoted by o can be written as

$$o = \sigma\Big(\sum_{i=0}^{9} w_{1i0} \cdot x_{1i} + b\Big), \quad \text{where} \quad x_{1i} = \sigma\Big(\sum_{j=0}^{9} w_{0ji} \cdot x_{0j} + b_{1i}\Big).$$

Here the x_{kl} are the input values for node l, layer k and the w_{klm} are the weights connecting node l, layer $(k-1)$ with node m, layer k. d_{kl} denotes the biases for the input values x_{kl}. The program outputs training and validation errors which are calculated as mean value of $|\, o_{true} - o \,|$ of training or control sample respectively. This is not the final classification error, which is obtained taking $int(o+1-c)$ with c being the cut value and $int(.)$ the integer operator as estimated class (and not o itself). One must also take care of the input values to be scaled to $[\,0, 1]$ because the program uses a default clipping operation. For the benchmark, a learning rate of 0.1 and 10.000 training cycles were used. In figure 6 one can see (left) the evolution of training and control sample errors with the training cycles, and (right) a histogram of the ANN output (signal probabilities) for signal and background events.

4.4.2 NNSU - Neural Network with Switching Units

The NNSU is a combination of a classical neural network architecture and a classification tree. This network is actually an oriented acyclic graph whose nodes are structures called building blocks. This acyclic graph will be refered to as the outer graph. Each building block is a neural network consisting of

[6] http://root.cern.ch/

two types of nodes. These nodes are connected such that they form an acyclic graph again, with the restriction that the output dimension of each building block is the same for all building blocks in the outer graph. The first type of node, which we call functional unit, makes a predefined mapping from the input space corresponding to this node to the output space of this node. Hence such a node can be described by a tuple of integers, input vector dimension and output vector dimension, and by a transfer function. The definition of this transfer function includes the parameters of this functional unit (weight vectors, threshold etc. in current neural network terminology).

Functional units map corresponding inputs to outputs, by an internal transfer function. This transfer function is linear, the coefficients differ for each functional unit. The second type of nodes, switching units, collect all outputs from parent functional units, concatenate them to form one vector, and search a predefined number of clusters in the set of such input vectors. We use a non-deterministic procedure known as Jancey cluster algorithm.

After clustering, each cluster is joined with a corresponding child functional unit; subsequently the parameters of this functional unit are adjusted using patterns in the corresponding cluster only. In fact, the division of input patterns into two or more disjoint sets, and the consecutive learning over these subsets of patterns, put a separating hyper-surface into the input space. The type of these hyper-surfaces is defined by the type of transfer functions of the switching unit parents.

So each building block in turn is optimized, the output from each building block is propagated to all children, and the output of the last building block is considered to be the final output from the neural network.

4.4.3 GMDH - Group method data handling

The Group Method Data Handling is a polynomial approximator, with the polynomial degree controlled according to the quality of the approximation reached. The selection of an appropriate degree of polynomial is a well-known problem of polynomial approximation. Because the degree of the polynomial is inherently controlled according to the character of the task solved, also the size of corresponding neural net is controlled. In description of the original GMDH algorithm we quote [20] as follows (we use *approximation* instead of *prediction*): "We start by computing the regression equations $y = A + Bx_1 + Cx_2 + Dx_1x_1 + Ex_1x_2 + Fx_2x_2$ for each pair of input variables $x1$ and $x2$ and the output y. This will give us $m(m-1)/2$ higher-order variables for predicting (approximating) the output y instead of the original m variables $x_1, x_2, ...x_m$. After finding these regression equations (from a set of input/output observations), we then find out which one to save. This will give

us a collection of quadratic regression models (say, m_2) which best approximate y (note that each approximation depends on two independent variables).

We now use each of the quadratic equations that we have obtained and generate new independent observations (which will replace the original observations of the variables $x_1, x_2, ...x_m$). From these new independent variables we will obtain combinations exactly as we did before. That is, we compute all of the quadratic regression equations of y versus these new variables (two at a time). This will give us a new collection of $m_2(m_2 - 1)/2$ regression equations for approximating y from the new variables, which in turn are estimates of y from the previous equations. Essentially what we have now is a collection of polynomials of degree 4 in four variables. We now merely select the best of these estimates, generate new independent variables from the selected equations to replace the old, and combine all pairs of these new variables."

The process will continue until some convergence criterion is reached. Each regression equation arises from combining two variables from the preceding set. One can see the process as building the neural net by adding two-input neurons to the preceding. Instead of the standard summation function and a transfer function, the quadratic function above is used [21,22].

4.4.4 MRS and MLP Neural Networks

Two more analyses with neural networks are included in our results (table 1 in section 5): a feed-forward network [23] with multistart random search (MRS), and a multilayer perceptron fit [24] (MLP). The MRS approach uses a subset of parameters considered optimal, and selects multiple intial weights and net configurations, of which it retains the one with the best result (on the control sample). The MLP is part of the TerraFerMA package [25] and puts the accent on powerful training methods and double-precision calculation.

4.5 Nearest Neighbours

The kernel density (see above) assigns weights to events in the reference sample, which quickly vanish, if the distance from the new point exceeds certain radius controlled by the parameter h. A similar effect may be reached even more directly by looking only at the events in a window of a given radius. However, the density of points in different regions of the parameter space is different. This has prompted us to also attempt a method using a fixed number of unweighted nearest neighbours. In fact, nearest-neighbour methods are simple and hence popular, their problem being that of defining a valid metric. Often, variables are simply scaled over their range so that they cover the domain from zero to one or normalized to have zero mean and unit variance. We

have used a metric used also in the Gaussian kernel function above. We call C_r the covariance matrix of the variables in the overall sample, and form the sums $S_{g,p}$ of squared distances $\Delta = ((x - x_r)^T C_r^{-1}(x - x_r)$, summing over the k smallest distances in the reference samples g and p, and using $R_g = S_p/S_g$ as discriminating test statistic (*gammaness*).

One should note that a principal component transformation (PCA) of our ten image parameters (making them linearly independent) and a suitable scaling of the new variables making use of the eigenvalues, results in variables whose variances all are the same and covariances are zero. The Euclidean distance in this space fully corresponds to the kernel distance as introduced above:

$$\Delta = (\xi - \xi_r)^T(\xi - \xi_r) = (x - x_r)^T C_r^{-1}(x - x_r),$$

where

$$(\xi - \xi_r) = (x - x_r)E_r.$$

E_r is the eigenmatrix of C_r, with each eigenvector (column) divided by the square root of the corresponding eigenvalue.

The computational gain by this orthogonalization is, however, a minor one, and partly compensated by the need to go through the linear transformation using E_r. The computation time in our implementation is entirely dominated by the access to every event in the reference samples, which is the same as for the kernel method.

A possible reduction of parameter space has also been explored with this method, albeit superficially: there is only a small loss in classification quality when reducing from ten to eight parameters, but beyond that, losses become visible. The choice of which parameters (in PCA-transformed space) are left out does not seem to be relevant; in particular, using this metric it does not seem associated with the eigenvalues (regularization viz. reduction in dimensionality often eliminates the eigenvectors with small associated eigenvalues).

The nearest-neighbour method requires as only choice that of a reference sample (like the kernel method), and in addition the number of nearest neighbours to consider. We have used the training samples as defined for all other methods, but the results do not appreciably change if reference samples down to 1500 events are used, or if the control and training samples are the same. Results are also remarkably robust with respect to the choice of the number of nearest neighbours: for our benchmark, we show results for 25 nearest neighbours (also tried were 15, 50, and 100).

4.6 Support Vector Machines

Support Vector Machines (SVM-s) (see [26]) are currently under very active research within the fields of neural computation and machine learning. SVMs are examples of a broader category of learning approaches which utilize the concept of kernel substitution, thereby making the task of learning more tractable by exploiting an implicit mapping into a high dimensional space. Motivated by statistical learning theory they have been successfully applied to numerous tasks within data mining, computer vision, and bioinformatics. An application in particle identification has been reported about [27]. On our samples, only a rudimentary first attempt was made with SVM, which was not conclusive.

4.7 Composite probabilities

This unpublished method [28] uses event probabilities obtained by comparing the event data to two-dimensional probability densities obtained from a training sample. Densities are determined by histogramming the training data in two dimensions, using bins that give constant bin content for signal data. All 2D projections are used that can be made from the image parameters, i.e. in our case of ten parameters 45 projections. Each bin of each projection ends up with a probability to be signal (due to the bin definition always nearly the same, $1/n_{bins}$), and a probability to be background. An incoming event thus has to be binned; the probabilities associated by the training sample to the bins in each projection are multiplied, and the product, the composite probability, is taken as a single test statistic ('signalness').

The method was previously applied to data from running experiments (Whipple, Hegra CT1), and (unpublished) results at least did match the best results then existing (from tuned supercuts). On the benchmark data set, the performance is inferior to several of the other methods tested.

Results on the training sample are clearly better with this method than on the control sample. This effect is enhanced, if a systematic reduction of projections is attempted: a major gain is achieved in the training sample results, by eliminating projections one at a time (typically, more than half the projections turn out to be 'useless' or even 'harmful'); this gain in the training sample, however, translates into a small loss of quality in the control sample.

A method using bins for likelihood classification in n dimensions does also exist [25] , but has not been applied to the data used in this study.

4.8 Linear discriminant analysis (LDA)

This is a popular method mostly because it results in an elegant parametric calculation. Its objective is to find a linear combination of the original image parameters such that the hyperplane defined by the transformation maximizes the distance between the means of signal (gamma) and background (hadron) samples, simultaneously minimizing the variance inside each sample. The method is fast, simple and robust; it also does not depend on training samples. However, it ignores non-linear correlations in n-dimensional space (because of the linear transformation). The inferior results we achieved with LDA indicate that at least higher-order variables must be introduced (e.g. parameters x, y can be used to obtain the additional parameter x^2y). There are variants to LDA like Quadratic Discriminant Analysis (QDA) and Regularized Discriminant Analysis (RDA) which partly respond to this criticism; we have not explored these.

The formalism of LDA is simple [29]: the transformation into the 'best separable space' is performed by the eigenvectors of a matrix readily derived from the data (for our application: in two classes, gammas g and hadrons p) Given samples $g_i(i = 1, n_g)$ for gammas and $p_j(j = 1, n_p)$ for hadrons, with nvar elements each, find a linear transformation vector a such that the transformed samples are $g' = a \cdot g$ and $p' = a \cdot p$, and the discriminating power $d = (y^T S_b y)\ /\ (y^T (S_b + S_w) y)$ gets maximized, where y is the joint set of g' and p'. S_b(between-class variance) and S_w (within-class variance) are defined by:

$$S_w = \sum_{observations} (x_i - \mu_{class})(x_j - \mu_{class}) \quad \text{and} \quad S_b = \sum_{classes} (\mu_i - \mu_{tot})(\mu_j - \mu_{tot}),$$

where μ_{class} = class mean, μ_{tot} = overall mean, and x is the joint set of observations g and p. This leads, for two classes, to the result:

$$a = \frac{\sqrt{n_g n_p}}{(n_g + n_p)} (S_b + S_w)^{-1} (\mu_{tot_1} - \mu_{tot_2})$$

Like Principal Component Analysis (PCA, [3]), LDA is used for dimensionality reduction: LDA reduces the high-dimensional space to a single variable of best separation; PCA, reduces a high-dimensional space to a space in which not all variables have the same significance, allowing to ignore some of them (regularization). The prime application difference between LDA and PCA is that PCA performs feature classification (e.g. some of the image parameters from our Cherenkov telescope data are obtained by PCA), while LDA performs sample classification.

5 Comparing the different methods

Results are shown graphically in figure 7, and numerically in table 1 giving several figures of merit. The figure relates acceptances for signal (ϵ_γ) and background (ϵ_p): the Neyman-Pearson diagram or ROC curve mentioned earlier (section 3). Table 1 gives as figures of merit *loacc*, *hiacc*, significance σ, and quality factor Q; their meaning is the following: *loacc* is the arithmetic mean of values of signal (gamma) acceptances ϵ_γ obtained by interpolating the curve at the points 0.01, 0.02, and 0.05 for background (hadron) acceptance ϵ_p; *hiacc* is obtained in a similar way by averaging ϵ_γ at the points 0.1 and 0.2 (again for ϵ_p); Q is defined by $Q = \epsilon_\gamma \ / \ \sqrt{\epsilon_p}$, the value given is that obtained at $\epsilon_\gamma = 0.5$ (Q often takes very large values at low ϵ_γ); the significance σ is defined by $\sigma = S/\sqrt{2B + S}$, where $S = \epsilon_\gamma N_S$ and $B = \epsilon_p N_B$ are the number of signal and background events that would be obtained by selecting events in samples with $N_B = 10000$ and $N_S = 500$. We give the value of σ obtained at $\epsilon_\gamma = 0.5$, and the maximum value along the curve, along with the value of ϵ_γ where it is found (in some cases this is at an unacceptably low ϵ_γ).

Method	*loacc*	*hiacc*	$Q_{0.5}$	$\sigma_{0.5}$	σ_{max}	ϵ_γ
Random Forest	0.452	0.852	2.8	8.44	8.74	0.412
C5.0	0.441	0.830	2.7	8.14	8.96	0.408
CART	0.414	0.810	2.6	7.94	8.03	0.538
Nearest Nb.	0.448	0.816	2.6	8.03	9.12	0.317
Kernel	0.443	0.803	2.8	8.43	8.64	0.390
NNSU	0.472	0.731	3.5	9.74	9.82	0.483
NeuNet	0.445	0.839	3.0	8.73	8.75	0.483
MRS	0.348	0.779	2.3	7.16	7.31	0.431
MLP	0.300	0.767	2.2	6.93	7.22	0.576
GMDH	0.280	0.736	2.0	6.55	6.77	0.574
Comp. prob.	0.332	0.728	2.1	6.78	6.83	0.585
Direct Sel.	0.306	0.636	1.8	5.91	7.52	0.153
LDA	0.195	0.638	1.6	5.47	5.80	0.710
SVM	0.124	0.586	1.4	4.81	5.76	0.784

Table 1: Summary of results (six quality parameters, or figures of merit) for all methods, using the full sample. The quality parameters are described in the text.

6 Dependence of results on energy

We have recorded in our sample, along with the parameters, the energy of the incident particle (Monte Carlo value). This has been used to split our samples into three bins of energy, using boundaries that give approximately the same event count for the signal sample, resulting in boundaries at 63 and 126 GeV. The discrimination level obtained for events selected by energy is different for the three bins, when using the overall result obtained for the full sample containing all energies; the best results are obtained for high energies; as shown here for the nearest-neighbor method (the quality parameters have the same definition as shown above):

	loacc	*hiacc*	$Q_{0.5}$	$\sigma_{0.5}$	σ_{max}	ϵ_γ
high energy ($\geq 126GeV$)	0.672	0.885	10.0	14.44	14.47	0.522
medium energy	0.455	0.780	2.5	7.77	10.25	0.280
low energy ($\leq 63GeV$)	0.277	0.707	1.6	5.43	6.72	0.672
full sample (as in table 1)	0.448	0.816	2.6	8.03	9.12	0.317

Table 2: Results in three energy bins, for the nearest-neighbour method, using as reference events the full training sample. The quality parameters are described in the text.

When splitting the samples and calculating separate selection algorithms, the effect is very clearly enhanced, and all results are substantially improved: better separation is possible for events in an energy range if a selection procedure is derived for events in this range, than when mixing them with events of different energies:

	loacc	*hiacc*	$Q_{0.5}$	$\sigma_{0.5}$	σ_{max}	ϵ_γ
high energy	0.798	0.936	13.9	15.06	15.80	0.549
medium energy	0.539	0.825	4.0	10.56	12.42	0.384
low energy	0.325	0.759	2.3	7.18	7.44	0.535

Table 3: Results in three energy bins, for the nearest-neighbour method, using as reference events training samples of the same enery bin. The quality parameters are described in the text.

7 Issues of algorithm implementation

The results of several methods seem to be qualitatively similar. Among the well performing methods, ease of implementation, robustness, computer time for classification, and maybe other secondary criteria may eventually have to decide the method(s) to be used. We have been most convinced by the simple implementation of the kernel and nearest-neighbour methods, and by the fact that both do not need training samples, whose data are to be used with some caution in the analysis. On the other hand, the substantially shorter computer time per event decision may make classification forests an attractive alternative.

8 Conclusions and caveats

We observe from the above results that the results from classification trees, kernel, and nearest neighbour methods are very close to each other; Neural Net (ANN) methods give results over a wide range, between the very best and rather mediocre. This effect is not fully understood, and probably show that under the ANN umbrella name, very different methods exist, and that ANN methods need deeper understanding, they certainly can not be used off the shelf. They also seem more sensitive to highly correlated parameters in the input (some of the results shown were obtained after reducing the number of parameters). The last group of methods (composite probabilities, direct selection, LDA and SVM) can be considered inferior.

The three classification tree methods all use multiple trees, results from single trees are inferior and not shown. The Random Forest method clearly outperformed the two more classical tree growing methodologies C5.0 and CART.

A systematic comparison of methods, as we intended to perform, may give a conclusive result for the given data set. Some of the methods under study show to be superior to direct cuts in parameters, used so far in the analysis of all Cherenkov telescope data, which confirms earlier results: [30,31,28]. The best results also are in good agreement among themselves, most likely demonstrating that close to those results, there is a limit (the Bayesian limit) in separability that can be attained.

However, the conclusions that can be drawn from the results are valid for our input data, Monte Carlo events for a Cherenkov telescope. An extrapolation of those conclusions to different data samples like observations in real-life Cherenkov telescopes, must be validated anew. Generalization to other problems, obviously, is even less addressed by a case study like the present one. A publication of results may, however, facilitate a future validation process, and the study does perhaps allow to discard some of the inferior methods in some generality.

The methods under study all assume an abstract space of image parameters, which is well adapted to Monte Carlo situations - and maybe only to these: real data are subject to influences that distort this space. In the case of Cherenkov telescopes, the star field in the field of view and the night sky background change during observation, the atmospheric conditions vary considerably, unavoidable detector changes and malfunction will occur. Some of these observational effects are taken care of by the transformation from pixel contents to parameters, i.e. pre-processing. None of these distortions of the parameter space have been the subject of our study, of course.

No classification method itself can, of course, substitute improvements in pre-

processing of the image, or, for that matter, invent new, independent parameters containing more information. They may be derived from the image, but could also be derived from new, independent observations, e.g. arrival time or energy of (Cherenkov) photons: to find these requires intuition in physics and good understanding of the detector.

9 Acknowledgements

We want to express our thanks to members of the MAGIC collaboration who have produced and provided the early Monte Carlo programs and the data used in this study. We are particularly indebted to O.Blanch (IFAE Barcelona) and T.Bretz (Universität Würzburg). E. Kotrč was supported by the Ministry of Education of the Czech Republic (Research and Development project LN00A056, Institute of Theoretical Computer Science - ITI). J.Klaschka and P.Savický were supported by grant no. 201/00/1482 from the Grant Agency of the Czech Republic. M.Jiřina and F.Hakl were supported in part by the Ministry of Education of the Czech Republik, project LN00B096.

References

[1] Proceedings of the Conference on Advanced Statistical Techniques in Particle Physics, Durham, 18-22 March 2002, Durham IPPP/02/39.

[2] D.J.Fegan: Gamma/hadron separation at TeV energies, Topical Review, J.Phys.G: Nucl. Part. Phys. 23 (1997) 1013.

[3] I.T.Jollife: Principal Component Analysis, Springer, New York, 1986. Many textbooks explain PCA, e.g. C.M.Bishop: Neural Networks for Pattern Recognition, Clarendon Press Oxford, 1995, or the web site http://www.statsoftinc.com/textbook/stathome.html

[4] D.Heck et al., CORSIKA, A Monte Carlo code to simulate extensive air showers, Forschungszentrum Karlsruhe FZKA 6019 (1998).

[5] J.P.Egan: Signal Detection Theory and ROC Analysis, Academic Press, New York, 1975.

[6] D.Kranich: The temporal and spectral characteristics of the active galactic nucleus Mkn 501 during a phase of high activity in the TeV range, Dissertation an der Technischen Universität München, 2001.

[7] M.Kestel, A method to correct HILLAS parameters of imaging Cherenkov telescope data taken at different background light levels, Proceedings of the 27th International Cosmic Ray Conference, ICRC 2001 Hamburg, Copernicus Gesellschaft 2001.

[8] M.Gaug, AMANDA Event Reconstruction and Cut Evaluation Methods, 2^{nd} Workshop on Methodical Aspects of Underwater / Ice Neutrino Telescopes, Hamburg, August 15-16, 2001, pp.123 - 130, DESY-PROC-2002-01.

[9] L.Breimann, J.H.Friedmann, R.A.Olshen, C.J.Stone: Classification and Regression Trees, Wadsworth, 1983.

[10] J.R.Quinlan, C4.5: Programs for Machine Learning, San Mateo: Morgan Kaufmann (1993).

[11] L.Breiman: Bagging Predictors, Machine Learning 24 (2) pp. 123-140 (1996).

[12] J.R.Quinlan, Boosting, Bagging, and C4.5. Proceedings of *AAAI'96* (1996).

[13] L.Breiman, Random Forests, Machine Learning 45 (1), pp. 5-32 (2001).

[14] L.Breiman, FORTRAN program Random Forests, Version 3.1, available at http://oz.berkeley.edu/users/breiman

[15] L.Breiman, in: Combining Artificial Neural Nets, Amanda J.C.Sharkey (Ed.), Springer-Verlag 1999.

[16] L.Breiman, Manual On Setting Up, Using, And Understanding Random Forests V3.1, available at http://oz.berkeley.edu/users/breiman

[17] T. Hastie et al, The Elements of Statistical Learning, Springer Verlag (2001).

[18] B. Knuteson et al, (2001), abstracts in physics, http://arxiv.org/abs/physics/0108002.

[19] J.P.Ernenwein, NeuNet package for ROOT, available at http://e.home.cern.ch/e/ernen/www/NN

[20] S.J.Farlow, Self-Organizing methods in Modelling - GMDH-Type Algorithms. Marcel Dekker Inc. New York, 1984.

[21] M.Jiřina, The Modified GMDH: Sigmoidal and Polynomial Neural Net. Proc. of the 10th IFAC Symp. on System Identification SYSID'94, Copenhagen, 4-6 July 1994, Preprints Vol. 2, pp. 309-311.

[22] M.Jiřina, M.S.Krayem, Convergence of the Learning with Small Learning Sets in GMDH Neural Net. Neural Network World Vol. 5 (1995), No. 3, pp. 329-339.

[23] A.Chilingarian et al., Multivariate approach for selecting sets of differentially expressed genes, Mathematical Biosciences 176 (2002) p.59.

[24] L.Ametller et al., Discriminating signal from background: Top quark search at the Fermilab Tevatron, Phys.Rev. D54 (1996) p.1233.

[25] S.Towers, TerraFerMA, a suite of Multivariate Analysis Tools, available at http://www-d0.fnal.gov/~smjt/ferma.tar.gz.

[26] N.Christianini, J.Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press 2000.

[27] A.Vaiciulis, Support Vector Machines in the Analysis of top quark production, in Proceedingsa of the Conference on Advanced Statistical Techniques in Particle Physics, Durham, 18-22 March 2002, Durham IPPP/02/39.

[28] R.K.Bock, Some thoughts on gamma/hadron separation using composite probabilities, Internal MAGIC note 99-04 (April 1999).

[29] Web site http://www.statsoftinc.com/textbook/stathome.html, look for discriminant function analysis. For an application in HEP: B.Fabbri, LDA with stepwise method, ALEPH 97-012 (internal report).

[30] P.Moriarty and F.W.Samuelson: Kernel Analysis in TeV Gamma-Ray Selection, Gamma-Ray Astrophysics Workshop, Snowbird, Utah 1999, AIP Conference Proceedings 515 p.338.

[31] S.Dunlea, P.Moriarty, and D.J.Fegan: Selection of TeV Gamma-Rays using the Kernel multivariate technique, Proceedings of the 27th International Cosmic Ray Conference, ICRC 2001 Hamburg, Copernicus Gesellschaft 2001, p.2939.

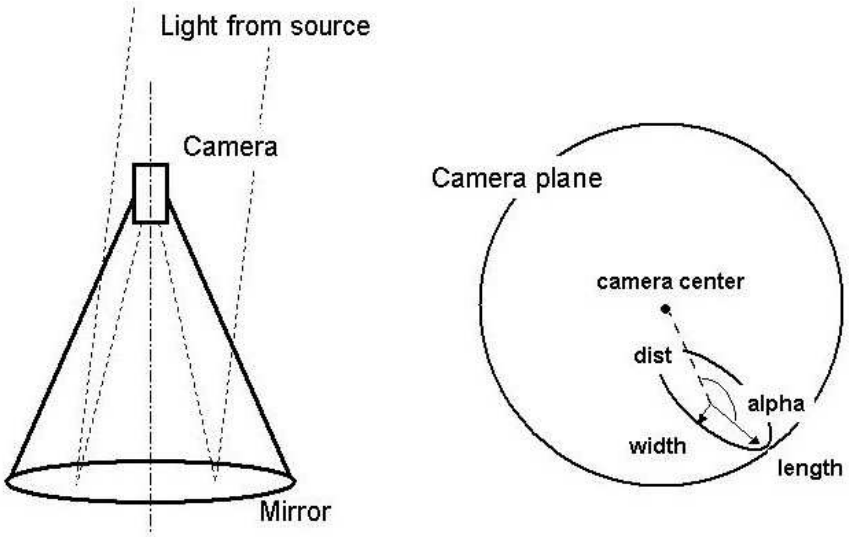

Fig. 1. Cherenkov telescope: sketch and definition of some image parameters

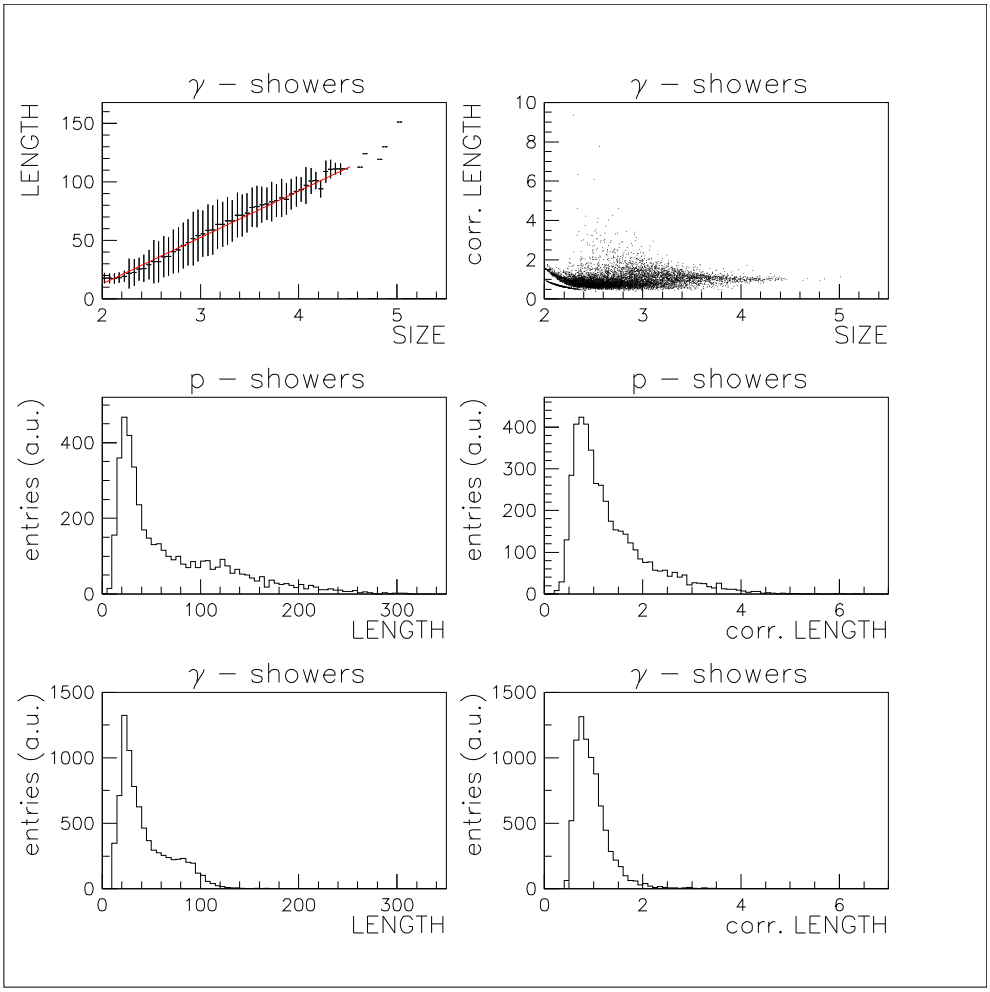

Fig. 2. Energy correction of the parameter *length*: A first order polynomial was fitted to the distribution of *length* values vs. *size* obtained from simulated gamma showers (top left). The corrected values show a smaller energy dependence (top right) and better separation between the proton and gamma samples. The lower graphs show the one-dimensional projections of the parameters *length* (left) and *corr.length* (right).

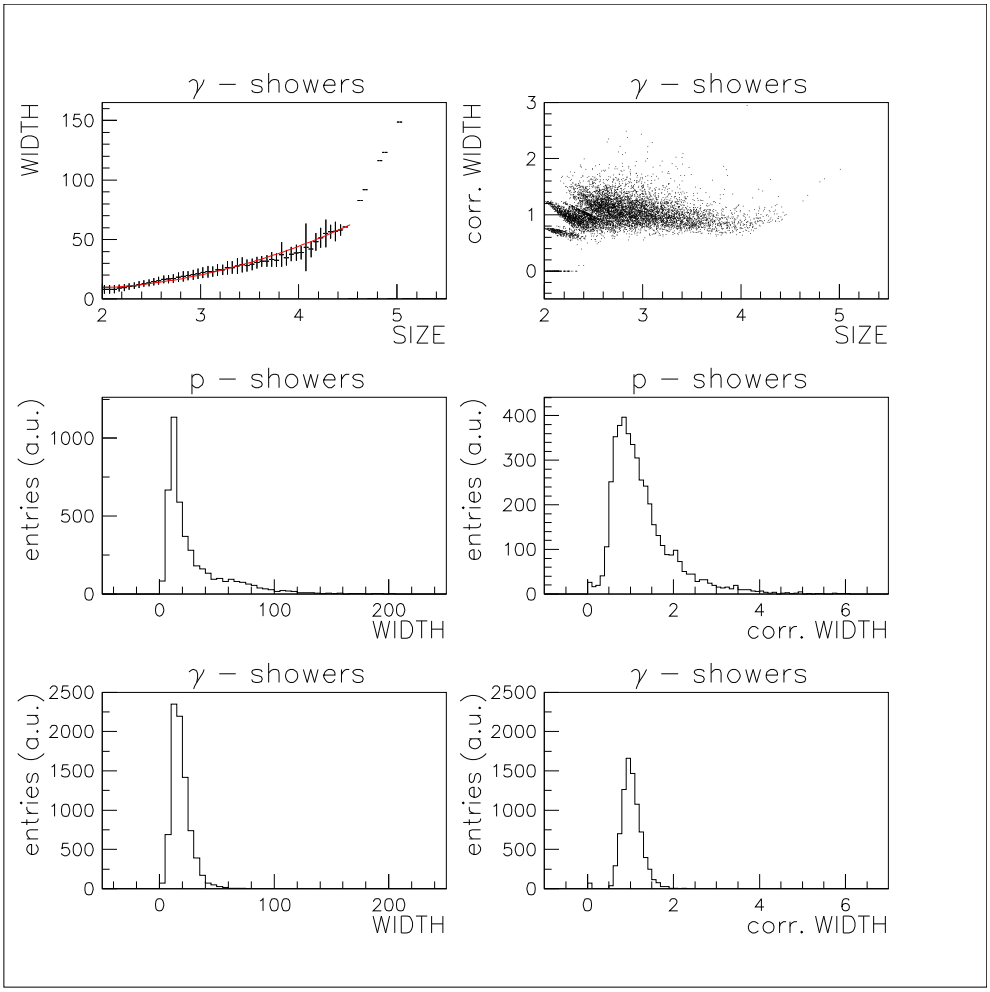

Fig. 3. Energy correction of the parameter *width*: A second order polynomial was fitted to the distribution of *width* values vs. *size* obtained from simulated gamma showers (top left). The corrected values show a smaller energy dependence (top right) and better separation between the proton and gamma samples. The lower graphs show the one-dimensional projections of the parameters *width* (left) and *corr.width* (right).

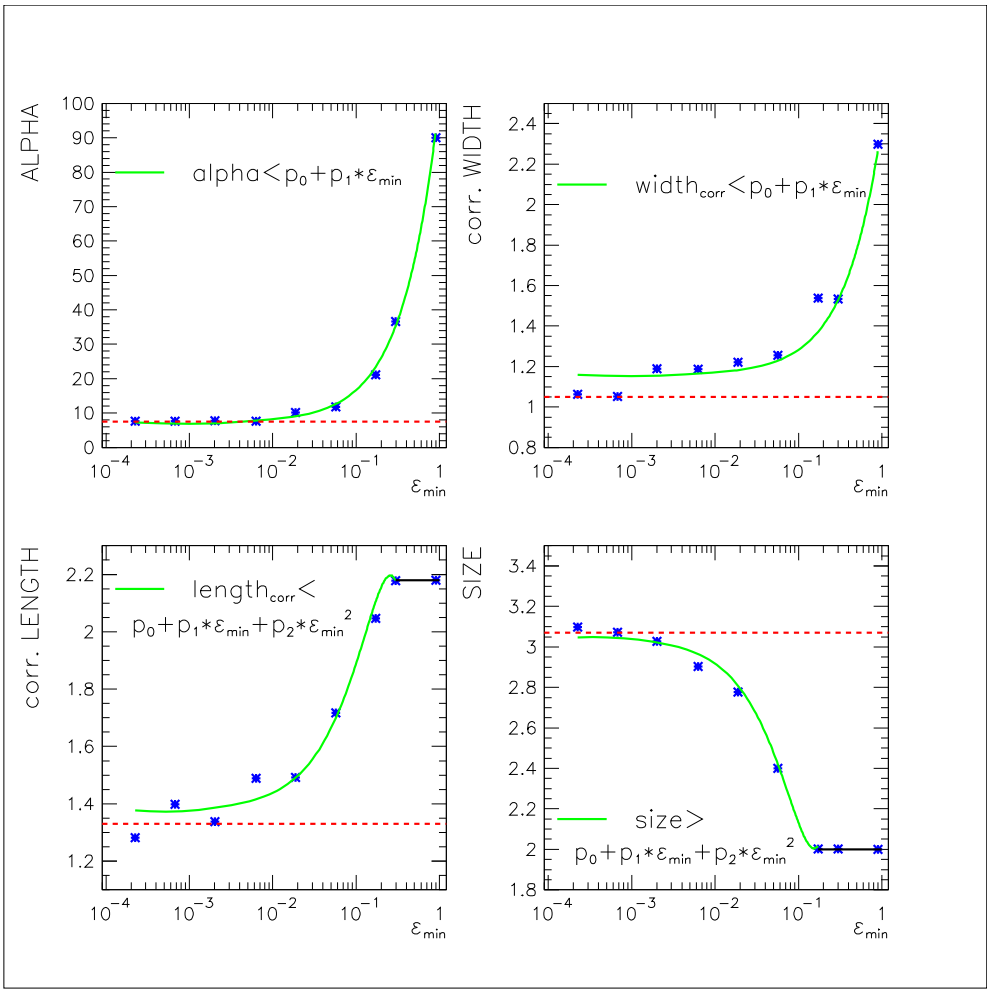

Fig. 4. Obtained cut values of the four "independent" cut parameters by imposing a boundary condition $\epsilon_p > \epsilon_{min}$. Two parameters (*size* and *corr.length*) saturate which is indicated by the horizontal lines. Also, the results of the experimental optimization are drawn as dashed horizontal lines (they are independent from ϵ_{min}).

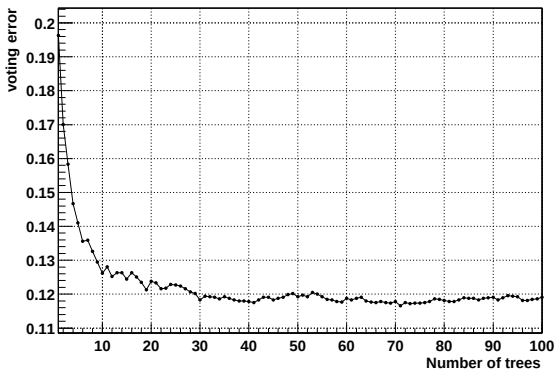

Fig. 5. Voting error as function of number of trees

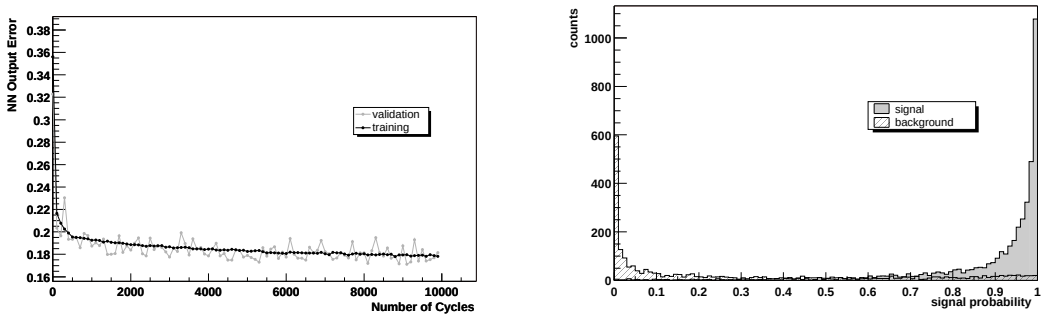

Fig. 6. Two graphs typical for ANN methods. Left: evolution of training and validation errors during the training process; right: NeuNet output for validation (control) data

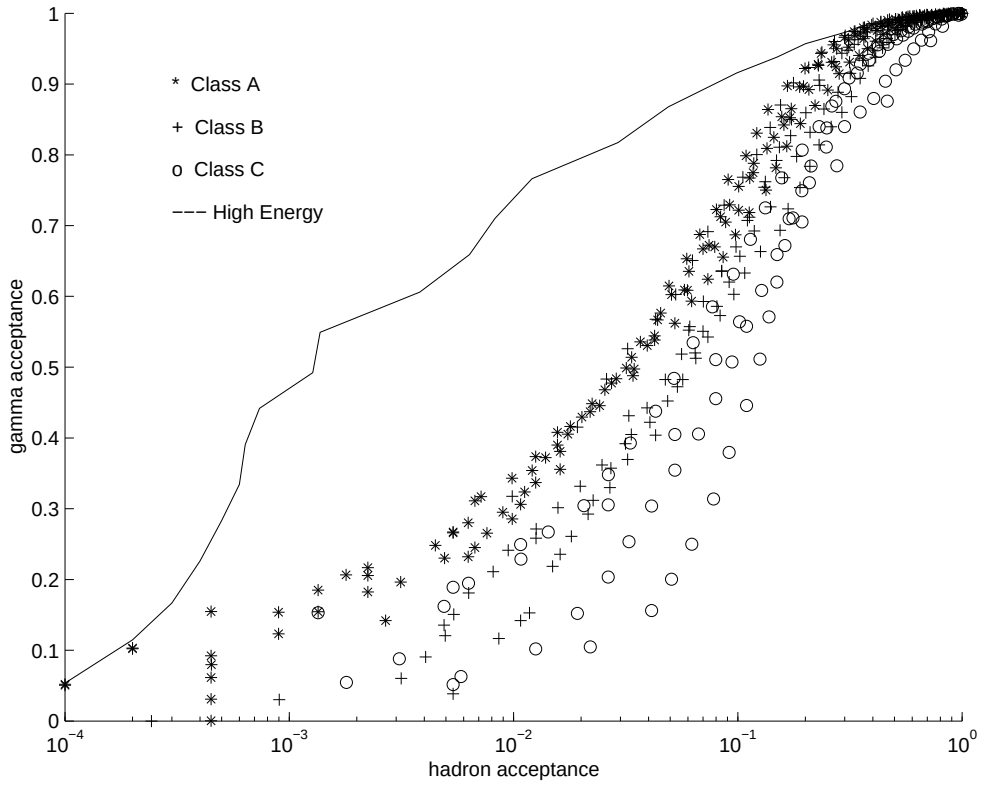

Fig. 7. Several classification methods compared; a logarithmic axis is used to better show the low-acceptance region. Results are grouped: class A are regression tree and PDE methods, class B contains various Neural Net methods, class C comprises SVM, direct selection, LDA, and composite probabilities. For comparison with present telescopes, a (nearest-neighbour) result for events with incident energies above 120 GeV is also shown.