# A COMPARISON OF MULTI-VARIATE PDE METHODS IN NEUTRAL PION DISCRIMINATION

A. Beddall, A. Beddall, A. Bingül, Y. Durmaz

Department of Engineering Physics, University of Gaziantep
Gaziantep 27310, Turkey

*(Received July 13, 2006)*

A number of multi-variate PDE (probability density estimators) methods are compared for the discrimination of signal from background in the selection of neutral pion candidates reconstructed at the ALEPH experiment at CERN. In this case-study, the question "Which method is the best choice?" reveals that the answer depends strongly on the size of the data set used to train and optimise the method, and the required simplicity of the algorithm.

PACS numbers: 29.85.+c

## 1. Introduction

Multi-variate probability density estimators have become important methods in the effective discrimination of signal and background in high energy physics data analysis. Several alternative multi-variate PDE methods have appeared in the literature in recent years, demonstrations of these methods often include a comparison with results from a neural network as neural networks generally give the best performance. A direct comparison between some of these PDE methods, using data generated from a toy Monte Carlo, can be found in Ref. [1].

For the study presented in this paper, a number of PDE methods are employed for the three-dimensional discrimination of signal from background in the selection of neutral pion candidates reconstructed at the ALEPH experiment [2] at CERN using a full physics and detector simulation. This study therefore, provides a real-life case-study with data that contains the complexities of particle dynamics, and the correlations that are often present between variables in high energy physics. The study aims to provide the reader with some useful experience and discussion to aid the researcher in the choice of a PDE method. The issues covered are: algorithm simplicity, discrimination performance, importance of statistics, and the time it takes to train and optimise the algorithm (the run-time).

The PDE methods under investigation are described in Sec. 2, results for discrimination performance and run-time are given in Sec. 3, and a discussion and conclusion is given in Sec. 4.

## 2. PDE methods

The PDE attempts to describe the signal probability, or purity, of a distribution that contains both signal and background data. The signal probability $P(\vec{x})$ for a multi-variate data $\vec{x}$ is shown in Eq. 1 where $f_{\mathrm{S}}(\vec{x})$ and $f_{\mathrm{B}}(\vec{x})$ are estimates of the underlying feature functions of the signal and background distributions respectively.

$$P(\vec{x}) = \frac{f_{\mathrm{S}}(\vec{x})}{f_{\mathrm{S}}(\vec{x}) + f_{\mathrm{B}}(\vec{x})} \,. \tag{1}$$

Here the data is obtained from Monte Carlo simulations of the processes that are under investigation.

The function $P(\vec{x})$ is used in the selection or rejection of events by applying a purity cut, $P_{\mathrm{cut}}$: events with $P(\vec{x}) \geq P_{\mathrm{cut}}$ (high signal probability) are accepted, and events with $P(\vec{x}) < P_{\mathrm{cut}}$ are rejected. The value of $P_{\mathrm{cut}}$ is optimised with respect to a performance measure which in this study is chosen to be the product of the selected signal purity and efficiency, $\varepsilon \times \mathcal{P}$, and optimal is defined by the maximisation of this quantity. At least one other parameter, specific to the method used to form the feature function estimates, is involved in the optimisation of the PDE, optimisation therefore involves a parameter search in at least two-dimensions. To avoid biases in the training data (over-training), the function $P(\vec{x})$ is formed using a training data set, and the optimisation is performed using an independent test data set.

In high energy physics, Monte Carlo simulations generally involve time-consuming modelling of signal and background processes, and detector simulations. Therefore, because of computing constraints, training and test data are limited in statistics. If statistics are not sufficient, the resultant functions $f_{\mathrm{S}}(\vec{x})$ and $f_{\mathrm{B}}(\vec{x})$ will be poor estimates leading to an inaccurate PDE and therefore less than optimal discrimination performance. The problem of limited statistics becomes greater as the dimensionality of the data increases, the so-called "curse of dimensionality".

To combat the problem of statistics, a PDE method employs an algorithm for the formation of the feature function estimates such that the data is generalised, or in other words, smoothed. However, over-smoothing leads to loss of detail, the algorithm and its optimisation must be chosen carefully. A number of such algorithms are summarised in the following sub-sections.

### *2.1. The histogrammed PDE, HPDE*

The histogram is the traditional algorithm for representing the distribution of data. In a simple histogramming algorithm, the signal and background data are binned to create multi-dimensional histogrammed functions $f_{\mathrm{S}}(\vec{x})$ and $f_{\mathrm{B}}(\vec{x})$. Training of the PDE simply involves the formation of the histograms $f_{\mathrm{S}}(\vec{x})$ and $f_{\mathrm{B}}(\vec{x})$, and then a PDE histogram formed by dividing the histogrammed functions according to Eq. 1. Optimisation of the PDE is performed with respect to the binning: the histogramming procedure is repeated for different total number of bins $N$, and the binning that gives the best PDE performance is chosen. This procedure can be performed with a relatively small number of iterations by searching through values of $N$ in the sequence: $N = 2^i$ with $i = 1, 2, 3, 4, \ldots, 16$ (a total number of bins $N = 2^{16} = 65536$ should be enough for most data, the value can be increased if required). The time taken to train and optimise the PDE is proportional to the number of data $n$, *i.e.* the time complexity of the algorithm is $O(n)$.

The optimal number of bins represents a balance between large $N$ that allows the histogram to model the form of the data though possibly with large statistical bin-to-bin fluctuations, and small $N$ that reduces (smooths) bin-to-bin fluctuations but at the possible expense of losing the underlying form of the data. At low statistics, or high dimensions, a good balance might not be achievable resulting in a poor PDE performance. The requirement that the data be bounded by upper and lower limits of the histogram can also give rise to performance problems when statistics are low.

### *2.2. The smoothed histogrammed PDE, SHPDE*

The simple histogramming method can be improved by, for example, applying a smoothing procedure to the feature function histograms, or applying variable bin width to increase binning in regions of high statistics and high gradients, and reduce binning in regions of low statistics. To demonstrate the improvements that can be gained, a simple Laplace histogram smoothing algorithm (averaging of neighbouring bins) is employed: a mask containing three consecutive bins (multiplied by each dimension) is scanned along the histogram, the value of the central bin of the mask is replaced with the mean of all values in the mask, in order to conserve the total number of entries in the histogram, the change in the value of the central bin is divided and subtracted equally from each member of the mask, all changes in the mask are reversed if any negative values result, this procedure is repeated five times.

The result of this procedure is the smoothing of bin-to-bin fluctuations; this enables a larger number of bins to be used in cases where statistics are limited.

### *2.3. The kernel PDE, KPDE*

Smoothing of the data in the kernel PDE method [3, 4] is obtained by replacing each data point with a multi-variate Gaussian kernel; the width of the kernel is optimised with respect to the PDE performance measure. The functions $f_{\mathrm{S}}(\vec{x})$ and $f_{\mathrm{B}}(\vec{x})$ are built by summing all Gaussian kernel functions in the training data. During optimisation the computation of functions $f_{\mathrm{S}}(\vec{x})$ and $f_{\mathrm{B}}(\vec{x})$ is repeated for every point in the test data; the time complexity of the KPDE algorithm is therefore $O(n^2)$. In its simplest form the KPDE is simple to implement, and has the advantage of not requiring a knowledge of the range of the data. Improvements to the KPDE performance can be gained using a variable (adaptive) kernel width [5] such that a wider kernel is used in regions of low statistics.

### *2.4. The histogrammed kernel PDE, HKPDE*

In an attempt to reduce the run-time of the KPDE method, the functions $f_{\mathrm{S}}(\vec{x})$ and $f_{\mathrm{B}}(\vec{x})$ may be histogrammed prior to the optimisation phase thereby reducing the time complexity of the algorithm to $O(n)$. However, this technique is only useful at high statistics, and inherits some of the disadvantages of histogramming: the range of the data needs to be defined, and some uncertainty in the PDE values is introduced. Additional optimisation of the binning is also required.

### *2.5. The range-search PDE, RSPDE*

Smoothing of the data in the Range-Search PDE method [6] is obtained by counting signal and background data in a volume around the point of interest. The volume shape may be a hyper-cube, or ellipsoid, with dimensions typically taken from the range, or the RMS, of the data, in each dimension. The scale of the volume is optimised with respect to PDE performance. Counting is performed in the training data while the "points of interest" are taken from the test data. As for the KPDE algorithm, the time complexity is $O(n^2)$. In its simplest form the RSPDE is simple to implement, and again, has the advantage of not requiring a knowledge of the range of the data. In Ref. [6] the run-time is reduced by storing events in two multi-dimensional binary trees, though this is achieved at the expense of a much more complicated algorithm.

## 3. PDE performances

The performance of the above five PDE methods, in their simple forms, are compared using data from a sample of 250,000 hadronic decays of the $Z$ boson, generated with the JETSET 7.4 Monte Carlo [7]. The events are passed through the ALEPH detector simulation and reconstruction program.

Reconstructed photons are selected and combined in pairs to create pion candidates for the decay $\pi^0 \to \gamma\gamma$. Details of these data in the context of $\pi^0$ selection can be found in the study of the reaction $\omega \to \pi^+\pi^-\pi^0$ [8].

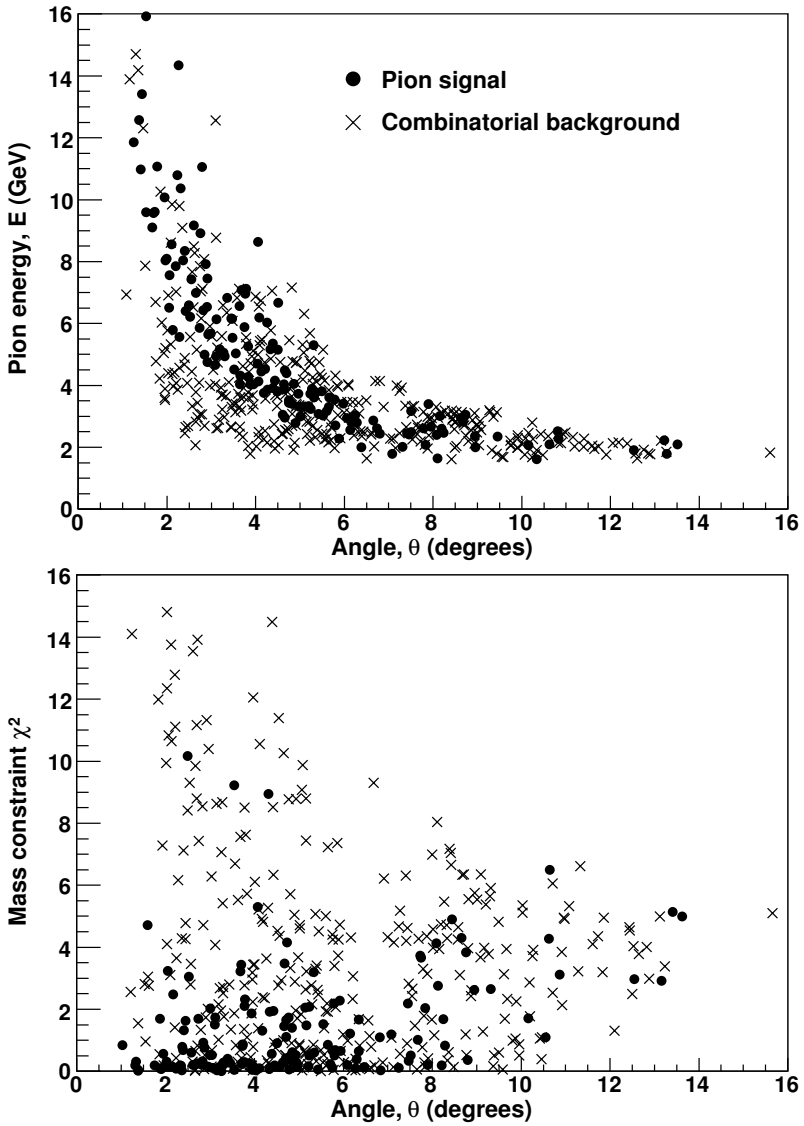


Fig. 1. Parameters used in the discrimination between signal and background; pion candidates are more likely to be signal if they have smaller values of $\theta$ and $\chi^2$, energy provides some discrimination information as the distribution of $\theta$ is energy- and mass-dependent.

In this comparison, the discrimination of pion signal from its background is optimised such that the product of the selected signal efficiency and purity is maximised. The pion candidates have already been selected from an invariant mass window giving an initial $\varepsilon \times \mathcal{P}$ value of 0.32. To improve further the discrimination between signal and background, three-dimensional PDEs are formed from the following discriminators: the $\chi^2$ from a constraint of the mass of a candidate to the nominal $\pi^0$ mass, the angle $\theta$ between the daughter photons in the lab frame, and the energy $E$ of the pion candidate. The distributions of these discriminators are shown in Fig. 1.

For this study the data is separated into forty data sets each containing 500 000 pion candidates. Training is performed on one data set, and, optimisation and selection is performed on a second independent data set. The procedure is repeated twenty times to obtain mean values and their standard deviations. To obtain results for both low and high statistics, the total number of pion candidates $n$ is varied from 50 to 500 000 according to $n = 5 \times 10^d$ where $d$ is a data scale that takes the values $1, 2, 3, 4$ and 5.

### *3.1. $\varepsilon \times \mathcal{P}$ performance*

The mean maximum $\varepsilon \times \mathcal{P}$ values for each PDE method are shown in Fig. 2 for different number of data. The figure also includes a comparison with results from a feed-forward artificial neural network (ANN); one hidden layer with seven nodes was found to be optimal for this data. The error bars represent the standard deviation of the twenty mean values. Results for the

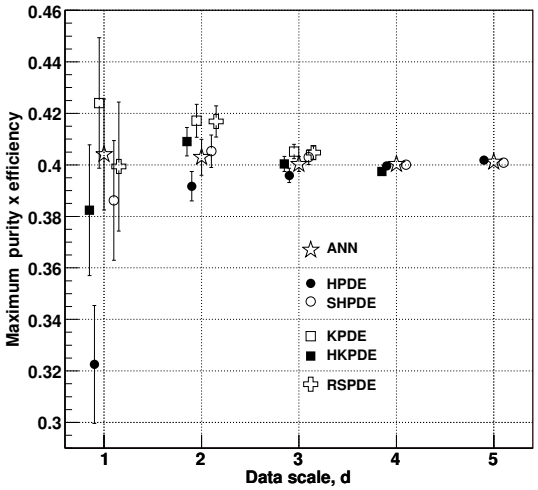


Fig. 2. Comparison of optimal $\varepsilon \times \mathcal{P}$ values for five PDE methods. The total number of data points scales as $n = 5 \times 10^d$. Results from an artificial neural network (ANN) are also shown.

KPDE, HKPDE, and RSPDE are not computed at high statistics due to the very large run-times necessary for these methods.

From Fig. 2, it can be seen that given enough statistics all methods perform equally well, raising the $\varepsilon \times \mathcal{P}$ value from an initial 0.32 to about 0.40, this represents a significant improvement in the selected signal significance. At lower statistics, results begin to fluctuate, this is apparent in the larger standard deviations, and scattering of the points.

The form of the results can be explained by considering two main effects. First, the fluctuations in the results, *i.e.* the standard deviations, appear to be only statistical in nature, *i.e.* they are independent of the method as each method experiences fluctuations of the same size for the same number of data. Second, some $\varepsilon \times \mathcal{P}$ values exceed 0.4, this does not represent improved performance[1], but instead over-training during optimisation using the test data set.

At the lowest statistics, $d = 1$ ($n = 50$), the three poorest performers are the methods involving histogramming, though only the simple histogram method has a performance that is significantly lower, this method shows no discrimination performance for this number of data. At first inspection, the highest performer appears to be the KPDE method, but assuming that its high $\varepsilon \times \mathcal{P}$ value of about 0.42 is due to over-training then we can conclude that the performance of this method when applied to real data is actually poorer and a resultant efficiency calculation may contain a significant systematic error.

The results from the neural network appear to exhibit, on average, neither over-training nor performance loss; the $\varepsilon \times \mathcal{P}$ values are consistent throughout the range of data scale. This is not surprising as neural networks are well known for their good discrimination behaviour. However, the "black box" nature of the neural network is not preferred by researchers due to the difficulty in assessing systematic errors.

### *3.2. Timing*

Run-time can be an important factor when choosing a PDE algorithm, especially in shared environments where, for example, jobs may be controlled by a queue manager, or on desktop computers that may have low CPU power. The average run-times[2] for each PDE method, including appropriate scales representing the number of repetitions employed in the optimisation procedure, are shown in Fig. 3 as a function of the data scale. For the Kernel and Range-Search methods the highest data scale values are predicted.

[1] We know, from the high statistics data, that the optimal performance is measured as $\varepsilon \times \mathcal{P} \approx 0.40$.

[2] The CPU times are recorded using an Intel 3.73 GHz Pentium 4 PC running the Linux operating system.

From this log–log plot, the $O(n)$ time complexity of the HPDE, SHPDE, HKPDE, and ANN methods, and the $O(n^2)$ time complexity of the KPDE and RSPDE can be seen. At largest data scale, $d = 5$, ($n = 500\,000$ pion candidates), the RSPDE and KPDE methods are predicted to take of the order of days, and weeks, respectively to train. By far the fastest methods are the simple and smoothed histogrammed PDEs that take only a few minutes to process 500 000 three-dimensional data points. However, at low statistics the run-times of any method are only a few seconds or less and so run-time for small data sets is not likely to be an issue for the researcher. Also, at low statistics, techniques to speed up $O(n^2)$ algorithms may not be effective, as is the case for HKPDE method where the procedure of histogramming introduces and extra optimisation parameter which in turns makes the algorithm relatively slower at small and medium data scales.

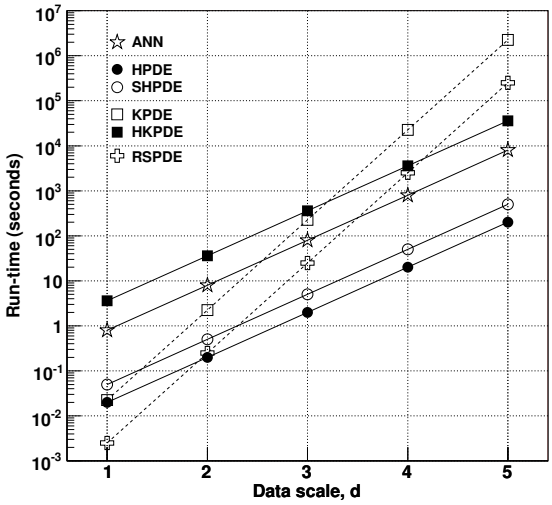


Fig. 3. Comparison of average run-times (Intel 3.73 GHz Pentium 4 PC) for five PDE methods, and the ANN. The total number of data points scales as $n = 5 \times 10^d$. The time complexity of the methods are seen to be either $O(n)$ or $O(n^2)$.

## 4. Discussion and conclusion

It is found, for a real high energy physics discrimination problem, in three-dimensions, that a number of PDE methods perform equally given enough statistics. At very high statistics, above 10 000 events, a simple histogramming method may be sufficient, and is of the order of magnitude faster than basic implementations of other PDE methods. Histogramming also carries the advantage that the results can be visualised directly.

As statistics reduce, PDE performance becomes unstable sometimes resulting in over-training and sometimes in under-training; in either case the discrimination performance in real data will be diminished, and efficiency calculations will contain errors. This fundamental problem applies equally to all the PDE methods studied in this paper down to 50 events. The exception is the HPDE method that fails to discriminate with this number of events.

At low statistics the histogram PDE method does not perform well unless additional smoothing is applied. Here, the Kernel and Range-Searching PDE methods show significant improvements. Although advanced implementations of some PDE methods are available, for example in the ROOT package [9], the simplicity and performance of the KPDE and RSPDE methods in their basic forms may be attractive to researchers who prefer, for various reasons, to implement routines themselves. These algorithms are also very fast if the number of data is less than 1 000. Also, as these two algorithms are conceptually quite different with respect to the way they form estimates of feature functions, they can be used as a systematic check on each other.

## REFERENCES

[1] A. Höcker *et al.*, TMVA toolkit for parallel multivariate data analysis, `http://tmva.sourceforge.net`.

[2] D. Buskulic *et al.*, *Nucl. Instrum. Methods* **A360**, 481 (1995).

[3] L. Holmstrom *et al.*, *Comput. Phys. Commun.* **88**, 195 (1995).

[4] K. Cranmer, *Comput. Phys. Commun.* **136**, 198. (2001).

[5] A. Askew *et al.*, Conference on Statistical Problems in Particle Physics, Astrophysics and Cosmology, PHYSTAT 2003, SLAC, Stanford, California, September 8–11, 2003, `http://www.slac.stanford.edu/econf/C030908/papers/TUHT002.pdf`.

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

[7] T. Sjöstrand, *Comput. Phys. Commun.* **67**, 74 (1994).

[8] A. Heister *et al.*, *Phys. Lett.* **B528**, 19 (2002).

[9] R. Brun, F. Rademakers, *Nucl. Instrum. Methods* **A389**, 81 (1997), `http://root.cern.ch`.