PROCEEDINGS OF SCIENCE

Optimization of tau identification in ATLAS experiment using multivariate tools

Marcin Wolter[1]
Institute of Nuclear Physics PAN,
ul. Radzikowskiego 152, 31-342 Kraków
E-mail: Marcin.Wolter@ifj.edu.pl

Andrzej Zemła
Institute of Nuclear Physics PAN,
ul. Radzikowskiego 152, 31-342 Kraków, Poland.
Jagiellonian University,
ul. Reymonta 4, 30-059 Kraków, Poland.
E-mail: a_zemla@o2.pl

Elementary particle physics experiments, searching for very rare processes, require the efficient analysis and selection algorithms able to separate signal from the overwhelming background. In the last years a number of learning machines have been developed. Three of such algorithms have been applied to identify τ leptons in the ATLAS experiment: Probability Density Estimator with Range Searches (PDE-RS), Neural Network and Support Vector Machine (SVM).

In the PDE-RS method the signal probability estimation is based on counting the signal and background events within a multidimensional hypercube surrounding the vector under classification. In the SVM approach to signal and background separation a separating hyperplane defined by a limited number of vectors from the training sample (support vectors) is created. The extension to a non-linear separation is performed by mapping the input vectors into a high dimensional space, in which data can be linearly separated. The use of kernel functions allows to perform computations in a high dimension feature space without explicitly knowing a mapping function. We have implemented an SVM algorithm and integrated it with the CERN TMVA/ROOT package.

All three methods have similar performance, which is significantly better than the baseline cut analysis. This might indicate, that the achieved background rejection is close to the maximal achievable performance.

XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research
Amsterdam, the Netherlands
23-27 April, 2007

[1] Speaker

PoS(ACAT)033

1.Introduction

Tau leptons play an important role in the physics to be observed at LHC (Large Hadron Collider at CERN, Geneva). They enter in electroweak measurements, studies of the top quark and as a signature in searches for new phenomena such as Higgs, Supersymmetry and Extra Dimensions. However, the tau reconstruction and identification is not an easy task. The QCD multi jet events dominating the backgrounds have much larger cross section, therefore the efficient selection is needed.

In this contribution, we describe multivariate methods used for τ-jet identification in the ATLAS experiment: a Neural Network (NN), Probability Density Estimator with Range Searches (PDE_RS) and Support Vector Machine (SVM). The analysis is performed on the simulated ATLAS data from the central CSC production, the channels $Z->\tau\tau$, $W->\tau\nu$ (with hadronic τ decays) are used as signal events and the events with QCD jets as background. In total about 30 k signal events and 1226 k background events are available.

2.Physics processes with τ leptons at ATLAS detector

The ATLAS experiment (A Toroidal LHC Apparatus) measures 22 m high, 44 m long and weights 7000 tons. The ATLAS detector is composed of a tracker, a calorimeter system (electromagnetic and hadronic) and of a large muon spectrometer. More details about the detector can be found elsewhere [1].

Detection of many processes depends on the efficient reconstruction of hadronic τ: light Standard Model (SM) Higgs produced in Vector Boson Fusion (VBF) $qqH \rightarrow qq\tau\tau$, charged SUSY Higgs production $H \rightarrow \tau\nu$, neutral SUSY Higgs $H/A \rightarrow \tau\tau$ at large $tan\beta$, SUSY signatures with τ in the final state as well as Extra Dimensions. The well known processes $Z \rightarrow \tau\tau$ and $W \rightarrow \tau\nu$ will be also used to calibrate the calorimeters.

Tau leptons decay to hadrons in 64.8% of the cases and to electron or muon the rest of the time. In 77% of hadronic τ decays only one charged track is produced: $\tau \rightarrow \nu\tau + \pi^{\pm} + n\pi^0$ and in 23% there are 3 charged tracks: $\tau \rightarrow \nu\tau + 3\pi^{\pm} + n\pi^0$. The τ candidates with a single identified charged track are called 1-prong, with three tracks – 3-prong. $3\pi^{\pm}$ candidates with one track missing or candidates with a single $\pi^{\pm}$ and one fake track are referred as 2-prong.

A τ lepton decaying hadronically generates a narrow τ jet. The background misidentified as a τ is mainly a QCD multi jet event, but also electrons that shower late or with strong Bremsstrahlung, or muons interacting in the calorimeter are contributing. A τ-jet can be identified through the presence of a well collimated calorimeter cluster with a small number of associated charged tracks . In ATLAS two methods of τ reconstruction and identification are used: TauRec, which is based on calorimeter clusters and Tau1P3P starting from a good leading hadronic track and creating a τ -jet candidate based on tracks and also on an additional calorimeter information. All of the multivariate identification methods presented here refer to the Tau1P3P algorithm [2].

For Tau1P3P algorithm several discriminating variables to separate real τ jets from background are defined:

1) tracking part:
- N_{track}: number of associated tracks in an isolation cone,
- W_T^{Tracks}: weighted width of track with respected to tau axis (2 or 3 tracks only),
- M_{inv}^{Tracks}: invariant mass of tracks (2 or 3 tracks only),

2) calorimetry part:
- N_{strip}: number of strips fired,
- W_{Strip}: energy weighted width in strips,
- ΔE_T: fraction of energy in half core cone to energy in the entire core cone,
- R_{EM}: energy weighted radius in EM part,

3) mixture of calorimetry and tracking information:
- ΔE_{iso}: fraction of energy in isolation cone to energy in core cone,
- E_{inv}^{Calo}: invariant mass calculated from energy-flow,
- E_T/p_T: ratio of energy in HAD part to energy of tracks,
- E_T^{vis}: visible transverse energy.

The variables are not independent and no single variable provides a really good signal and background separation (see Fig. 1 for three prong data), which emphasizes a need for efficient selection algorithms. Beside the standard cut analysis three multivariate algorithms are applied to select τ candidates: Probability Density Estimator with Range Searches (PDE_RS), Neural Network (NN) and Support Vector Machine (SVM). For testing these techniques the data are splited into two parts: one is used for training and the other one for validation.

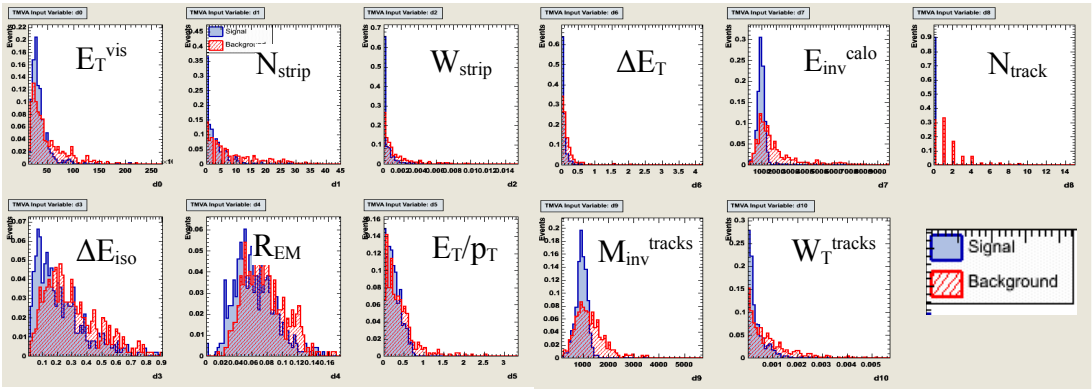

Fig. 1: Discriminating variables for 3-prong τ candidates.

3.PDE-RS method

The probability density estimation technique (PDE) was used for optimization of the Tau1P3P algorithm [3]. Method and implementation is based on publication [4]. As most of the standard multivariate algorithms the technique combines the input observables into a single one, called a discriminant, on which a cut separating signal from background is applied. Calculation of the discriminant is based on sampling the signal and background densities in a multi-dimensional phase space built out of discriminating variables. Taking number of signal events n_S and number of background events n_B in a small volume $V(\boldsymbol{x})$ around point $\boldsymbol{x}$ in the multidimensional space, a discriminant defined as:

$$D(x)=\frac{n_S}{n_S+c\,n_B} \qquad (1)$$

is a good approximation of probability that given candidate is a signal τ. Parameter $c=N_S/N_B$ is the ratio of the total number of signal events N_S to the number of generated background events N_B. The event counting is done using multi-dimensional binary trees. As stated in [4], this method is supposed to give significantly better results than the cut analysis and comparable to other multivariate techniques.

4.Neural Network

Neural network is a non-linear discriminating method (we refer reader to [5] for detailed description of the neural network techniques). The Stuttgart Neural Network Simulator [6] is used for the identification of τ candidates.

In the feedforward network, as used for the τ identification, the information propagates from input to output without any loops. To each neuron *j* in the hidden layer *n* inputs x_k and one output variable (the answer of the neuron) z_j are associated. For the first hidden layer the inputs are the discriminating variables, for next layers the inputs are the outputs of the preceding layer.

The architecture of the network is optimized to give the proper classification of signal and background and to avoid over-fitting at the same time. The neural network used for τ identification is built with 9 (1-prong candidates) or 11 (2 or 3-prong) input nodes and two layers of hidden nodes, each with 14 nodes (Fig. 2)

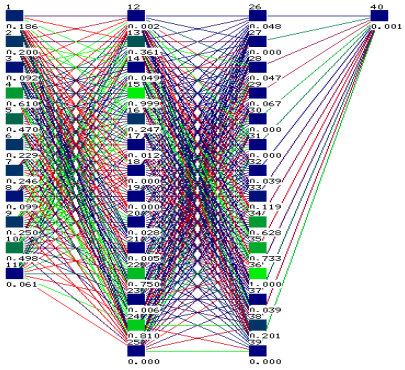

Fig. 2: Schematic view of the neural network.

The neuron sums up the input variables y_k, weighted by a factor w_{jk}, plus a threshold Θ_j. This defines the signal Z_j:

$$Z_j=\sum_{k=1}^{N} w_{jk}\,y_k+\Theta_j \qquad (2)$$

The output of the neuron is a function of *Z*: $z_j = a(Z_j)$, where $a(Z_j)$ is called an activation function, and is chosen to be of the form:

$$a(x)=\frac{1}{1+e^{-Z_j}} \quad \text{(logistic function).} \qquad (3)$$

The training phase of the neural network consists in determining the weighting factors w_{jk} and the thresholds Θ_j. This is done by minimizing the error function defined as:

$$E=\frac{1}{2}\sum_{i=1}^{n}(X_i-t_i)^2 \quad , \tag{4}$$

where t_i is the expected output (0 for background, 1 for signal), X_i the actual value returned by the network and n is a number of events used for training.

In the process of training patterns are presented to the network which generates an output. The output is compared with the desired output from the training sample and the cost function is calculated. Than the weights in nodes is adjusted to decrease the value of the cost function. The errors are propagated backward using the current weights (the backpropagation algorithm [7,8]).

5.Support Vector Machine

In the early 1960s the linear support vector method was developed to construct separating hyperplanes for pattern recognition problems [9, 10]. The main idea of the SVM approach is to build a separating hyperplane which maximizes the margin. The position of the hyperplane is defined by the subset of all training vectors called support vectors. The extension into non-linear SVM [11, 12] is performed by mapping input vectors into a high dimensional feature space in which data can be separated by a linear procedure using the optimal separating hyperplane.

5.1Linear Support Vector Machine

A detailed description of SVM formalism can be found for example in [13], here only a brief introduction is given. Consider a simple two-class classifier with oriented hyperplanes. If the training data is linearly separable, then such a set of $(\vec{w}, b)$ pairs can be found that the following constraints are satisfied:

$$\forall_i y_i(\vec{x}_i\cdot\vec{w}+b)-1\geq 0 \tag{5}$$

where x_i are the input vectors, y_i the desired outputs $(y_i=\pm 1)$ and $(\vec{w}, b)$ define a hyperplane. Intuitively, the classifier with the largest margin will give better generalization. Hence, in order to maximize the margin, one needs to minimize the cost function W:

$$W=(1/2)|\vec{w}|^2 \tag{6}$$

with the constraints from Eqn. 5. At this point it would be beneficial to consider the significance of different input vectors $\vec{x}_i$. The training data points laying on the margins, which are called the support vectors (SV), are the data that contribute to defining the decision boundary (Fig. 3). If the other data are removed and the classifier is retrained on the remaining data, the training will result in the same decision boundary. To solve this constrained quadratic optimization problem, we first reformulate it in terms of a Lagrangian:

$$L(\vec{w}, b, \vec{\alpha})=1/2|\vec{w}|^2-\sum_i \alpha_i(y_i((\vec{x}_i\cdot\vec{w})+b)-1) \tag{7}$$

where $\alpha_i\geq 0$ and the condition from Eqn. 5 must be fulfilled.

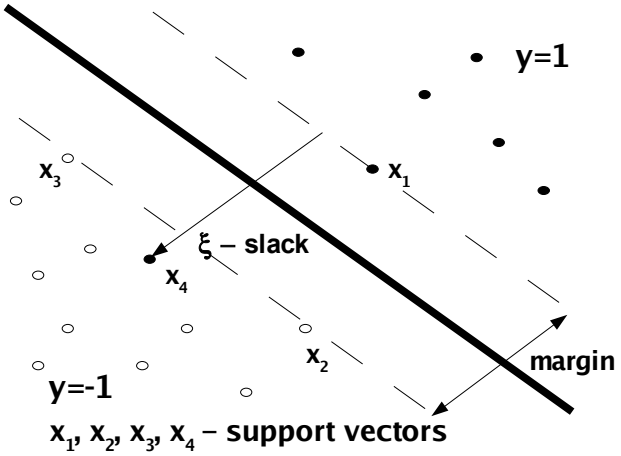

Fig. 3: Hyperplane classifier in two dimensions. Points x_1, x_2 and x_3 define the margin, i.e. they are the support vectors.

Lagrangian L should be minimized with respect to $|\vec{w}|$ and b and maximized with respect to $\vec{\alpha}$. The optimization problem becomes the one of finding the $\vec{\alpha}$ which maximize:

$$L(\vec{\alpha})=\sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j) \tag{8}$$

Both the optimization problem and the final decision function depend only on dot products between input vectors, which is crucial for the generalization to the nonlinear case. For non-separable data the correct classification constraints in Eqn. 5 are modified by adding a slack variable ξ_i to it ($\xi_i=0$ if the vector is properly classified, otherwise ξ_i is a distance to the decision hyperplane).

$\forall_i \; y_i(\vec{x}\cdot\vec{w}+b)-1+\xi_i \geq 0$ The training algorithm needs to minimize the cost function, i.e. a trade-off between maximum margin and classification error:

$$W=(1/2)|\vec{w}|^2+C\sum_i \xi_i \tag{9}$$

The selection of C parameter defines how much a misclassification increases the cost.

5.2Nonlinear Support Vector Machine

The formulation of SVM presented above can be further extended to build a nonlinear SVM, which can classify nonlinear data. Consider a function Φ which maps the training data from $\Re^n$ to some higher dimensional space $\Re^N$. In this high dimensional space, the data can be linearly separable, hence the linear SVM formulation above can be applied to these data.

In the SVM formulation data appear only in the form of dot products $(\vec{x}_i \cdot \vec{x}_j)$ (see Eqn. 8). The dot product $\Phi(\vec{x}_i)\cdot\Phi(\vec{x}_j)$ appears in the high dimensional feature space, where it is replaced by a kernel function:

$$K(\vec{x}_i, \vec{x}_j)=\Phi(\vec{x}_i)\cdot\Phi(\vec{x}_j) \tag{10}$$

By computing the dot product using a kernel function, one avoids the mapping $\Phi(\vec{x})$. This is desirable, because $\Phi(\vec{x})$ can be tricky or impossible to compute. Using a kernel

function, one does not need to know explicitly what the mapping is. The most frequently used kernel functions are the Gaussian, polynomial and linear.

The optimization problem becomes well defined convex quadratic programming problem, which assures us that there exists a global minimum. This is an advantage of SVMs compared to neural networks, which may fall into one of the local minima.

6.Implementation of SVM in the ROOT/TMVA framework

We have implemented the SVM algorithm in the CERN ROOT/TMVA framework [14, 15]. This implementation uses a Sequential Minimal Optimization (SMO) [16] to solve the quadratic problem. Further modifications proposed by Keerthi [17] speed up the algorithm. To speed up the minimization most of the algorithms divide a set of vectors into smaller subsets. The SMO method puts the subset selection to the extreme by selecting subsets of two vectors.

Let us give a brief description of the SMO algorithm, the details can be found in [16] and [17]. The pairs of vectors are chosen, using heuristic rules, to make the largest possible minimization step. Because the working set is of the size of two it is straightforward to write down the analytic solution. The minimization procedure is repeated recursively until the minimum is found. The SMO algorithm has proved to be significantly faster then the other methods like chunking [18] or SVMlight [19], and has become the most common minimization method used in the SVM implementations.

The implemented by us SVM algorithm performs the classification tasks using linear, polynomial or Gaussian kernel function. The Gaussian kernel allows to apply any, even very complicated, discriminant shape in the input space.

7.Application to the identification of τ particle in the ATLAS experiment

The Neural Network, PDE-RS and the SVM implemented within TMVA package have been used for identification of τ leptons in ATLAS experiment. The results are shown in Fig. 4. Signal efficiency is defined as a ratio of accepted and all signal events $\epsilon_s = N^{sig}_{accep} / N^{sig}_{all}$ and background rejection as a ratio of rejected and all background events $R = 1 - \epsilon_b = N^{bkg}_{rej} / N^{bkg}_{all}$.

All three multivariate algorithms perform significantly better than the basic cut analysis. The best performance is achieved with a Neural Network, however all three methods have very similar performance. This might indicate, that the achieved background rejection is close to the Bayesian limit.

For the SVM classification the Gaussian kernel function was chosen. The performance of the SVM with radial kernel depends on two parameters: the width of the Gaussian kernel and the cost parameter *C*. A grid search in the space of these two parameters was performed to maximize the background rejection. It must be pointed out, that SVM was trained on a small subsample of about 10% of the data available. This shows a good generalization performance of the SVM technique.

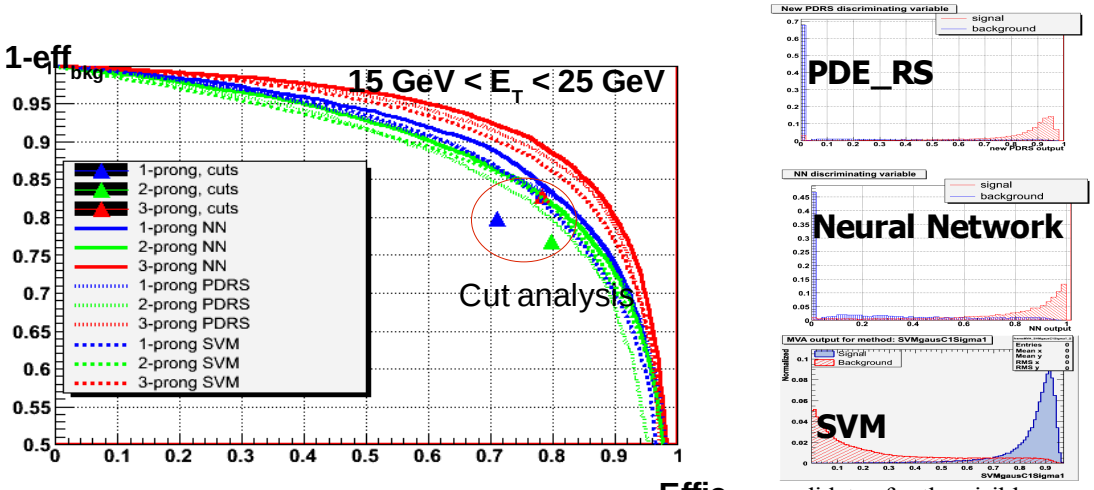

Fig. 4. The ROC curve ($1-\varepsilon_{bkg}$ vs. ε_{sig}, left plot) for 1, 2 and 3-prong candidates for the visible transverse energy interval 15 GeV < E_t^{vis}. The multivariate methods perform significantly better than the cut analysis. In the right plot the distributions of discriminant for all three methods for 3-prong candidates are shown.

8.Summary

Identification of τ candidates by the Tau1p3p algorithm is significantly improved by using multivariate analysis tools. All of the applied classification methods are performing well giving similar results. The analysis based on cuts is robust, transparent for users and doesn't require CPU consuming training. Neural network is in our case giving the best performance. It allows, after a costly training, a very fast classification while the trained network is converted to the C code. PDE-RS is robust and transparent for users, but large samples of reference candidates are needed, also the classification is slower than for other methods. The SVM algorithm wasn't, up to now, commonly used in HEP. We have shown that Support Vector Machine can be successfully used to analyze high energy physics data. The implementation described above is included in the ROOT package, therefore it is easily available to the entire particle physics community. This implementation extends the range of multivariate analysis tools available within the ROOT framework.

Acknowledgments

Authors would like to thank the entire ATLAS Tau WG for their help and support. Special thanks to D. Cavalli and E. Richter-Wąs for carefully reading this contribution and useful suggestions. Thanks also to A. Höcker for introducing us to the TMVA package and for his help while dealing with TMVA problems.
The work was supported in part by Polish Ministry of Science And Higher Education grants 154/6.PRUE/2007 and PBS NR 132/CER/2006/03 .

References

[1] ATLAS collaboration, *Detector and Physics Performance Technical Design Report*, Volumes 1 and 2, CERN/LHCC/99-14, ATLAS TDR 14, (1999).

[2] E. Richter-Was, T. Szymocha, *Hadronic τ identification with track based approach: the z→ττ,w→τν and di-jet events from dc1 samples.* ATLAS Note ATL-PHYS-PUB-2005-005.

[3] L. Janyst, E. Richter-Was, *Hadronic tau identification with track based approach : optimisation with multi-variate method*, ATL-COM-PHYS-2005-028; Geneva : CERN, 03 Jun 2005 .

[4] T. Carli and B. Koblitz. *A multi-variate discrimination technique based on range-searching.* Nucl. Instrum. Meth. A, (501):576, 2003.

[5] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford: Oxford University Press, Oxford, 1995.

[6] A. Zell and et al. *http://www-ra.informatik.uni-tuebingen.de/SNNS/.*

[7] P. J. Werbos. *Beyond regression: new tools for prediction and analysis in the behavioural sciences*. Ph.D. thesis, Harvard University, Boston MA, 1974.

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation.* Vol. 1 of Computational models of cognition and perception, Cambridge, MA: MIT Press, chap. 8:pp.319–362, 1986.

[9] V. Vapnik, and A. Chervonenkis, *A note on one class of perceptrons*. Automation and Remote Control, (25), 1964.

[10] V. Vapnik and A. Lerner *Pattern recognition using generalized portrait method.* Automation and Remote Control, (24), 1963.

[11] B. Boser, I. Guyon, and V. Vapnik, *A training algorithm for optimal margin classifiers*. Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, ACM Press, pages 144–152, 1992.

[12] C. Cortes and V. Vapnik *Support vector networks.* Machine Learning, (20):273–297, 1995.

[13] C. Burges *A tutorial on support vector machines for pattern recognition.* Data Mining and Knowledge Discovery, 2(2):1–47, 1998.

[14] R. Brun, and F. Rademakers, *ROOT - An Object Oriented Data Analysis Framework*, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/

[15] A. Höcker, H. Voss, K. Voss, J. Stelzer, *TMVA (Toolkit for MultiVariate Analysis),* http://tmva.sourceforge.net

[16] J. Platt (1999) *Fast training of support vector machines using sequential minimal optimization.* In B. Scholkopf, C. Burges & A. Smola, eds, Advances in Kernel Methods - Support Vector Learning. MIT Press.

[17] S. Keerthi, S. Shevade, C. Bhattacharyya and K. Murthy, I*mprovements to Platt's SMO algorithm for SVM classifier design.* Tech Report, Dept. of CSA, Bangalore, India, 1999.

[18] V. Vapnik, *Estimation of Dependences Based on Empirical Data*, Springer-Verlag, (1982).

[19] T. Joachims, *Making large-scale support vector machine learning practical*, in B. Scholkopf, C. Burges, A. Smola. Advances in Kernel Methods: Support Vector Machines, MIT Press, Cambridge, MA, December 1998.