Journal of Physics: Conference Series

Testing TMVA software in b-tagging for the search of MSSM Higgs bosons at the LHC

To cite this article: T Lampén *et al* 2008 *J. Phys.: Conf. Ser.* **119** 032028

Testing TMVA software in b-tagging for the search of MSSM Higgs bosons at the LHC

T Lampén, F Garcia, A Heikkinen, P Kaitaniemi, V Karimäki, M J Kortelainen, S Lehti, T Lindén, and L Wendland

Helsinki Institute of Physics, P.O.Box 64, FIN-00014 University of Helsinki, Finland

E-mail: Tapio.Lampen@cern.ch

Abstract. We test the usage of a Toolkit for Multivariate Data Analysis (TMVA) in b tagging. Tagging b jets associated with heavy neutral MSSM Higgs bosons at the LHC can be used to extract the Higgs bosons from the Drell-Yan background, for which the associated jets are mainly light quark and gluon jets. Achievable b tagging efficiency is studied with more than ten MVA classifiers at 1% mistagging rate. Most classifiers were found to perform better than the simple track counting algorithm.

1. Introduction

At the LHC, the dominant Higgs boson production mechanism in the Minimal Supersymmetric Standard Model (MSSM) at large values of tanβ is the heavy neutral Higgs boson production in association with two b quarks. These associated b jets can be used to extract the Higgs events from the Drell-Yan Z/γ^* background [1], for which the associated jets are mostly light quark and gluon jets.

Due to the relatively long lifetime of the B-hadrons, a jet can be identified as a b jet using lifetime based tagging algorithms [2, 3], which rely on displaced secondary vertices and the track impact parameter, *ip*. The impact parameter is the closest approach of the track trajectory to the primary vertex. In a b jet tracks originate typically from a displaced secondary vertex, as shown in Figs. 1 and 2, while the tracks in light quark (uds) and gluon jets originate from the primary vertex. One of the most simple b-tagging algorithms is counting tracks with high enough impact parameter significance within the jet cone. The impact parameter significance is defined as the impact parameter value divided by its estimated error.

In this study we test the usage of the Toolkit for Multivariate Data Analysis (TMVA) software in the b tagging problem. The b tagging efficiency is estimated and optimized for 1% mistagging rate. The background discriminating power is estimated for various methods available in TMVA, such as rectangular cut optimization, projective and multi-dimensional likelihood estimators, linear discriminant analysis with Fisher discriminants, artificial neural networks and boosted/bagged decision trees. The effect of event preselection and variable transformations applied on data is also investigated. The efficiency of the simple track counting algorithm [1] using three tracks with best impact parameter significance is given as comparison.

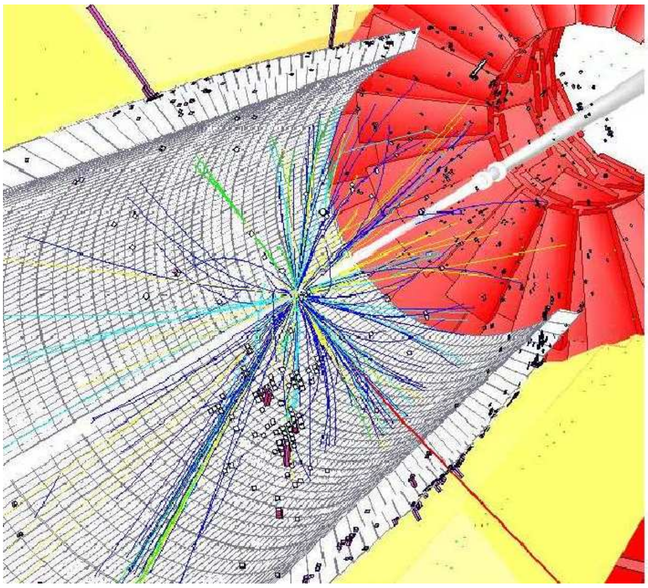

Figure 1. Geant4 based simulation of a SUSY event in the CMS detector containing missing transverse energy, jets and several leptons in the barrel detector. (Picture: IguanaCMS.)

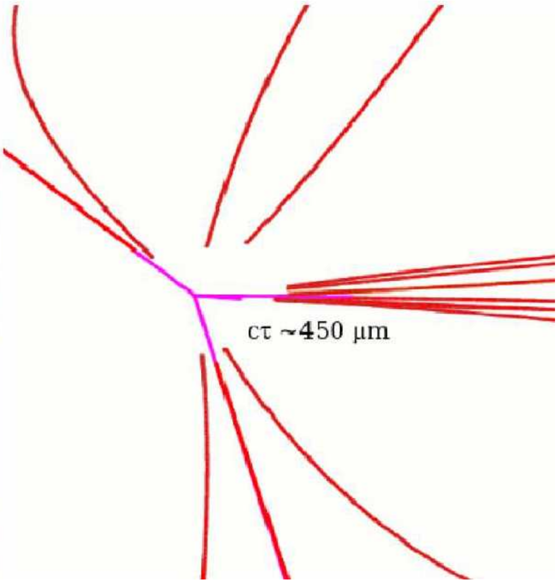

Figure 2. A displaced secondary vertex in a $b\bar{b}H$ event with $H \rightarrow \tau\tau$ in the CMS detector. The second b jet is not reconstructed due to a low jet energy and track multiplicity.

2. Key features of TMVA

The Toolkit for Multivariate Analysis for ROOT (TMVA) [4, 5, 6] is a machine learning environment for sophisticated multivariate classifiers. It enables the use of various multivariate classifiers and their evaluation in a ROOT environment. The key features are:

- individual pre-processing of the data for each classifier as a linear transformation into a non-correlated variable space or projection upon their principal components
- providing the same training and test data for selected classifiers within the same execution job allowing an easy comparison between classifiers
- for standalone use of the trained classifiers, code for lightweight C++ response classes independent of ROOT and TMVA is generated
- visualization scripts with a graphical user interface providing e.g. signal efficiency vs. background rejection curves (ROC curves).

General characteristics of the classifiers are presented in Table 1. Their details are described in Refs. [5, 6]. As an example of the visualisation capabilities of TMVA, Fig. 3 shows examples of a decision tree of the BDT classifier and convergence of the neural network classifier.

3. Data description

3.1. Event generation and simulation

In this study we use b jets from top quark decays as signal. With real data the b tagging algorithms and efficiency will also be studied with $t\bar{t}$ events, which are copiously produced and easily identifiable. The signal and background jets are generated with TopREX [8] ($t\bar{t}$ events) and with PYTHIA [9] (Z/γ^* events), and selected using the available generator level Monte-Carlo

Table 1. Main characteristics of different classifiers [7].

Method	Pros	Cons
Cuts	Easy to understand	Possibly inefficient
Likelihood methods	Fast to train and evaluate	Non-linear correlations treated badly
HMatrix, Fisher	Very fast and transparent	fail if PDFs have same mean, and if non-linear correlations
PDERS, kNN	Handles well complex class boundaries	Impractical with more than 10 variables
ANN	Very good with non-linear correlations	Black box, needs tuning
BDT	Very good out-of-the-box performance	Needs tuning to avoid overtraining
RuleFit	Like BDT but simpler, fast evaluation	Often needs some tuning
SVM	Good with non-linear problems, insensitive to overtraining	Not transparent
FDA	Very good classification if boundary is known	Classification boundary function needed

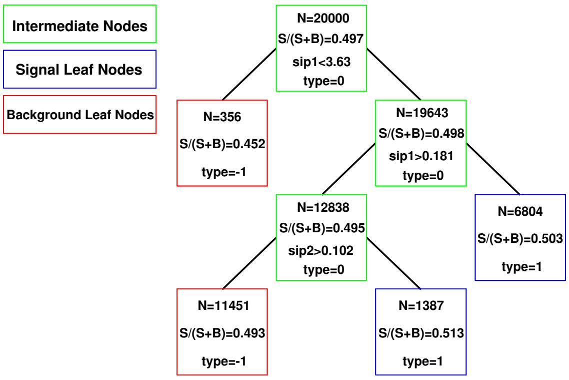
Intermediate Nodes
Signal Leaf Nodes
Background Leaf Nodes
N=20000 S/(S+B)=0.497 sip1<3.63 type=0
N=356 S/(S+B)=0.452 type=-1
N=19643 S/(S+B)=0.498 sip1>0.181 type=0
N=12838 S/(S+B)=0.495 sip2>0.102 type=0
N=6804 S/(S+B)=0.503 type=1
N=11451 S/(S+B)=0.493 type=-1
N=1387 S/(S+B)=0.513 type=1

Figure 3. Example of a decision tree for the BDT classifier. Type of the node, number of events, purity and the cut are displayed for each node.

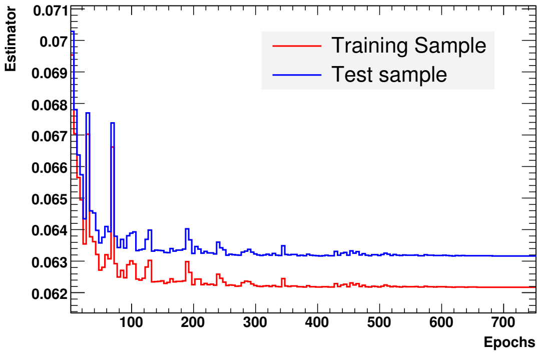

Figure 4. Convergence of the neural net estimator as a function of iteration cycles (epochs).

truth. The signal consists of 162k b jets from $t\bar{t}$ events and 588k light quark and gluon jets from Z/γ^* events. The event reconstruction is based on official CMS digitized datasets [1] with pile-up included. The pile-up consists of on the average 3.4 minimum bias events superimposed per event crossing for luminosity $2 \times 10^{33}\text{cm}^{-2}\text{s}^{-1}$. The detector simulation has been done with full GEANT4 [10] simulation. The CMS detector is simulated with complete ideal detector, no staging and no misalignment of the detector elements is assumed. The jets, tracks and vertices are reconstructed using standard methods available in the CMS reconstruction software. A more detailed description of the event simulation can be found in Ref. [11].

3.2. Test scenarios and variables

We divide our analysis into two test scenarios. In the first scenario, we approach the b tagging problem by feeding the different MVA classifiers the same set of variables, which are used in the simple track counting algorithm. Using this fixed set of variables and the same set of events for training and testing the classifiers, we compare various classifiers with each other as well as with results from previous studies with neural networks [12, 13, 14]. The input variables used in this scenario are:

- transverse impact parameter significance, σ_{ip}, of the track with highest σ_{ip}
- σ_{ip} of the track with the second best σ_{ip}

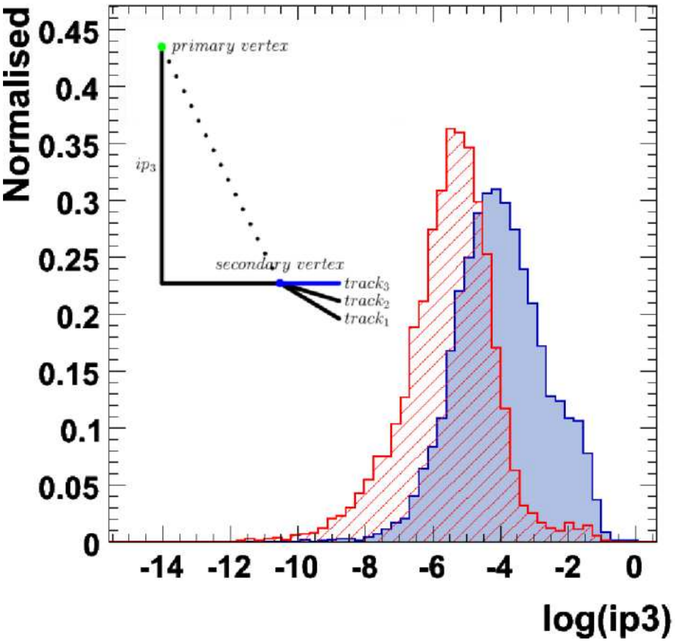

Figure 5. Definition of impact parameter ip and distribution for ip_3.

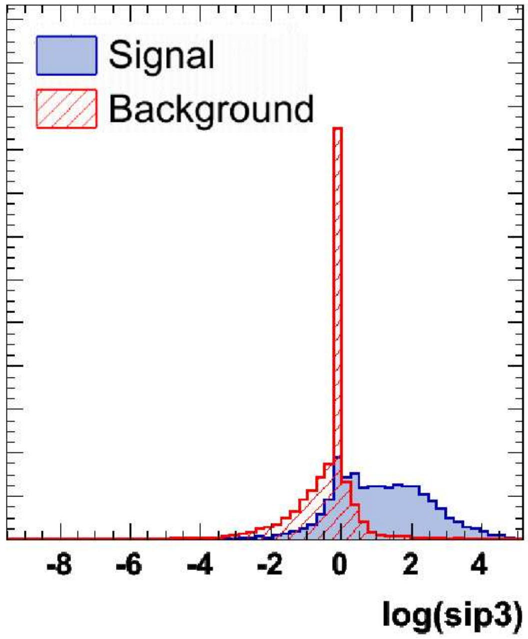

Figure 6. Transverse impact parameter significance for highest $\sigma_{ip,3}$.

- σ_{ip} of the track with the third best σ_{ip}.

Example of ip and σ_{ip} distributions are shown for the signal and background jets in Figs. 5 and 6. Decorrelation and principal component analysis (PCA) were tested as input variable preprocessing options.

In the second scenario, a larger set of input variables is used, from which one can freely choose the optimal combination of variables in order to maximize the separation of the signal and background events at the operating point. These variables include transverse momenta p_T, transverse impact parameter ip and the σ_{ip} for the three tracks with highest σ_{ip}, the number of tracks n_{tracks} with $p_T > 0.5$ GeV in a cone of $\Delta R = 0.7$ around the jet axis, the jet E_T, the number of secondary vertices n_{vtx} and the best vertex significance σ_{vtx}. Preprocessing of the variables (e.g. logarithm) and combinations of variables were studied and used in addition to decorrelation and PCA in order to optimize the performance of each classifier.

4. Computing environment

The computations have been performed in Helsinki using a 64-bit 1.8/2.2 GHz AMD Opteron M-grid cluster called *ametisti*, which has 260 CPUs in 130 computational nodes with 2/4 GB RAM. Ametisti has a dedicated Gb/s network for communication and another dedicated Gb/s network for NFS-traffic to enhance the performance of the shared NFS disk system. In addition to this there is also a fast Ethernet network used for remote management.

NPACI Rocks Cluster Distribution is a cluster management software for scientific computation based on Red Hat Linux. It supports cluster installation, configuration, monitoring and maintenance [15]. We use Rocks v. 4.1 on ametisti and the Sun Grid Engine (SGE) batch queue system, which supports advanced features like back filling, fair share usage and array jobs. Root version 5.14/00d and TMVA 3.8.5 has been used in this work.

5. Results

The b tagging efficiency for each tested classifier is shown in Table 2 for the first scenario with transverse impact parameter significances of the three best tracks as input. 10k events of both signal and backround data were used for training the classifiers, while the remaining 152k signal events and 578k background events were used for testing.

No input variable preprocessing (other than decorrelation for BDT) was used as they did not improve the results, probably because the key quantities are exponential like and far from Gaussian.

The signal separation power of H-Matrix and ANN classifiers is shown in Figs. 7 and 9. The corresponding efficiency and purity graphs are shown in Figs. 8 and 10, respectively. The errors of Table 2 were estimated by running the program for each classifier with different random number generator seed values *SplitSeed*, which affects the event sampling. The run was repeated 20 times, out of which one run did not succeed. The mean values of the signal efficiencies of 19 successful runs are taken as the reported efficiencies with the standard deviation representing the errors. A 5% (systematic) uncertainty of the efficiency of the simple track counting algorithm was taken according to Ref.[1].

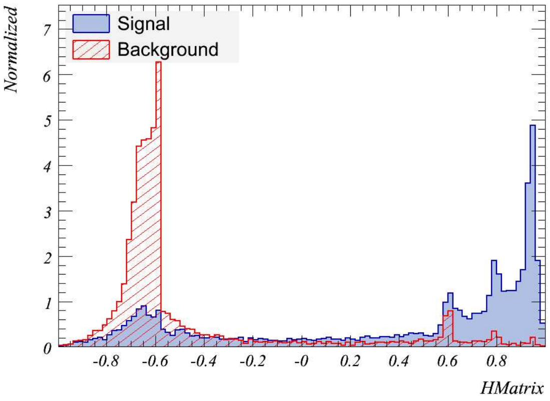

Figure 7. The signal separating power of the H-Matrix classifier for test data.

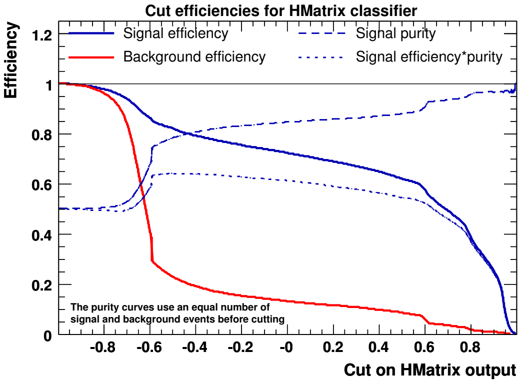

Figure 8. H-Matrix classifier efficiencies for different cut values.

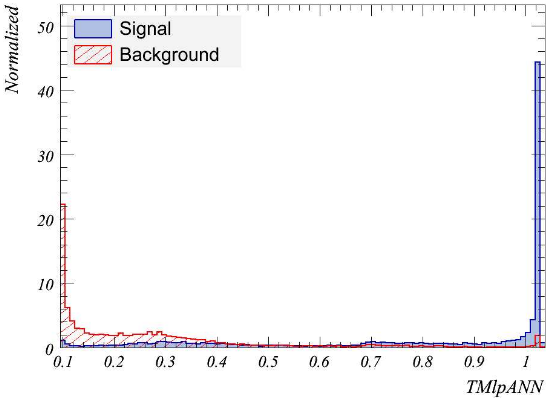

Figure 9. The signal separating power of the TMlpANN classifier for test data.

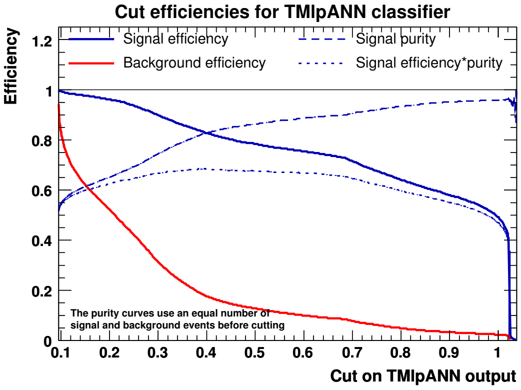

Figure 10. TMlpANN classifier efficiencies for different cut values.

Table 2. Signal efficiencies of the studied classifiers at 1 and 10% mistagging rates using track impact parameter significances as input. Result for the simple Track counting algorithm is shown for comparison. CPU times measured with the *time* command for training and evaluation are also given. Errors are calculated as explained in the text, except for RuleFit, for which the value computed by TMVA is given.

		train	train		
	1%	efficiency	10%	efficiency	CPU
	bkg eff.	at 1%	bkg eff.	at 10 %	time [s]
Cuts GA	26.9±0.6	27.2±2.0	74.7±0.4	74.7±0.6	484
PDEKDE	22.1±1.1	26.4±1.7	70.0±0.3	70.2±0.3	123
PDERS	25.0±0.7	52.2±0.8	73.9±0.1	76.8±0.4	14848
HMatrix	28.2±0.4	28.5±2.1	63.8±0.9	63.8±1.2	32
Fisher	12.5±0.4	12.7±0.9	62.1±0.7	62.2±1.0	9
FDA GAMT	20.6±5.4	20.6±4.8	72.5±1.4	72.6±1.4	125
TMlpANN	26.6±0.3	27.7±3.2	74.9±0.1	75.0±0.4	158
BDT	28.3±1.3	36.9±1.4	70.2±1.2	72.3±1.2	241
RuleFitJF	31.8±0.4	26.3	74.9±0.4	75.2	72
SVM	27.1±0.2	27.5±1.9	71.8±0.8	71.9±1.0	3542
Track counting b-tagging	26.6±1.3				

The ROC curves for the first scenario is shown in Figs. 11 and 12. The latter shows in more detail the area close to the 1% mistagging rate. Since the ROC curves are almost horizontal in that area, a small variation in the background efficiency may result in a large variation in the signal efficiency.

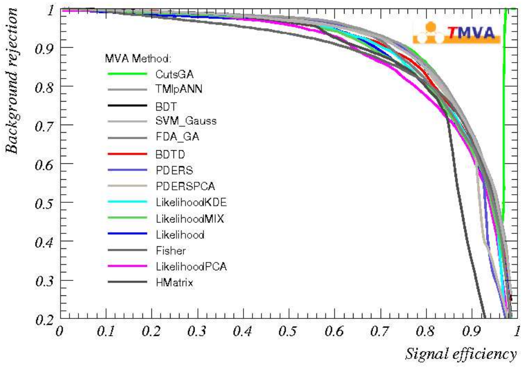

Figure 11. ROC curves for the studied classifiers.

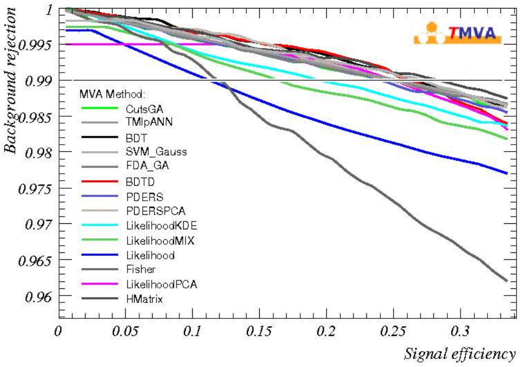

Figure 12. ROC curves near the 1% background efficiency region. A horizontal line is drawn at the 1% background rejection rate.

The results for the second scenario are shown in Table 3. In this scenario, 10-15k of signal and background events were used for training and the remaining 730k events for testing. For the rectangular cuts method, however, only 150k signal and background events were used, since its implementation was not stable with larger test samples. The input variables differ in this scenario for each classifier, as the set of input variables was optimized separately for each classifier. This optimization shows clear improvement in the b tagging efficiency compared to the first scenario. For most classifiers, improved separation power was obtained by taking the logarithm of selected input variables, possibly since their distribution was more more exponential like than Gaussian. The error estimate is calculated as in the first scenario, but now with 10 runs only.

In both scenarios, most classifiers did not experience overtraining. Only PDERS and BDT were affected, but were still capable of producing good results. The overtraining was partly reduced by adding more variables and by changing the parameters of the classifiers. Nevertheless, overtraining is a feature inherent in these two classifiers and therefore even increasing the training sample size cannot remove it completely.

Table 3. Signal efficiencies of the studied classifiers at 1 and 10% mistagging rates when the combination of variables and parameters of each classifier was optimized for the 1% efficiency. The optimal combination of variables is also shown.

	1% bkg eff.	training eff. at 1%	10% bkg eff.	training eff. at 10%	variables
Cuts GA	48.7±0.8	48.5±1.1	74.3±0.3	74.2±0.5	$\sigma_{ip,2}$, $\sigma_{ip,3}$, σ_{vtx}
PDEKDE	41.2±0.8	42.5±1.6	75.5±0.3	75.8±0.4	$\log(\sigma_{ip,k}), k = 1, 2, 3,\ n_{vtx},\ \sigma_{vtx}$
PDERS	40.1±1.2	62.7±0.4	76.5±0.3	79.9±0.2	$\log(\sigma_{ip,k}),\ n_{vtx},\ \sigma_{vtx}$
kNN	49.1±0.5	49.0±1.9	79.7±0.1	79.6±0.3	$\log(\sigma_{ip,k})$,$\log(\sigma_{vtx})$, E_T
HMatrix	32.4±0.7	32.3±1.9	73.5±0.2	73.4±0.3	$\log(\sigma_{ip,3})$,$\log(ip_3)$,n_{vtx},ip_1,n_{tracks},$p_{T,3}$
Fisher	41.1±0.4	41.5±1.7	73.5±0.1	73.4±0.2	$\log(\sigma_{ip,3})$,$\log(ip_3)$,n_{vtx},ip_1,ip_2
FDA GAMT	36.8±0.7	36.8±1.8	77.3±0.1	74.0±0.4	$\log(\sigma_{ip,2})$,$\log(\sigma_{ip,3})$,$\log(\sigma_{vtx})$, n_{tracks}
TMlpANN	48.0±0.5	53.0	77.5±0.5	80.5	$\log(\sigma_{ip,k})$,$\log(ip_k)$,$\log(p_{T,k})$,$\log(\sigma_{vtx})$, n_{track}, n_{vtx}
BDT	49.2±0.5	58.4±1.4	79.1±0.1	82.0±0.4	$\log(\sigma_{ip,k})$,$\log(ip_k)$,$\epsilon_k = ip_k/\sigma_{ip,k}$, $\log(p_{T,k})$, $\log(\sigma_{vtx})$, n_{tracks}, n_{vtx},$\Sigma_k\sigma^2_{ip,k}$, $\Sigma_k ip^2_k$, $\Sigma_k\epsilon^2_k$, $\Sigma_k p^2_{T,k}$
RuleFitJF	51.8±0.2	53.8±1.2	79.0±0.1	79.3±0.3	$\sigma_{ip,k}, ip_k, k = 1, 2, 3, \sigma_{vtx}$
SVM	35.0±0.7	35.8±1.7	76.7±0.2	76.8±0.5	$\sigma_{ip,k}, ip_k, p_{T,k}, \sigma_{vtx}$,$n_{vtx}, n_{track}$

The following observations were made for individual classifiers:

- The best results for the rectangular cut optimization using genetic algorithm (Cuts GA) method were achieved with a small number variables with best signal separating power. The variables used were the impact parameter significance of the second and third track, and the secondary vertex significance. The default set of optimization parameters were used. The rectangular cut suffered from problems with very large number of events in the test tree, hence only 150k signal and 150k background events were used for testing.
- As to the likelihood classifiers the decorrelation preprocessing did not help, probably because the impact parameter distributions are exponential and far from Gaussian. The KDE smoothing was slightly better option than the spline smoothing. The option with range search (PDERS) worked well, but took about three orders of magnitude more CPU time than the PDE classifiers with smoothing.
- The kNN classifier is fairly fast. The user also does not have to give complex input. These features make kNN easy to use. The classifier itself, however, could not handle certain type of input data. It had problems with impact parameter significances as input variables. The

runs with these variables stopped with error message "kNN result list is empty or has wrong size" which prevented us from getting the results of Table 2 for kNN.

- Extremely small CPU requirements and no need for guiding made the H-Matrix and Fisher methods best candidates for the very first studies with the data. Efficiency of the H-Matrix method was of the average level of the classifiers studied. The Fisher method performed rather poorly in the first study in which track impact parameter significances were used.
- FDA classifier is very sensitive to the user defined discriminating function which is not always easy to find for non-linear problems. In the optimized case a third degree polynomial was used as the formula and it seems to produce results which do not change much as we change the random seed. FDA with genetic algorithms and Minuit (FDA GAMT) seems to be stable.
- From three neural networks implementations supported by TMVA we found TMlpANN giving systematically best results. Our initial findings supported previous results [11, 12, 13], so we decided to focus on an extended set of input variables, thus challenging core functionalities of TMVA. When we added previously unused variables p_T, σ_{vtx}, and n_{vtx}, which we knew to be relatively weak signal-background -separators, a small, but systematic, improvement was found in signal efficiencies.
- The BDT classifier was found to be robust enough to handle a large number of input variables. Although it is prone for overtraining, it was found to be one of the best classifiers in performance. The trees were allowed to grow on average to 10 nodes with the σ_{ip} as input and to 250 nodes in the full optimization case. The number of trees was chosen as 400 in all results.
- Both of the RuleFit implementations gave good results in the first scenario, but both suffered from instabilities, preventing the use of the same error estimation as for the other classifiers. RuleFitJF seems to be the more robust of the two giving slightly better results in the first scenario. In the second scenario RuleFitJF performed significantly better.
- The SVM method was found to be CPU intensive, especially with large number of training events. The efficiency was adequate compared to other classifiers. The best performance for the standard scenario was found with Gaussian kernel with $\sigma = 0.5$ and $C = 2.9$. For the free scenario (Table 3) the parameters were $\sigma = 2.0$ and $C = 3.0$.

6. Conclusions

We have succesfully tested ROOT based multivariate analysis package TMVA with MC data consisting of signal events with Higgs topology and corresponding background events.

The TMVA toolkit provides several classification methods to extract signal from background with fairly little effort. Easy application makes it possible to find the suitable method and optimal set of parameters and their transformations with a finite amount of work. Furthermore, the automatic C++ code generation makes straightforward to embed the trained classifier in external code.

Our results indicate improved classification power in comparison with earlier work [1, 12] and the reference algorithm. Some classifiers suffer from instabilities, yet TMVA shows good potential to be used for the LHC data analysis, and it even sets a new standard for easy application of MVA classifiers in HEP.

Acknowledgments

A. Heikkinen and T. Lindén would like to gratefully acknowledge the Waldemar von Frenckell Foundation. T. Lampén would like to express his gratitude to Magnus Ehrnrooth Foundation and Waldemar von Frenckell Foundation.

References

[1] CMS Physics Technical Design Report, Volume II, CERN/LHCC 2006-021 CMS TDR 8.2 26, June 2006.

[2] G. Segneri and F. Palla, "Lifetime based b-tagging with CMS", CMS NOTE 2002/046, November 2002.

[3] C. Weiser, "A Combined Secondary Vertex Based B-Tagging Algorithm in CMS", CMS NOTE 2006/014, January 2006.

[4] J. Stelzer, *TMVA - Toolkit for Multivariate Data Analysis*, these proceedings.

[5] *TMVA homepage*, (`http://tmva.sourceforge.net`)

[6] *TMVA Users Guide, Version 4*, arXiv::physics/0703039 (`http://arxiv.org/abs/physics/0703039`)

[7] F. Tegenfeldt, "TMVA - Toolkit for multivariate data analysis with ROOT, presentation at PHYSTAT-LHC Workshop on Statistical Issues for LHC Physics, June 27–29 2007.

[8] S. Slabospitsky and L. Sonnenschein, "TopReX", Comput. Phys. Commun. 148, 2002, hep-ph/0201292.

[9] T. Sjostrand, L. Lonnblad, S. Mrenna and P. Skands, "Pythia 6.3 Physics and Manual", LU TP 03–38, 2003, hep-ph/0308153.

[10] GEANT4 Collaboration, S Agostinelli et. al.,"GEANT4: A simulation toolkit", Nucl.Instrum.Meth. A506 (2003) 250-303.

[11] S. Lehti, "Study of MSSM H/A→ $\tau\tau \rightarrow e\mu + X$ in CMS", CMS Note 2006/101.

[12] A. Heikkinen and S. Lehti, "Tagging b jets associated with heavy neutral MSSM Higgs bosons", Nuclear Instruments and Methods A 559 (2006) 195–198.

[13] A. Heikkinen and S. Lehti, "Self-organized maps for tagging b jets associated with heavy neutral MSSM Higgs bosons". (To be published in the proceedings of the CHEP 2006, Mumbai, India, February 13–17, 2006.)

[14] T. Linden, F. García, A. Heikkinen, and S. Lehti, "Optimizing Neural Network Classifiers with ROOT on a Rocks Linux Cluster". (To be published in the Lecture Notes in Computer Science.)

[15] Papadopoulos, P., Katz, M., Bruno, G., NPACI Rocks: Tools and Techniques for Easily Deploying Manageable Linux Clusters, Concurrency Computat. Pract. Exper. 2002; 00:1–20.

[16] A. Hocker *et al.*, "TMVA - Toolkit for Multivariate Data Analysis", arXiv::physics/0703039.

[17] J. Zimmermann and C. Kiesling, "Statistical learning methods in high-energy and astrophysics analysis", Nuclear Instruments and Methods A 534 (2004) 204–210.

[18] K. Hultqvist *et al.*, "Using a neural network in the search for the Higgs boson", Nuclear Instruments and Methods A 364 (1995) 193–200.