Nuclear Instruments and Methods in Physics Research A 838 (2016) 137–146

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Performance and optimization of support vector machines in high-energy physics classification problems

M.Ö. Sahin*, D. Krücker, I.-A. Melzer-Pellmann

Deutsches Elektronen-Synchrotron, Notkestr. 85, 22607 Hamburg, Germany

ARTICLE INFO

Article history:
Received 18 May 2016
Received in revised form
8 September 2016
Accepted 8 September 2016
Available online 13 September 2016

Keywords:
Support vector machine
Multivariate analysis
Beyond the standard model search
High energy physics
Supersymmetry

ABSTRACT

In this paper we promote the use of Support Vector Machines (SVM) as a machine learning tool for searches in high-energy physics. As an example for a new-physics search we discuss the popular case of Supersymmetry at the Large Hadron Collider. We demonstrate that the SVM is a valuable tool and show that an automated discovery-significance based optimization of the SVM hyper-parameters is a highly efficient way to prepare an SVM for such applications.

1. Introduction

Data analysis in High Energy Physics (HEP) is a genuine multivariate problem. Despite the fact that multivariate techniques have been used in HEP for a long time, the explosive growth of machine learning (ML) techniques during the last two decades had only a limited impact on the accustomed style in which data analysis is performed in this field. TMVA, the Toolkit for Multivariate Data Analysis with ROOT [1], is probably the most commonly used software package in this context and especially Boosted Decision Trees (BDT) and Artificial Neural Networks are applied to explore the large and complex datasets delivered by present-day experiments. Only recently an increased interest in the machine learning expertise acquired in other areas of science can be observed [2–4].

In this paper we promote the application of Support Vector Machines (SVM) [5–7] for new-physics searches. Support Vector Machines are a competitive and widely used approach to binary classification. The search for new physics can be considered as a classification task where the rare new-physics signal and the dominant Standard Model (SM) background constitute the two distinct classes. Although there are a few HEP papers on SVMs [8–13], this approach seems to be much underrepresented in HEP analyses when compared to the more common BDT.

For our studies we use the popular SVM implementation LIBSVM [14] which is also used in many statistics and machine learning environments, e.g. R [15] or scikit-learn [16], and which can also be accessed by the new RTMVA and PyTMVA [17] interfaces through TMVA.

After an introduction to SVMs in Section 2, our approach to tuning the parameters that define the SVM model (hyper-parameters) is described in Section 3.

In Section 3.3 we first discuss a toy model and then in Section 4 an example for an actual new-physics search, targeted at a s Supersymmetric partner of the top quark at the LHC. We demonstrate that the SVM is a valuable tool for HEP searches and that the not-common use of the SVM approach within the HEP community can be related to the limited use of its implementation in TMVA without an automated hyper-parameter search. Moreover, we show that a significance-based optimization of the hyper-parameters is a highly efficient way to prepare an SVM for a HEP search.

Our software package, called SVM-HINT, that performs such a significance-based optimization of hyper-parameters and interfaces ROOT [18] trees with LIBSVM is freely available [19].

2. Support vector machines

A HEP search typically starts with a set of physical variables and cuts on these variables. The cuts are defined to select a

* Corresponding author.
E-mail addresses: ozgur.sahin@desy.de (M.Ö. Sahin), dirk.kruecker@desy.de (D. Krücker), isabell.melzer@desy.de (I.-A. Melzer-Pellmann).

http://dx.doi.org/10.1016/j.nima.2016.09.017
0168-9002/© 2016 Elsevier B.V. All rights reserved.

new-physics signal against the background of known physics and are often chosen in an ad-hoc style. The optimal use of these variables is a typical machine learning problem. Monte Carlo (MC) simulation samples for the signal and background class can be used to train a supervised[1] machine learning algorithm which is potentially a more efficient classifier than a set of cuts. Typically, a subset of the signal and background events are chosen for the training of the machine learning algorithm, while a second subset is used to test the classifier obtained after the training.

We write for a set of N training events:

$$(y_1, \mathbf{x}_1), (y_2, \mathbf{x}_2), \ldots, (y_i, \mathbf{x}_i), \ldots, (y_N, \mathbf{x}_N) \quad y_i \in \{-1, 1\}, \tag{2.1}$$

$$\mathbf{x}_i = \left(x_i^{(1)}, \ldots, x_i^{(n)}\right) \tag{2.2}$$

where for an event $i = 1 \ldots N$ the label y_i distinguishes between signal and background and $\mathbf{x}_i$ is an n-dimensional vector formed from the physical variables under consideration. These vectors constitute an n-dimensional real vector space $\mathbb{V}$.

A support vector machine is a supervised binary classifier based on the intuitive concept of an n-dimensional hyperplane separating two distinct classes. In this approach, finding the best separating hyperplane is considered to be a convex optimization problem. In its simplest form a SVM defines the eponymous *support vectors* as those elements of the training sample which are closest to the hyperplane. The separation margin between the classes is completely defined by the support vectors and maximized by the algorithm.

This idea can be extended to overlapping distributions and eventually, by an implicit transformation of the variables, known as the *Kernel trick*, to non-linear problems. The last two modifications introduce additional hyper-parameters that must be set to some best value before the SVM training which is discussed in more detail in Section 3. In the following we give a short introduction to the concepts behind the SVM algorithm and to the hyper-parameter tuning. The reader who is more interested in physical applications may continue with Section 4.

2.1. Linearly separable distributions

A linear SVM separates the elements of two classes by an optimal hyperplane. Optimal in this approach is a hyperplane that maximizes the margin between the two classes for a given training sample. Those elements of the training sample sitting on the maximum margin boundaries are called support vectors. The support vectors are sufficient to construct the optimal hyperplane.

A separating hyperplane can be described as $\mathbf{w}\cdot\mathbf{x} + b = 0$, $\mathbf{w} \in \mathbb{V}$ and $b \in \mathbb{R}$. The vectors of the training sample are either *above* or *below*[2] this plane. We can always choose the scale of $\mathbf{w}$ and b such that for the vectors which are closest to the hyperplane, i.e. the support vectors $\mathbf{x}_k$, we obtain $\mathbf{w}\cdot\mathbf{x}_k + b = \pm 1$. Multiplied with the class label y_i, this expression must always be positive for correctly classified points:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \geq 0, \tag{2.3}$$

and the equality is satisfied by the support vectors (circled in Fig. 1). With this scale, the separation margin ρ between the support vectors on both sides is given by:

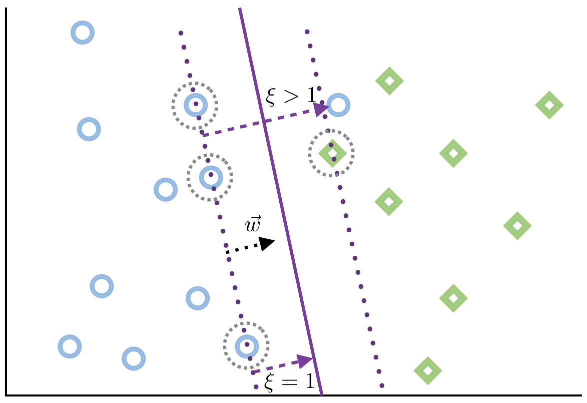

Fig. 1. The events represented by blue circles belong to the first class ($y=-1$), whereas the green rectangles belong to the second class ($y=1$). The dotted straight lines represent the maximum margin boundaries, and the corresponding support vectors are marked by dotted circles. From all possible hyperplanes dividing the two samples, the one with the largest margin is chosen. The blue circle at $\xi > 1$ is not linearly separable, as discussed in Section 2.2.

$$\rho(\mathbf{w}, b) = \frac{2}{|\mathbf{w}|}. \tag{2.4}$$

Maximizing the margin ρ is equivalent to minimizing $|\mathbf{w}|^2$. Finding the optimal separating hyperplane is therefore identical to solving the following quadratic optimization problem where the correct classification is enforced by the constraints from Eq. (2.3):

$$\min_{\mathbf{w}\in\mathbb{V},\, b\in\mathbb{R}} \frac{1}{2}|\mathbf{w}|^2$$
$$\text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + \mathrm{b}) \geq 1 \ \text{ for all } \mathrm{i} = 1\ldots\mathbf{N}. \tag{2.5}$$

This is a quadratic optimization problem with inequality constraints and can be solved using Lagrange multipliers α_i (with $\alpha_i \geq 0$).

Most SVM implementations search for a numerical solution to the so-called dual problem [20] which is described by the Lagrangian

$$\mathcal{L} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j\, \mathbf{x}_i\cdot\mathbf{x}_j + \sum_{i=1}^{N} \alpha_i. \tag{2.6}$$

Due to the constrains all non-support vectors are forced to have vanishing Lagrange multipliers $\alpha_i = 0$, and only the support vectors contribute in the calculation of the optimal $\mathbf{w}_0$ and b_0. The decision function, i.e. the expression to predict the class label $\hat{y}$ of a new vector $\mathbf{u}$, then becomes:

$$\hat{y} = \mathrm{sign}\left(\sum_{k=1}^{N_{SV}} y_k\alpha_k\mathbf{x}_k\cdot\mathbf{u} - b_0\right), \tag{2.7}$$

where N_{SV} is the number of support vectors.

It is important to note that the dual form in Eq. (2.6) only depends on the scalar products of input vectors, and the same is true for the decision function Eq. (2.7). This advantage of the dual form is essential for the non-linear case in Section 2.3.

2.2. Overlapping distributions

The method described so far assumes non-overlapping signal and background distributions. By allowing misclassification, the *hard margin* separation above can be modified into a *soft margin* approach. This is done by introducing *slack* variables [6] ($\xi_i \geq 0$, $i = 1\ldots N$) which measure for each training vector the relative distance by which they are on the wrong side of the separating hyperplane, as illustrated for one misclassified point in

[1] A ML algorithm is called *supervised* when the class membership of all training vectors is known.

[2] For simplicity we use a 3 dimensional way of speaking. All described concepts are valid in n dimensions.

Fig. 1. They are used to weaken the constraints (2.3):

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \tag{2.8}$$

and allow to introduce the sum of the slacks $\sum_i^N \xi_i$ as a penalty term into the optimization problem. The modified Lagrangian becomes:

$$\mathcal{L} = \frac{1}{2}|\mathbf{w}|^2 + C\sum_{\mathbf{i}} \xi_i - \sum \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x} + b) - 1 + \xi_i] - \sum \beta_i \xi_i, \tag{2.9}$$

and the extremum condition $\partial\mathcal{L}/\partial\xi_i = 0$ implies a relation between the Lagrange multipliers, $\beta_i = C - \alpha_i$ and we are left with an optimization problem that is identical to the hard margin case up to the modified constraints.

The constant C that controls the strength of the penalty term appears now only as an upper limit on the Lagrange multipliers $0 \leq \alpha_i \leq C$, restoring the hard margin case in the limit of $C \to \infty$. Furthermore, it controls the trade-off between simplicity of the decision rule and error frequency and is one of the hyper-parameters that must be set to a sensible value *before* the SVM training.

2.3. Non-linear distributions

For HEP searches we expect complicated, non-linear hypersurfaces separating the two classes. The linear SVM presented in Sections 2.1 and 2.2 can be extended to create non-linear decision boundaries. The basic idea for a non-linear SVM [5] is to map the input vectors $\mathbf{x}_i$ into a higher dimensional *feature space F* where the problem becomes linearly separable: $\mathbf{x}_i \mapsto \phi(\mathbf{x}_i) \in F$. The construction of a linear SVM in this feature space follows the same lines as before and the dual Lagrangian from Eq. (2.6) will contain an inner product $\langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$ of elements of F. The peculiar fact that the input vectors only appear in the dual Lagrangian, as well as in the decision function of Eq. (2.7), in form of scalar products allows us to avoid the explicit mapping and to use instead a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$, such that the dual Lagrangian and the decision function become

$$\mathcal{L} = -\frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j\, K(\mathbf{x}_i, \mathbf{x}_j) + \sum \alpha_i, \tag{2.10}$$

$$\hat{y} = \text{sign}(\hat{f}), \quad \hat{f} = \sum_{k=1}^{N_{SV}} y_k\alpha_k\, K(\mathbf{x}_k, \mathbf{u}) - b_0. \tag{2.11}$$

The existence of the mapping $\mathbf{x} \mapsto \phi(\mathbf{x})$ is guaranteed by Mercer's theorem, given that the kernel function fulfills certain conditions [5,21]. In general, the feature space F may be an even infinite dimensional Hilbert space. For an in-depth exposition on kernel techniques in machine learning see for example [20]. A common and in many applications successful choice [22] is the Gaussian radial basis function (RBF) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma|\mathbf{x}_i - \mathbf{x}_j|^2}. \tag{2.12}$$

The width of this kernel function is controlled by the value of γ which is the second hyper-parameter that must be set to a sensible value *before* the training of the SVM. We note that the RBF kernel only contains one parameter and the components of the data vectors are added quadratically. It is therefore necessary to normalize the individual components of the input vectors. Such a scaling ensures that all components of the input vectors may contribute equally. Within this paper we always normalize by the range between the minimum and maximum values for each dimension of the input data vector (2.2).

2.4. Probabilistic output

The SVM described so far is a binary classifier that applies a signal or background label to each event. In many cases a posterior probability P that quantifies the belief in the class label is useful and offers an easier interpretation. In Section 3 a probability cut, $P > P_0$, is used to modify the signal-to-background ratio and the total number of selected events. Such a classification probability can be estimated [23,24] by fitting a two-parameter sigmoid model mapped to the range $[0, 1]$. In general, especially for the non-linear SVM, the result will be biased if the SVM training data itself is used for the fit. In LIBSVM the bias is avoided by internally performing a 5-fold cross-validation.[3] It is important to note that a strictly increasing sigmoid function of the decision value does not change the order of any sequence of decision values. Since the cross-validation increases the computational burden by a factor of five, we do not calculate probabilities during the parameter scan but only for presenting the final results.

3. Hyper-parameter tuning

The two parameters C and γ, introduced in the previous section, must be set to sensible values before the training of the support vector machine. These values are crucial for the performance of the algorithm, and different strategies to optimize the parameter choice are possible. The easiest approach is a simple grid search. A two dimensional grid is defined, at each grid point the SVM training is performed on a training dataset, and the trained machine is applied to an independent test data sample where some performance measure is evaluated. Eventually, the $(C, \gamma)-$ pair with the highest performance index is used. We first consider what could be an appropriate performance measure for a new physics search and describe then the tuning algorithm in some detail.

3.1. Performance measures

3.1.1. Machine learning performance measures

From a machine learning perspective, a natural performance measure describes how well a classifier separates the two distinct classes of inputs. On a sample of test data we know the true class labels. There are 2×2 categories formed by the true label $y \in \{-1, 1\}$ and the label $\hat{y} \in \{-1, 1\}$ estimated by the SVM. The matrix they form is known as confusion matrix. The relative amount of test data in these categories can be used to quantify the performance of a machine learning algorithm. Typical ML measures are the *accuracy* which gives the percentage of correctly predicted labels, or the *precision* which, in our case, is the percentage of correctly predicted signal events. Another frequently used measure is the *AUC*, the area under the receiver operator curve (ROC). The ROC curve shows the background rejection (false positive) against the signal efficiency (true positive) at various threshold values of the decision function. While the use of these and other performance measures is common also in HEP [1], we will follow a different approach to optimize the SVM.

3.1.2. Physics motivated performance measures

The maximum number of correctly classified events is of secondary importance for a HEP search. There is a much more physically and statistically motivated measure: the discovery

[3] The procedure of k-fold cross-validation splits the training data randomly into k equal sized subsamples. One subsample is retained for validation while the remaining k-1 subsamples are used for training. The training is repeated k times with changing roles such that each subsample is used exactly once for validation.

significance. Optimizing the significance is a common procedure in HEP. Typically the search area is optimized for a statistically relevant signal to background ratio that allows to prove or reject a certain hypothesis. Here, we consider the case of a cut-and-count analysis for which several significance estimators are commonly used [25,26]. Optimizing a certain, statistically motivated, figure of merit is common practice in HEP to select different ML algorithms or different sets of input variables. The new insight of this paper is that such a procedure can successfully be applied in the stage of model selection, i.e. during the hyper-parameter tuning.

The exact numerical calculation of the statistical significance may become computationally costly. A well performing estimate for the discovery significance, known as *Asimov* estimate, has been given in [25]. For the case of Poisson distributed background and signal events (s,b) with background uncertainty σ_b the approximated median discovery significance becomes

$$Z_A = \left[2\left((s+b)\ln\left[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2} \right] - \frac{b^2}{\sigma_b^2}\ln\left[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2)} \right] \right) \right]^{1/2} \quad (3.1)$$

3.2. Hyper-parameter search

The SVM with RBF kernel requires two hyper-parameters: C (Section 2.2) and γ (Section 2.3). In addition to the hyper-parameters, the number of selected signal and selected background events (s, b) depends on the probability cutoff P_0 (Section 2.4), or a corresponding decision value $\hat{f}_0$. The easiest algorithm to find the optimal values for these parameters is a grid search. At each point of a logarithmically spaced grid in (C, γ) a SVM is trained and, on an independent test dataset, the Asimov significance Z_A is calculated as function of P_0. In general, the best cut P_0 is selected as the value with the highest significance. To avoid artificially high significance values due to statistical fluctuation of a small signal at very low values of b, a further requirement of at least 5 signal events is applied. While conceptually the plain grid search is sufficient to find good values for (C, γ), computationally it may be advantageous to use a more refined algorithm for the hyper-parameter tuning. The details of the iterative algorithm used for the results in this paper are given in Appendix A.

3.3. Performance comparison on a toy model

Comparing speed and classification performance of different classifiers is not always straightforward. In order to have simple and well defined conditions, we start with a toy model and compare our SVM-HINT framework with a BDT and an SVM, both implemented with the TMVA 4.2.0 library. The toy model includes the following variables V_i generated with the random numbers x_i (the *tilde* means sampled from):

$$\begin{aligned} V_1 &= \sin(x_1); \quad x_1 \sim g(x_1|a, b) \\ V_2 &= x_2; \quad x_2 \sim \exp(-x_2/c) \\ V_3 &= x_3; \quad x_3 \sim g(x_3|d, e) \\ V_4 &= \sqrt{x_4}; \quad x_4 \sim \exp(-x_4/f) \end{aligned} \quad (3.2)$$

where $g(x|\mu, \sigma)$ is a Gaussian distribution with mean μ and width σ, and $a, b, c, d, e, f > 0$ are constants with different values for signal and background samples.

This model does not have any hidden correlation between the variables and each ML algorithm needs only to find a set of independent optimal cuts. Due to its simplicity, the toy model enables us to generate large quantities of events to study the training time as function of the training sample size for the different codes.

As explained in Section 3, SVM-HINT provides a hyper-parameter search. The hyper-parameter search is performed beforehand and is not part of the timing performance study. The SVM implementation provided by the TMVA library lacks such an automated hyper-parameter search. We therefore apply the same hyper-parameter values as obtained by the SVM-HINT tuning algorithm. The out-of-the-box performance of the TMVA-BDT cannot compete in most cases with the automatically tuned SVM-HINT. There is a trade-off between classification performance and time consumption of the BDT which can be controlled by an appropriate choice of the BDT parameters, e.g. the number of trees, minimum node size, and cut values, as introduced in Appendix B. For comparing the training times, we follow the strategy to optimize the BDT parameters manually[4] to accomplish a similar Asimov discovery significance as achieved with the SVM-HINT. The following parameters have been optimized: the number of trees, the minimum node size in a tree, the maximum depth of a tree, and a parameter for AdaBoost algorithm. This allows us to compare the time consumption of equally performing algorithms. Blindly optimizing for maximal performance on a fixed evaluation training sample could otherwise produce slow, over-sized trees and would place a disadvantage on the BDT implementation.

Fig. 2 shows the results for the different classifiers applied to the same sets of toy data ranging from 10,000 to 50,000 events. At low numbers of training events the BDT performs better than the other single-threaded classifiers. With increasing training sample size the number of trees needed to achieve a competitive classification performance becomes larger with negative impact on the training time. The TMVA-SVM does not scale well in terms of timing performance and it performs poorly on bigger samples. Overall the SVM-HINT performs similar or better compared to TMVA-BDT and TMVA-SVM. In addition, the SVM-HINT can efficiently take advantage of multi-core architectures. Naturally, the multi-threaded performance of the SVM-HINT with 12 threads outperforms the other implementations.

4. Application in a third-generation Supersymmetric partner search

As a real-world physics example we consider a search for Supersymmetry [27–34] at the LHC. The search is designed for the case of direct top-squark pair production with subsequent decay of the top squarks into the lightest Supersymmetric particle (LSP) and a top quark. Several searches for such a scenario have been performed at Tevatron as well as at the LHC [35–42]. These searches are typically interpreted in simplified models [43–45]. In our model we assume that only the lighter top squark and the lightest Supersymmetric particle, the neutralino, have low-enough masses to be accessible at the LHC. Limits are then set as a function of the top squark and the neutralino masses. For our study, we chose one set of mass parameters that are expected to be on the border of discovery with about 300 fb^{-1} of data taken at the LHC, with the top squark mass fixed at 900 GeV and the neutralino mass at 100 GeV. We consider only the dominating background, top-antitop ($t\bar{t}$) production. A more detailed study covering a large top-squark neutralino mass parameter space and all relevant backgrounds can be found in Ref. [46]. A leading-order simulation is sufficient for our purpose and we only take into account next-to-leading order results for the total cross sections of signal [47] and the $t\bar{t}$ background [48,49]. The signal cross section is 17.2 fb.

[4] Configuration files are available at [19].

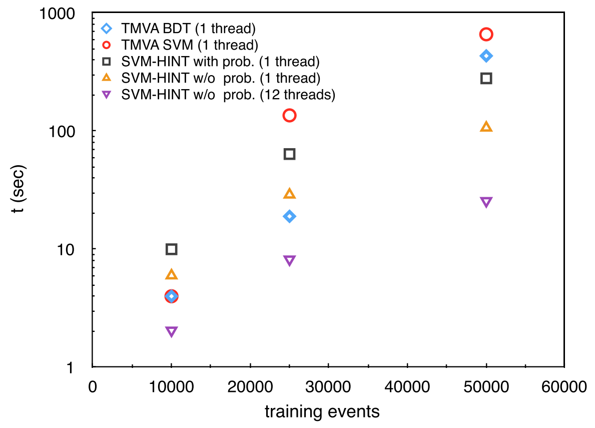

Fig. 2. The timing performance of the classifiers are compared on a computer with two Intel® Xeon® E5-2440 CPUs and 12 physical threads running at 2.40 GHz clock speed. Number of threads used by each classifier implementations are stated within parentheses in the legend.

Table 1
Top-squark search: List of baseline selection requirements.

$\lvert\eta_{l,\,\mathrm{jet}}\rvert < 2.4$
$p_{\mathrm{T},l} > 30$ GeV
$p_{\mathrm{T,jet}} > 40$ GeV
$p_{\mathrm{T,jet1}} > 80$ GeV
$p_{\mathrm{T,jet2}} > 60$ GeV
$\not{E}_{\mathrm{T}}$>200 GeV
$H_{\mathrm{T}} > 300$ GeV
$n_{jet} > 3$
$n_{b-jet} > 0$

Table 2
Summary of all low-level and high-level variables used in the analysis. Set 1 includes all variables. Set 2 and set 3 consist of low- and high-level variables, respectively. Set 4 is a smaller subset of high- and low-level variables.

	Variable	Set 1	Set 2	Set 3	Set 4
Low-level	$p_{\mathrm{T},l}$	•	•		
	η_l	•	•		
	$p_{\mathrm{T},jet(1,2,3,4)}$	•	•		
	$\eta_{jet(1,2,3,4)}$	•	•		
	$p_{\mathrm{T},b\,jet1}$	•	•		
	$\eta_{b\,jet1}$	•	•		
	n_{jet}	•	•		
	$n_{b\,jet}$	•	•		
	$\not{E}_{\mathrm{T}}$	•	•		•
	H_{T}	•	•		•
High-level	m_{T}	•		•	•
	$m_{\mathrm{T2}}^{\mathrm{W}}$	•		•	•
	$\Delta\phi(W, l)$	•		•	
	$m(l, b)$	•		•	
	Centrality	•		•	
	Y	•		•	
	H_{T}-ratio	•		•	
	$\Delta r_{\mathrm{min}}(l, b)$	•		•	
	$\Delta\phi_{\mathrm{min}}(j_{1,2}, \not{E}_{\mathrm{T}})$	•		•	

Pythia6 [50] is used for the event simulation and Delphes3 [51] to model the detector response. The detector model is a *combined* ATLAS and CMS detector, as it had been used for the 2013 Snowmass effort [52]. Jets are anti-k_T [53] with a cone parameter of $R=0.4$ and we use lepton as generic term for electrons and muons.

In order to reduce the time required for training and optimization, a pre-selection, summarized in Table 1, is applied to the signal and background samples. We require at least one lepton with transverse momentum p_{T} of 30 GeV within a pseudorapidity of $\lvert\eta\rvert < 2.4$. Each event must contain at least four jets with $p_{\mathrm{T}} > 40$ GeV, while the highest-p_{T} (called 'leading') jet is asked to have $p_{\mathrm{T}} > 80$ GeV, and the sub-leading jet $p_{\mathrm{T}} > 60$ GeV. At least one of the jets is required to be tagged as originating from a bottom quark. In addition, we require the missing energy perpendicular to the beam direction, $\not{E}_{\mathrm{T}}$, to be above 200 GeV, and the scalar sum over the transverse momenta of all preselected jets, H_{T}, to be above 300 GeV. The selected sample, which is used for training and tuning, consists of about 90,000 events with a sample composition of 1/3 signal and 2/3 background events. An independent, equal-sized sample is used in the evaluation step.

After this preselection, the remaining background is still several orders of magnitude higher than the investigated signal.

Top quarks decay to almost 100% to one b quark and a W boson, with the latter decaying either to two quarks or to one lepton and a neutrino. When requiring one lepton in the final state, we mainly expect to select semi-leptonic decays of top quarks (where one W boson decays to two jets and the other to a lepton and a neutrino), but dileptonic top decays (where both W bosons decay to a lepton and a neutrino) might be selected as well, if one lepton is not identified (lost) for various reasons.

We categorize the physical variables with respect to their mathematical complexity as high-level and low-level variables. The low-level variables consist of basic properties of the reconstructed physics objects measured by the detector, while the high-level variables are constructed from the low-level variables using physical insight to improve the classification performance. The physics objects are jets and leptons (electrons and muons). For simplicity, we do not consider tau leptons since their experimental reconstruction is more complex.

Low-level variables are the transverse momentum and the pseudorapidity of the single lepton, of the 4 jets with the highest transverse momenta, and of the leading b-quark jet. In hadron collider experiments $\not{E}_{\mathrm{T}}$ is commonly reconstructed as an independent variable. It is therefore treated as a low-level quantity, as well as H_{T}. In many SUSY models, we expect large $\not{E}_{\mathrm{T}}$ due to the LSP, which is expected to be neutral and weakly interacting and will therefore not be detected. As SUSY particles are heavy, we also expect a large amount of energy in the detector leading to large H_{T}. In addition, the multiplicities of jets (n_{jet}) and b-quark jets ($n_{b-\mathrm{jet}}$) are included.

As high-level variables we consider the following quantities: the transverse mass m_{T}, defined as $m_{\mathrm{T}} = \sqrt{2\,p_{\mathrm{T},l}\,\not{E}_{\mathrm{T}}(1-\cos\Delta\phi(l,\not{E}_{\mathrm{T}}))}$, where $\Delta\phi(l,\not{E}_{\mathrm{T}})$ is the azimuthal angle between the lepton and the $\not{E}_{\mathrm{T}}$ vector, can be used to suppress the background from W boson production, as m_{T} of leptonic W decay events does not exceed the W mass. Dileptonic-$t\bar{t}$ events with one lost lepton are an important background since the lost lepton mimics large missing energy from the LSP. The $m_{\mathrm{T2}}^{\mathrm{W}}$ variable [54] is constructed exploiting the knowledge of the $t\bar{t}$ – decay kinematics to separate such events. Top-squark production is a high-mass process with large missing energy. High-mass production is typically related to more centrally distributed particles in the detector, such that the *Centrality*, defined as $\sum_{\mathrm{jets,l}} p_{\mathrm{T}} / \sum_{\mathrm{jets,l}} p$, can be used to enhance such events. Commonly used relations between the hadronic activity and the missing energy are $Y{=}\not{E}_{\mathrm{T}}/\sqrt{H_{\mathrm{T}}}$, often referred to as $\not{E}_{\mathrm{T}}$ significance, and the H_{T} ratio, the normalized hadronic activity in the hemisphere of $\not{E}_{\mathrm{T}}$. The last group of variables exploit topological relations between the event particles: $\Delta\phi(W, l)$ is the azimuthal angle between the W

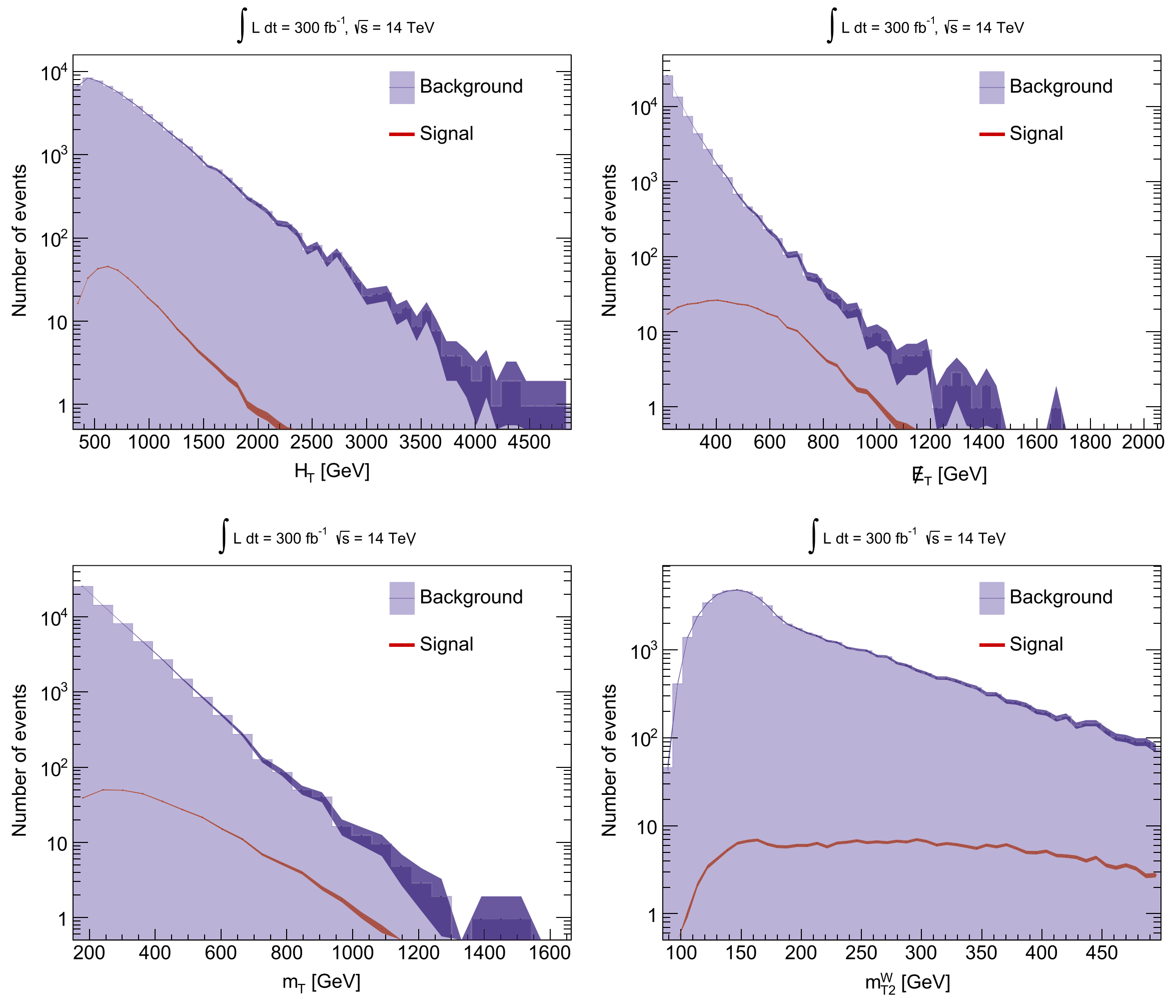

Fig. 3. The distribution of signal (red line) and background (blue filled histogram) after the baseline selection for two low-level and two high-level variables that are used in the analysis: H_T (top left), $\not{E}_T$ (top right), m_T (bottom left) and m_{T2}^W (bottom right). The *y* axis shows the number of events (normalized to the integrated luminosity), and the *x* axis shows the variable value for a given bin. Statistical errors are represented by transparent bands. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

boson and lepton, $\Delta r_{min}(l, b)$ is the distance ($\Delta r = \sqrt{\Delta\phi^2 + \Delta\eta^2}$) between the *b* jet and the closest lepton and $m(l, b)$ is the invariant mass of the *b* jet and the closest lepton.

A compilation of all low-level and high-level variables is given in Table 2, together with the definition of four sets of variables which are considered to investigate the influence of the variable multiplicity and complexity in the multivariate analysis. We define one set containing all variables, one using only low- or high-level variables, respectively, and a subset of two low-level and two high-level variables with relatively large separation power.

Fig. 3 shows the distribution of signal and background for two low-level and two high-level variables, H_T, $\not{E}_T$, m_T and m_{T2}^W after the baseline selection, normalized to the expected luminosity at the end of the LHC run in the year 2023, corresponding to 300 fb^{-1}. The background is several orders of magnitude higher than the signal, and the distributions of signal and background are quite similar due to their similar kinematics.

The samples are separated into three independent subsamples: training, test and evaluation sample. Each classification method is optimized over the training and test samples and the best-performing configuration is applied to the independent evaluation sample for the final performance assessment.

The TMVA-BDT has been manually trained and tested over 32 different settings for each of the four variable sets independently in order to obtain optimal parameters as described in Appendix B, while the SVM-HINT is auto-tuned by the iterative grid search, as described in Section 3 and Appendix A. Without modifying the default SVM-HINT settings, the two step grid search hyper-parameter optimization function provides the optimal parameters using test and training samples. We calculate the final significance with an independent evaluation sample.

Results. Figs. 4 and 5 show the performance of the SVM-HINT and the TMVA-BDT for the four different variable sets. The numerical results are summarized in Tables 3, 4.

Both SVM-HINT and TMVA-BDT obtain the highest performance with the largest number of variables (set 1) in Table 3. Both methods perform much worse with only low-level variables (set 2), while the high-level variables (set 3) are clearly able to separate signal and background. Despite of the poor performance of the low-level variables, they add a substantial amount of extra information and enlarge the significance when they are combined with the high-level variables. We conclude that even with a number of 25 variables at least in our example, SVM-HINT as well as TMVA-BDT do not require a preselection of variables. We would like to note here that this study is not meant to compare the two methods, but to put our SVM-HINT results into the context TMVA-BDT which is a common tool in HEP. SVM-HINT with automated hyper-parameter tuning needs no further manual optimization and can make use of high numbers of variables simultaneously

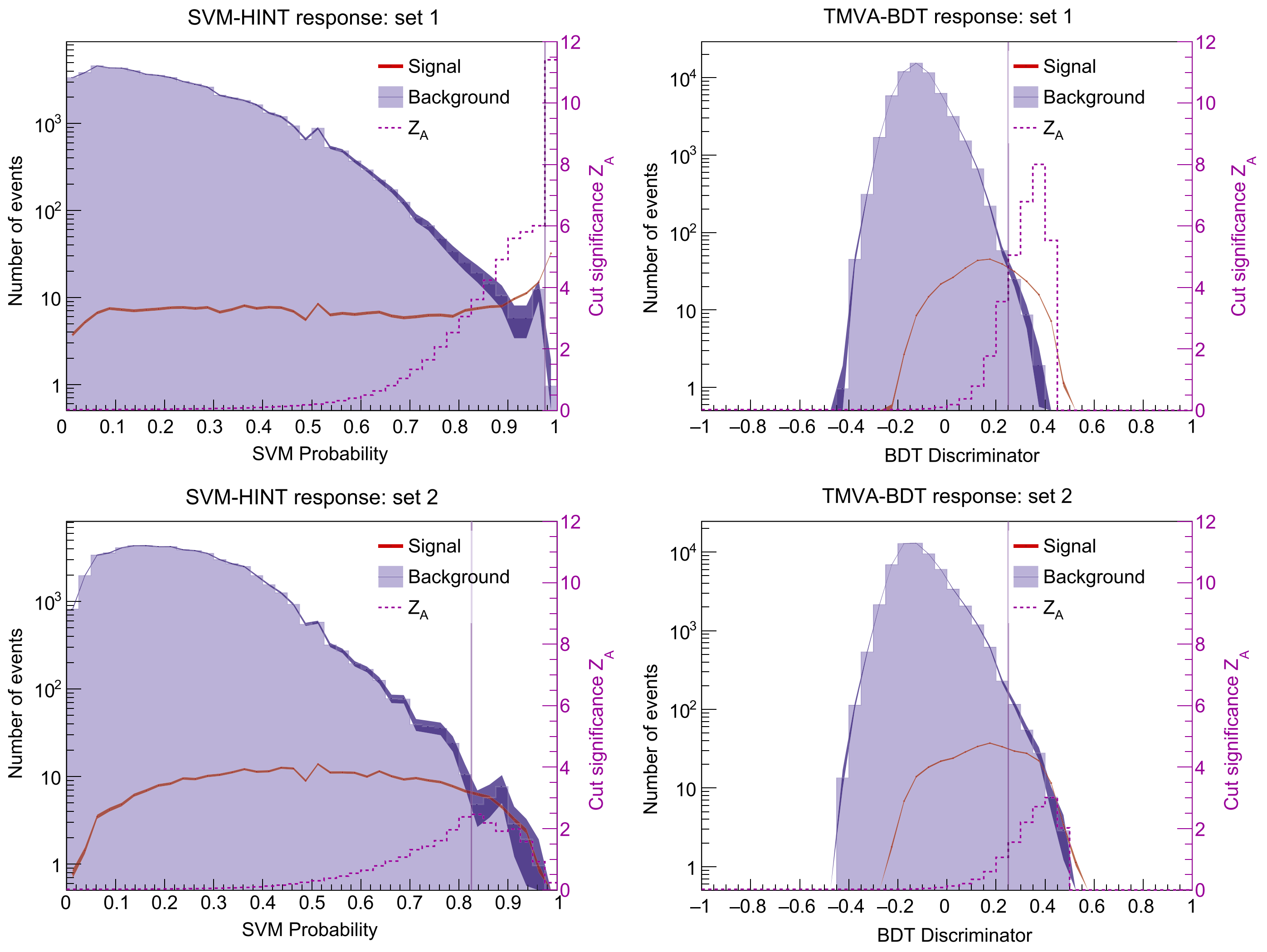

Fig. 4. The SVM-HINT and TMVA BDT responses trained with the variable set 1 and set 2, as defined in Table 2. Even though the optimal Z_A efficiency information is not available in data, it is included to demonstrate the reliability of the optimal discriminator cut (vertical line) output from the classifier implementations. The *y* axis on the left shows the number of events (normalized to the aimed integrated luminosity), whereas the *y* axis on the right shows the Asimov significance for the discriminator cut per bin.

with an increasing classification power.

The importance of the choice of a good performance measures is visible in Table 4. The common approach to use the accuracy for tuning the SVM, i.e. the percentage of correctly classified events, is not well suited to select the optimal hyper-parameters. In Table 4 all values are equally high since they are dominated by the large number of correctly classified background events. Precision, the number of correctly classified signal events only, is more sensitive but not a useful choice since, as a ratio of event numbers, it is insensitive to the absolute number of selected signal events. The statistical correct choice of using the Likelihood ratio for the discovery significance implemented in the Asimov estimate Z_A gives a clearly different selection of hyper-parameters. For set 1, SVM-HINT is able to select an almost background-free region. This is only possible since the Z_A estimate remains valid even at vanishing background where $s/\sqrt{b}$ would diverge [25]. The Asimov estimate has also the advantage to provide a clear optimum since it correctly describes the trade-off between reduced background and reduced signal statistics. Due to the choice of $s/\sqrt{s+b}$, TMVA-BDT does not pick the statistically optimal discriminator cut-off. In Table 3, the discovery significance calculated with the discriminator cut provided by TMVA is shown as Z_A^{prov}, while Z_A^{max} is realized at the optimal discriminator cut, i.e. the cut that maximizes the Asminov significance. Also the TMVA-BDT event numbers in Tables 3, 4 are given for this optimal cut to allow a fair comparison.

5. Conclusions

In this paper we present an automated method for the hyper-parameter tuning of the Support Vector Machine classifier. The choice of an appropriate set of hyper-parameters is essential for the SVM performance and we show that our algorithm is able to find such a parameter set. We have studied the CPU needs of our algorithm on a toy data set and demonstrate its ability to take advantage of multi-core architectures. As a real-world physics example we have performed a search for Supersymmetry at the LHC on simulation data and we compare our approach to the Boosted Decision Trees classifier.

Our results show that a Support Vector Machine is an efficient machine-learning algorithm for new-physics searches in high-energy physics. The rare applications of this tool in this field may be related to its implementation provided by the popular TMVA library, which does not provide an automated hyper-parameter tuning. An appropriately-designed automated search for the two hyper-parameters easily overcomes these limitations and reveals

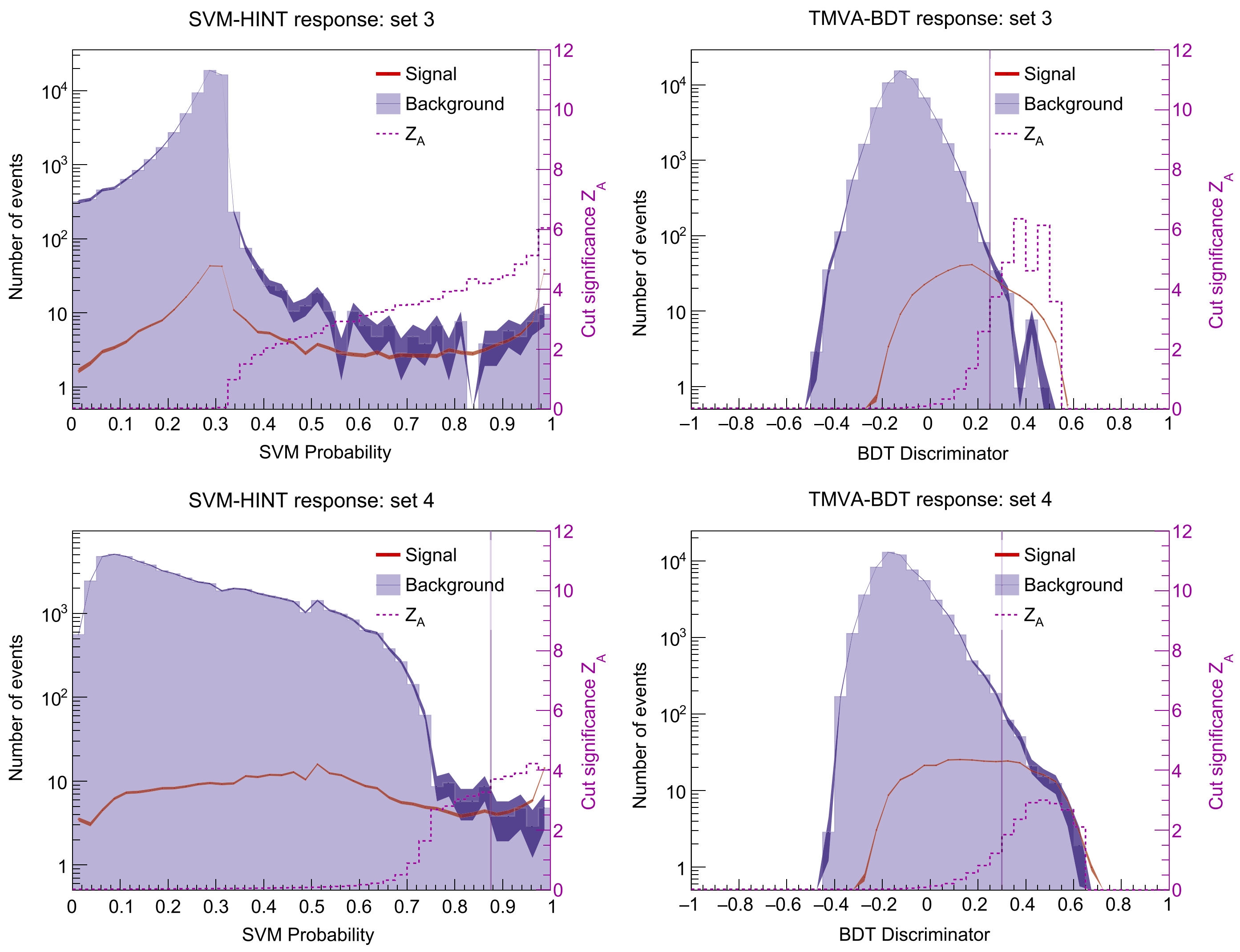

Fig. 5. The SVM-HINT and TMVA BDT responses as described in Fig. 4 trained with the variables set 3 and set 4.

Table 3
Summary of the number of signal (N_s) and background (N_b) events expected for the different variable sets and the two multivariate methods under study. In addition, we quote the achievable discovery significance (Z_A^{max}). For comparison, we also quote the significance for the cut value suggested by TMVA-BDT (Z_A^{prov}).

Set	SVM-HINT			TMVA-BDT			
	N_s	N_b	Z_A^{max}	N_s	N_b	Z_A^{max}	Z_A^{prov}
1	32.1 ± 0.6	1.0 ± 1.0	11.4	24.1 ± 0.5	1.9 ± 1.4	8.0	5.1
2	23.2 ± 0.5	23.9 ± 4.8	2.5	16.2 ± 0.4	10.5 ± 3.2	3.0	1.5
3	37.8 ± 0.6	9.6 ± 3.0	6.1	40.5 ± 0.7	9.6 ± 3.0	6.4	3.7
4	33.4 ± 0.6	20.1 ± 4.4	3.7	40.5 ± 0.7	35.4 ± 5.8	3.0	1.8

Table 4
The first 4 elements of each set are the elements of the confusion matrix for the example search. From this values accuracy, precision and discovery significance as defined in Section 3.1 are derived.

Classified as:	Signal		Background		Accuracy	Precision	Z_A^{max}
Sample:	Signal	Bgrd.	Signal	Bgrd.			
SVM-HINT							
Set 1	32.1	1.0	286.7	58701.5	99.51%	97.0%	11.4
2	23.2	23.9	295.6	58678.6	99.46%	49.3%	2.5
3	37.8	9.6	281.0	58692.9	99.51%	79.7%	6.1
4	33.4	20.1	285.4	58682.4	99.48%	62.4%	3.7
TMVA-BDT							
Set 1	24.1	1.9	294.7	58700.6	99.50%	92.7%	8.0
2	16.2	10.5	302.6	58692.6	99.47%	60.7%	3.0
3	40.5	9.6	278.3	58692.9	99.51%	80.8%	6.4
4	40.5	35.4	278.3	58667.1	99.47%	53.4%	3.0

the full potential of this approach. The performance of Boosted Decision Trees depends on a larger number of parameters, which complicates the construction of an automated tuning procedure. The clear advantage of the Support Vector Machine with an RBF kernel is the rather straightforward hyper-parameter tuning. While both approaches deliver compatible results when comparing the training times with a single thread, SVM-HINT has the ability to use multi-thread architectures, leading to significantly reduced training times.

We demonstrate that the approximated median discovery significance (Asimov significance) is an effective figure of merit for the parameter tuning, and that only two parameters need to be adapted to define a well performing search tool.

The SVM shows good generalization properties and the maximum-margin concept guarantees that the best separating hyperplane is selected. Furthermore, Support Vector Machines are known to be robust against a large number of even partially correlated input variables. This is in agreement with our studies

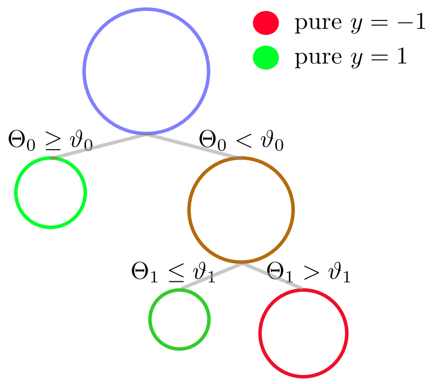

Fig. 6. Representation of a simple binary decision tree structure: Each node is split with respect to a variable Θ_i and a cut value ϑ_i determined by the performance measure.

where the algorithm reliably exploits information available from 25 input variables.

Overall, we show that the Support Vector Machine algorithm is able to compete with the Boosted Decision Tree algorithm which is currently the prevalent machine learning tool in high-energy physics.

Acknowledgments

M. Ö. Sahin would like to thank the Joachim Herz foundation for the support.

Appendix A. Iterative grid search

As shown in Section 3.2 an SVM with RBF kernel requires two different hyper-parameters to be adjusted: C and γ. While in principle a *brute force* grid search is sufficient to find the best hyper-parameters, we used an adaptive search strategy for the results in this paper which we describe here for completeness. The SVM-HINT grid search algorithm uses a modified version of the Asimov Significance (3.1), a significance score $\tilde{Z}_A$ based on the difference between the significance value observed in the test sample and the significance value from the training sample.

$$\tilde{Z}_A = Z_A^{(test)}\left[1 - \frac{|Z_A^{(test)} - Z_A^{(train)}|}{Z_A^{(test)} + Z_A^{(train)}}\right]. \tag{A.1}$$

This way, the extreme significance values observed due to fluctuations or overtraining can be penalized without a high computational effort. $Z_A^{(train)}$ is evaluated on the training sample and $Z_A^{(test)}$ on an independent test sample. The search algorithm can be formalized as follows:

1. For the given initial parameters $\gamma_{initial}$ and $C_{initial}$, the iterative grid search algorithm produces an array of logarithmically spaced γ values with a step size K_t around the mid-value $\gamma_m^{(1)} = \gamma_{initial}$ such that:
$$\gamma_k^{(l)} = K_t \cdot \gamma_{k-1}^{(l)}, \quad \text{where } K_t = \tfrac{1}{2}(1 + \ln(t/2)),$$
$$k = 0, \ldots, m, \ldots, 2m = 18, \quad t = \mathrm{int}(l/4), \quad l = 1, \ldots, 20 \tag{A.2}$$
where l indicates the number of iterations, the variable t is a *focus parameter* that decreases the step size factor K_t every fourth iteration.
$\tilde{Z}_A$ is then evaluated for all of these C-γ pairs.
2. For the next step C is increased to $C^{(l+1)} = 1.5 \cdot C^{(l)}$ and $\tilde{Z}_A$ is again calculated with each value in the γ array.
3. If the maximum $\tilde{Z}_A^{(l)}$ value is at least 30% larger then the best $\tilde{Z}_A^{(l-1)}$ from the previous iteration the higher C parameter is accepted. The 30% hurdle is introduced to stabilize against fluctuations.
4. After each fourth iteration, the C-γ pair corresponding the highest significance score is taken as the new initial γ and the algorithm returns to the first step; now with a smaller step size factor K_t such that the new γ array has a tighter stepping around the new initial γ value.
5. When the number of iterations reaches the pre-defined maximum value, the algorithm stops and the γ-C pair with the maximum $\tilde{Z}_A$ in the final iteration are returned as the best hyper-parameter values.

The procedure assumes that a sufficiently small $C_{initial}$ had been chosen. In case that the found best C value is identical with the $C_{initial}$ the algorithm is restarted with a smaller value of $C_{initial}$.

Appendix B. Boosted decision trees

Boosted decision trees (BDT) are probably the most common ML classifier in experimental particle physics. We therefore compare the performance of our SVM framework to a BDT implemented with TMVA. We shortly introduce the relevant BDT concepts used in this comparison.

Decision trees. A binary decision tree [55] separates signal and background by a sequence of binary cuts (Fig. 6). The terminating branches, or leaves, correspond to a cubical separation in the multi-dimensional space of the input variables and the tree as a whole forms a complex separation boundary between the two classes. Each node of the tree is connected to two branches that are split with respect to only one of the variables $x_i^{(1)} \ldots x_i^{(n)}$ defined in (2.2). At each node the algorithm selects one of the available physical variables and searches for a best-cut value.

This process requires a suitable goodness-of-split measure and a common choice is the *Gini-impurity* index which is also used for the TMVA-BDT in this paper. For the two class case the *Gini* index is given by $g = 2p(1-p)$, where the purity p of a node is defined as the ratio of signal events over all events.

The training starts with the root node, and the tree is constructed recursively while at each split the reduction in *Gini* impurity is maximized. The splitting stops when a node falls below a predefined minimum of events.

Pruning. The constructed decision tree is sensitive to statistical fluctuation. To avoid overtraining it must be pruned to remove statistically insignificant nodes. *Cost-complexity pruning* [55] removes branches which increase the misclassification cost. The misclassification rate $R = 1 - \max(p, 1-p)$ is used as a cost estimate at each node and compared to the cost of the subtree below the node. The cost complexity is defined as $\rho = (R_{\text{node}} - R_{\text{subtree}})/(N_{st} - 1)$, where N_{st} is the number of nodes in the subtree. The node with the smallest ρ is recursively removed from the tree as long as ρ is below a certain pruning strength value ρ_0.

Boosting. A single decision tree is seldom an efficient classifier. Boosting is a powerful iterative algorithm which improves the performance of weak classifiers. The boosting of a decision tree extends this concept from one tree to a ensemble of trees. The trees are derived from the same training data by reweighting events, and are finally combined into a single classifier of considerably enhanced performance. For this paper AdaBoost [56] is used.

For the TMVA-BDT with AdaBoost we consider 5 parameters to be optimzed for best performance: `NTrees` (100…1000), `MinNodeSize` (5%…0.005%), `MaxDepth` (2…5 and in addition the default choice of TMVA-BDT), `AdaBoostBeta` (0.1…0.6) and `nCuts` (0.1…0.6 and the default choice). The Asimov significance is used as performance measure. The tuning is performed in an $n-1$ approach: one parameter is changed while the others are fixed. Starting with default settings of TMVA-BDT, we work through the listed parameters and the given parameter ranges in 5 logarithmically-spaced steps. For each parameter the two best performing values are recorded. This gives in total 32 parameter combinations from which we choose the final best performing set.[5]

References

[1] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, H. Voss, TMVA toolkit for multivariate data analysis, PoS ACAT (2007) 040 arXiv:physics/0703039.
[2] The HiggsML challenge, May to September 2014. ⟨https://higgsml.lal.in2p3.fr/⟩.
[3] HEPML 2014 Proceedings, vol. 42, 2015.
[4] Data science @ LHC 2015 Workshop, November 2015. ⟨https://indico.cern.ch/event/395374/⟩.
[5] B.E. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM Press, 1992, pp. 144–152.
[6] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.
[7] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York Inc., 1998.
[8] P. Vannerem, K. Muller, B. Scholkopf, A. Smola, S. Soldner-Rembold, Classifying LEP Data with Support Vector Algorithms, arXiv:hep-ex/9905027 [hep-ex].
[9] A. Vaiciulis, Support vector machines in analysis of top quark production, Nucl. Instrum. Methods A 502 (2003) 492–494 arXiv:hep-ex/0205069 [hep-ex].
[10] Ł. Janyst, A. Kaczmarska, T. Szymocha, M. Wolter, A. Zemła, Optimization of tau indentification in atlas experiment using multivariate tools, Comput. Sci. 9 (2008) 35–45.
[11] CDF Collaboration, T. Aaltonen, et al., Search for the standard model Higgs boson produced in association with a $W^{\pm}$ boson with 7.5 fb^{-1} integrated luminosity at CDF, Phys. Rev. D 86 (2012) 032011 ⟨http://link.aps.org/doi/10.1103/PhysRevD.86.032011⟩.
[12] F. Sforza, V. Lippi, G. Chiarelli, Rejection of multi-jet background in $p\bar{p} \rightarrow e\nu + j\bar{j}$ channel through a SVM classifier, J. Phys.: Conf. Ser. 331 (3) (2011) 032045 ⟨http://stacks.iop.org/1742-6596/331/i=3/a=032045⟩.
[13] F. Sforza, V. Lippi, Support vector machine classification on a biased training set: multi-jet background rejection at hadron colliders, Nucl. Instrum. Methods A 722 (2013) 11–19 arXiv:1407.0317 [hep-ex].
[14] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27 ⟨http://www.csie.ntu.edu.tw/~cjlin/libsvm⟩.
[15] R Core Team, R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2015. ⟨https://www.R-project.org⟩.
[16] F.e.a. Pedregosa, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830 arXiv:1201.0490.
[17] See ROOT 6.06 Release Notes, December 2015.
[18] I. Antcheva, et al., ROOT: a C++ framework for petabyte data storage, statistical analysis and visualization, Comput. Phys. Commun. 182 (2011) 1384–1385.
[19] M.O. Sahin, D. Kruecker, I.A. Melzer-Pellmann, SVM-HEP Interface, 2015. ⟨https://www.github.com/ML-hint/svm-hint⟩.
[20] B. Schölkopf, A. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond, in: Adaptive Computation and Machine Learning, MIT Press, 2002.
[21] J. Mercer, Functions of positive and negative type, and their connection with the theory of integral equations, Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci. 209 (441–458) (1909) 415–446.
[22] C.-W. Hsu, C.-C. Chang, C.-J. Lin, A Practical Guide to Support Vector Classification, ⟨https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf⟩.
[23] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, 1999, pp. 61–74.
[24] T.F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (2004) 975–1005.
[25] G. Cowan, K. Cranmer, E. Gross, O. Vitells, Asymptotic formulae for likelihood-based tests of new physics, Eur. Phys. J. C 71 (2011) 1554, arXiv:1007.1727 [physics.data-an].
[26] R.D. Cousins, J.T. Linnemann, J. Tucker, Evaluation of three methods for calculating statistical significance when incorporating a systematic uncertainty into a test of the background-only hypothesis for a Poisson process, Nucl. Instrum. Methods Phys. Res. A 595 (2008) 480–501, physics/0702156.
[27] P. Ramond, Dual theory for free fermions, Phys. Rev. D 3 (1971) 2415.
[28] Y.A. Golfand, E.P. Likhtman, Extension of the algebra of Poincaré group generators and violation of P invariance, JETP Lett. 13 (1971) 323.
[29] A. Neveu, J.H. Schwarz, Factorizable dual model of pions, Nucl. Phys. B 31 (1971) 86.
[30] D.V. Volkov, V.P. Akulov, Possible universal neutrino interaction, JETP Lett. 16 (1972) 438.
[31] J. Wess, B. Zumino, A Lagrangian model invariant under supergauge transformations, Phys. Lett. B 49 (1974) 52.
[32] J. Wess, B. Zumino, Supergauge transformations in four dimensions, Nucl. Phys. B 70 (1974) 39.
[33] P. Fayet, Supergauge invariant extension of the Higgs mechanism and a model for the electron and its neutrino, Nucl. Phys. B 90 (1975) 104.
[34] H.P. Nilles, Supersymmetry, supergravity and particle physics, Phys. Rep. 110 (1984) 1.
[35] D0 Collaboration, V.M. Abazov, et al., Search for 3- and 4-body decays of the scalar top quark in pp collisions at $\sqrt{s} = 1.8\ TeV$, Phys. Lett. B 581 (3–4) (2004) 147–155.
[36] D0 Collaboration, V.M. Abazov, et al., Search for pair production of the scalar top quark in muon+tau final states, Phys. Lett. B 710 (2012) 578–586, arXiv:1202.1978 [hep-ex].
[37] D0 Collaboration, V.M. Abazov, et al., Search for the lightest scalar top quark in events with two leptons in $p\bar{p}$ collisions at $\sqrt{s} = 1.96$ TeV, Phys. Lett. B 659 (2008) 500–508, arXiv:0707.2864 [hep-ex].
[38] CDF Collaboration, T. Aaltonen, et al., Search for the supersymmetric partner of the top quark in $p\bar{p}$ collisions at $\sqrt{(s)} = 1.96$ TeV, Phys. Rev. D 82 (2010) 092001, arXiv:1009.0266 [hep-ex].
[39] CDF Collaboration, D. Acosta, et al., Search for the supersymmetric partner of the top quark in dilepton events from $p\bar{p}$ collisions at $\sqrt{s} = 1.8$ TeV, Phys. Rev. Lett. 90 (2003) 251801, arXiv:hep-ex/0302009 [hep-ex].
[40] ATLAS Collaboration, G. Aad, et al., ATLAS Run 1 searches for direct pair production of third-generation squarks at the Large Hadron Collider, Eur. Phys. J. C 75 (10) (2015) 510, arXiv:1506.08616 [hep-ex].
[41] CMS Collaboration, S. Chatrchyan, et al., Search for top-squark pair production in the single-lepton final state in pp collisions at $\sqrt{s} = 8$ TeV, Eur. Phys. J. C 73 (12) (2013) 2677, arXiv:1308.1586 [hep-ex].
[42] M. Berggren, A. Cakir, D. Krücker, J. List, I.A. Melzer-Pellmann, B.S. Samani, C. Seitz, S. Wayand, Non-Simplified SUSY: Stau-Coannihilation at LHC and ILC, Eur. Phys. J. C 76 (2016) 183, arXiv:1508.04383[hep-ph].
[43] N. Arkani-Hamed, P. Schuster, N. Toro, J. Thaler, L.-T. Wang, et al., MARMOSET: The Path from LHC Data to the New Standard Model via On-Shell Effective Theories, 2007., arXiv:hep-ph/0703088.
[44] J. Alwall, P. Schuster, N. Toro, Simplified models for a first characterization of new physics at the LHC, Phys. Rev. D 79 (2009) 075020, arXiv:0810.3921.
[45] D. Alves, et al., Simplified models for LHC new physics searches, J. Phys. G 39 (2012) 105005, arXiv:1105.2838.
[46] M.O. Sahin, Search for Supersymmetric Top-quark Partners Using Support Vector Machines and Upgrade of the Hadron Calorimeter Front-end Readout Control System at CMS, PhD thesis, Hamburg University, Hamburg, Germany, 2016.
[47] C. Borschensky, M. Krämer, A. Kulesza, M. Mangano, S. Padhi, T. Plehn, X. Portell, Squark and gluino production cross sections in pp collisions at $\sqrt{s} = 13$, 14, 33 and 100 TeV, Eur. Phys. J. C 74 (12) (2014) 3174, arXiv:1407.5066 [hep-ph].
[48] J.M. Campbell, R.K. Ellis, MCFM for the Tevatron and the LHC, Nucl. Phys. Proc. Suppl. 205–206 (2010) 10–15, arXiv:1007.3492 [hep-ph].
[49] P.M. Nadolsky, H.-L. Lai, Q.-H. Cao, J. Huston, J. Pumplin, D. Stump, W.-K. Tung, C.P. Yuan, Implications of CTEQ global analysis for collider observables, Phys. Rev. D 78 (2008) 013004, arXiv:0802.0007 [hep-ph].
[50] T. Sjostrand, S. Mrenna, P.Z. Skands, PYTHIA 6.4 physics and manual, JHEP 0605 (2006) 026, arXiv:hep-ph/0603175 [hep-ph].
[51] S. Ovyn, X. Rouby, V. Lemaitre, DELPHES, A Framework for Fast Simulation of a Generic Collider Experiment, arXiv:0903.2225 [hep-ph].
[52] J. Anderson, et al., Snowmass energy frontier simulations, in: Community Summer Study 2013: Snowmass on the Mississippi (CSS2013) Minneapolis, MN, USA, July 29–August 6, 2013, 2013. arXiv:1309.1057 [hep-ex].
[53] M. Cacciari, G.P. Salam, G. Soyez, JHEP 04 (2008) 063 arXiv:0802.1189 [hep-ph].
[54] J.G. Yang Bai, Hsin-Chia Cheng, J. Gu, Stop the Top Background of the Stop Search, arXiv:1203.4813 [hep-ph].
[55] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, CRC press, Boca Raton, 1984.
[56] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.

[5] Configuration files are available at [19].