Usage of SVM for a Triggering Mechanism for Higgs Boson Detection

Klemen Kenda
Jožef Stefan Institute, Aritificial Intelligence Laboratory
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
klemen.kenda@ijs.si

Dunja Mladenić
Jožef Stefan Institute, Aritificial Intelligence Laboratory
Jožef Stefan International Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
dunja.mladenic@ijs.si

ABSTRACT
Real-time classification of events in high energy physics is essential to deal with huge amounts of data, produced by proton-proton collisions in ATLAS detector at Large Hadron Collider in CERN. With this work we have implemented a triggering mechanism method for saving relevant data, based on machine learning. In comparison with the state of the art machine learning methods (gradient boosting and deep neural networks) shortcomings of Support Vector Machines (SVM) have been compensated with extensive feature engineering. Method has been evaluated with special metrics (average median significance) suggested by the domain experts. Our method achieves significantly higher precision and 8% lower average median significance than the current state of the art method used at ATLAS detector (XGBoost).

Categories and Subject Descriptors
H.2.8 [**Database Applications**]: Data mining, scientific databases

General Terms
Algorithms, Measurement, Experimentation.

Keywords
Support Vector Machine, Gradient Boosting, Classification, High Energy Physics, Higgs Boson

1. INTRODUCTION
ATLAS and CMS experiments have announced discovery of the Higgs boson in 2012 [1]. Experiments have been conducted on Large Hadron Collider (LHC) in CERN in Geneva. The discovery has been succeeded by a Nobel Prize in Physics, awarded to François Englert and Peter Higgs. The existence of the particle, which gives mass to other elementary particles, has been predicted around 50 years ago [6][7][8].

Higgs boson decays almost instantly and can be observed only through its decay products. Initially the particle has been observed through $H \to \gamma\gamma$, $H \to Z^0Z^0$ and $H \to W^+W^-$ decays. These decays leave a signature that is relatively easy to interpret. The next steps required analysis of Higgs boson decay into fermion pairs: τ leptons or b quarks.

In this paper we focus on a special topology of $H \to \tau^+\tau^-$ decay [9]. Due to similarities with other decays this particular decay is very difficult to classify. Distinguishing background (events that do not belong to the $H \to \tau^+\tau^-$ decay) from signal (events that belong to Higgs boson decay) requires the use of state of the art machine learning methods.

In the past the task has often been solved with simple cut-off techniques based on statistical analysis, performed by expert users. Today advanced classification methods based on machine learning are used regularly.

State of the art methods for this type of problems include deep neural networks and gradient boosting [10][11][12]. Experiments at CERN prefer the usage of gradient boosting classifiers as they are able to evaluate large amounts of data (more than 20×10^6 events/s) [4].

The success of both methods is based on their intrinsic property of introducing non-linearity into the system. In our work we want to compare basic linear methods and Support Vector Machines (SVM) with different kernels to the state of the art models. Additionally, we want to enrich the data by intensive feature engineering.

The results of feature engineering can be used for further physical interpretation of relevant physical phenomena.

2. DATA
Dataset has been made public by the ATLAS collaboration for the Higgs Boson Machine Learning Challenge on Kaggle in 2014 [3]. It contains data from the ATLAS detector simulator (real labelled data would be impossible to obtain). The winning method from the challenge is being used in the ATLAS experiment today [4].

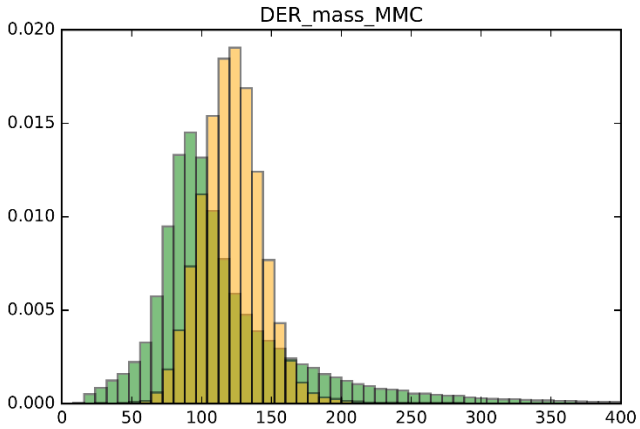

Figure 1. Distribution of signal (yellow) and background (green) according to most informative attribute DER_mass_MMC (mass of Higgs boson candidate) [5].

2.1 Data Description
Dataset consists of 250,000 instances. 85,667 represent signal, 164,333 represent background. Each instance consists of 32 attributes and 1 target variable. All the attributes are numerical (continuous), target variable is nominal (binary). 2 of the attributes

should not be used for classification purposes, as they represent id of the instance and probability of such an event happening in the experiment [4].

There are missing values in the data. 11 attributes could not always be measured due to characteristics of the detector. Distribution of the missing values is different for signal and for background.

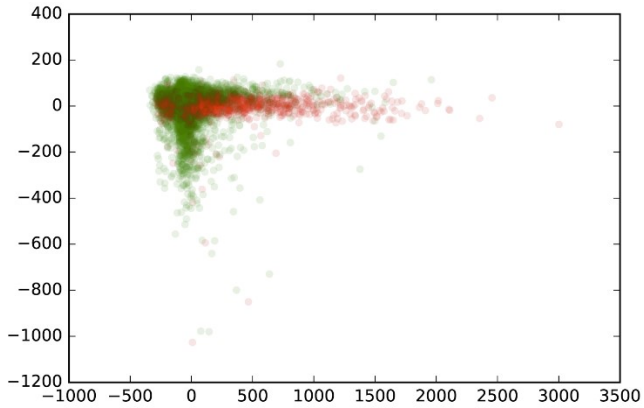

Figure 1. Plot of 1st PCA component against the 3rd. Red dots represent signal instances, green dots represent background instances [5].

The signal is limited to the events representing only one possibility for $\tau^+\tau^-$ pair decay [4].

2.2 Data Understanding

The main task of our method is to separate the signal from the background, based on the ATLAS detector measurements. As vast amounts of data (a few terabytes/day) are generated within the process it is crucial that only the relevant events are detected and stored [4].

Exploratory analysis has shown (see Figure 1) that this task can not be successfully accomplished with simple cut-off techniques based on a single attribute. Figure 2 depicting PCA components plot is a bit more promising as parts of phase space can clearly be assigned to one of the classes.

Attributes are divided into 3 groups. First group contains 18 primary attributes (measured in the detector), second group contains 12 derived attributes (relevant physical phenomena calculated from primary attributes) and 2 metadata values (weight and event id). Detailed exploratory data analysis can be found in [5].

2.3 Data Preprocessing

ATLAS detector enables good precision of all measurements, therefore expected noise in the data is very small and it can not be further filtered. Missing values have been dealt with in two different ways. Firstly – we used "replacement with average" strategy to fill in the missing data and secondly, we generated additional binary features, representing missing attribute values.

SVM expects input data to be normalized, therefore the features have been normalized with average and standard deviation values set to 1. Data transformation has been handled with Pandas library in Python.

2.4 Feature Engineering

The main task of our work has consisted of extensive feature engineering, where non-linear combinations of features were introduced to overcome the shortcomings of linear SVM in comparison with gradient boosting or deep neural networks.

We have built new features from original attributes by transforming them with some common functions like e^x, x^2, x^3, $\sqrt{x}$ and $log(x)$. Additionally we have used k-means clustering to generate an additional attribute (cluster id). All the generated feature sets are shown in Table 1.

Table 1. Attribute sets used for SVM.

Set	Description
1	Original feature set.
2	Added missing values.
3	Filtered missing values and all e^x derivatives.
4	Filtered missing values, e^x and all x^2 derivatives.
5	Filtered missing values, e^x, x^2 and all x^3 derivatives.
6	Filtered missing values, e^x, x^2, x^3 and all $\sqrt{x}$ derivatives.
7	Filtered missing values, e^x, x^2, x^3, $\sqrt{x}$ and all $log(x)$ derivatives.
8	Selection of most relevant transformations by one attribute.
9	Unfiltered set of transformations by one attribute.
10	Unfiltered set of $x_i x_j$.
11	Set of attributes by one of HiggsML winners (Tim Salimans, DNN).
12	Unfiltered set of $x_i^2 + x_j^2$.
13	Unfiltered set of $e^{x_i^2 + x_j^2}$.
14	Unfiltered set of $\sqrt{x_i^2 + x_j^2}$.
15	Unfiltered set of $(1 + x_i x_j)^2$.
16	Filtered set of transformations by 1 and 2 attributes.
17	(8) with k-means cluster id.

Filtering of the features has been done manually, with a simple cut-off technique based on feature importance as obtained from linear SVM model.

3. MACHINE LEARNING METHODS USED

Baseline experiments have been carried out with simple cut-off techniques and linear methods like logistical regression and Naïve Bayes classifier. As state of the art methods we included gradient boosting and gradient boosting adjusted for the approximate median significant metrics (see Section 3.2) [11].

We are proposing to use SVM method [12]. Linear SVM can be used for feature selection with large number of attributes. It can discover most relevant features in a large feature set.

3.1 Brief Description of SVM

In our setting we are solving a binary classification problem. Let us assume, that the classes are linearly separable in our space. In general, there are many different hyper planes that can separate the two classes. Support vector machine (SVM) method is also called maximum margin classifier. There exists only one hyper plane that maximizes margin between the two classes [12].

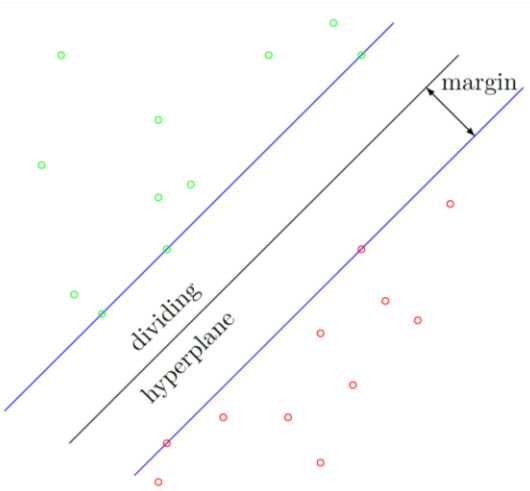

Figure 2. Maximum margin of dividing hyper plane in SVM [5].

SVM classifier is derived from maximization of the margin, which can be translated into minimization of $||w||^2$ [5][12]. As we are dealing with data sets, where classes are not separable, we need to consider a soft margin method that would take into account classification error. SVM is therefore solving minimization problem of $||w||^2 + C\sum_{i=1}^{n}\xi_i$, where ξ_i is a classification error metrics and C is a parameter that controls the influence of ξ.

3.2 Brief Description of the Evaluation Criteria

Evaluation of the results is to be done with measures derived from confusion matrix (accuracy, precision, recall, F_1). The evaluation metrics (*approximate median* significance) is defined as

$$AMS = \sqrt{2\left(s + b + b_{reg}\right)\ln\left(1 + \frac{s}{b + b_{reg}}\right) - 2s}$$

s represents sum of event probabilities of true positives (signal), b represents sum of event probabilities of true negatives (background), b_{reg} is set to 10 and represents a pre-set regularization parameter. The metrics favorizes recall before precision. In real setting this algorithm is used as a triggering mechanism for saving relevant data. Probability for a positive example in the real data is only around $p \approx 2 \times 10^{-5}$, therefore we do not want to lose many of them.

4. EVALUATION

Experiments have been carried out in Python. Data loading and cleaning has been accomplished with `Pandas` library, implementation of SVM, scaling and other methods have been taken from `scikit-learn` package. Default parameters for SVM have been used.

On our system SVM learning phase took ~1 hour. For time optimization purposes normal evaluation with training and test set has been performed. Training set consisted of 225,000 and test set of 25,000 instances.

Table 2. Evaluation of different attribute sets on SVM with linear kernel.

Attribute set	Prec.	Rec.	Acc.	F_1	AMS
1	0.665	0.548	0.749	0.600	1.999
3	**0.748**	0.655	0.805	0.698	2.526
4	**0.748**	0.654	0.805	0.698	2.528
5	0.740	0.657	0.802	0.696	2.478
6	0.743	0.683	0.809	0.712	2.547
7	0.734	0.690	0.807	0.711	2.516
8	0.732	0.670	0.802	0.700	2.482
10	0.744	0.705	0.815	0.724	**2.582**
11	0.694	0.584	0.768	0.634	2.201
12	0.744	0.705	0.815	0.724	**2.583**
13	0.744	0.709	**0.816**	**0.726**	**2.581**
14	0.744	0.705	0.815	0.724	**2.583**
15	0.744	**0.710**	**0.816**	**0.726**	2.578
16	0.740	0.684	0.809	0.711	2.553

Results from extensive feature engineering are shown in Table 2. Linar SVM performed similar to linear baseline methods (logistic regression, Naïve Bayes). AMS score was ~2.00. Best feature sets for linear SVM were (10), (12), (13) and (14). These feature sets include two-attribute transformations, e.g., $x_i x_j$. It is interesting to notice that filtered feature sets performed slightly worse. Extensive feature generation achieved almost 30% better AMS results (1.999 on basic feature set compared to 2.583).

Table 3. Evaluation of different methods and attribute sets compared to baseline and state-of-the-art methods.

Method and attribute set	Prec.	Rec.	Acc.	F_1	AMS
simple window	0.560	**0.824**	0.716	0.667	1.579
log. reg. (1)	0.668	0.535	0.749	0.594	2.015
SVM-LIN (13)	0.744	0.709	0.816	0.726	2.581
GBC (8)	0.787	0.703	0.832	0.742	2.856
SVM-r (8)	**0.791**	0.718	**0.837**	**0.752**	2.940
opt. SVM-r (8)	**0.907**	0.446	0.793	0.598	3.451
XGBoost (1)	0.665	**0.806**	0.793	0.729	**3.735**

Table 3 contains results of baseline, state-of-the-art and the proposed SVM. Best feature sets for selected methods were chosen. Baseline methods are simple window (based on cut-off technique on candidate particle mass) and logistic regression. As state of the art methods we included: gradient boosting (GBC) and current state of the art (XGBoost, gradient boosting optimized for AMS).

Proposed methods are linear SVM, SVM with RBF kernel (SVM-r) and optimized SVM with RBF kernel (opt. SVM-r).

Usage of kernels (RBF and polynomial kernels have been tested) improved AMS score for another ~15%. Because of the nature of SVM kernels in this setting 2-attribute transformations were less efficient than 1-attribute transformations. Selection of most relevant transformations by 1 attribute (set (8)) gave the best results. Method behaved better than gradient boosting classifier (GBC) on the same training set. However, methods were not optimized to maximize AMS score. The difference however suggests that the usage of SVM might be a promising way to proceed.

Finally we optimized SVM with RBF kernel for AMS score and compared it to XGBoost method, which implements gradient boosting, optimized for AMS. Optimization has been done based on threshold for SVM confidence score. Our method performs approximately ~8% worse than the state of the art. There is, however, a big difference with XGBoost. Our method yields higher precision than the other methods and still preserves very high AMS score. The proposed method also performs ~20% better than other SVM based methods reported in HiggsML Challenge [3].

5. CONCLUSION

In our work we have examined the potential of SVM for a triggering mechanism in high-energy physics domain. With extensive feature engineering we have also provided an interesting input for high energy physics experts, where most effective generated features could be analyzed through domain knowledge.

Our method achieves more than 200% better AMS score compared to cut-off techniques, based on statistical approach. Further, our methods achieves ~20% better AMS score than other SVM based methods reported by HiggsML Challenge competitors, but performs ~8% worse than current state of the art (XGBoost). There is however a significant difference between our method and state of the art. Although achieving comparable AMS score, our methods achieves much better precision. This might make SVM based methods valuable members of an ensemble method.

Beside adding SVM methods to ensembles and trying to improve state of the art, further work could be done with adapting the SVM optimization to AMS metrics. In our work features were selected based on weight-importance. Often different transformations of the same attributes have been selected. Features that could improve our models only by little have potentially been left out. This should be studied further. Optimization of SVM parameters should also be performed.

6. ACKNOWLEDGMENTS

This work was partially supported by the Slovenian Research Agency.

7. REFERENCES

[1] G. Aad et al. *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*. Physics Letters B, 716(1):1 – 29, 2012.

[2] S. Chatrchyan et al. *Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC*. Physics Letters B, 716(1):30 – 61, 2012.

[3] HiggsML challenge. https://www.kaggle.com/c/higgs-boson, 2014. [Online; access April 20, 2016].

[4] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl in D. Rousseau. *The Higgs boson machine learning challenge*. In Workshop on High-energy Physics and Machine Learning, HEPML 2014, held at NIPS 2014, Montreal, Quebec, Canada, December 8-13, 2014 [30], pages 19–55.

[5] Kenda, K., Podobnik, T., Gorišek, A. *Uporaba metod strojnega učenja pri analizi podatkov, zajetih z detektorjem ATLAS*. Diploma thesis. 2016. Faculty of Mathemathics and Physics, University of Ljubljana

[6] P. W. Higgs. *Broken symmetries, massless particlees and gauge fields*. Physics Letters, 12:132–133, September 1964.

[7] F. Englert, R. Brout. *Broken Symmetry and the Mass of Gauge Vector Mesons*. Physical Review Letters, 13:321–323, August 1964.

[8] P. W. Higgs. *Broken symmetries and the masses of gauge bosons*. Phys. Rev. Lett., 13:508–509, Oct 1964.

[9] G. Aad et al. *Evidence for the Higgs-boson Yukawa coupling to tau leptons with the ATLAS detector*. JHEP, 04:117, 2015.

[10] T. Chen, T. He. *Higgs boson discovery with boosted trees*. In HEPML 2014, held at NIPS 2014, pages 69–80.

[11] T. Chen, C. Guestrin. *XGBoost: A scalable tree boosting system*. CoRR, abs/1603.02754, 2016.

[12] Boser, B. E., Guyon, I. M., Vapnik, V. N. *A training algorithm for optimal margin classifiers*. Proceedings of the fifth annual workshop on Computational learning theory – COLT '92. p. 144.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot in E. Duchesnay. *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12:2825–2830, 2011.