PROCEEDINGS OF SCIENCE

Automated selection of particle-jet features for data analysis in High Energy Physics experiments

Andrea Di Luca,[a,b,c,*] **Francesco Maria Follega,**[a,b] **Marco Cristoforetti**[b,c] **and Roberto Iuppa**[a,b]

[a] *Dipartimento di Fisica, Università di Trento, Via Sommarive 14, 38123 Trento, Italy*
[b] *TIFPA, Via Sommarive 14, 38123 Trento, Italy*
[c] *FBK, Via Sommarive 18, 38123 Trento, Italy*

E-mail: andrea.diluca@unitn.it

We show that it is possible to reduce the size of a classification problem by automatically ranking the relative importance of available features. Variables are importance-sorted with a decision tree algorithm and correlated ones are removed after ranking. The selected features can be used as input quantities for the classification problem at hand.
We tested the method with the case of highly boosted di-jet resonances decaying to two b-quarks, to be selected against an overwhelming QCD background with a Deep Neural network. We make it explicit the relation between different importance rankings obtained with different algorithms. We also show how the signal-to-background ratio changes, varying the number of features to feed the Neural Network with.

40th International Conference on High Energy physics - ICHEP2020
July 28 - August 6, 2020
Prague, Czech Republic (virtual meeting)

*Speaker

PoS(ICHEP2020)907

1. Introduction

In current and future high-energy physics experiments, the sensitivity of selection-based analysis will increasingly depend on the choice of the set of high-level features determined for each collision. The complexity of event reconstruction algorithms escalated in the last decade and thousands of parameters, often correlated to each other, are available for the analysts. An example is the set of observables associated with tracks, both of instrumental and algorithmic origin. Training multivariate algorithms with all available features is often impossible, due to lack of calibration or computing power limitations. In addition, it is known that increasing the dimensionality of a classification problem, not always increases the performance of the model. This issue is often referred to as the "*curse of dimensionality*" [1]. Moreover, the higher is the complexity of the problem we want the model to learn, the longer the training will take. Therefore, it is always desirable to reduce the number of input observables. Nevertheless, such reduction is often pursued by trial and error, without a systematic approach guaranteeing that that particular set of observables is the optimal one available at hand. There is a need for a robust method, allowing to control the selection of features, to account for their correlation, as well as to evaluate gains and losses in sensitivity when uncertainties are taken into consideration.

We propose here a possible method based on ranking the importance of observables, i.e. their weight on the predictive power of the model, and estimating the performance of the model versus the number of considered features.

2. Automated feature selection

The automated feature selection procedure can be divided into three main steps.

1. **Preliminary steps**: selections are applied to the raw dataset and the main starting variables are selected. At this stage, any prepossessing can be applied to the dataset.

2. **Feature ranking extraction**: after having chosen a machine learning algorithm capable to rank features, different models are trained using cross validation. The final ranking is produced by ordering the variables by the mean value of all the obtained rankings. This procedure furnishes an estimate of the ranking position uncertainty.

3. **Correlated variable removal**: it is expected to have highly correlated variables close in positions in the feature ranking. Removing correlated variables improves the significance and prevents introducing undesired noise. A threshold on the correlation value is defined and correlated variables are removed following the obtained feature ranking.

After this procedure, the data-set is ready to be used to train the final model. In the following, we show an example of the application of the automated feature selection for the specific case of tagging events containing a boosted Higgs boson decaying to two b quark in a proton-proton (pp) collision experiment.

3. Benchmark application: Boosted $H \to bb$ tagging

The problem of tagging events containing a boosted Higgs boson decaying to two b-quarks is chosen as a benchmark for the feature selection procedure. This represents an appealing channel to study Higgs boson properties since it accounts for 58% of the total Higgs boson decays [2]. However, in pp collision experiments, it is not easy to distinguish these events against the huge irreducible background coming from QCD multi-jet production [3]. In this regime, the Higgs decay products are very collimated and it is not always possible to resolve the di-jet structure [4]. Therefore a single large and massive jet containing both the b quark originated jets is reconstructed. Moreover it is expected to observe larger deviations from Standard Model prediction in the boosted regime [5]. Given the importance of the process, we developed an $H \to bb$ tagger for pp collision experiments based on a deep neural network to identify jets that contain both the b-quarks from boosted Higgs decay.

3.1 Simulated data and object reconstruction

The dataset is produced using a fast and reliable framework developed by combining tools such as Pyhtia8 [6], to generate high-energy physics events, Delphes [7], to mimic the detector response and RAVE [8] for secondary vertex reconstruction. We reconstruct large radius anti-kT jets [9] (large-R jets) with $R = 1$ and variable radius track jets [10] with $R_{\mathrm{MAX}} = 0.4$, $R_{\mathrm{MIN}} = 0.02$ and $\rho = 30$. For the large-R jets, we defined kinematic variables, plus 11 jet substructure variables. For the variable R track jets we defined kinematic variables plus the b-tagging information and 5 variables connected to the secondary vertex. We selected large-R jets with $p_T > 450\mathrm{Gev}/c^2$ and $\eta < 2$. Then we look for the 2 highest p_T track jets contained in a selected large-R jet. We produced the simulation on a Microsoft Azure Virtual Machine with 6 CPUs core and 56 GB RAM and 1 NVIDIA K80 GPU board.

3.2 Feature ranking

Given the generated data, we move to rank features. We present a comparison between the results obtained with two different tree based algorithms: CATBOOST [11] and Random Forest [12]. To extract the ranking, we performed a 10-fold cross validation training. Then, we ordered the variables by the median value of the obtained importance. Random Forest orders features by measuring Gini importance [13], while CATBOOST measures how much on average the prediction changes if the feature value changes: the bigger the value of the importance the bigger on average is the change in the predicted value. We compared the ranking obtained with the two algorithms by producing the scatter plot in Fig 1. Clusters of variables with compatible feature importance are highlighted. On the top right of the plot it is possible to distinguish the red cluster that corresponds to the most important variables. On the bottom left there is a green cluster, containing the least relevant variables for both the algorithms. The presence of these clusters is important because it shows how similar are the rankings obtained with these two different algorithms. In the future we plan to compare these results with other non-tree based algorithms.

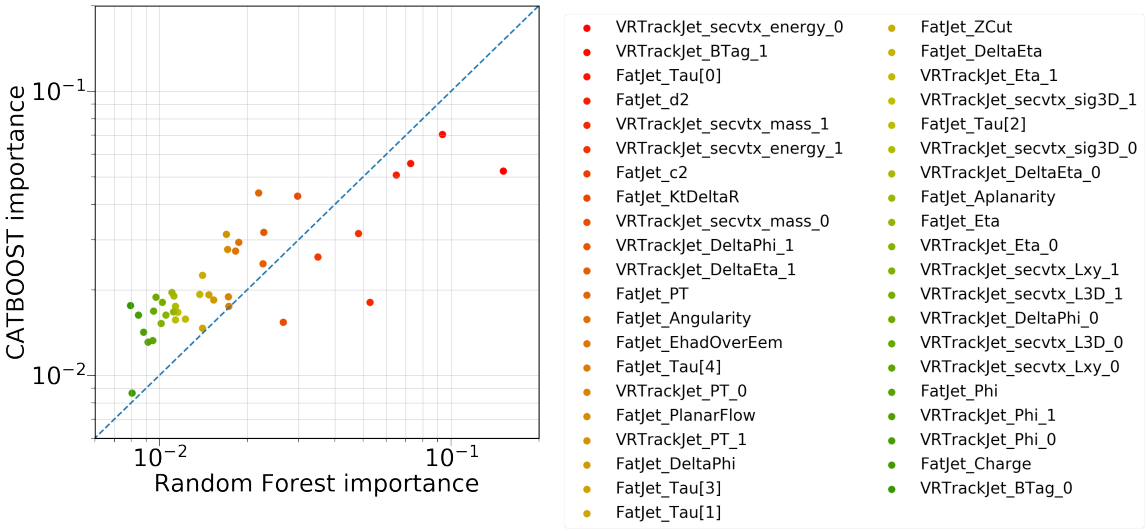

Figure 1: Comparison between feature importance obtained with Random Forest and CATBOOST. On x axis we have RF importance while on y axis we have the CATBOOST one.

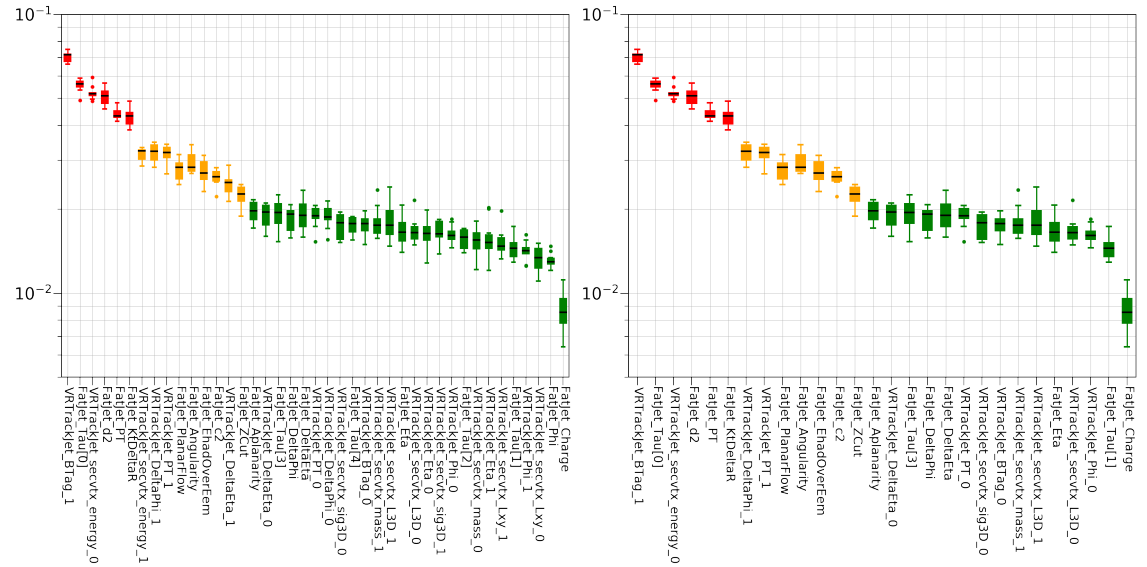

Figure 2: CATBOOST feature ranking before (left) and after (right) correlated variables removal. Red points has importance median greater than 0.04, orange points between 0.04 and 0.02, while green points below 0.02.

3.3 Correlated variable removal

After having fixed the ranking, we proceed with the removal of correlated variables. We computed the feature correlation matrix by evaluating the Pearson correlation coefficients [14] and we removed correlated variables following the order defined by the feature ranking. It is expected that correlated variables occupy close position in the feature ranking. Indeed, Fig. 2 shows how, after removing correlated variables, the width of the plateaus in the feature ranking plot, that corresponds to features that share common relevance, are reduced in width.

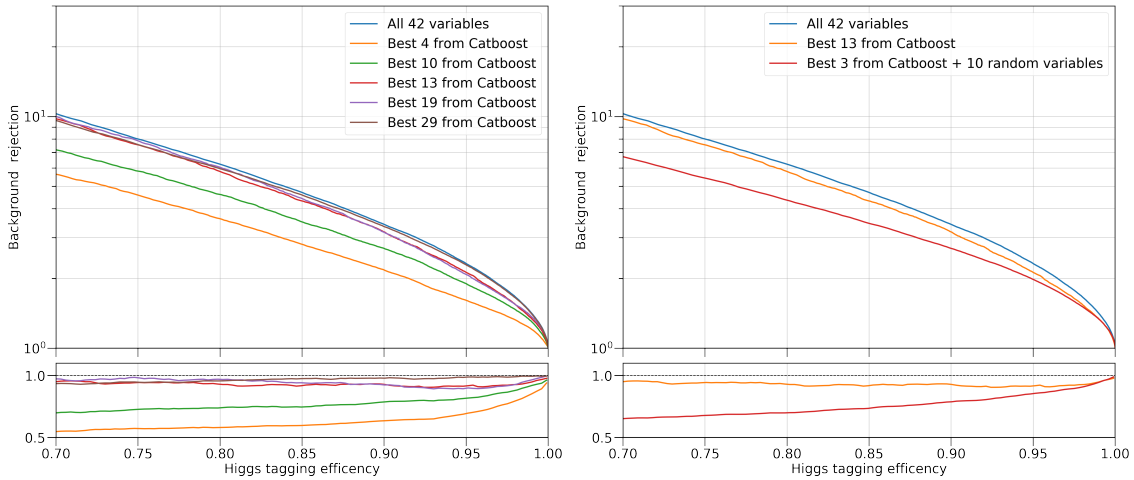

Figure 3: Background rejection rate versus Higgs tagging efficiency with large-R jets. (left) Comparison between different models trained adding variables as ranked by CATBOOST algorithm. (right) Comparison with a model trained using the best 3 ranked variable plus 10 random variables is shown.

3.4 Training of the Neural network classifier

After all the steps above described, we move to train the final model. We choose a deep neural network with fully connected layers. We built 6 hidden layers with 128 nodes per layer and SELU activation function [15]. The output of the network will be a number between 0 and 1 so that it will be possible to choose a working point for Higgs tagging efficiency. The framework used to develop and train the classifier is Pytorch [16]. We optimized the neural network architecture, i.e. number of hidden layers and nodes, when using all the features. Then, we changed the input layer size according to the number of selected variables while keeping the hidden network architecture constant.

3.5 Results

Fig. 3 shows the performances of different models trained by varying the number of input features . Performances are evaluated by looking at the Higgs tagging efficiency against background rejection. Each point on the plot is obtained by varying the threshold on the output of the classifier. The blue line corresponds to the model trained using all the 42 available variables. The plot on the right shows how, adding variables to the model, improves the background rejection rate at a fixed Higgs tagging working point. The improvements reduce in size after having reached variable 13 (red line), which has performances compatible with the complete model within 6%. In order to assess the relevance of the ordering, we also produced a model trained using the first 3 non correlated variables from the feature ranking and we added 10 random variables from all the remaining ones. The performances of this model are shown in the right side plot of Fig. 3 (red line). This model is performing 10% worse than the model trained with the first 13 variables (orange line) at almost every working point.

4. Conclusion

We showed a possible approach for feature selection, containing the most relevant non-correlated features as ranked by a decision tree algorithm. We chose as a benchmark model the tagging of large-R jets that contain both the b-quarks coming from the Higgs boson decay, against the QCD jets originated in pp collisions. We showed that two different rankings obtained using different algorithms, i.e. CATBOOST and Random Forest, share common features as the most and least relevant features. An estimate of the stability of the ranking can be obtained by running a n-fold (in this work $n = 10$) cross validation training. We showed that the feature ranking obtained using the decision tree-based algorithm can be consistently applied to a neural network-based classifier. Indeed, using the most important features allows to perform as efficiently as the complete model within a few percent: little or no improvement is obtained by adding the lowest ranked features. The automated feature selection is shown in action starting from 42 variables, but it is easily scalable to a larger number of variables. Other ranking algorithms are currently under test for a generalization of the method.

References

[1] R. Bellman, R. Corporation and K.M.R. Collection, *Dynamic Programming*, Rand Corporation research study, Princeton University Press (1957).

[2] LHC Higgs Cross Section Working Group collaboration, *Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector*, 1610.07922.

[3] J.M. Campbell, J. Huston and W. Stirling, *Hard Interactions of Quarks and Gluons: A Primer for LHC Physics*, *Rept. Prog. Phys.* **70** (2007) 89 [hep-ph/0611148].

[4] ATLAS Collaboration collaboration, *Performance of large-R jets and jet substructure reconstruction with the ATLAS detector*, Tech. Rep. ATLAS-CONF-2012-065, CERN, Geneva (Jul, 2012).

[5] M. Grazzini, A. Ilnicka, M. Spira and M. Wiesemann, *Modeling BSM effects on the Higgs transverse-momentum spectrum in an EFT approach*, *JHEP* **03** (2017) 115 [1612.00283].

[6] T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten et al., *An introduction to PYTHIA 8.2*, *Comput. Phys. Commun.* **191** (2015) 159 [1410.3012].

[7] DELPHES 3 collaboration, *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057 [1307.6346].

[8] W. Waltenberger and F. Moser, *Rave - an open, extensible, detector-independent toolkit for reconstruction of interaction vertices*, in *2006 IEEE Nuclear Science Symposium Conference Record*, vol. 1, pp. 104–109, 2006.

[9] M. Cacciari, G.P. Salam and G. Soyez, *The anti-ktjet clustering algorithm*, *Journal of High Energy Physics* **2008** (2008) 063.

[10] D. Krohn, J. Thaler and L.-T. Wang, *Jets with Variable R*, *JHEP* **06** (2009) 059 [0903.0392].

[11] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush and A. Gulin, *Catboost: unbiased boosting with categorical features*, .

[12] L. Breiman, *Random forests*, *Mach. Learn.* **45** (2001) 5–32.

[13] G. Louppe, L. Wehenkel, A. Sutera and P. Geurts, *Understanding variable importances in forests of randomized trees*, in *Advances in neural information processing systems*, pp. 431–439, 2013.

[14] *Pearson's correlation coefficient*, in *Encyclopedia of Public Health*, W. Kirch, ed., (Dordrecht), pp. 1090–1091, Springer Netherlands (2008), DOI.

[15] G. Klambauer, T. Unterthiner, A. Mayr and S. Hochreiter, *Self-normalizing neural networks. arxiv 2017*, *arXiv preprint arXiv:1706.02515* .

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., *Pytorch: An imperative style, high-performance deep learning library*, .