Automated, Scalable Signal Region Identification with Applications to the ATLAS $W^{\pm}W^{\pm}W^{\mp}$ Analysis

Nicolò Foppiani Jonah Philion Baojia Tong

Abstract

The aim of this work is to use machine learning to maximize the capability of particle physics experiments to discover new phenomena.

In a typical particle physics experiment, detector data from trillions of particle collisions are recorded. During each collision, various physics processes randomly occur, each with some probability. The time steps are data points, and the particles measured by the detector are the features of the data points.

To discover a new phenomenon, physics processes are divided into two categories: the known and already discovered *background*, and a new, undiscovered *signal*. All events are assumed to be generated from either a background or signal process. Discovery is claimed when the distribution of observed data points is incompatible with the background expectation.

Our approach is to use machine learning techniques to identify the *signal region* in the multi-dimensional space of detector features. The signal region is the volume of parameter space with the largest significance of signal with respect to the background. Instead of enforcing that the signal region be rectangular as is traditionally done in particle physics experiments, we leverage the output of contemporary classification algorithms to identify a more flexible signal region.

The approach is tested on simulations of $W^{\pm}W^{\pm}W^{\mp}$ production in the presence of a WZ background at the ATLAS experiment. The statistical models improve expected significance over rectangular cuts by 43%. Physics-driven interpretations of the classifiers are discussed. Code for this paper is available here

1. Introduction

Given a dataset from a particle collider, particle physicists would like to determine the phenomena that generated the dataset. A collision - known as an *event* - is a data-point in the dataset. Events are assumed to be independent of each other because they represent distinct collisions. Each event has a label identifying the physical process that generated the event and features measured by the detectors. In a real dataset from the ATLAS detector, the labels must be inferred.

Synthetic datasets also come equipped with a set of weights. Weights are computed in simulation such that the synthetic dataset has the same proportion of signal and background events as real datasets collected by the experiment. All histograms, ROC curves, and significance are computed using these weights instead of raw counts.

Typically, in a real dataset collected by the experiment, a mixture of background and signal data-points are collected. The number of signal and background data-points follow the Poisson distribution. If the number of expected signal events is greater than the statistical fluctuations of the background, the signal process is verified. This is formally quantified using the *significance*, which is defined in terms of a statistical test.

The problem is to identify the region in the feature space with the largest possible significance. In particle physics, identification is often performed by choosing hyper-rectangles in the feature space. The choice of the hyper-rectangle is often guided by the physical understanding of the processes and of the meaning of the features that are collected. However, this procedure cannot a efficiently select a non-linear region in the feature space. It is neither automated nor scalable. This work provides a procedure based on machine learning to select the optimal non-linear signal region to compute the significance. The problem has been factored into two steps. In the first step, we use machine learning to determine a representation of the parameter space which separates the signal events from the background events. In the second step, the region with the highest significance is built with the points of higher signal to background ratio.

This method has been tested on a dataset provided by the ATLAS experiment, affiliated with the $W^{\pm}W^{\pm}W^{\mp}$ analysis. For this analysis a previous study using a hyper-

rectangular signal region is present. We define the result from that paper to be the baseline for this project. Three classification algorithms have been studied, namely logistic regression (one-layer neural network), deep neural networks, and boosted decision trees. The result of our algorithm is compared with the baseline, both in terms of classifier performance and in terms of significance, showing a good improvement with respect to the baseline.

The power of these adaptive basis features are evaluated against features which the Standard Model suggests should be strong discriminators for WWW production analysis.

2. Background

In particle physics, discovering a new process is an important step. A discovery is performed on a statistical basis, in terms of a statistical test. This means to understand if the number of expected signal data-points is compatible with the statistical fluctuations of the number of expected background events. The output of the statistical test is called *significance*: this is a well established metric in physics, and its optimization represents the final goal of this work. A detailed description about significance and theoretical way to optimize is given in reference (Punzi, 2003). However, for the purpose of this study, a summary of the required knowledge is following. Given a certain region in the feature space space delimited by a non-linear contour, it is possible to estimate the number of expected events belonging to that region which are labeled as signal (S) and the number of events labeled as background (B). The problem is to understand if the signal is significant, or in other words, if the signal is likely to be a statistical fluctuation of the background. To quantify this significance, usually a statistical test is performed. In the hypothesis in which there is no signal, the number of events in the selected region is drawn from a Poisson with mean B.

$$n \sim Pois(\mu = B)$$

In case that B is large, so that the distribution of n can be approximated with a normal distribution:

$$n \sim \mathcal{N}(\mu = B, \sigma^2 = B)$$

Now it is possible to quantify the expected significance. Since n is drawn from this distribution

$$z = \frac{n - B}{\sqrt{B}} \sim \mathcal{N}(0, 1)$$

If the signal is present, we expect $n = S + B$, thus,

$$z = \frac{S}{\sqrt{B}}.$$

If $z \simeq 0$, the expected number of events of signal S is compatible to be a statistical fluctuation of the background B. On the contrary, if z is large, it is unlikely for the background to fake the signal: thus the experiment will be able to observe and discover the signal process. Usually in particle physics a significance of at least 3 is required for making an observation, and $z > 5$ to claim a discovery.

It is worth noting the importance of the p-value of the distribution of the z variable. In fact, it is possible to compute the significance even in the case of small number of background expected events B (Cowan, 2016), and the result can be translate into a z variable, equivalent in meaning to the one before, although not Gaussian distributed:

$$z = \sqrt{2\left((S+B)\log\left(1+\frac{S}{B}\right) - s\right)}$$

This equation is derived from the Poisson PDF and makes no assumption of large n. This forms our final evaluation metric.

Generally speaking, particle physics experiments collect big dataset (several millions of data-points) and produce even larger amount of synthetic data. These dataset contains several simple features, that corresponds to the properties of all the particles produced during a collision, which are measured in the detector. These feature can be discrete and continuous. The number of particles produced in a collision can be very large, and usually only the most important ones are considered. Additionally, the large amount of simple features gives the possibility to build constructed features, usually physically driven. The main challenge in using these datasets to observe and discover new physical processes is represented by the small signal to background ratio. Typical signal of interests are very tiny, and the background can be even 10^5 larger.

In this specific project, a synthetic dataset from the ATLAS experiment has been used. It is related to the analysis for the observation of the $W^{\pm}W^{\pm}W^{\mp}$ process. The ATLAS detector is one of the two main experiments at the Large Hadron Collider (LHC), currently operating at CERN (Geneva, Switzerland). The ATLAS collaboration has previously published an analysis (Aaboud et al., 2017) on this process in 2016. This study was done in the traditional way of selecting a signal region using an hyper-rectangular region. This optimization led to a remarkable result, yet not enough to observe the process of interest. An important point of this previous work was the splitting of the data-points into three different categories, labeled by the *SF0S* variable, which is chosen by a physics insight on the process of interest. In this work it has been decided to keep this splitting, in order to make an easier comparison with respect to the previous work.

3. Related Work

Several theoretical work has been written about the significance and its optimization, yet most studies with experimental data relies on the technique which identifies the signal region with an hyper-rectangular border. This is obtained by looking at the distribution of each feature and by the deciding, feature by feature, which region contains more signal. The approach discussed in this work differs from the previous one because it allows the signal region to be non linear, and it construct it with machine learning techniques, in terms of the output of a classifier algorithm.

For the specific dataset which has been used for this work, the paper relative to the previous analysis is available (Aaboud et al., 2017). It is based on the previous approach with a manual optimization, but leading with a remarkable result. The result of this previous work, which has been reproduced for the sake of comparison, makes up the baseline for this project.

The two methods will be compared into two steps, which resemble the way the problem has been decomposed to. First of all, the classification algorithms are compared with the baseline in terms of true and false positive rates. Then the result will be further compared in terms of the significance of the signal.

4. Model

The definition of the problem is to define the region of the feature space in which the significance, z, is maximum. This is achieved by dividing the problem into two steps: the first one requires the introduction of a classification problem to separate signal and background.
To reach this goal the problem is modeled as a logistic regression: we model the probability of the labels y (signal or background) given the features x.

$$y \sim Ber(f(x)) \tag{1}$$

where $f(x)$ is any function $\mathbb{R}^n \to [0, 1]$.

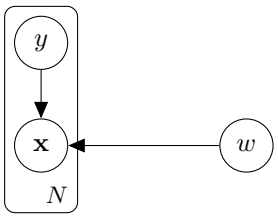

Figure 1. Directed graphical model for the logistic regression: $\mathbf{x}$ represents the vector of features, y is the class label, and w represents the parameters of the architecture.

The classifier is used to predict the probability for a certain event to be signal for each point of the feature space. The region which maximize the significance is then chosen optimizing a selection on this variable. It is expected that a region with abundant signal with respect to the background is more likely to deliver a large significance.

These models have several parameters, depending of the chosen architecture, that are estimated with a supervised learning approach.

Three different architectures have been studied.

- Simple logistic regression (one layer neural network): this architecture has one weight for each input feature plus a collective bias.
- Deep neural networks
- Boosted Decision trees: the XGBoost library (Chen & Guestrin, 2016) has been used in this context.

The parameters of the neural networks are weights and biases for each neuron. For the case of decision trees, the parameters are the values of the cuts on the various variables in each step of the tree.

5. Training

The training is performed in a supervised way, using MLE inference. Assuming equation 1, the loss function is the well known binary cross-entropy:

$$L = \sum_i w_i \left(y_i \log(f(x_i)) + (1 - y_i) \log(1 - f(x_i)) \right)$$

The training is done using backpropagation through the network nodes for the two cases of logistic regression and deep neural network. For the boosted decision tree, it has been decided to rely on the black-box algorithm of XGBoost.

6. Methods

The dataset consists of 570k events of synthetic data, which are simulated with a Monte Carlo technique using the ATLAS software. About one third of the data-points has label *signal*, whereas the rest is labeled as *background*. The dataset has been divided in training, validation, and test sets, containing about 70%, 15%, and 15% of the data-points, respectively. The dataset is pre-processed in order to transform all the discrete features in one-hot variables and to standardize all the continuous variables, by subtracting the mean and rescaling to unitary variance. The significance is further evaluated in the three independent SFOS regions. It is noteworthy that, if computing the significance selecting as region the whole feature space, the results would be 0.46, no significant at all.

There are 21 *simple features* for each data-point, plus the label and the weight. These features are related to the quantities measured experimentally related to the various particles produced in each event. On top of these, 24 additional *constructed features* have been computed. These features are computed by combining the simple features in physically meaningful variables, that are known to carry information about the process. This computation should improve the performance of the adaptive basis in the classification. Moreover, a dedicated discussion to understand if the ML methods could recover some of these higher level features, in case those are not directly in the training set, has been carried out in the following. In the next, the words *simple* and *constructed* will be used to refer and distinguish the two different kind of features.

The training is optimized differently for the different architectures. In the case of the neural networks (one layer and three layers) the PyTorch library has been used. The training is performed in batches of 200 data-points, run for 10 epochs. At each epoch the dataset is reshuffled. The weights and the biases are initialized randomly. The optimization is performed with Adam (Kingma & Ba, 2014), set with learning rate = 10^{-3} and weight-decay = 10^{-5}. There parameters are found to be optimal for a reliable and fast convergence. Concerning the boosted decision tree, the training parameters are chosen to be learning rate of 0.1, number of estimators of 100, maximum depth of 5, minimum child weight of 1. Both the training of the neural network and of the boosted decision tree last for about 5-10 minutes on a laptop with a Intel Core i5 cpu.

7. Results

The first important result is the improvement in the performance with respect to the baseline. The various classifiers are compared by looking at the ROC curves, and quantifying the area underlying it, as shown in figure 2.

This plot shows that there is a significant difference between the various architectures. The XGBoost classifier (green line) works much better than the one-layer neural network (red line). The 4-layers neural network (blue line) works better than XGBoost, showing a very good improvement with tuning the parameters. The baseline (black dot) compares well with the two most powerful architectures, although the machine learning algorithms work better than the baseline.

The second important result is the improvement with respect of the baseline in terms of the significance. The plot 3 shows the result, for the three different physical regions determined by the value of $SFOS$. This result is computed using the test set and the deep neural network architecture.

For each value of $SFOS$, an optimization study of the significance versus the classifier output is performed. Defining the classifier output as θ, the significance is computed considering all the event with $\theta \leq \bar{\theta}$ for different values of $\bar{\theta}$ between 0 and 1. Then the optimal $\bar{\theta}$ is computed. This requirement defines a non-linear boundary in the feature space, in which the significance is computed. In all the three regions the baseline is surpassed by about 25/30%,

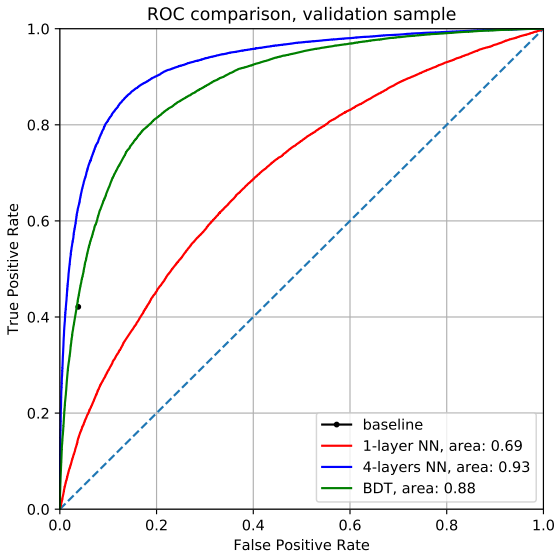

Figure 2. ROC curves on the validation sample. The x-axis is the false positive rate, the y-axis is the true positive rate. The baseline is shown as a dot; the logistic regression is shown as the red curve, the BDT is shown as the green curve and the Neutral Network is shown as the blue curve.

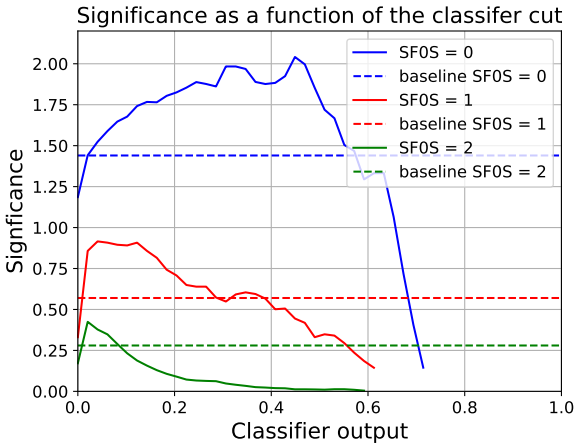

Figure 3. Significance as a function of the region determined by the selection on the classifier cut. The study is produced for the three different regions determined by the value of the variable *SF0S*.

showing a remarkable improvement with respect to the previous technique. The result is summarized in table 1.

Method	SF0S=0	SF0S=1	SF0S=2	combined
Baseline	1.44	0.57	0.28	1.57
DNN	2.05	0.85	0.39	2.25

Table 1. Summary of the result of the significance optimization with respect to the baseline. The result is shown for the three different regions identified by the variable *SF0S*. The column combined indicates the significance obtained by combining the three different regions. The improvement of about 43% is a remarkable result of this work.

8. Discussion

The machine learning method shows improvements relative to the baseline analysis, as expected. It is therefore important to further understand the output of the classification algorithm.

For this purpose, the XGBoost algorithm has been chosen, since it has been found easier to investigate the output of the decision tree with respect to the neural network, and because of the simple built-in functions of the library. Figure 8 shows the ranking of the input features. The features are ranked by XGBoost by summing up how many times each feature is split on. It is worth noting that most of the high-ranking features are actually part of the constructed features, which has the result of the feature engineering performed on top of the simple ones.

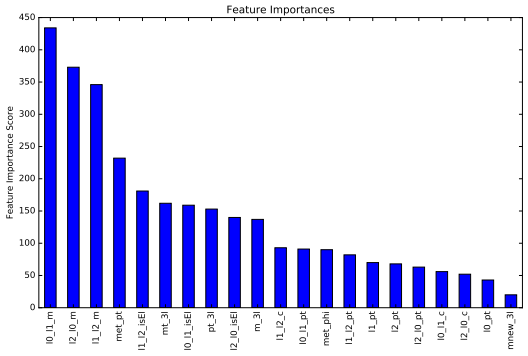

Figure 4. Ranking of features performed with XGBoost.

To further evaluate the high-ranking features, those constructed features are removed from the training pool, indicated with $l_a l_b$ labels. A separate XGBoost classifier is trained using only the simple features. In principle, if the XGBoost adaptive basis recovers all the non-linear transformation which has been done to produce the constructed features, the test output should be similar in the two cases. However, the performance results to be worse than the baseline XGBoost shown before. Specifically, the area under the ROC curve results to be 0.78 instead of 0.88, which is a significant difference.

To explore how much XGBoost has exploited the non-linear correlations among the features, the distribution of one of the constructed feature is shown versus the classifier output, for the two cases trained with and without constructed features. This comparison is shown in Figure 8. The feature shown in the plot is the highest ranked

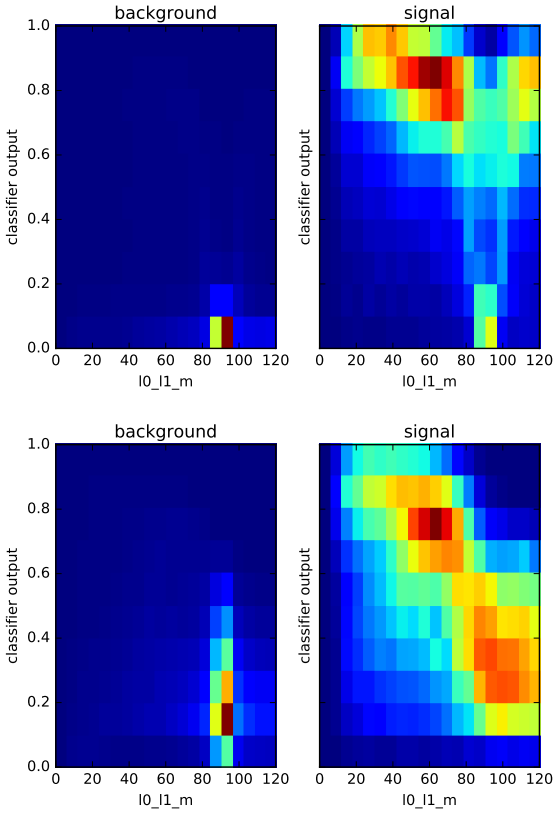

Figure 5. Classifier output as a function of $l_0 - l_1$ mass. On the x-axis one of the constructed feature is shown, whereas the y-axis refers to the output from the XGBoost classifier. The color-map shows to the number of events. The left column is for background, the right column is for signal. The top row is for the XGBoost classifier trained with all the features, while the bottom row is for the XGBoost classifier trained only with the simple features.

variable in the full-feature-trained XGBoost classifier. The distribution shows a great separation power between signal and background. It is noteworthy that the classifier trained only with simple features produces an output which favors a similar region of space as the classifier trained with all the constructed features. In other words, the classifier exploited part but not all of the correlation in the low level input variables. This means that more work needs to be done in tuning the training parameters, and in further exploring

the architectures for constructing better adaptive basis.

9. Conclusion

To summarize, this work has successfully shown the power of the application of machine learning techniques to the optimization of the signal region for a discovery of a new process in particle physics. The problem has been decomposed into two steps. First, classification algorithms are used to distinguish signal and background. For this step, three different architecture has been tested: one-layer neural network, four-layers neural network, and XGBoost decision trees. Subsequently, the signal region has been built by interpreting the classifier output as the signal to background ratio for each point of the feature space, and by optimizing the selection on this variable.

The project has been tested on a synthetic dataset produced by the ATLAS experiment, related to the $W^{\pm}W^{\pm}W^{\mp}$ analysis. Features are either reconstructed particle characteristics, which are classified as *simple* features, or higher level combination of the simple ones, on a basis of a physical motivation. The current methods from Neural Network or Boosted Decision Trees both outperform the previous baseline cut-based in terms of signal separation. The optimization of the signal region definition allowed eventually to obtain a total significance 43% larger than the baseline. Additionally, a comparison between the architecture trained with only simple features, or with the addition of constructed features, has been performed. This analysis has shown that the adaptive basis is able to partially the correlation among the variables to reproduce the constructed variables, although only partially.

Some future directions of this work involve adding more background processes, and refining the network structure.

Specifically, at the moment only one major background is considered in the dataset, and since different physics processes generate different distribution of the features, it is expected that the current architectures, trained with only one background process, will not perform in the same way with more background. More detailed studies need to follow.

Moreover, one ultimate goal would be to build a software package that could take input as signal and backgrounds with raw input variables, and return a list of higher order **re-combined** variables that will distinguish the signal and background. The work which has been done in understanding the capabilities of the adaptive basis show that more effort is needed in order to tune properly the architecture and the training hyper-parameters. This method could then be widely applied not only in high energy physics research, but in general on all kinds of real world problems. Due to the complexity, this goal may be hard to achieve by the end of this year but will worth continued exploration based on the result of this project.

References

Aaboud, Morad et al. Search for triboson $W^{\pm}W^{\pm}W^{\mp}$ production in pp collisions at $\sqrt{s} = 8\ TeV$ with the ATLAS detector. *Eur. Phys. J.*, C77(3):141, 2017. doi: 10.1140/epjc/s10052-017-4692-1.

Chen, T. and Guestrin, C. XGBoost: A Scalable Tree Boosting System. *ArXiv e-prints*, March 2016.

Cowan, Glen. Review of Statistics for Particle Physics. *Chin. Phys.*, C40(10):100001, 2016. doi: 10.1088/1674-1137/40/10/100001.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Punzi, Giovanni. Sensitivity of searches for new signals and its optimization. *eConf*, C030908:MODT002, 2003. [,79(2003)].