

# A Comparison of Machine Learning Techniques for Classification of the Higgs Boson

Jared Moskowitz & Julia Rowe  
(Dated: 8 May 2015)

## I. INTRODUCTION

One of the integral tasks of particle physics experiments is identifying elementary particles produced in high energy collisions. This can be especially problematic for the Higgs boson, given that the primary observed decay mode,  $W$ - $W$ +, results in the production of quark jets, which are difficult to distinguish from the background. Machine learning techniques are used to distinguish the signal from the background. The techniques depend on the features as defined by the standard model of the particles in question. The speed and the accuracy of the algorithm is of utmost importance as the data can be very large. The purpose of this analysis is to compare machine learning algorithms, looking at both the accuracy and how they scale with the number of features.

## II. METHODS

Various algorithms from the Waikato Environment for Knowledge Analysis (WEKA) suite of machine learning software were tested under the same conditions. The machines were trained using Monte Carlo data and tested on real data from ATLAS. The Monte Carlo data was sorted into runs with weights creating a sample that matches that of real data. Data was separated into four groups: even runs with nine features, odd runs with nine features, even runs with thirty-one features, and odd runs with thirty-one features. Predictive machine models were built from each of these groups and were used to test the opposite group of runs with the same number of features.

### A. The Algorithms

#### 1. Decision Trees

Decision trees creates a flowchart-like structure consisting of three types of nodes. The most important of these nodes are the decision nodes which act as tests or comparisons of a feature of an event. The branches represent the outcome of the test, which continues down until an end node or leaf is reached, representing a classification.

#### 2. Random Forest

Random forests are made up of multiple decision trees using random feature selection and bootstrap aggrega-

tion to correct for the common problem of decision tree overfitting. This algorithm creates multiple decision trees by selecting a smaller subset of the features and building trees. Our random forests created ten trees. For runs with nine features each tree selected four of these features and for runs with thirty-one features six features were selected.

#### 3. Naïve Bayes

This algorithm creates simple probabilistic classifiers by applying Bayes' theorem under the assumption of statistical independence between the features. Although highly simplified, the Naïve Bayes algorithm has been successful in highly complex situations. It is also highly scalable, making it an efficient algorithm.

#### 4. Support Vector Machine

Support vector machines builds a hyperplane that best separate the two categories of data. This hyperplane is built from support vectors from the training data. An advantage of SVM is it uses a fixed number of support vectors from the sample, avoiding overfitting. Often the data is projected into higher dimensional feature space, if the set is not separable in the given dimension. The hyperplane is defined the set of points  $\mathbf{x}$  which satisfy the equation:

$$\mathbf{w} \cdot \mathbf{x} - b = 0 \quad (1)$$

where  $\mathbf{w}$  is the vector normal to the hyperplane. Anything above the hyperplane should have the label of 1 and anything below -1. For a nonlinear problem, to avoid having to go into higher dimensions, a kernel trick can be applied replacing the inner product of the basis functions with a kernel function. The kernel is used to map the problem to a new space by applying a nonlinear transformation. Data was tested without a kernel and with a gaussian (radial) kernel.

### III. RESULTS

Algorithm	9 Even	9 Odd	31 Even	31 Odd
Decision Tree	10.056s	8.605s	26.672s	29.490s
Random Forest	21.346s	23.487s	37.249s	38.542s
Naïve Bayes	1.381s	1.687s	2.989s	3.099s
SVM - linear	17.110s	17.572s	86.40s	82.086s
SVM - radial				

Algorithm	9 Even	9 Odd	31 Even	31 Odd
Decision Tree	88.41 %	87.98%	87.70%	87.75%
Random Forest	88.72%	88.57%	88.82%	88.53%
Naïve Bayes	87.52%	87.79%	84.94%	84.97%
SVM - linear	87.07%	87.01%	87.33%	87.26%
SVM - radial				

Algorithm	9 Even	9 Odd	31 Even	31 Odd
Decision Tree	0.907	0.904	0.855	0.861
Random Forest	0.929	0.928	0.929	0.929
Naïve Bayes	0.931	0.929	0.915	0.913
SVM - linear	0.792	0.794	0.803	0.803
SVM - radial				

### IV. CONCLUSION