Tevatron Higgs

Daniel C E Bunting & Amalia Madden

June 15, 2015

Abstract

1 Introduction

2 Theory

3 Method

3.1 Event Simulation

The Monte Carlo simulation of the events used in the training and evaluation of the neural networks is beyond the scope of this project, so will only be briefly summarised here. Signal $\phi b \to b\bar{b}b$ events were simulated using the PYTHIA [1] to leading order with a mass hypothesis of $M_H = 110\text{GeV/c}^2$ and then corrected to next-to-leading order with MCFM [2]. The background multijet events were generated using ALPGEN [3]. The simulated events were then passed through a model of the D∅ detector [4]. Finally various preprocessing cuts were applied to the data excluding unlikely signal events.

3.2 Feature Selection

In any scenario where statistical methods are used to interpret data selecting features in order to maximise the discriminatory or explanatory power of the model plays a critical role. From the b-tagged jets, the jets with the highest and second highest transverse momentum P_t were identified as being the putative product of the decaying Higgs boson, with all the following variable differences are considered to be taken between these leading jets. From the raw simulation data, following [5] seven feature variables were extracted and used as inputs to the classification neural network.

Pseudorapidity difference $\Delta\eta$

Te pseudorapidity of a jet is defined as

$$\eta = -\ln\left[\tan\left(\frac{\theta}{2}\right)\right] \tag{1}$$

where θ is the angle between the jet and the beam axis. The pseudorapidity is preferred to the θ as it is Lorentz invariant.

Momentum balance $P_{balance}$

The momentum balance of the leading b jet pair is defined

$$P_{balance} = \frac{|p_1 - p_2|}{|p_1 + p_2|} \tag{2}$$

Sphericity S

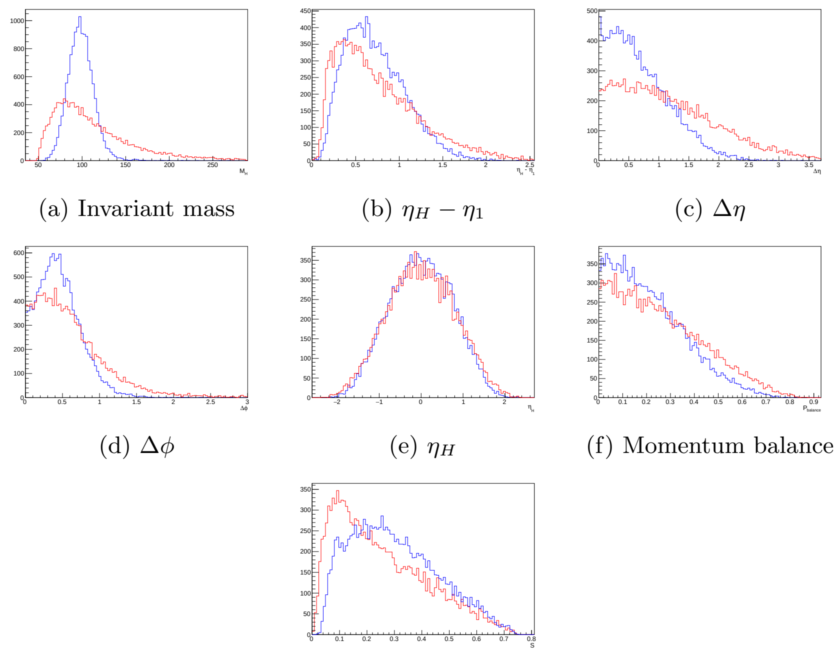
The sphericity is defined as

$$S = \frac{3}{2}(\lambda_2 + \lambda_3) \tag{3}$$

where λ_2 and λ_3 are the second and third largest eigenvalues of the sphericity tensor $\hat{\mathbf{S}}^{\alpha\beta}$

$$\hat{\mathbf{S}}^{\alpha\beta} = \frac{\sum_i p^i_\alpha p^i_\beta}{\sum_i |p^i|^2} \tag{4}$$

Additionally we define the **azimuthal angle difference** $\Delta\phi$, the **combined pseudorapidity** η_H, the **difference between the leading jet and the Higgs** $\eta_H - \eta_1$ and the **invariant di-jet mass** M_H.



(a) Invariant mass

(b) $\eta_H - \eta_1$

(c) $\Delta\eta$

(d) $\Delta\phi$

(e) η_H

(f) Momentum balance

(g) Sphericity

Figure 1: Distributions of the input features. All y scales are events. The signal is in blue and the background in red.

3.3 Artificial Neural Networks

Artificial neural networks (ANNs) are a machine learning approach to classification/regression problems, consisting of stacked layers of neurons implementing a nonlinear activation function interlinked with linear synapses. For our purposes, the activation function $g(x)$ is taken as a sigmoid and the output node a single logistic regression node, this class of ANNs are called Multilayer Perceptrons (MLPs).

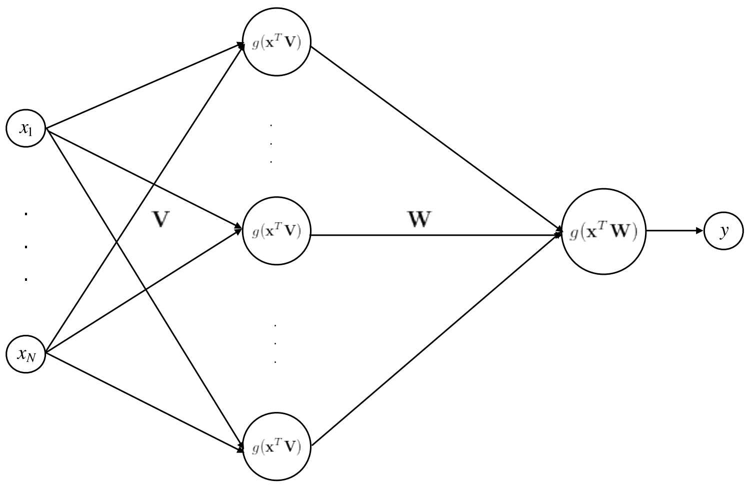


Figure 2: An example MLP with one hidden layer, N input variables $\mathbf{x}$ and single output $0 < y < 1$. The transfer function $g(\mathbf{x^T V}) = \frac{1}{1+e^{-\mathbf{x^T V}}}$ is sigmoid and the weight matrices are $\mathbf{V}$ and $\mathbf{W}$

As a supervised machine learning problem, ANNs are trained by minimising the negative log likelihood (NLL) function $L(\mathbf{W})$ w.r.t. the parameters,

$$L(\mathbf{W}) = -\log\left[P(\mathbf{W}|\mathcal{D})\right] \tag{5}$$

$$= -\sum_{i=1}^{N_{train}} \left[y^i \log \hat{y}^i + (1-y^i)\log(1-\hat{y}^i)\right] \tag{6}$$

where $\mathbf{W}$ denotes the weight matrices collectively and $\mathcal{D}$ the data. Due to the nonlinearity of the activation function the NLL is a non-convex function of $\mathbf{W}$, so it is minimised using random restarts of stochastic gradient descent. A strength of ANNs is that the gradient of the NLL can be efficiently computed by the backpropagation algorithm [6], which allows that gradient at the nth hidden layer to be calculated in terms of the error at the $(n+1)$th layer, so computation can start at the output layer and work backwards. Once the gradient has been calculated the weights are optimised and the output estimates $\hat{y}$ updated by forward propagating the inputs through the network. A single backpropagation, optimisation, forward propagation cycle is called an epoch. In order to avoid overfitting, the input data is split and 50% reserved as a "test set" . After each epoch the network is evaluated on the on the test set and the misclassification error

$e = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathbf{1}\{y^i \neq \hat{y}^i\}$ calculated, training is stopped when e stops declining significantly. We used the Cern ROOT [7] implementation of an MLP, `TMultiLayerPerceptron` due to its tight integration with the rest of the ROOT data analysis frame, despite it lacking some desirable features.

4 Discusion

5 Conclusion

References

[1] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. Pythia 6.4 physics and manual. *Journal of High Energy Physics*, 2006(05):026, 2006.

[2] John M Campbell and RK Ellis. Mcfm for the tevatron and the lhc. *Nuclear Physics B-Proceedings Supplements*, 205:10–15, 2010.

[3] Michelangelo L Mangano, Fulvio Piccinini, Antonio D Polosa, Mauro Moretti, and Roberto Pittau. Alpgen, a generator for hard multiparton processes in hadronic collisions. *Journal of High Energy Physics*, 2003(07):001, 2003.

[4] Rene Brun, Federico Carminati, Simone Giani, et al. Geant detector description and simulation tool. *CERN program library long writeup W*, 5013:1993, 1993.

[5] The D∅ Collaboration. Search for neutral higgs bosons in the multi-b-jet topology in of collisions at. *Physics Letters B*, 698(2):97 – 104, 2011.

[6] Kevin P Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

[7] Rene Brun and Fons Rademakers. Root - an object oriented data analysis framework. In *AIHENP'96 Workshop, Lausane*, volume 389, pages 81–86, 1996.