Using Artificial Neural Networks in the *stop squark* Search

R. Brockman, J. DeVito, and R. LeVan

ELEC 502, Spring 2013

Table of Work

Name	Percentage
Robert	40
Justin	30
Ricky	30

Outline

Contents

1 Statement of Problem 1

2 Objectives 1

3 Technical Approach 1

3.1 Physics of the Signal Events 1

3.2 Physics of the Background Events 2

3.3 Derived Scalars - Physics Perspective 2

3.4 Derived Scalars - Machine Learning Perspective 3

3.5 Classification Basics 5

3.6 Significance 5

3.7 Multilayer Perceptron with Back-Propagation 5

3.8 SOM 6

4 Results 6

4.1 Summary of Significance 6

4.2 Back-Propagation Results 7

4.3 Self-Organizing Map 8

4.4 Two-Stage BP-SOM Classifier Results 12

4.5 Next Steps 13

4.6 Data Problems 13

A Theoretical Limitations of Back-propagation 14

1 Statement of Problem

With the discovery of the Higgs-like particle at the Large Hadron Collider (LHC), all of the important predictions of the Standard Model of particle physics have been tested. The focus of particle physics has now shifted to the search for new particles associated with extensions of the Standard Model. One such search involves the search for a particular particle associated with an extension to the Standard Model known as supersymmetry (SUSY). In particular, physicists at the Compact Muon Solenoid (CMS) detector group at the LHC are interested in verifying a SUSY model in which specific p-p collision events generate the super-symmetric partner to the top quark known as the *stop* squark. Verification of these models requires an efficient classifier capable of separating signal *stop* squark events from background events. More efficient classifiers allow discoveries of fundamental particles to be made with fewer events, less expensive beam time, and thus lower cost.

2 Objectives

Our main objective for this project was to use Artificial Neural Networks to build a classifier for distinguishing two types of collision events. One class of event indicates the presence of supersymmetric particles (specifically, a *stop squark*), and the other class of event represents a troublesome background event.

Our goal is to make a classifier that is better and more efficient at separating the signal from the background than models currently being developed and used. Another method being developed by theoretical particle physicists is a method using a series of cuts on a set of eight derived parameters [1]. The efficiency of the classifiers will be compared by measuring the discovery significance per unit beam luminosity.

Onkur Sen, a physics senior working with Dr. Paul Padley at Bonner Lab here at Rice, is attempting to improve upon this with a boosted decision tree algorithm, using the same derived parameters. Dr. Padley also provided invaluable assistance with understanding the underlying physics.

3 Technical Approach

Our classifier will be trained on simulated data generated by PYTHIA, a particle physics simulator which can be tuned to represent the appropriate SUSY model. If the SUSY theory under consideration is correct, a classifier which works on the simulated data will detect the *stop* squark on the real data from the CMS detector.

3.1 Physics of the Signal Events

The event type we are trying to detect is shown below:

$$\begin{aligned} p+p &\to \tilde{t}+\bar{\tilde{t}} \\ \tilde{t} &\to t+\tilde{\chi}_1^0 \\ \bar{\tilde{t}} &\to \bar{t}+\tilde{\chi}_1^0 \\ t &\to b+j+j \\ \bar{t} &\to \bar{b}+j+j \end{aligned}$$

Here components of two protons colliding roughly head on down the accelerator beam axis combine to form two *stop squarks*, represented by $\tilde{t}$ and $\bar{\tilde{t}}$. (The bar on top represents an antiparticle.) The stop squarks decay into top quarks and neutralinos. Neutralinos escape the experiment undetected, carrying missing energy with them. In the signal event, the top quarks then decay into three other quarks, one of which is a b quark and the other two are lighter quarks from the 1st two particle generations. The quarks form recognizable jets of particles in the detector, and the jets from the b quarks can be differentiated from the others. Thus the detector will see this event as 6 jets, 2 of which are from b quarks, and there will be considerable missing transverse energy caused by the escape of the neutralinos. (The total transverse momentum is the vector sum of all of the momenta perpendicular to the beam axis of the detected particles . If it is not zero, momentum conservation demands that there be undetected particles which have escaped with the missing momentum.) This is shown in figure 1.

The relevant raw numbers from this event are therefore the energy-momentum four-vectors for the four normal jets (j) and two b jets (b), for a total of 24 scalars per event.

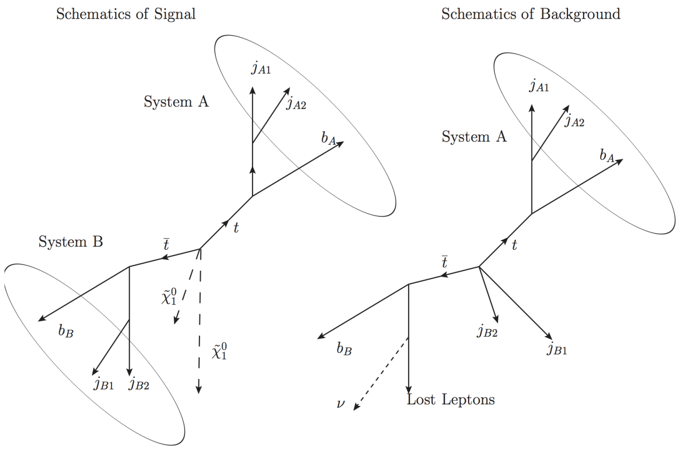

Figure 1: Schematic of the Signal and Noise events [1]

3.2 Physics of the Background Events

Unfortunately, there is a troublesome background event that looks very similar to the signal event, but contains no interesting new physics:

$$\begin{aligned} p+p &\to t+\bar{t}+j+j \\ t &\to b+j+j \\ \bar{t} &\to \bar{b}+W^- \\ W^- &\to e^- + \nu_e \end{aligned}$$

Here the initial collision produces two top quarks directly as well as two low-mass quarks which form jets. One of the top quarks decays into a b quark and two low mass quarks as in the signal event. The other top quark decays into a b quark, a lepton (electron or muon) and a neutrino which carries off missing energy. The big problem is that lepton detection is not perfect, and when the lepton is missed, this event also appears as 6 jets, 2 of which are from b quarks, and with missing transverse energy, just like the signal event. Our task is to study the four-vectors of the 6 jets in each of the event types to see if we can distinguish them.

3.3 Derived Scalars - Physics Perspective

Dutta, et. al. have derived 8 scalar features to help in identifying the background event [1]. The basic strategy comes from the observation that two of the jets from low-mass quarks (hereafter *j*) emerge directly from the initial collision in the background event, whereas in the signal event the *j*'s are paired and each pair is associated with a b quark. Thus, the patterns in the angles between jets and the energy of the jet groupings should be different.

Python scripts provided by Onkur Sen compute the scalar features for each event and output them in a format which we can feed into a classifier. The scripts group the quark jets from each event into two groups of 3 jets each, referred to as System A and System B. Each group contains one b quark and 2 normal jets. The scripts then compute the following:

- M3 : The invariant mass of all three jets in a group. This scalar does not change with reference frame, and should correspond to the rest mass of the top quark that spawned the jets. This is computed for both groups, so we have M3A and M3B.
- M2: The invariant mass of the two non-b jets in a group. This should correspond to the difference between the top and b quark rest masses in the signal case. In the background case, two non-b jets emerged from the initial collision and thus there should be less correlation. This is computed for both groups, so we have M2A and M2B.
- B_ANGLES: The azimuthal angle for the b quark in System B.
- J1_ANGLES: The azimuthal angle for the 1st non-b quark in System B.
- J2_ANGLES: The azimuthal angle for the 2nd non-b quark in System B.
- MISSING_E: The missing transverse energy.

3.4 Derived Scalars - Machine Learning Perspective

The data for we plan to classify with an ANN thus consists of vectors of 8 real-valued scalars, 5 of which are energies in GeV (range 0-1000, mostly around 200) and 3 of which are angles in radians (range 0-π). Each of the vectors is labeled as signal or background, so we have only 2 classes. Right now we have 7595 signal events and 18292 background events generated by Onkur's scripts. We have enough data to split both signal and noise evenly into training, cross-validation, and test sets. The cross-validation sets are used to determine the correct filter strength for the SOMs, as well as to tune the output cutoff threshold for signal and noise for back-propagation.

We include two figures showing the statistics of the input data, for both the derived input data (Figure 2)and the raw input data (Figure 3).

In both cases, we include boxplots for both the signal and the noise events, which describe the quartiles and the median values of each parameter. In the figures, the verticies of the lines describe the means.

For the eight derived parameters, all values are positive, and thus scaling the maximum values to 1 results in a scaling between 0 and 1. For the 24 general parameters, some values (such as momentum) can be negative. Again we scale all values to have a maximum of 1, but in this case this is no longer a scaling between 0 and 1.

In both cases, the purpose of the plots is to demonstrate the complexities of the data. The signal and the noise events resemble each other very closely, and there are extreme outliers across all input parameters.

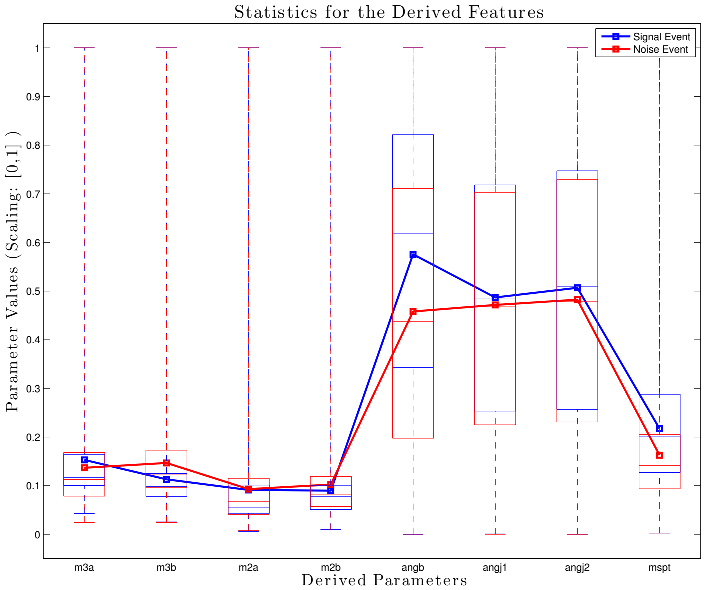

Figure 2: Distribution of Scaled 8-D Input Values

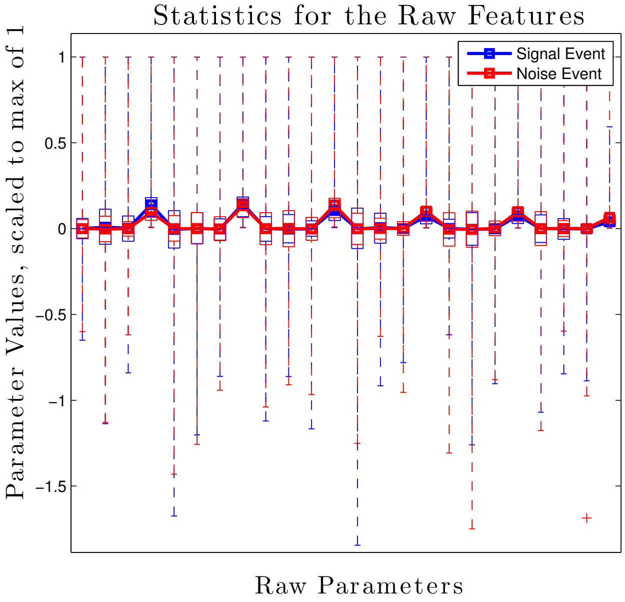

Figure 3: Distribution of Scaled 24-D Raw Data Values

3.5 Classification Basics

The complexities of particle physics reduce (at least for the purposes of machine learning methods) to a classification problem. There are two target outputs—that of signal, and that of noise. We could have had one output PE, with signal corresponding to 1 and noise to -1. But in accordance with standard practice, we instead had two output PE's, and had signal correspond to [1;0] and noise to [0;1] (appropriately scaled). Thus our network performs a nonlinear mapping from 8-D (or 24-D if we are using the raw data) space to 2-D space. From the simulated data, with the help of Onkur's code, we were able to create a length 24 vector of raw data for each signal or background event and a length 8 vector of derived variables from the raw data [1]

Naturally, upon training our network, all outputs will not fall perfectly on [1;0] or [0;1]. Instead, we expect them to typically fall between the two extremes. For an unbiased classification scheme, we simply draw a 45 degree line through the origin, declaring points below to be signal and those above to be noise. However, we can vary this line to improve the ultimate significance ratings. For example, for our purposes false positives are more detrimental than false negatives. For this reason we can bias the classifier so that it is more eager to perform signal or noise classifications.

Earlier classification methods, such as cuts (i.e., thresholding as outlined in Dutta, et al. [1]) used by the physicists, are more simple and primitive than neural network methods. An alternative technique, used by Onkur Sen for his physics senior thesis, is boosted decision trees. In each case, the ultimate goal is improving significance, because more significance is extremely valuable because it translates into less time needed at the LHC to achieve an equal certainty of results.

3.6 Significance

A measure of discovery confidence based on a certain number of high energy particle collisions used commonly in physics is significance. Every significance measure in the Results Section is based on a standard number of collisions. Significance is defined as:

$$\text{significance} = \frac{N_{\text{Signal}}}{\sqrt{N_{\text{Background}}}}$$

where N_{Signal} := number of signal events that come through the filter and $N_{\text{Background}}$:= number of noise events that come through the filter.

Our goal is to maximize the statistical significance, defined as the ratio of signal to the square root of background events. We do this through obtaining a larger ratio of signal to noise than would otherwise be possible without machine learning techniques. We attack the problem with a Multilayer Perceptron trained through back-propagation, as well as with an SOM. In the case of the SOM, prototypes attached to the lattice neurons naturally find clusters of patterns in the input space. Upon completing the SOM training, we can see which lattice neurons become associated with regions of especially rich signal concentration. We then look to the where in the input space the corresponding prototype points to find a region of higher significance.

3.7 Multilayer Perceptron with Back-Propagation

We used the parameters shown in Table 1 for training back-propagation [2]. Experimenting with more layers and/or more hidden units has not yielded improved results so far. These parameters were also used for the back-propagation stage of our two-stage filter.

ARCHITECTURE	
Topology	(8 +1_{Bias}) - (30 +1_{Bias}) - 2_{output}
Transfer Function	tanh with slope $b = 1$
LEARNING PARAMETERS	
Initial weights	$w \sim U[-0.1, 0.1]$
Learning rate, $\gamma(t)$	$\gamma(t) = 0.01(1-0.0001)^t$
Momentum, α	$\alpha = 0.3$
Epoch size	$K = 1$
Stopping criteria	learning step > 100,000
Error measure (Err)	RMSE
Monitoring frequency (m)	1,000 Learning Steps
INPUT/OUTPUT SCALING	
Input Scaling	(-0.9,0.9)
Output Scaling	(-0.9,0.9)
PERFORMANCE EVALUATION	
Accuracy measure (Acc_X)	Significance = $\frac{S}{\sqrt{B}}$

Table 1: Back-Propagation Filter Settings

3.8 SOM

We used the parameters shown in Table 2 traning an SOM [3].

ARCHITECTURE	
Topology	10 x 10
LEARNING PARAMETERS	
Initial weights	$w \sim U[-0.1, 0.1]$
Learning rate, $\gamma(t)$	$\gamma(t) = 0.3(1-0.00001)^t$
Neighborhood, $\sigma(t)$	$\sigma(t) = 1.5 + 3.5\,(1-0.00001)^t$
Epoch size	$K = 1$
Stopping criteria	learning step > 750,000
Monitoring frequency (m)	1,000 Learning Steps
INPUT/OUTPUT SCALING	
Input Scaling	Angles in Degrees, Otherwise None
Output Scaling	None
PERFORMANCE EVALUATION	
Accuracy measure (Acc_X)	Significance = $\frac{S}{\sqrt{B}}$

Table 2: SOM Filter Settings

4 Results

4.1 Summary of Significance

As seen in the table below (Table 3), the physicists' methods of thresholding are actually counter-effective in terms of significance. Although they do improve classification, they cut out so many events that significance remains small. With back-propagation, we created an effective filter that was able to classify enough events as signal to still generate substantial significance. The same holds true for the SOM trained on the derived variables. In the richest lattice PE, the gain was sufficiently high to compensate for the other events in the input space, which were cut out. Finally, combining different learning paradigms gives us the best results of all. We achieve these results by feeding the signal-rich input space (as determined by our SOM) into a back-propagation network.

METHOD	TEST SET SIGNIFICANCE
Thresholding [1]	1.93σ
No Filter	2.62σ
Back-Propagation	3.79σ
Self-Organizing Map for Derived Variables	3.69σ
SOM then Back-Propagation	4.36σ
Self-Organizing Map for Raw Data	2.62σ

Table 3: Table of the significance values produced using each method

4.2 Back-Propagation Results

As shown in Figure 4, training an MLP with back-propagation results in steadily increasing significance, up to 10^5 epochs. The back-propagation network itself is being trained under the objective function of sum-of-squared-error. However, this indirectly results in improved significance. As seen in the figure, significance also increases, as desired, when the cross-validation data is run through the network taught with the training data. In this case, significance happens to be even higher for the cross-validation data, but this is simply an artifact of the cross-validation data having more signal events to begin with.

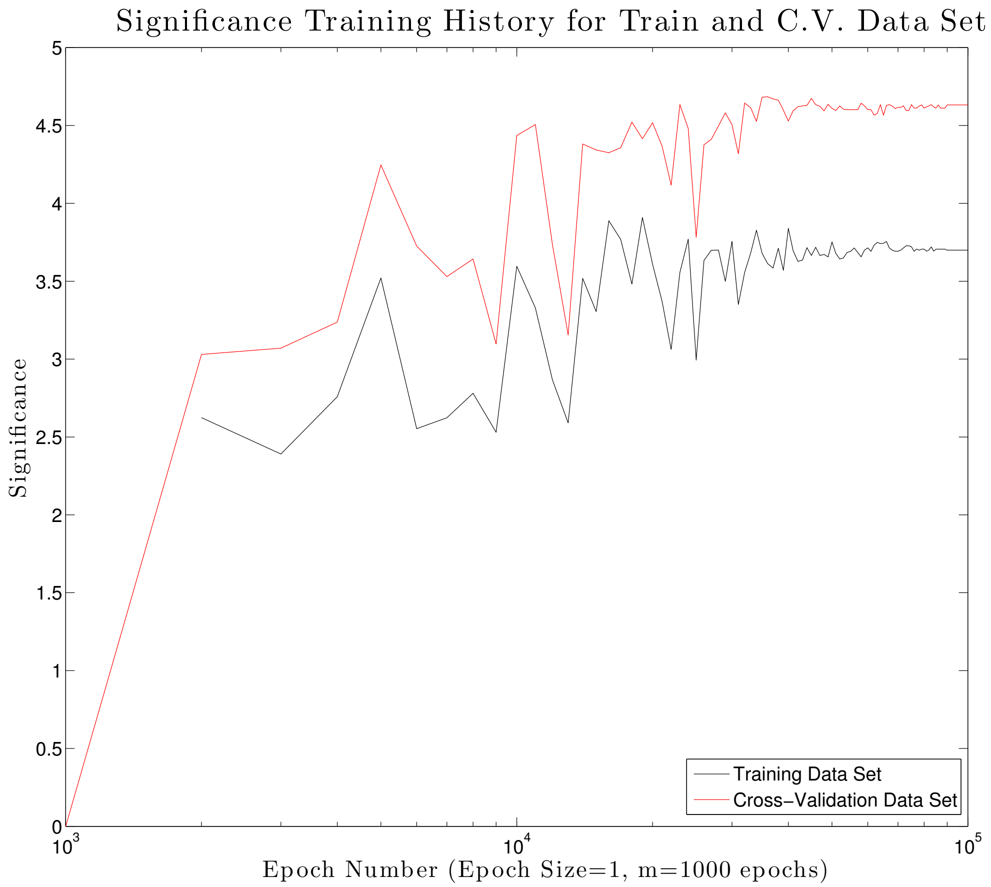

Figure 4: Back-Propagation Results for Derived Variables

4.3 Self-Organizing Map

The SOM for the derived variables converged as shown in Figure 5. There are a cluster of lattice cells that have a high concentration of signal events (visualized in red), and much the remaining space is solidly noise (visualized in green). The yellow square in the top-left corner we interpret as an anomaly, but one which does not seem to detract from our results. The *gain* for each PE, defined to be the output signal to noise ratio (SNR) divided by the input SNR, is shown in Figure 6. Note that the highest gain for a lattice cell is over 20. It is important that the gain be large, because we necessarily sacrifice significance when we cut out many of the input data points. Thus, we need a large concentration of signal in a given lattice cell to compensate for this loss.

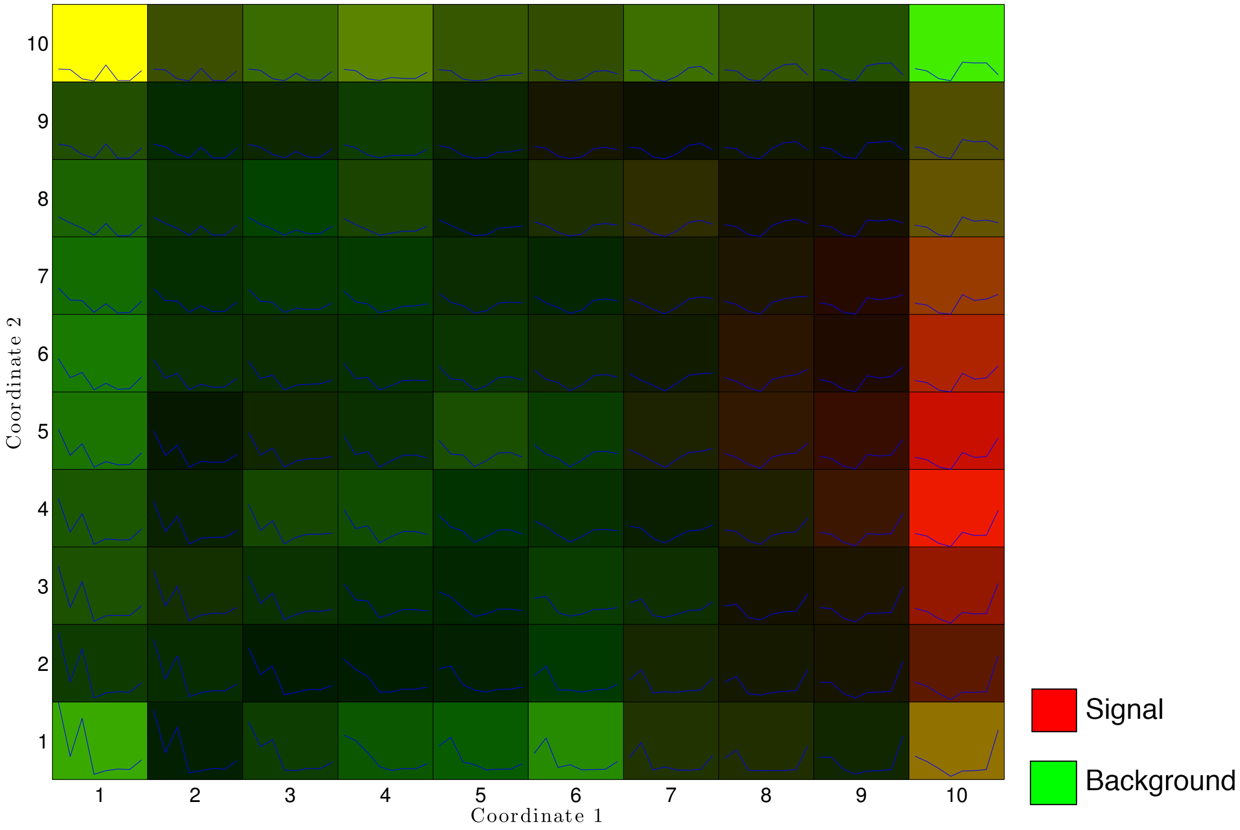

Figure 5: SOM Signal and Noise Density plots with weights for the Derived Variables

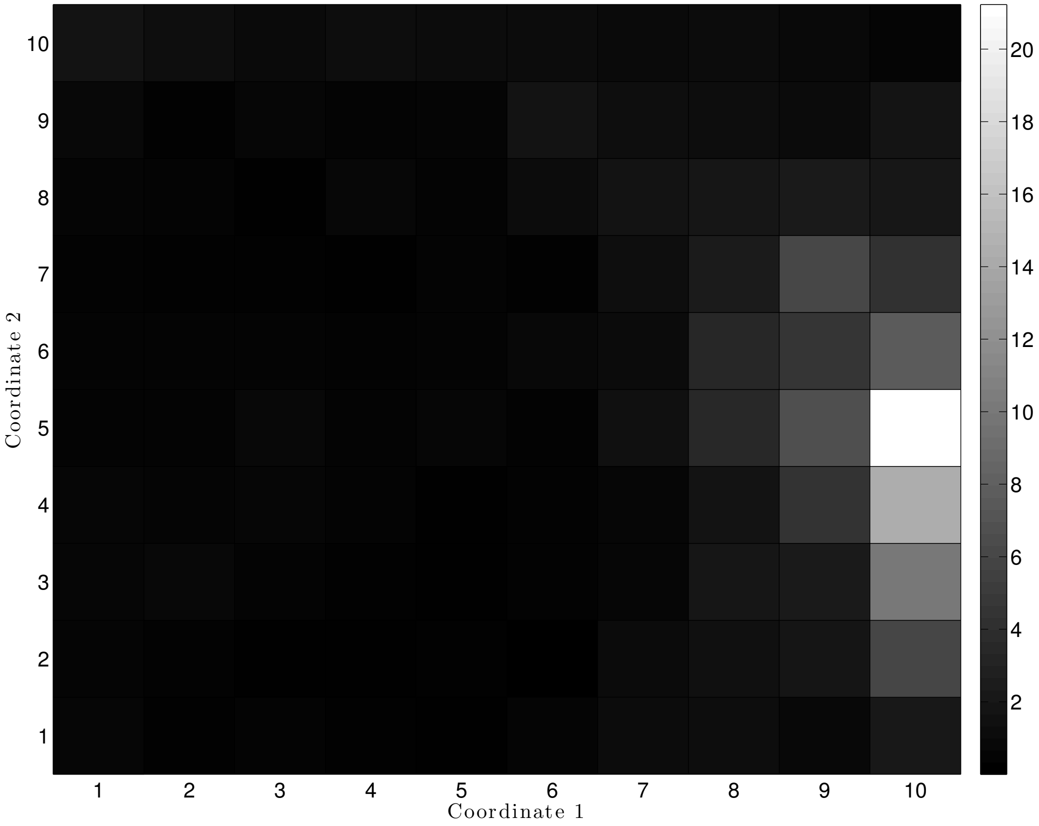

Figure 6: SOM Normalized Signal to Noise Ratios for the Derived Variables

In Figures 7 and 8, we repeat the SOM analysis, except this time we use the raw data. Thus, our prototypes extend into 24-D space, corresponding to the 6 particle jets each with their accompanying 4-vector of energy and momentum. We see the increase in input space visually because the blue lines in each cell (representing the prototypes) have more "wiggles". This is because they are connecting 24 points, instead of merely 8. Superficially, these results look very similar to the results from the 8 derived inputs case, however, the gain is much weaker. Where before our gain extended past 20, here it barely reaches 5. Although this SOM does succeed in identifying signal clusters in input space, it fails to do this effectively enough to make up for the large number of data-points cut out by the filter. This filter does not increase the significance at all.

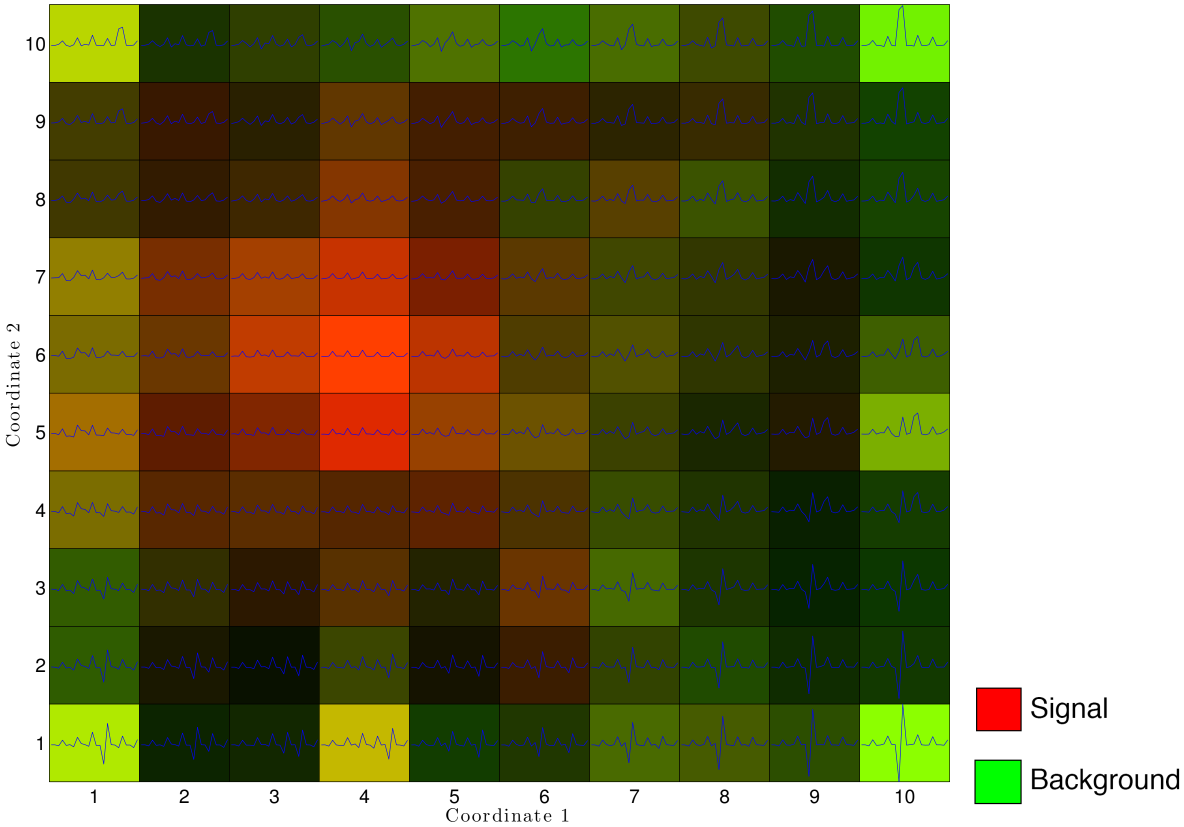

Figure 7: SOM Signal and Noise Density plots with weights for the Raw Data

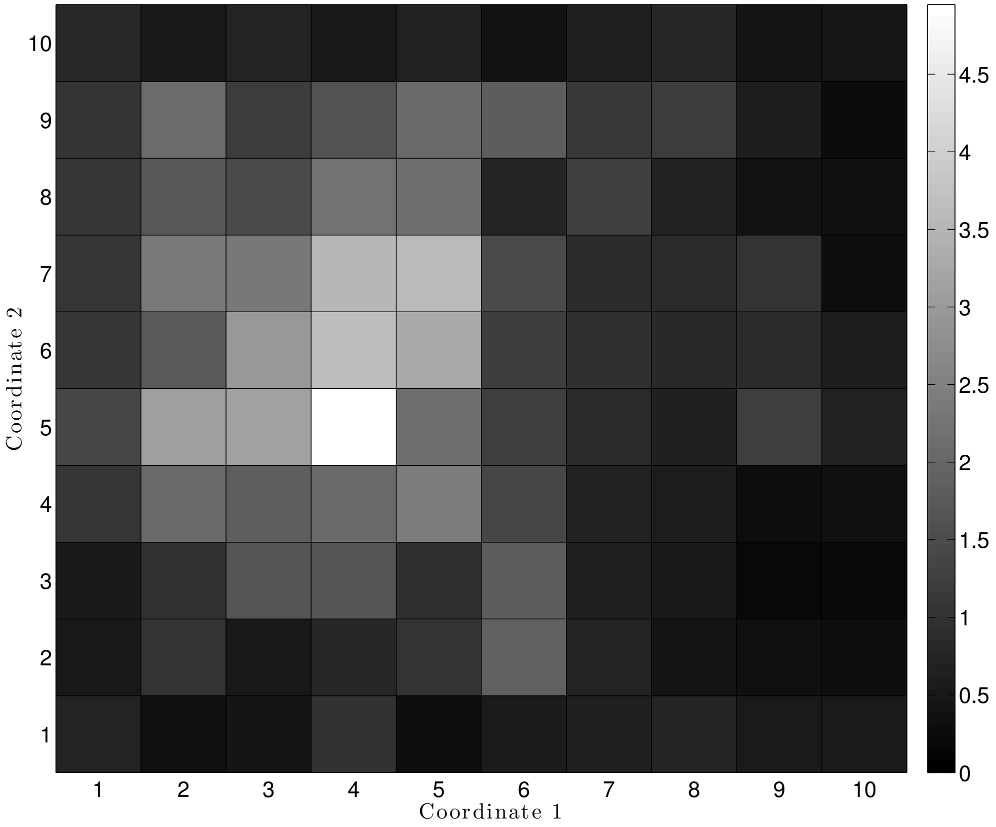

Figure 8: SOM Normalized Signal to Noise Ratios for the Raw Data

Ideally, we would want the SOM to perform as well or better on the raw parameters as on the derived parameters. There are several reasons why this might not be the case initially, and why more sophisticated analysis could fix the problem. Firstly, the physicists presumably already have a good intuition for what sorts of variables are the “defining” variables of a system. So by using their derived variables, we are leveraging their intuition. If we make the very reasonable assumption that their intuition is effective, then we reduce the amount of work the SOM must do in finding underlying patterns, because much of the work has already been done.

A primary cause of this lack of effectiveness is that the derived variables include 3 angles and already normalizes momentum vectors regardless of direction. These angles were independent of the absolute direction of the resulting quark jets, whereas the 24 raw data variables depended on the absolute angular direction of the jets. For the purposes of the analysis from a physics perspective, an event where the detected jets are just a rotation of the jets of another event should be considered the same event. However, the neural networks examined are unable to perform this rotation automatically by themselves (which is not expected, either). In continuing this analysis we need to attempt to align the x, y, and z momenta of the jets in order to make sure similar events are classified together by the neural network. We believe that this will produce even better results as we cannot be sure the 8 derived variables take into account all of the data of the 24 raw data variables.

4.4 Two-Stage BP-SOM Classifier Results

Although our attempt to implement a two stage classifier by applying back-propagation to the output of the SOM filter did seem to increase the significance compared to SOM or back-propagation alone, this may have been a result of chance. As seen in Figure 9, the neural network did not train well, with no meaningful increase in the significance on the training data.

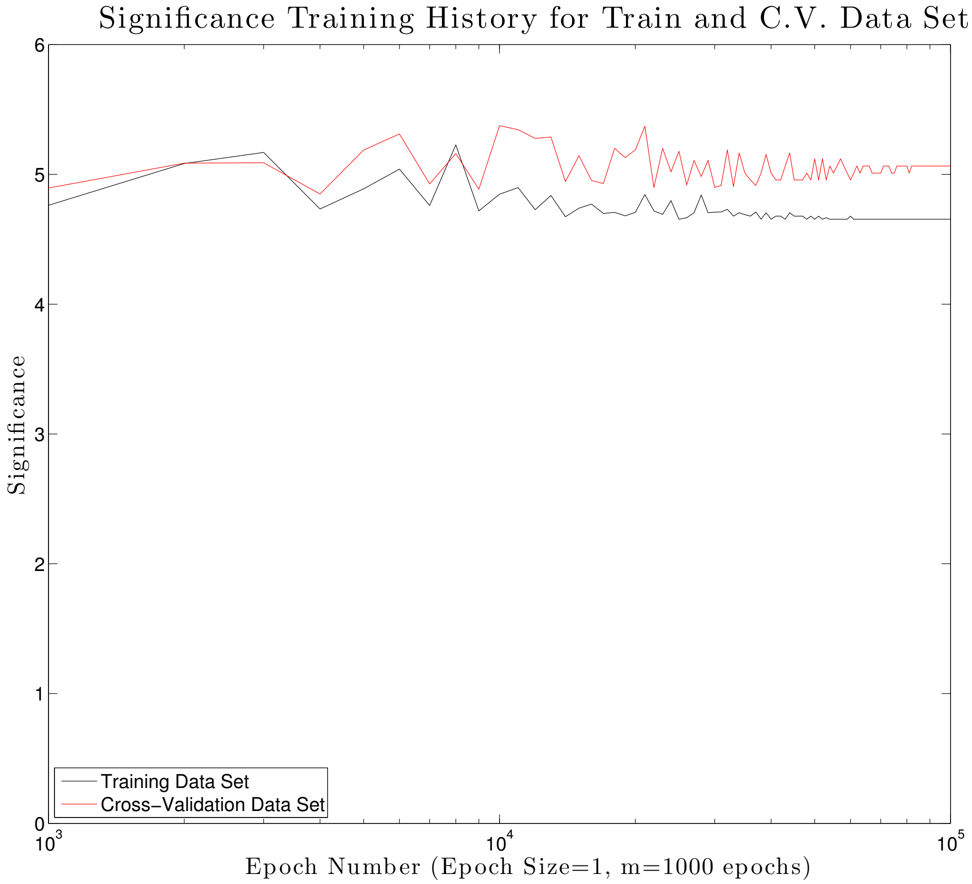

Figure 9: SOM followed by BP for Derived Variables

4.5 Next Steps

With our main results acquired, we look to how we can improve our results, with the aim of publication in mind. First, as described earlier, we would like to better align the 24 non-derived variables to take into account angle variation. Second, we would like to experiment further with the training parameters in our learning machines. Perhaps other combinations are more optimal than those used so far. Third, we would like to run a second SOM only on the SOM cells of the first SOM with a high gain. In this manner we hope to further improve significance with an even higher gain.

Finally, we would like to develop an alternative learning machine that would use significance directly as an objective function. In our current system, the MLP seeks to minimize sum-of-squared-errors, and this indirectly benefits significance. Likewise, the SOM simply tries to find clusters in the input space. This improves significance to the extent that certain clusters are signal-rich. Perhaps we could train an SOM superimposed with supervised learning methods, causing the SOM to favor clusters of signal, especially large clusters that will maintain high significance. Another option is to use an LVQ, an already existing supervised learning paradigm.

4.6 Data Problems

At the very last minute, Dr. Padley and Onkur Sen discovered serious problems with both Onkur's feature extraction code as well as the data generated from PYTHIA. These problems may partially explain why our unfiltered results are so much better than the results of the physics paper we used [1]. Fortunately, regenerating the results in this project once these problems have been corrected is very straightforward.

A Theoretical Limitations of Back-propagation

From Bishop, et al. [4], we know that there are theoretical limitations on the usefulness of a multilayer perceptron trained with back-propagation. These considerations help explain the limited usefulness of the MLP in classifying the particle collision data.

The analysis begins with a reworking of the error measure in the limit of an infinite training data set. In this limit, we can move from finite sums of input patterns to integrals over the distribution of the data in the input space. Bishops analysis considers the conditional averages of the target data (conditional here refers to conditional on the given input pattern). With some simplifications (described on p. 202), he achieves the following terse but powerful result: $y_k(x, w*) = \langle t_k \mid x \rangle$. Put into English, this states that the output of the neural network corresponds to the conditional average of the target outputs. Thus, the trained network is essentially returning the conditional mean of the target outputs given some input pattern. This is not an artifact of a particular network architecture, or even the use of a neural network (p. 203). Rather, the importance of the conditional average comes from the training via the sum of squares objective function.

The relevance of this theory to our project is that certain types of nonlinearity in the distribution of the data will necessarily be ignored by a learning machine trained to minimize a sum of squares error. If, for example, $\langle t_k \mid x \rangle$ is an expected value that few (if any) input patterns actually map to (just as the expected value of a die roll is the impossible 3.5), then the network will try to find structure in the conditional average where such structure is not actually present.

Our data and their corresponding target values may suffer from this sort of nonlinearity.

References

[1] B. Dutta, T. Kamon, N. Kolev, K. Sinha, and K. Wang, “Searching for top squarks at the lhc in fully hadronic final state,” *High Energy Physics - Phenomenology*, 2012.

[2] C. Fyfe, *Artificial Neural Networks and Informaton Theory*, Course Notes., Dept. of Computing and Information Systems. University of Paisley., 2000.

[3] T. Kohonen, *Self-Organization and Associative Memory*. New York: Springer-Verlag, 1988.

[4] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.