Optimising SUSY searches at CMS using Machine Learning techniques

Yuting Li
Mentor: Prof. M. Spiropulu
Co-mentors: Javier M. G. Duarte, Dr. Jean-Roch Vlimant, Dr. Si Xie

September 25, 2015

Abstract

The Higgs detection with its 126 GeV mass is hinting at physics beyond the standard model. We are at the stage where we are unclear as to what discovery story awaits and need to cater for an extremely small amount of signals. There have been studies demonstrating the use of supervised machine learning techniques in signal detection. However, supervised methods have the risks of overtraining on Monte Carlo imperfections as well as the difficulties of getting statistical interpretations. In this paper, the Self Organising Map (SOM) was applied as an unsupervised data-driven clustering algorithm. To tackle the problem of the lack of statistics, the Neural Autoregressive Density Estimator (NADE) was first introduced for background estimation in the cells in the SOM. Its ability to detect outliers was also investigated in this paper.

1 Introduction

The widespread use of TMVA, a ROOT package that gives black-box implementation of a wide range of multivariate algorithms, have increased the popularity of machine learning in particle physics. The standard approach is to use “shallow” learning techniques (regression analysis, shallow neural nets, boosted decision etc) combined with specially engineered kinematic variables that are better discriminates for signal vs. background. In recent years, rapid developments in Deep Learning have made it possible to construct deep architecture for probing complicated non-linear structure in the data. [13]

In May 2014, ATLAS released the Higgs Boson Machine Learning Challenge on applying Machine Learning techniques to Higgs Boson identification against a large amount of background noise. The winning solution consists of a deep neural net with dropout trained on random shuffles of the training data. In addition, papers have been published on applying Deep Learning Techniques to a wide range of HEP problems including jet identification, Higgs benchmark model and SUSY benchmark model. [6] [1]

All the techniques mentioned above belong to the category of "supervised machine learning"- learning an unknown function through examples of labelled data. This requires accurate knowledge of both the signal and the background. At the moment, the kind of labelled data that we could get hold of are Monte Carlo generated simulated data. Despite the amount of studies going into Monte Carlo simulation, we know that it is still not perfect- due to the complexity of the simulation of the detector response, different tunings of the parameters could lead to different simulated trace. In addition, even if simulation techniques could be perfected, to account for all possible signals, we would need to train and test specifically for each kind of signal, which is a very time-consuming process.

Unsupervised machine learning, on the other hand, does not require labelled data, implying that training could be applied directly to detector data, saving the learning process from Monte Carlo imperfections. Although we would still need to reply on simulated data for background estimation, it is much simpler to device correction methods for testing than training. In addition, unsupervised methods get rid of the need to train separately on different signals- instead we could train on detector data and then do hypothesis tests with all the possible signals. The two techniques studied are the Self Organising Map (SOM), an unsupervised nonlinear dimensionality reduction algorithm, and the Neural Autoregressive Density Estimator (NADE), a density estimator for multidimensional data.

In section 2, I will briefly introduce Supersymmetry and the Razor Variables; in section 3, the methods of data selection will be discussed; section 4 contains a description of the statistical tools used to compare the sensitivities; detailed accounts of the unsupervised clustering algorithms are given in section 5 and section 6, with results and conclusion in section 7 and section 8 respectively.

2 Background and Previous Work

2.1 Supersymmetry (SUSY)

Initially motivated by the hierarchy problem, supersymmetry is a proposed extension to the symmetry of spacetime that relates bosons and fermions. It predicts that each boson in the standard model (SM) would have a new fermionic superpartner, and vice versa.

The observed mass of the Higgs Boson, measured by the ATLAS and CMS experiments at the Large Hadron Collider (LHC), has prioritized searches for SUSY. According to the SM, the Higgs boson is extremely sensitive to quantum corrections. Without new physics to offset the effect, the Higgs mass is pushed up to the Planck scale unless there is an almost-perfect cancellation due to the fine tuning of certain parameters. SUSY provides such a cancellation mechanism. Moreover, SUSY may explain the existence of dark matter. To date, no SUSY signals have been detected at any particle collider but some lower bounds on the masses of SUSY particles have been set with the data from the previous two runs of the LHC and other colliders. [2] [10].

Following an energy upgrade to 13 TeV, the LHC will be able to probe a broad range of SUSY scenarios during its next run [5]. The CMS detector consists of many subdetectors, composed of various materials to measure the energy and momentum of outgoing particles, from which particle tracks can be reconstructed and retraced back to the original collisions [11].

2.2 Razor Variables

Simplified Models usually assume that only two new SUSY particles are accessible at the LHC energy scale: a heavy particle such as squark (superpartner of quark) and the lightest SUSY particle (LSP), the lightest neutralino. The benefit of such models is that they can be well described by a few parameters related to measurable particle physics observables. Moreover, the limits defined through simplified models can be used to derive constraints for more general models. In decay chains proposed by simplified models, one of the final-state particles - the LSP, is assumed to be weakly-interacting, leading to missing transverse momentum. The razor kinematic variables [9] M_R and M_T^R are defined as follows,

$$M_R \equiv \sqrt{(| \vec{p}^{j_1} + | \vec{p}^{j_2})^2 - (p_z^{j_1} + p_z^{j_2})^2}$$

$$M_T^R \equiv \sqrt{\frac{E_T^{miss}(p_T^{j_1} + p_T^{j_2}) - \vec{p}_T^{miss} \dot{} (\vec{p}_T^{j_1} + \vec{p}_T^{j_2})}{2}}$$

where $\vec{p}_{j_i}$ is the four momentum of the ith jet and E_T^{miss} is the missing transverse energy. M_R is closely related to particle mass whereas M_T^R is related to missing transverse momentum. The razor dimensionless ratio is then defined as:

$$R \equiv \frac{M_T^R}{M_R}$$

The distribution of R^2 and M_R of the collider data can be compared with the predictions given by SM background and SUSY simplified models using statistical methods, providing a way to detect SUSY signals. My mentor, Professor Maria Spiropulu, together with her students and colleagues, has performed searches with razor variables on 7TeV and 8TeV CMS data from the 2011 and 2012 runs, extending the upper bounds on the mass of top squack and gluino [3] [4].

3 Data Selection

I used three types of backgrounds (QCD, $Z(\nu\nu)$+jets and TTJets) and one type of signal (T1bbbb) in my project to ensure the variety of data but at the same time focusing the efforts more on the testing of the machine learning techniques rather than processing datasets. Initially $W(\ell\nu)$ + jets dataset was also used for some test. See appendix A for the exact set of Monte Carlo data I used.

The input variables I used were transverse momentum (PT) and eta of the three leading jets, two leading muons and two leading electrons, Missing Transverse Energy (metPt and sumMET) and the Razor Variables (M_r and r^2). Due to the fixed size of the input array, some variables were missing: for example, when there was only one electron, all the information about the second electron was absent. This could be solved by the creation of boxes and conducting a separate analysis in all the boxes. Alternatively, some place-holder could be inserted in place of those missing values: one option is to make all of the absent values zero or some unrealistic value, e.g. negative value when it is meant to be a positive number, which is how the Higgs Machine Learning data were delivered; the second option is to use some noisy distribution around an unrealistic value away from all the other data points. Both options have their merits and issues and the choice depends on the following data process procedures.

The first stage of selection requires at least two jets with (Pt ¿ 40 GeV, eta ¡ 2.4) and no muon with (Pt ¿ 2000 GeV). Here's a table of different ways I the datasets were processed at the second stage:

Cut	Method
Hadronic	Selecting all events with no leptons and at least two jets
R2 0.1	Selecting all events with R^2 larger than 0.1
R2 0.05	Selecting all events with R^2 larger than 0.05
metPt 65	Selecting all events with metPt larger than 65

4 Statistical Test

To compare the sensitivities of different tools, likelihood test was introduced to set the exclusion and discovery limits on signal strength with the same types of signal and background. In this hypothesis test, the null hypothesis - background only - is tested against the alternative hypothesis - signal with strength μ, assuming that μ has a flat prior from 0 to ∞. Flat prior implies that $L(data \mid \mu)$ is a constant multiple of $L(\mu \mid data)$ for $\mu > 0$.

Given a set of data binned in a certain way (e.g. on the M_r and R^2 plane), the likelihood of the dataset given signal strength μ is:

$$L(data \mid \mu) = \prod_{bins} P(data \mid background + \mu \times signal) \tag{1}$$

Assume that the counts in each bin are Poisson distributed around some true value of $b + \mu s$, the test statistic - the log of the ratio of the likelihoods of an alternative hypothesis and the null hypothesis - is as follows:

$$\begin{aligned}
\lambda &= \log(\frac{L(data \mid \mu_{test})}{L(data \mid \mu_{best})}) && (2)\\
&= \log(\frac{\prod_{bins} Poisson(n_i \mid b_i + \mu_{test} s_i)}{\prod_{bins} Poisson(n_i \mid b_i + \mu_{best} s_i)}) && (3)\\
&= \sum_{bins} [-(b_i + \mu_{test} s_i) + (n_i) \log(b_i + \mu_{test} s_i) + (b_i + \mu_{best}) - n_i \log(b_i + \mu_{best})] && (4)
\end{aligned}$$

where b_i, s_i are expected background counts and signal counts in bin_i, n_i is the counts from data, μ_{best} is the value of μ that maximises $L(data \mid \mu)$, μ_{test} is a test value of interest.

The exact form of the likelihood as a function of μ is unknown but in most situations it is well approximated by a Gaussian centred around μ_{best}. [8] Taking into account the flat prior, the likelihood can be treated as the probability distribution of μ given the data. Therefore the log of the likelihood ratio (λ) gives $-\sigma^2/2$ where σ is the number of standard distributions away from the mean, inverting the formula gives $-2\lambda = \sigma^2$. By convention, C.L. values are also converted into sigmas with the following formula:

$$\sigma = \sqrt{2} erf^{-1}(C.L.) \tag{5}$$

The common 95% C.L. quoted in particle physics corresponds to a value of σ of $\sqrt{2 \times 1.92}$.

As a preliminary comparison of the sensitivities, two toy tests were devised to approximate what happens in a search with detector data. Exclusion test: what is the exclusion limit we could set on signal strength if there were truly only background ($\mu_{best} = 0$). Discovery test: with what significance we can claim the discovery if there were truly signal with strength μ ($\mu_{best} = \mu$). In both tests, the luminosity was assumed to be 10 fb^{-1}.

For exclusion test, for simplicity, assume we get exactly the expected counts of background (i.e. $n_i = b_i$ in each bin). $\mu_{test} = mu$ since for each value of μ we are testing how many sigmas away n_i is from $b_i + \mu s_i$. The test statistics is therefore:

$$\lambda_{exc} = \sum_{bins} [-(b_i + \mu s_i) + (b_i) \log(b_i + \mu s_i) + b_i - b_i \log(b_i)] \tag{6}$$

The μ value where $-2\lambda_{exc}$ crosses 2×1.92 gives the lower limit of μ we can exclude if we observed only background.

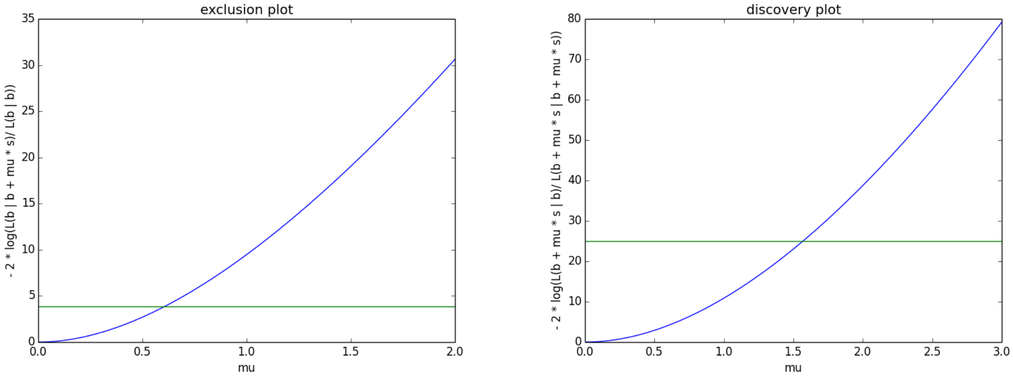

Figure 1: The figure shows the exclusion plot (left) and the discovery plot (right) of data binned according to nodes in a Self Organising Map trained with data in hadronic box with cut on R^2 at 0.1. Both plots show the respective test statistics as a parabolic (by eye) function of μ as expected.

For discovery test, again under the simplified assumption that for each value of μ, exactly the expected amount of signal is observed (i.e. $n_i = b_i + \mu s_i$). $\mu_{test} = 0$ since we are testing how many sigmas away the observed is away from expected from background. The test statistics is therefore:

$$\lambda_{dis} = \sum_{bins} [-b_i + (b_i + \mu s_i) \log(b_i) + b_i + \mu s_i - (b_i + \mu s_i) \log(b_i + \mu s_i)] \tag{7}$$

A 5σ discovery corresponds to a crossing point of 25.

5 Clustering with Self Organising Map

One unsupervised clustering algorithm I used is the Self Organising Map. The Self Organising Map (SOM) [7], or Kohonen Map, is a lower dimensional, discretised and non-linear grid representation of data. Once trained, data points that are close together in the input space would map to neighbouring grid-points on the SOM. In the context of HEP, events from the same process should have similar input parameters and therefore projected onto some connected region in on the SOM.

A map is a grid with vectors attached to each node. The vectors correspond to points in the input space that the nodes map to. The search method is as follows: train a SOM on actual data from the detector and count occupancies at each node; pass Monte Carlo generated background samples into the map and count node occupancies; perform a likelihood test on the map. See figure 2 for the node occupancies of a trained Self Organising Map.

The training algorithm for a SOM is as follows:

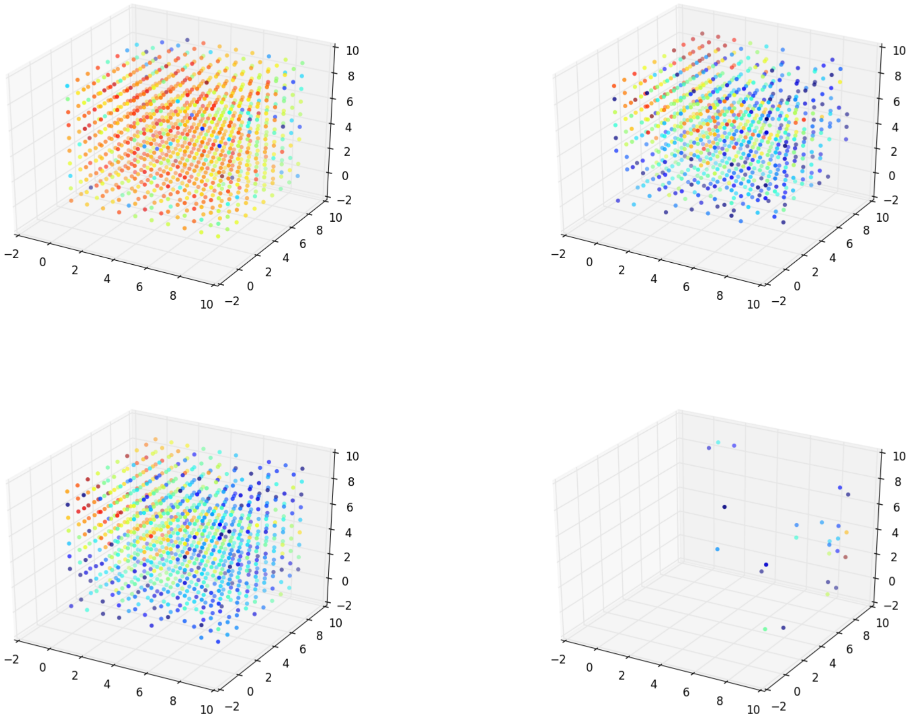

Figure 2: This is a trained SOM with data in the hadronic box and cut on R^2 at 0.1. From left to right, top to bottom, the plots correspond to QCD, ZJets, TTJets and SMS-T1bbbb respectively. The colour bar goes from blue to red indicating the increase of the number of hits on the node. We can see from the plots that there is some clear clustering, with SMS signal near the edge of the map occupying the nodes with few hits from the background data.

Initialisation randomise the map's nodes' vectors

Sampling select a sample

Matching use Euclidean distance formula to find the node with the closest vector, or the Best Matching Unit (BMU)

Update update the vectors of the nodes in the neighbourhood of the BMU with a Gaussian according to pre-defined learning rate and neighbourhood decay rate.

Stopping condition once a datapoint has been attached to the same node a certain number of times, pass on the update step

One of my co-mentors, Dr Vlimant, presented the possibility of using Self Organising Map to do clustering at a workshop at Simons Foundation. He showed that with equal amount of data, four types of events (QCD, $Z(\nu\nu)$+jets, $W(\ell\nu)$+jets, T1tttt) form distinguishable clusters on 2, 3 and 4 dimensional maps.

Following this promising results, I changed the number of data points from each type of data to be more realistic. This brought about the problem of not having enough Monte Carlo samples for the QCD dataset used. Due to its large cross section, QCD was an important component of background that could not simply be ignored. I introduced a way to do weighted-training by updating the vectors as if the same sample has been passed through the map n times where n is the weight. Admittedly this is different from training with actual data but before the running speed of the Self Organising Map improved or more Monte Carlo data could be generated this is the best way to mimic training with the right amount of data.

Despite rewriting the code using a faster numerical Python library named Theano, the running speed of the Self Organising Map training algorithm is not fast enough to handle the total amount of data expected. On a CPU (1.8 GHz Intel Core i5), the processing time per sample is about 0.0005. With GPU (k40), the performance improves by approximately a factor of 2. Without any cuts in the dataset and a luminosity of 10 fb^{-1} (the total amount of data collected at 8TeV was 20 fb^{-1}), the expected amount of data would be at least of the order of 10^9, making the processing time approximately 3 days. Due to the amount of tuning required to get a Self Organising Map to deliver satisfactory performance, we would expect the process to take a few weeks at least. This means that some cut in the data is necessary to reduce the total amount down to some realistic level. I tested the performance of Self Organising Map on both unboxed data and hadronic box with R2 0.1 cut or metPt 65 cut. With weighted training implemented due to the lack of QCD data, the total number of training samples was 100000, i.e. training time per epoch 50s on CPU.

On top of making the datasets a reasonable size for the Self Organising Map, pre-scaling the datasets to have mean of 0.5 and standard deviation of 1 also helped. This is due to the way the node vectors were initialised- the vectors were randomly numbers following a uniform distribution between 0 and 1. Another factor I have found to might have influenced SOM training is the absolute number of signals. See figure /reffig:lumi. For the unboxed datasets, all the missing values were replaced with zero before pre-scaling.

Once training is complete, test data were passed into the map to obtain expected amount of counts from background and signal at each node. One problem encountered was the presence of zero nodes- nodes where there are no hits from any type of background, which makes hypothesis test problematic. My solution was to combine all the zero nodes with the least occupied node as a way of binning.

Due to the random initialisation of vectors, there is some variation in the discovery limits. The values quoted in Section 7 are averages of five independent training with the same training dataset. It is also possible to train a Self Organising Map several times on detector data to pick the one that has the best performance in setting either discovery or exclusion

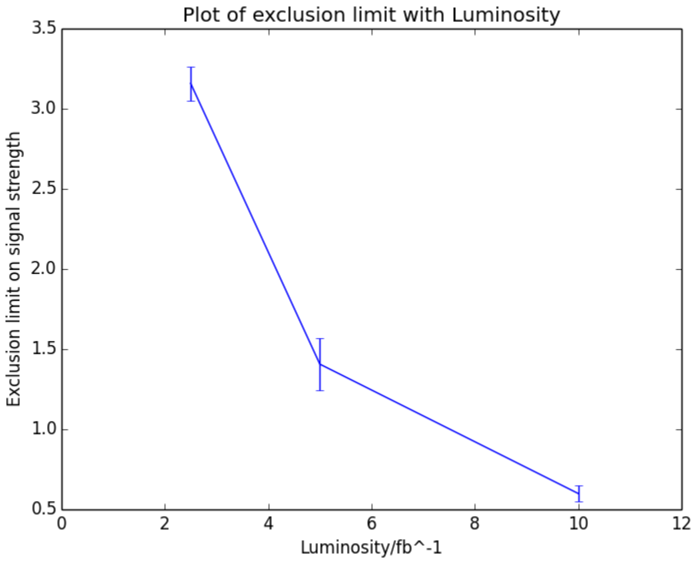

Figure 3: This shows a plot of exclusion limit on signal strength against luminosity. As luminosity increases, the exclusion limit gets lower, implying that the SOM is better at discriminating signals against background as the luminosity increases.

limits with Monte Carlo simulated data. This is similar to varying the binning on the razor variable plane and as long as detector data were not used for testing purpose this does not violate the "blind-testing" principle of SUSY group.

6 The Neural Autoregressive Density Estimator

The other technique employed is a density estimator named the Neural Autoregressive Density Estimator (NADE) [12]. It is applicable to multidimensional data described by a joint probability distribution. The basic structure is a deep neural network that resembles staged Restricted Boltzmann Machines and connected according to some prior ordering of the input variables. For each input sample, the corresponding output generated by NADE is an estimated log density. NADE averages the log density of all the training samples to obtain estimated average log likelihood for the entire training dataset, which it then uses as the cost function for back propagation. To avoid the bias introduced by the ordering, the more advanced structure, orderlessNADE generates a large ensembles of NADEs with different orderings and train them simultaneously.

With a NADE trained on the background signals, we can clearly see from figure 4 that the signals have log densities much more negative than backgrounds. NADE can also be used to generate pseudo data that are meant resemble the true distribution of the training data.

At first we thought this could be a solution for lack of QCD data; however as it turned out, the pseudo data generated were not good enough approximation to used in the context of SOM (see section 6.1).

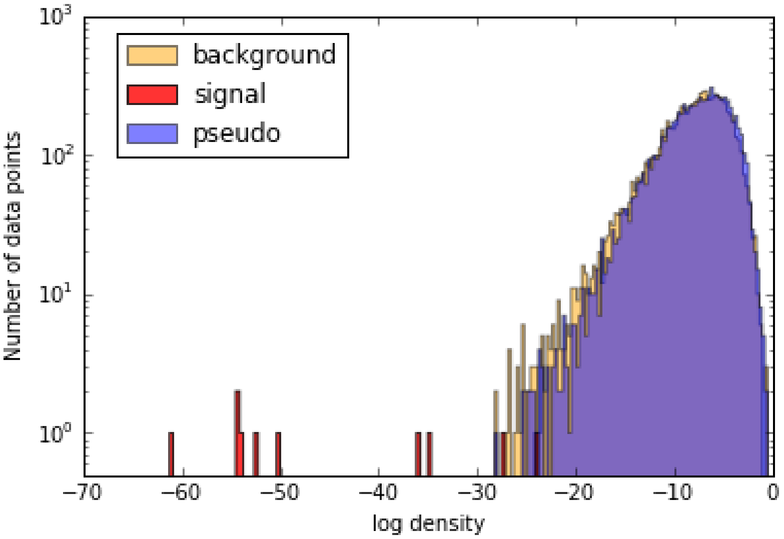

Figure 4: This is a plot of signal, background test data and pseudo data. The NADE was trained on Monte Carlo simulated background data. Test data were drawn from the same distribution as the training data. Pseudo data were drawn from NADE once it is trained. We could see that pseudo data roughly follows the same distribution as test data in log density but displays a fatter tail than test data. Signals were well separately from both test data and pseudo data, supporting the statement that log density is a good discriminant.

Other than the hyper parameters of neural net training (learning rate, momentum, weight decay, shape parameters etc), another factor that has been found to influence NADE training is the shape of the distributions input variables. Through trials and errors, it became evident the more Gaussian-like the training data are, the better the modelling performance. With some datasets, due to the cuts performed at the pre-processing stage, there are sharp edges in some of the variables. As an attempt to make the datasets more smooth, at least in the individual variables, a set of functions were used to transform each variables. See Appendix B for the functions used.

Initially, to keep all the data together, the missing values were placed as a noisy distribution away from the true values. Later on, with the transforms applied, it became problematic as to where to put the noisy distribution. For example, eta follows a symmetric distribution around zero and putting the noisy at either side would create an artificial bias. Hence it was decided that NADE testing would only be done with boxed datasets.

To test whether the modelling ability of NADE is influenced by the amount of training data, i.e. whether NADE can "extrapolate" into tails of the distribution, the ratio of

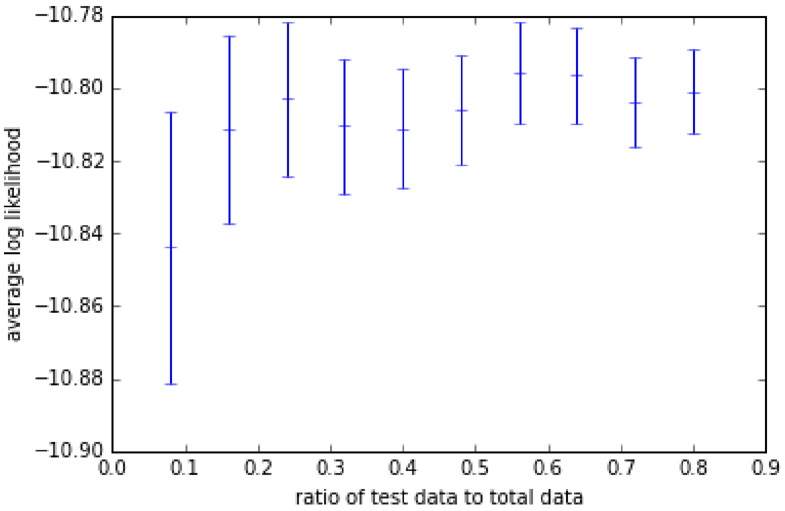

Figure 5: The mean and error of the log likelihood values were obtained by selecting different subsets of the remaining 80% of the total data. From the graph it can be seen that log likelihood does not become more negative as the ratio between test data and training data increases, implying that the modelling ability of NADE does not vary.

training data to test data was varied to see the change in average log likelihood of dataset and the distribution of test data vs pseudo data in log density. The entire dataset of $W(\ell\nu)$ + jets (all data with zeros as noisy distributions and no cuts) was split into 20% training data and a fraction of the rest as test data. The plots in figure 5 and figure 6 supports the postulate that NADE can cope with lack of statistics and that the modelling ability does not decrease as more data are drawn.

6.1 Background Estimation of SOM with NADE

For the likelihood test, we will need to know the contribution of all the known process and simulated signal in each cluster identified by SOM. Since the signal cluster identified by the SOM will most likely be irregular, and the mapping to the SOM is unknown, there is no easy analytical way to obtain the amount of background in the box. Having shown that modelling ability of NADE does not appear to depend on the amount of training data, initially it was hoped that pseudo data generated by NADE could be used in place of Monte Carlo data as background estimation.

To test the agreement between NADE pseudo data and Monte Carlo data in the context of Self Organising Map occupancies, a SOM was trained with three types of backgrounds with 0.1% signal injection to obtain the expected amount of background count (b_{train}) at each node. Pseudo data of the background 100 times the amount of training data were then passed into the train SOM to see node occupancies (b_{pseudo}). Assume that the count

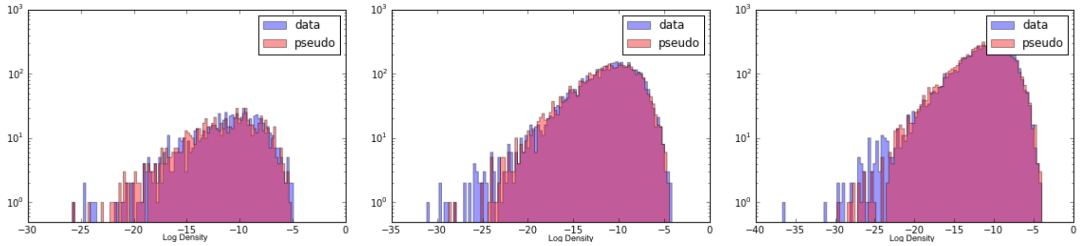

Figure 6: Three graphs of the distribution of test data and pseudo data in log density are shown. From left to right, the training data to test data ratios are 2.50, 0.50 and 0.28 respectively. The three graphs look identical by eye, suggesting that the quality of pseudo data does not vary.

at each node is Poisson distributed. λ can be approximated by b_{pseudo} due to large amount of excess. Various statistics were compared between a toy Poisson distribution thrown around the λ values and b_{train} values. For reference purpose, the same processes were also carried out with Monte Carlo background data 20 times the amount of training data. From figure 7 it is clear that NADE pseudo data are not a good enough approximation to the data it is trained on in the context of Self Organising Map.

6.2 NADE logdensity

Even through the log density produced by NADE is merely an approximation of the true log density and at the moment we cannot yet put statistical bounds on how good this approximation is, we could still use the log density as an indicator to distinguish signals from backgrounds.

It was found that the ordering of the input variables affect testing. Similar to the choice of binning, the ordering is also something we could choose by repeated trials until the ordering that maximises either exclusion limit or discovery limit is found. In addition, to deal with zeros, all the zero bins were combined with the bin with larger (less negative) log likelihood until a nonzero bin was created.

7 Results and Analysis

The best limits set by SOM, NADE log density and Razor Variables are listed in the tables. The sets marked by ”no razor” are those trained without the Razor variables.

The discovery limits are uniformly larger than exclusion limits as expected. The performance of the Self Organising Map is close to that of the Razor Variables, while NADE log density sets much looser limits than both.

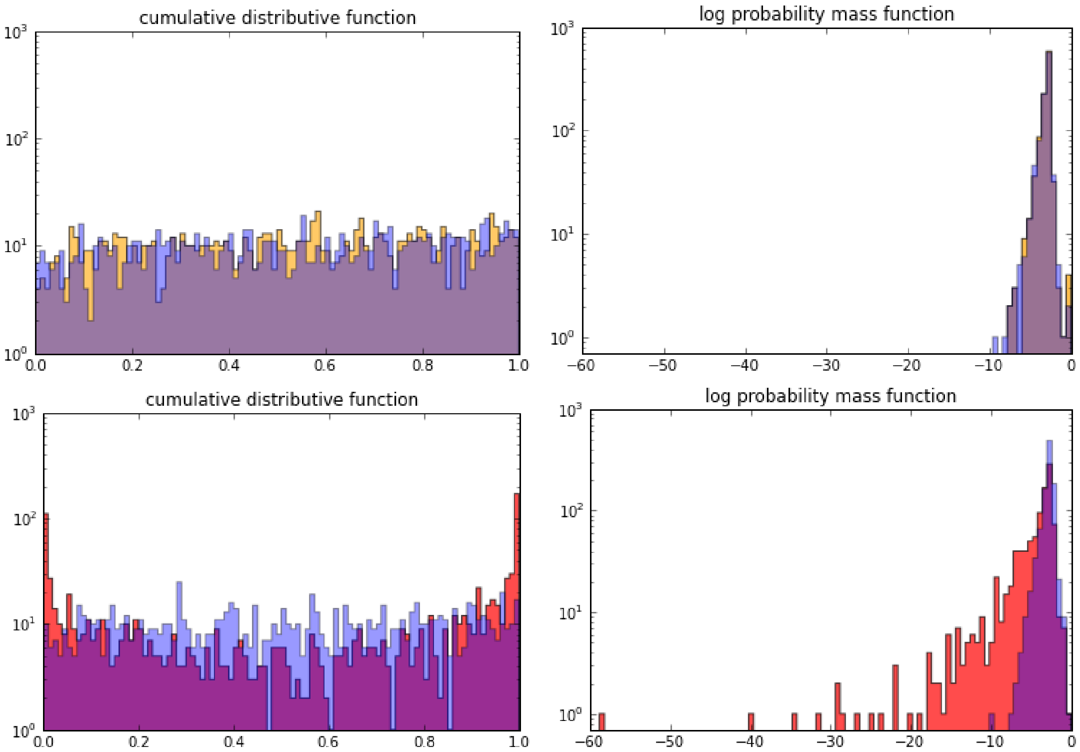

Figure 7: The top two plots compare a toy Poisson distribution with $\lambda = b_{MC}$ and b_{train}; the bottom two plots compare a toy Poisson distribution with $\lambda = b_{MC}$ and b_{train}. Good agreements between toy Poisson and Monte Carlo can be seen in both cumulative distribution function and log probability mass function, whereas for NADE pseudo data there are more data points with cdf near 0 or 1 and large negative log(pmf) values, implying that mismatch is more than a statistical fluctuation

Data selection	SOM	NADE	Razor
Hadronic, R2 0.1	0.65 ± 0.05	1.792	0.467
Hadronic, R2 0.1, no razor	0.621 ± 0.017	/	/
Hadronic, R2 0.05	/	1.376	0.377
Hadronic, metPt 65	0.46 ± 0.13	1.471	0.391
Hadronic, metPt 65, no razor	0.428 ± 0.019	/	/
Hadronic, no cuts	/	1.376	0.679
All, R2 0.1	0.89 ± 0.04	/	0.675
All, R2 0.05	/	/	0.880
All, metPt 65	0.67 ± 0.05	/	0.423
All, no cuts	/	/	0.455

Table 1: Exclusion limits ($-2\lambda_{exc} = 2 \times 1.92$)

Data selection	SOM	NADE	Razor
Hadronic, R2 0.1	1.67 ± 0.14	4.600	1.203
Hadronic, R2 0.1, no razor	1.61 ± 0.04	/	/
Hadronic, R2 0.05	/	3.543	0.969
Hadronic, metPt 65	1.21 ± 0.04	3.784	1.010
Hadronic, metPt 65	1.11 ± 0.05	/	/
Hadronic, no cuts	/	3.544	1.742
All, R2 0.1	2.27 ± 0.04	/	1.739
All, R2 0.05	/	/	2.271
All, metPt 65	1.730 ± 0.012	/	1.092
All, no cuts	/	/	1.174

Table 2: Discovery limits ($-2\lambda_{dis} = 5^2$)

For both the Self Organising Map and Razor, cutting the datasets helped setting tighter limits. In terms of the specific cuts used, the Self Organising Map performs best with metPt cut at 65 GeV whereas no obvious conclusions can be drawn for NADE and Razor.

Interestingly, even when the Razor Variables are masked, the Self Organising Map reaches similar limits within error. This suggests that Self Organising Map could arrange the itself such that it stretches along the a subspace that resembles the Razor plane in terms of sensitivity to signals.

The background estimation for both NADE and SOM are done with Monte Carlo simulated data, which have known to be inaccurate. Various methods have been devised to offset this difference. In the Razor Variable search, other than data-driven sideband fit, the control region method has also been proven to give good background estimation. The control region method requires finding two types of events with the same kinematics properties, one covering the signal region and the other known to purely background. Monte-Carlo-to-data scale factors in the razor plane derived from the control samples are then applied to the corresponding signal samples. Fundamentally, each node in the SOM corresponds to some irregular region in the input space, so are the bins in NADE log density. If we can prove that the two types of events indeed have the same Monte Carlo imperfection in the input space, this method can also be applied.

8 Conclusion and Outlook

For this particular kind of signal, the best limits set by the Self Organising Map have not exceeded razors; however since its clustering ability does not depend directly on the type of signal, for signals that do not yet have sensitive hand-engineered kinematic variables, the SOM could be applied to achieve similar sensitivity with low-level variables easily obtained from detector data. Further more, in the situation where the type of signal searched for is unknown, simply by doing the background estimation on a SOM trained on the detector data and testing which type of signal Monte Carlo fill the under-occupied nodes could give us some information as to what we should be looking for.

A multi-dimensional density estimator has great potential in particle physics. Although NADE does not given perfect estimation of densities, some systematics errors could be set to account for the difference. Despite the fact that NADE log density appears to have high discriminating ability (figure 4), in the context of binned likelihood test, the performance is not as well as Razor and SOM. More test data might help with likelihood test as the tails will be filled, where the signals usually lie. In addition, NADE log density plot has the potential to be used as a preliminary survey to hand-pick the anomalies for manual inspection to see whether the data point is a detector fault or obvious signal.

9 Acknowledgements

I would like to thank all my mentors for their helpful advice and guidance. I am grateful to Dr. Maurizio Pierini and Dustin Anderson for their insightful comments, and Dr. Adolf Bornheim for the air conditioner in his office.

10 Appendices

A Data

The Monte Carlo datasets used were given to me by the Caltech CMS group, stored under the following directories on eos file system at CERN:

Dataset	Path	Cross Section(pb)
QCD	eos/cms/store/group/phys_susy/razor/run2/Run2RazorNtupleV1.14/MC /RunIISpring15DR74_50ns /v3/sixie/QCD_Pt_170to300_TuneCUETP8M1_13TeV _pythia8/Run2RazorNtuplerV1p14_ToCERN_MC_50ns_RunIISpring15DR74-Asympt50ns_MCRUN2_74_V9A-v2_v3_v1/150724_042550/0000/	117276
$Z(\nu\nu)$+jets	eos/cms/store/group/phys_susy/razor/run2/RazorNtupleV1.5/PHYS14_25ns/v7 /sixie/ZJetsToNuNu_HT-200to400_Tune4C_13TeV-madgraph-tauola/razorNtuplerV1p5_PHYS14_25ns_v7_v1/150212_174727/0000/	100
$W(\ell\nu)$+jets	eos/cms/store/group/phys_susy/razor/run2/RazorNtupleV1.5/PHYS14_25ns/v7/ sixie/WJetsToLNu_13TeV-madgraph-pythia8-tauola/ razorNtuplerV1p5_PHYS14_25ns_Phys14DR-PU4bx50_PHYS14_25_V1-v1_v7_v2/150603_201201/0000/	60290*1.0195
TTJets	eos/cms/store/group/phys_susy/razor/run2/RazorNtupleV1.5/PHYS14_25ns/v7 /sixie/TTJets_MSDecaysCKM_central_Tune4C_13TeV-madgraph-tauola/razorNtuplerV1p5_PHYS14_25ns_v7_v1/150212_174432/0000/	424
SMS-T1bbbb	eos/cms/store/group/phys_susy/razor/run2/RazorNtupleV1.5/PHYS14_25ns/v7 /sixie/SMS-T1bbbb_2J_mGl-1500_mLSP-100_Tune4C_13TeV-madgraph-tauola/razorNtuplerV1p5_PHYS14_25ns_v7_v1/150212_173952/0000/	0.014

B Transforms

Here's a list of transforms I used on the the variables. cut_{metpt} refers to the cut on metPt and cut_{r2} refers to the cut applied on r2.

Variable x	Function f(x)
jetPt	$log(x - 39.5)$
jetEta	$arctanh(x/2.4 \times 0.99)$
jetMass	x
metPt	$log(x - (cut_{metpt} - 0.05))$
sumMET	$log(x + 1)$
MR	$log(x)$
R2	$log(x - (cut_{r2} - 0.0001))$

C Code

Here's the github repository for the codes written for this project. `https://github.com/Irene-Li/susyML`

References

[1] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5, 2014.

[2] CMS collaboration. Interpretation of searches for supersymmetry with simplified models. *Physical Review D*, 88(5), 2013.

[3] CMS Collaboration. Search for supersymmetry with razor variables in pp collisions at $\sqrt{s} = 7$ TeV. *Phys. Rev. D*, 90:112001, Dec 2014.

[4] CMS Collaboration. Search for supersymmetry using razor variables in events with b-tagged jets in pp collisions at sqrt(s)=8 tev. 02 2015.

[5] CMS Collaboration. Supersymmetry discovery potential in future LHC and HL-LHC running with the CMS detector. Technical Report CMS-PAS-SUS-14-012, CERN, Geneva, 2015.

[6] Luke Percival de Oliveira. *Deep Feature Learning for charm and bottom jet identification at s = 8TeV using the ATLAS detector at CERN.* PhD thesis, Yale University, New Haven, Connecticut, May 2014.

[7] Teuvo Kohonen. *Self-organizing Maps.* Springer Series in Information Sciences. Springer-Verlag Berlin Heidelberg, 3 edition, 2001.

[8] K. A. Olive et al. Review of Particle Physics. *Chin. Phys.*, C38:090001, 2014.

[9] Christopher Rogan. Kinematical variables towards new dynamics at the lhc. 06 2010.

[10] Maria Spiropulu. SUSY at LHC. *The European Physics Journal C*, 59:445–462, 2008.

[11] Maria Spiropulu and Steinar Stapnes. LHC's ATLAS and CMS detectors. *International Journal of Modern Physics A*, 23(25):4081–4105, 2008.

[12] Benigno Uria, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 467–475. JMLR Workshop and Conference Proceedings, 2014.

[13] Helge Voss. Successes, challenges and future outlook of multivariate analysis in hep. In *Journal of Physics: Conference Series*, volume 608, page 012058. IOP Publishing, 2015.