Version 1.0 as of April 24, 2014
Primary authors: Matthew M. Reid
Comment to mattreid99@gmail.com

Abstract

A multivariate analysis (MVA) can prove to be a powerful tool when dealing with highly non-linear datasets. These days they are common practice within the particle physics community such as the tools provied by CERN's TMVA analysis framework [1]. We provide an application using the TMVA framework applied to real financial market data. A simple buy-only trading stategy is employed on the SNP500 index using technical indicators as input parameters to the MVA. A MVA has many far reaching applications and we merely show one possibility, with other potential methods discussed in the conclusions.

Multivariate analysis using technical indicators

April 24, 2014

1 Introduction

`DISCLAIMER - If you decide to try and use this on real world trades then you do so at your own risk!! This is a toy strategy created simply as an academic exercise.`

Random phenomena dominate our complex world. With modern developments of high power computing and us entering an era where large datasets are openly available, more and more research is going into predicting what were once seemingly random processes. One attempt at such a feat is the use of neural networks to predict future stock market trends. We present a free C++ package you can install and then try for yourselves whereby you can employ your own neural networks (NN) on real market data using some wrappers to ROOT (`http://root.cern.ch/drupal/`). The strategy is based on buying a certain stock, in this case we are trading with a benchmark index known as the SNP500 (`http://us.spindices.com/indices/equity/sp-500`). The code is an extension to a wonderful package called Hudson written by Alberto Giannetti, he offers a GPL licensed end of day back testing framework using C++ stl, boost, ta-libs and gnu-gsl libraries (original code base is here `http://code.google.com/p/hudson/`) where you can load in various formats of csv file (Yahoo, Google etc). This extension allows one to perform a multivariate analysis (MVA) which can decipher linear and non-linear trends between input variables of a given dataset and then test its performance.

2 Why a multivariate analysis?

A specific type of market trading utilises technical indicators, called technical analysis, as opposed to a more fundamental approach, trading decisions can be made solely on the quantitative assessment of indicators. These indicators are built using historic data, such as the end of day close, high, low, volume `etc` under the assumption that the underlying market trend will persist. A list of useful technical indicators can be found at `http://ta-lib.org/function.html`, but there are many websites online posting far more. As an example we can consider the moving average convergence divergence (MACD) indicator (see here for an overview of the indicator and its usage, advantages and disadvantages `http://www.investopedia.com/articles/technical/082701.asp`). Various authors suggest indicators such as the MACD can be used by implementing specific trading rules, such as looking for cross over lines of the two moving averages. Lets say we want to trade using the MACD indicator, for instance we create a rule whereby we buy in when the indicator is greater than 0.01 and sell the same asset when the indicator drops below 0.01. This essentially splits our dataset into two since it remove all days where the MACD indicator has a value that is less than 0.01. In particle physics such behaviour is call placing a "cut", since it cuts out data that is deemed uninteresting (more specifically we call such cuts "rectangular cuts"). We could build a series of rules like this to encompass more than one indicator, such as including an additional "cut", e.g. an exponential moving average (EMA) when it is greater than some arbitrary value, say 30. Now we have two rules(cuts), which must both be satisfied in order to make a return on our initial investment.

But what is to say that any one of these rules will work best for the stock you are trading, at what value of each of these variables should you place a trade? Also what happens when you want to use three rules, or more? Have we even considered the way in which the application of one rule may affect another, for instance, perhaps applying a rule on the MACD will eradicate any gains one might have expected from applying another rule or even have an adverse effect on your profitability and lose you money? Frankly, there is no one rule fits all else we would all be millionaires, stock markets are combinations of highly non-linear correlations in an extremely complex web of interactions across a global economy. The postulate of the day is; can we use historic data to obtain information about future market moves. How can we do this taking multiple indicators into account and maximise our return whilst minimising our risk. In fact multiple rules can be correlated and often are. So an ideal approach would be if we can somehow look at multiple technical indicators and determine the optimum decision using a mathematical description. This is where ROOT and the Toolkit for MultiVariate data Analysis (TMVA) framework comes into play. This allows us to use multiple input variables such as technical indicators and optimise a trading decision specific to a stock of interest. In no way are we limited to using just technical indicators [1].

3 Generating signals - particle physics and markets

In particle physics we often want to distinguish between signal (our particle decay) and background (random combinations of different particle decays that masquerade themselves as our decay mode of interest). In general we do this by applying a series of "cuts" to our dataset that removes bad events and keeps the ones we are interested in so that we can study them further. It is our job to distinguish the difference between the two; signal and noisy background events. So how do we compare a particle physics analysis with a stock market analysis? In this laymans example we must first define what we believe is signal and what we believe to be background.

[1] To highlight this we included an example utilising Google trend data which I may elaborate on at a later date. The method is based on a multivariate form of the analysis found in this reference [2].

A comparison with markets could be that signal is a bull (positive sentiment) market and background is categorised in a similar manner for bear (negative sentiment) markets. The actual categorisation used in this example is based on looking an arbitrary number of days ahead, we take 7 days for the period and call this period the *look forward period n*. We can use end of day close price, $p(t+n)$ and the current days price $p(t)$ and determine if the market goes up or down. Following this we define the relative change in the price by

$$\Delta p(t) = p(t+n) - p(t) \tag{1}$$

with signal and background characterised using

$$f_{\mathrm{S/B}}(t) = \begin{cases} \Delta p(t) > 0 & \text{Signal (bull)} \\ \Delta p(t) < 0 & \text{Background (bear)} \end{cases} \tag{2}$$

where f_S denotes a signal like period and f_B a background like period. In addition to this one can use the percentage change between the current day close price and that for the look forward period as a weighting. We define the weighting as

$$w(t) = 1 + \left| 1 - \frac{\Delta p(t)}{p(t)} \right| \tag{3}$$

where $w(t)$ is the weight at time t and is always greater then unity. It is useful to apply some weighting to our dataset to make light of the fact everyday will not be the same and in some cases, large market movements should be given large weights. As an example, if the market goes down by 30 % after the look forward period then the weighting would be 1.3 and thus the values of the variables that cause this decrease are considered more important compared to some set of variables where the decrease was only 5 %. More complex weighting systems could be devised based on a separate evaluation of the dataset. Our approach, albeit very simplistic, is model independent and data driven.

The biggest difference between the markets and what we may do in a particle physics experiment is that physical laws of the universe stay the same, or so we may believe, whilst those of the free market do not. The stock market is a dynamic moving entity. What follows in this "paper" assumes that market conditions are static which is not the case in reality. In particular we can have large periods of volatility caused by global recession, terrorist attacks, tsunamis are just a few things that clearly cannot be easily predicted and may affect the psychology of peoples trading habits. Thus you will almost always be susceptible to losses. The task we must then do is minimise these based on what we learn by applying a MVA to our data and use this in conjunction with good judgement and common sense. For future reference we now refer to the output decision our MVA produces as the *MVA response*.

4 The strategy

We employ an extremely simple trading strategy for this analysis. The approach to this analysis follows some simple steps

- Apply some simple data driven logic for signal generation (`i.e.` using the 7 day look forward period).

- Train a MVA on multiple variables calculated using historic financial data.

- Implement a simple trading strategy that uses an optimised output from your MVA.

These three basics steps will allow us to train a MVA on real data. The first has already been addressed as we are using a 7 day look forward period and then weighting variables accordingly. The generation of the $f(t)_{S/B}$ allows us to build up two independent datasets, one for signal (bull) markets and one for background (bear) as shown in Eq. 2. Due to the nature of the signal generation when the relative change in price is very close to 0, there is some abiguity as to which category the data belongs to. To get around this and the fact there could be statistical fluctuations causing signal and background to look similar about zero we apply a separating cut such that the weight, given in Eq. 3 is greater 1 % for both signal and background (this really should be investigated further). Once we have these two datasets with the choice of variables we can feed them into the machinery of TMVA where we can choose from a series of MVA methods to run. Finally the optimisation of the MVA output response need be fine tuned, this really depends on what you as the investor care most about. For this example we chose a minimisation of risk factors and attempt to maximise the overall return.

4.1 SNP500 dataset

Our data is obtained from yahoo finance using the SNP500 over a total period ranging from 2001-2013. There are several subsets of the SNP500 dataset that have been chosen for this analysis, it is completely arbitrary and no optimisation went into the choice of dates, other than simply trying to keep it current (in the 21st century) with recent data. There are three main periods that we utilise;

1. *Training* 2001-01-01 to 2010-01-01 – train a MVA response using the TMVA toolkit.

2. *Back testing* 2010-01-02 to 2012-12-31 – evaluation of the response to find optimal cut value (risk Vs reward).

3. *Live trading* 2013-01-01 to 2013-12-01 – test our network on real "live" market data.

The main point behind the first two is that we do not want to use the data we trained on to select our optimal MVA response. If you think about it you will agree with this statement as it is silly since the MVA will already be optimised on the *training* dataset, hence the title of said dataset. Once the MVA is trained it will be able to produce an output/response given some initial input parameters. In order to choose the correct MVA response value we chose an independent period whereby we can see how our MVA response performs for a sub-sample of our live data, this is for 3 years from start of 2010. Finally, we switch to the live trading period. For the sake of this example the final data sample running over this last year is essentially real data and would be the output if we were to make live trades using this strategy. In summary, you will notice that there are no overlaps between any of these data samples to avoid introducing a bias (avoid testing on already optimised data).

5 Technical variables and correlations

The choice of variables to use in a multivariate analysis is up to you, we have not checked if there exists some universal magic combination of indicators that work for most markets, or indeed if you need a unique set. These variables give decent performance but are in no way the best or final answer. Note they're all technical indicators and thus not necessarily the best choice, there is a bunch of interesting analyses trying to use social media such as News report scrapping, Twitter posts, Google trends and other such input that may shed light onto market sentiment to help pick out market inefficiency. There are 7 applied in this example which are

- BOP Balance Of Power
- ADX Average Directional Movement Index
- APO Absolute Price Oscillator
- STDDEV Standard Deviation
- CCI Commodity Channel Index
- ADO Chaikin A/D Oscillator
- MACD Moving Average Convergence/Divergence

The interested reader may Google search any of these for more information, there are plenty more out there which can be added to the list if required (although too many variables can create too complex a situation and give poor performance). After collating the data you have to ask are there any linear correlations between these input variables? Figure 1 shows the correlation between the signal like variables and the background like ones respectively. If a variable is strongly correlated with another it means that it is heavily dependent on the value the other might take and so they are essentially the same variable. However we could have non-linear correlations that are not taken into account by this method. The variables presented are of a reasonable level of linear correlation, remove any that are extremely correlated (in general, anything over $\pm 90\,\%$ would not be needed).

5.1 Viewing variable distributions

Once the variables are chosen, TMVA provides some scripts you can use to determine the best ones. A powerful test you can do visually is to look at the distribution of the

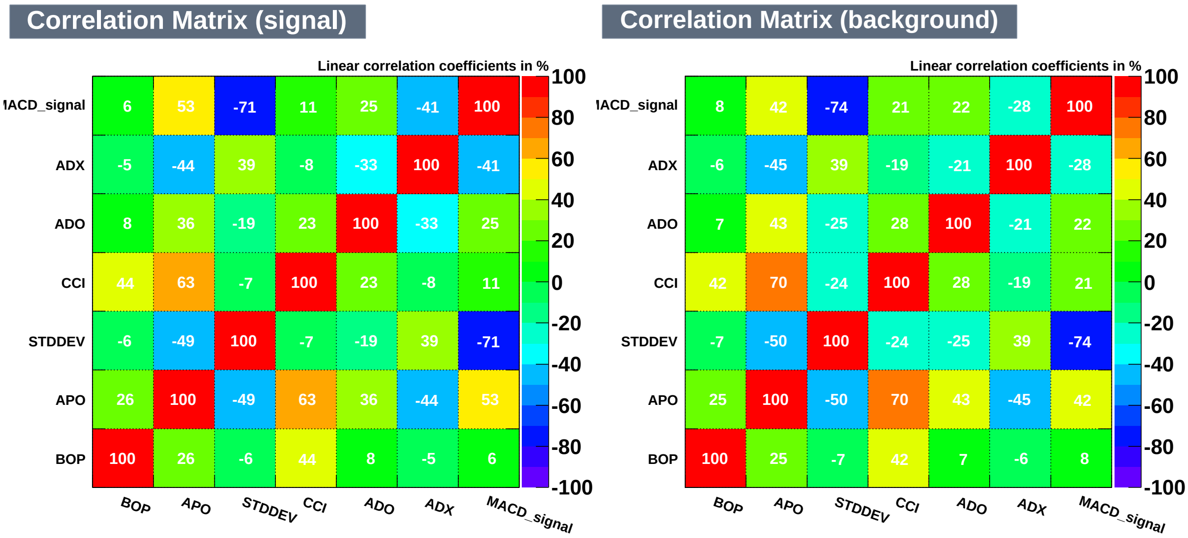

Figure 1: Correlation matrix for signal like (left) and background like (right) end of day close indicators.

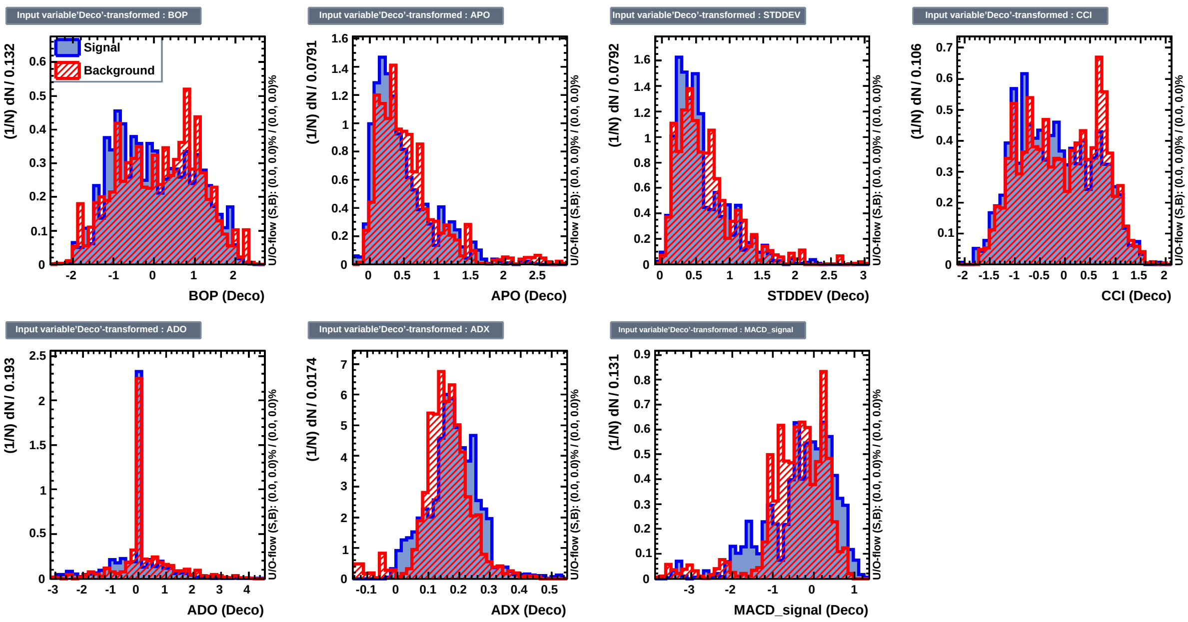

Figure 2: Separation between signal and background like variables after a decorrelation transformation has been applied.

signal and background variables after certain transformations have been applied such as a decorrelation via the square-root of the covariance matrix as seen in the next plot 2. In general the distributions for signal and background in figure 2 do not appear to have a large separation and remain quite similar. You can also apply multiple transformations

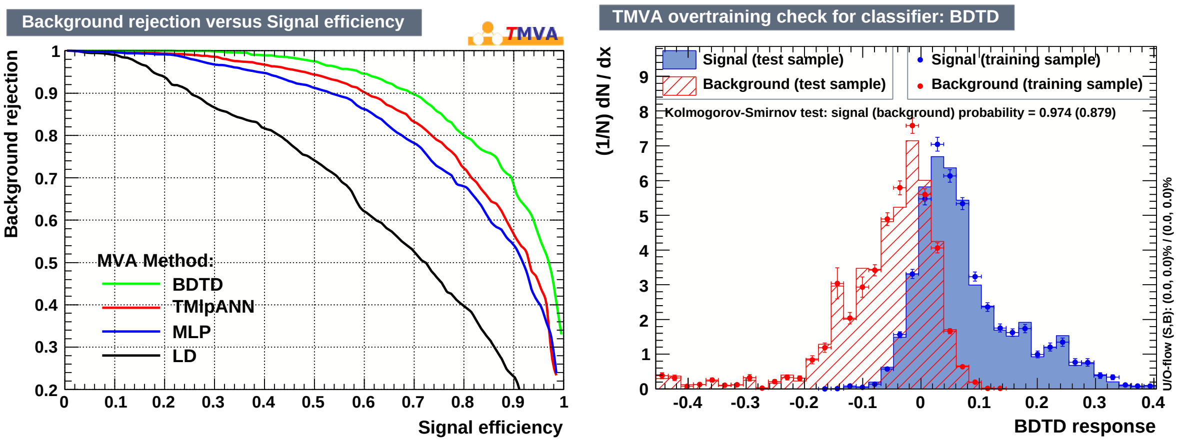

Figure 3: ROC curve for various MVA methods applied (left) and overtraining cross-check (right) for BDTD specifically.

one after another, such as a principal component analysis, Gaussian transformation or normalisation as looking at your dataset from different perspectives can sometimes separate the signal and background distributions.

6 Training the MVA

ROOTs TMVA package provides a whole plethora of built-in MVA types, from simple linear regression analysis to more complex neural networks. For reference on the inner workings of these and many other types available in TMVA check out the reference manual [1] (there is also a whole stack of references at the back regarding the theory of machine learning algorithms for instance!). Our MVA is trained over a 9 year period, from 2001-01-01 to 2010-01-01. There is no immediate logic for these dates, it encompasses a large set of data under several financial crashes, at the same time it is hopefully a large enough time period that the underlying trends will remain consistent, which is in itself, an interesting point to explore further. A small amount of this data is kept back for internal testing in TMVA. This testing is required to make sure any such analysis is not over trained. Over training is when your network becomes sensitive to statistical fluctuations and thus not the overall trend and can occur with neural networks or boosted decision trees if one is not careful. Luckily, there is a built-in cross check which we will discuss in a moment. In the example we train 4 separate networks, these are, BDTD, TMlpANN, MLP and LD. Do not worry about the names its just convention but they can all be found with names and how to optimise them on-line `http://tmva.sourceforge.net/optionRef.html` and can be tuned manually. The performance of these networks can be seen in figure 3. This is known as a receiver operating characteristic (ROC) curve and essentially plots the ability of each trained network to retain signal events and reject background. The best performer here is the BDTD where the closer the curve tends to the top right hand corner the more efficient the network is at its job.

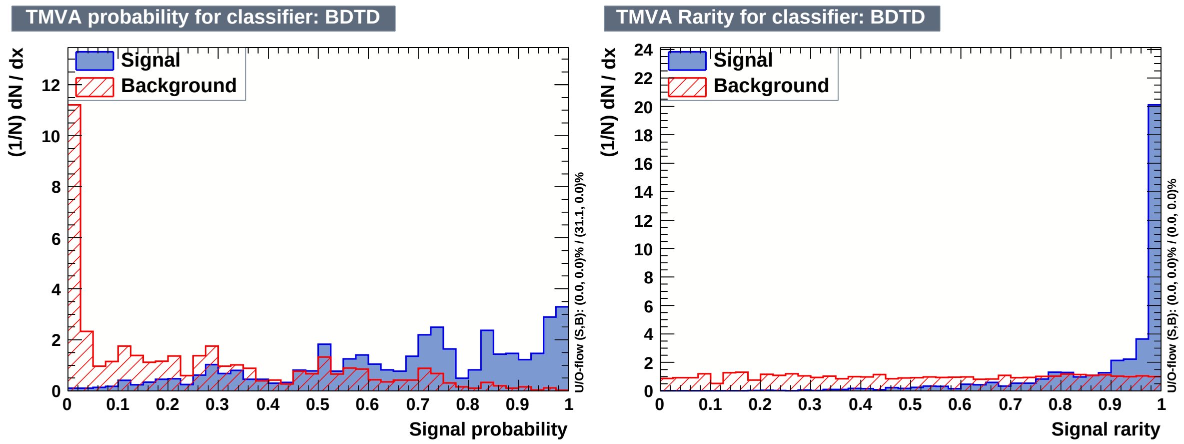

Figure 4: Signal probability (left) and signal rarity (right) classifiers displayed for signal (bull data) and background (bear data). Estabilishing how well our MVA can distinguish between the two.

TMVA has a built in over-training cross check tool that essentially takes a subset of the data to test that the network response is the same as that which it was trained on. If the data is stable and the network is not over trained then the value of this test should be close to 1 for both the signal and background testing samples, see figure 3. The test is called the Kolmogorov-Smirnov test [3] which compares two distributions and returns a probability of how similar they are. In this case that the training sample has the same distribution as the testing sample for both signal and background. These two values at the top right are 0.974(0.879) which are both close to 1 implying theyre almost identical, this is very good to see and means you should be able to rely on your network in the assumption that the data used to train upon is a realistic representation of future events [2].

7 MVA performance

We saw in figure 3 (left) that the best performer was the BDTD method. A different way of testing the performance of this method can be seen in figure 4, which is simply the number of signal and background events plotted as a function of the signal probability (left side) or in other words the probability the MVA will select a bullish event, and secondly (right side) as a function of signal rarity. For the signal rarity, the background should be flat and the signal events should bulk up towards 1, which they seem to do from inspection of figure 4. These two plots tell us that our network is behaving reasonably well so far. We must now make a choice on the cut value of the MVA. At this point a MVA has been trained and thus a decision on each day of the market can be made using the BDTD response. Instead of applying a trading rule such as sell if the $MACD > 10$,

[2] We discuss this further in the conclusion.

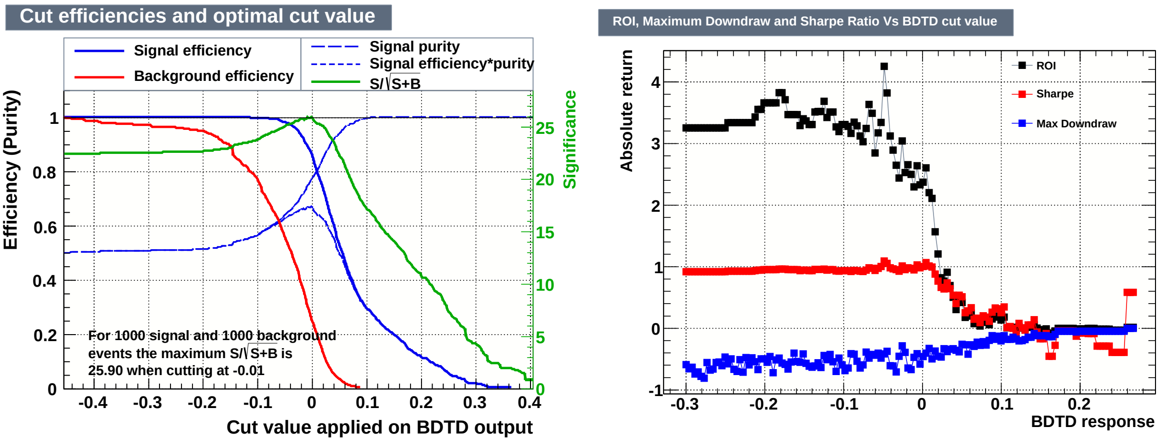

Figure 5: Efficiency of the trained BDTD network as a function of the BDTD response (left) and the ROI, sharpe ratio and maximum downdraw (all superimposed) as a function of BDTD response cut value (right). Note that the (left) is created using the training sample (2001-2010), (right) is a based on the testing sample (2010-2012).

the MVA takes all variable values for a given day as input and computes a response that allows you to cut on a single value. Essentially, I take variables x, y, z feed them into a machine and outcomes a single value m that has in intrinsic relationship to x, y, z. Up to now, we have not decided on a cut value to use for our BDTD. Figure 5 (left) shows the efficiency as a function of the MVA output for data used in the training phase. However since we are never going to use the data from the training phase (and to avoid introducing a bias) we switch to the testing data sample (2010-2012) to determine our actual MVA cut value. Figure 5 (right) shows the return on investment (ROI), sharpe ratio (with risk free rate of, $r_f = 3\,\%$) and maximum down draw all as a function of the MVA response cut value. As with all well trained MVAs using the TMVA framework, the larger the value of the netowkr response the larger the more signal like the event should be. We choose the cut value by visual inspection here and pick a value of 0.016 as it maximises the ROI (157.7 %) and sharpe ratio (0.9) but minimises the maximum down draw (36.21 %). We come back to this risk optimisation in the conclusions. Since this is our last chance to check the MVA before going live we can inspect the testing region in a little more detail by breaking it up as shown in section 8.

8 Closer inspection of the testing period

In this section we evaluate the testing period by breaking it up into smaller regions, this gives us snap shots of how our MVA could perform on live market data. Clearly from figure 5 we have shown that the BDTD response can make much larger returns for our chosen cut value over the entire testing time period. Inspecting smaller periods is a good way to see just how dynamic the market is and helps to highlight where our MVA perform better in some situations over others? We can use this testing period and examine it in a

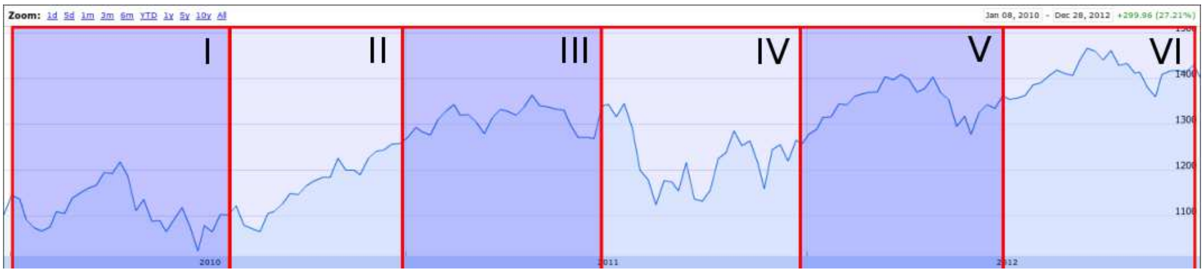

Figure 6: The testing period in which we optimise the neural network response where the dataset has been split into six.

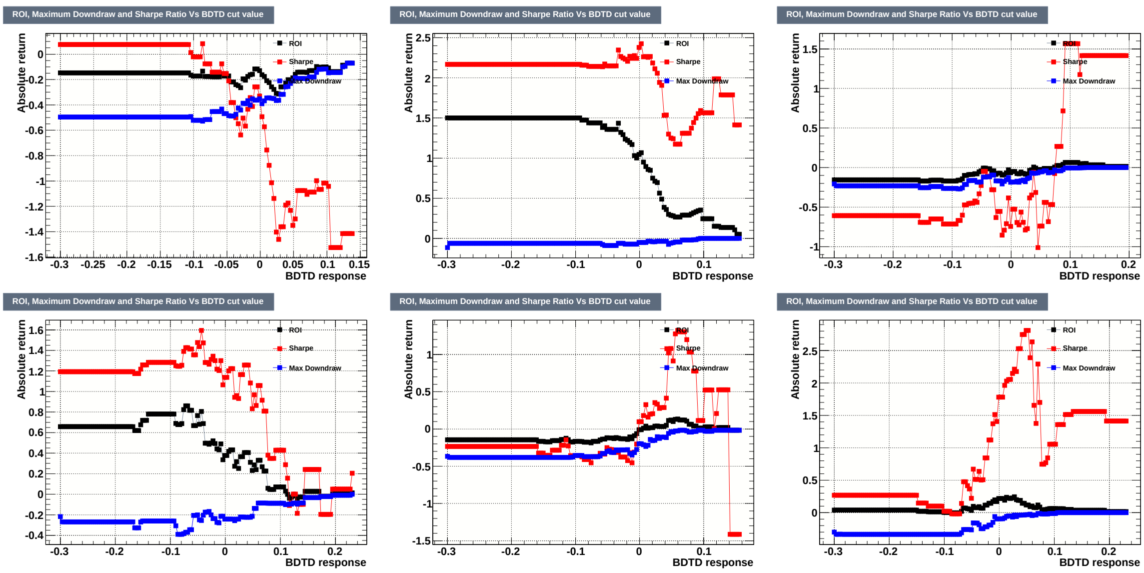

Figure 7: Shows the BDTD response for various cut values as functions of ROI, Sharpe and MaxDD. These plots relate directly to the size areas shown in figure 6 where the top row going from left to right counts regions I-III and the bottom row for regions IV-VI.

little more detail by checking how it varies over time. For this we divide up the training period into six separate regions as shown in figure 6. For this period we show the BDTD response in the same way we did for the total time period (figure 5) but for each of the six periods in turn.

All of the time slices can be found in figure **??**. Interestingly we can see that for some large values of the BDTD response we can begin to get negative returns. In this period it seems that the MVA find some events to have a high probability of being signal but in fact they are not very signal like and do not provide a positive return. This is a classic example of where to be careful with a network trained on volatile data. Simply put, you cannot cut too tight because the market conditions may change in the opposite direction to that which you trained your network on, but equally you cannot cut too loose else you buy in too often becomming exposed to larger risk.

During 2010-2012 many trend following quantitative hedge funds (CTAs) found it difficult make returns under such market conditions. These were extraordinary times of government quantitative easing and other circumstances that can break trends due to market intervention; in attempts to respond to the financial crash of 2008. However, what we do see is that at times of market rally, the MVA is able to capitalise rather well. The return on periods II, IV and VI are in general pretty good, the alternate period returns have negative returns or are very small, but the overall picture for the full period as shown in figure 5 shows we can make a good return over the entire testing period. The point of checking the variation over different time periods is to see if the network can be susceptible to losses in a way we might not have anticipated. It appears for some periods we will indeed occur losses and this should be accounted for in the maximum downdraw of 36.21 %. The method of training on bull and bear markets as an entwined figure of merit means we will always be susceptible to losses. The figure of merit used for the training is something one must thing hard about, equally as to how one optimises the network response. We chose a method that seems logical and easy to implement given our 7 day look forward period, this is most likely too simplistic.

9 Results

Taking a neural network value of 0.016 the BDTD network performance is reasonable over the live trading year and would provide a return of 38.18 % from 2013-01-01 to 2013-12-01, for the same period the SNP500 changed 23 %, making you a modest +15 % on the benchmark. Here are the results and statistical interpretation of running our stategy on the live data set is given in table. 1.

10 A randomised strategy

It is useful cross check to see how well a random strategy would have done on the market data. For this we take a uniform random number generator $[-1, 1]$ and then trade the stock if the random number generated is greater than our cut value of 0.016. This equates to roughly a 50 % probability of a trade occurring each time the generator is called. What we then do is run this randomised method over all the data 200 times. From the output of these 200 pseudo-experiments we assume that it will be Gaussian distributed (it is not in some cases and and exhibits some skew hence we provide the error associated with using this fit model) and provide the results in table 2, The performance here is much better than our MVA response in terms of risk, however, this is due to the market being bull and is based on no a priori knowledge of that fact. It would be hard to convince investors that such a random strategy is really a good idea... There is also still plenty of room for improvement with this network. An additional test was performed just to see what happens if one was to trade with no strategy at all, a strategy where by we buy in every day and sell after the look-forward period (one week later). The return on this type of strategy shows us that the market was bull with returns hitting 185.69 %, sharpe

Table 1: Backtesting information for a BDTD cut value of 0.016. This strategies has been applied to the “live” data period from 2013-01-01 to 2013-12-01.

Parameter	BDTD Strategy
Trades	79
Avg trade	0.42 %
Std dev	1.66 %
Skew	7.05
2SD Range	-2.90 % | 3.75 %
3SD Range	-4.56 % | 5.41 %
Pos trades	47 (59.49 %)
Neg trades	32 (40.51 %)
Avg pos	1.53 %
Avg neg	-1.20 %
Best	3.96 % [2013-Jul-03/2013-Jul-12]
Worst	-3.26 % [2013-Jun-17/2013-Jun-25]
Max cons pos	9 [2013-Jun-24/2013-Jul-12]
Max cons neg	7 [2013-Oct-02/2013-Aug-16]
Max drawdown	-9.40 % [2013-Aug-13/2013-Aug-08]
Adverse Excursion	
Average AE	-1.13 %
Worst AE	-4.77 % [2013-Jun-18/2013-Jun-24]
Favorable Excursion	
Average FE	1.44 %
Best FE	3.93 % [2013-Oct-09/2013-Oct-16]
ROI	38.18 %
CAGR	42.39 %
GSDm	24.40 %
Sharpe	1.48 %

ratio: 2.26, CAGR: 214.93 %, GSDm 65.44 % and Max drawdown -21.74 %. Again no investor would trust such a strategy but it shows there could be gains to be made should one tweak the knobs of the TMVA toolkit further.

Table 2: Parameters after a Gaussian ($\mathcal{N}(\mu, \sigma)$) fit to the distributions of the ROI, sharpe and maxDD after running 200 pseudo-experiments using a randomised trading strategy. Hence we quote the mean and sigma and associated uncertainties on these fitted parameters.

Parameter	Mean (μ)	Width (σ)
ROI	$38\,\% \pm 2\,\%$	$\sigma_{ROI} = 17\,\% \pm 2\,\%$
Sharpe	1.74 ± 0.08	$\sigma_{Sharpe} = 0.71 \pm 0.10$
MaxDD	$-6.0\,\% \pm 0.3\,\%$	$\sigma_{MaxDD} = 1\,\% \pm 0.2\,\%$

11 Conclusion

I am not for one moment suggesting that this is good performance, this is merely an exercise that shows how one can use an MVA in order to obtain a return even in extreme market conditions. The first and foremost aspect of this statement is that it is not best practice to trade an index, such as the SNP500 or FTSE100. The FTSE100 is based on 100 companies with the highest market capitalisation (the net worth of the shares issued by a company) on the London Stock Exchange, this does not mean that these stocks are the safest in terms of their volatility. For instance Lloyds banking group (as well as several other major banks) before the financial crisis has a share price of 600p/share, however after and still today it remains below 100p/share, a loss of 600 % in just two years. While this is just an example, such wild volatility means that big is not always better and even though in general the companies at the top will perform well, a more selective group of stocks could be chosen with much smaller volatility and thus risk. Finally one would want to capitalise on the market what ever the sentiment thus providing absolute returns for investors. The current strategy is not designed for this but could easily be implemented.

The ROI I quote is when choosing a BDTD cut value that maximises the signal yet not necessarily minimising the background. Thus you will always be exposed to risk, this is clearly unavoidable since you never have all the information at any one time. However you can minimise the loss by increasing the value at which you cut for you analysis, clearly a consequence is a reduced ROI. So I guess I have to ask what is a good volatility and maximum drawdown that would be deemed reasonable to an investor? If this measure is deemed a reasonable way to approximate the risk then of course one could find an optimal regression for the cut value of the MVA as opposed to the selection we obtain by eye as we have done.

I should point out that the parameters used for training these networks are not optimal and are thus using the default setting specified by the makers of the code (TMVA). This limits the performance a lot and is left to the eager to optimise further, feel free to post any results on this forum. Furthermore, the over simplification of the strategy quite horrendously limits your market participation, such that we can only "buy" in and are forced to exit exactly a week later at that days opening price. The list of variables used to train the network were chosen for their significance, they're a small enough list that the network converges well and provides good results but the user should be aware that there are many other variations one could try. What this network does not do

- The output of the network assumes that market conditions will remain the same, i.e. the dynamics over the period that you used to generate your training dataset, assumes that the underlying variable distributions will not change in the future. This of course may not be true, and is exactly why we did the testing phase and over training cross check. Perhaps one should train many networks under different market conditions or by binning in discrete time sets?
- You only have the past to go on and trends that have been seen to date, these trends may not be the same as peoples ideas and strategies evolve.
- The signal generation here (looking 7 days ahead) is simple and model independent, but signal generation need not be so simplistic. This is extremely limiting, your system is set up to sell after a pre-determined period of 7 days, which of course may not be a wise choice especially in a bull market.
- The strategy looses out completely on making short positions.
- We consider no transaction costs for a trade placed. This is not practical.
- Trading a benchmark is not necessarily a good idea. It is composed of the companies with the largest market capitalisation, thus not necessarily the smallest volatility and/or potential return for the risk. Also one cannot characterise a portfolio since they are usually ranked against a given benchmark such as this. So useful quatitative comparison tools such as the information ratio, tracking error, α and β become lost.

However, even with all these clear drawbacks, remarkably trends can be found in this data using multivariate techniques. The other use of such a tool is that it could be used solely as an indicator, `i.e.` it allows one to see the market direction. What we can clearly see is that the performance of the network capitalises on bull like market data. Hence such networks could be trained to aid the investor in picking out good times of positive sentiment and/or potential times of market inefficiency [3].

A network such as this may not be valid indefinitely, its effectiveness would need continuous evaluation, since like I say a market is more than likely not static and there will be periods of extraordinary activity (government intervention for instance). Remember the network will not be able to distinguish any new variations within the market since it was not trained to do so, thus if the market dynamics and/or correlations between the input variables change for what ever reason, the network may not be able to aid the investor and its effectiveness becomes invalid. It is for this reason that one should not rely solely on a network and strategy such as that presented as a dominant trading strategy.

[3] One neat transformation of the MVA output is that they can be converted to a more conventional probability on the bounded range $[0, 1]$, where by $P_{mva} = 0.5 * (1.0 + mva)$. This makes the output appear much more natural when talking about how bull like the market is given we can talk about it in terms of a probability measure.

However, it can clearly be used in addition to other techniques and strategies, or even using several networks.

11.1 Further work - ideas

This is really just a bit of brain dumping here. Comments are more than welcome!

One of the main issues with an approach such as this (in my opinion) is that the network is static and in the example is trained only on one period of data from 2000-2010. As has been mentioned market dynamics will change over time so it would be useful if one could do several things as a cross-check that our trained network is still valid. Unfortunately this requires rather large amounts of data. This really does raise an important question of when should one turn on a neural network. We have seen in figure 7 that the input parameters for the network response can vary over its lifetime.

1. continuous evaluation of the input parameters, their correlation and how similar they are to the training dataset.

2. Some of the TMVA networks can produce errors, this can be a useful indication of how accurate the networks thinks it is given a set of input parameters. One could likewise do this anyway by testing MVA value and seeing the return for a given input distribution for some range of variable spread.

3. Models - the introduction of models to generate additional input parameters could be very interesting here. For instance, derivatives markets (futures, options `etc`) are used throughout modern finance and the Black Scholes formula can be easily fitted to real market data (since the volatility is determined by a constant, σ). One could also try more complex models such as Heston model with an additional stochastic volatility term, although these are more difficult to set up quickly. Either way using models that are commonly found could aid future network trainings, inparticular is one wishes to trade on the derivatives market.

4. What do we care about? Is it to maximise the ROI whilst minimising the maximum drawdown? Maximal Sharpe ratio? Or is there some other set of risk variables more suitable that can be used to aid us in the choice of MVA response. One can pick any figure of merit they want, it just needs to be decided a priori.

5. Transaction costs and portfolio optimisation. I suppose this is really two separate items; we do not consider transaction costs of any sort and equally how much to bet the (Kelly criterion [4] could be a nice idea here with the probability interpretation of the network). Once we have a position we have no strategy in place to exit. In no way have we tried to make absolute returns using this simple strategy, of course this would be a wise investigation into the real benefit of a numerical analysis such as this.

6. A major point of criticism is that the extraction of trading rules from the use of a technical analysis is highly subjective therefore different analysts might extract different trading rules by studying the same data. We began this article by touching upon the idea of fundamental analysis. The analysts that use this method of prediction use fundamental data to determine if they will choose to invest or not. They are aiming to compute the "real" value of the asset that they will invest in by studying variables such as the growth, the dividend payout, the interest rates, the risk of investment and so on. Their objective is to calculate the intrinsic value of an asset (e.g. of a stock). One interesting idea could be the inclusion of fundamental analysis in a separate network which could be trained on correlated assets (`i.e.` from similar market sectors). This kind of study could be used for three things; the most significant and important fundamental varibles, as a trigger for when to start your main trend following network and picking out stocks that are at turning points. The down side is fundamental data is much more limited in supply when compared to daily stock quotes for instance, some data will only be released as a company announces its quarterly results.

7. Other input variables. We only provided an analysis based on equities. An interesting market sector to try networks would clearly be the foreign exchange market, which is also the largest market in the world. Here there are many input parameters one may try including; Gross Domestic Product (GDP), inflation, unemployment rates, government bonds, commodity prices `etc`.

8. Ultimately I think one could attempt to make use of several networks each performing a different task. In theory, one could have one tracking foreign exchange markets or bond markets, one following house prices, CPI and other domestic econometrics, and then some attempt at a sentiment indicator following Google trends for instance. All of these could be used to generate several optimised networks. It is also not uncommon to have networks trained on input from other networks, I like to call them super networks...

Most of all this is an extremely simple test setup. With a little more thought one could in theory find much better optimisations or methods for neural networks to become standard indicators for any financier.

References

[1] A. Hocker *et al.*, *TMVA - Toolkit for Multivariate Data Analysis*, PoS **ACAT** (2007) 040, `arXiv:physics/0703039`.

[2] T. Preis, H. S. Moat, and H. E. Stanley, *Quantifying trading behavior in financial markets using google trends*, Scientific Reports **3** (2013) .

[3] N. Smirnov, *Table for estimating the goodness of fit of empirical distributions*, The Annals of Mathematical Statistics **19** (1948) 279.

[4] J. L. Kelly, *A new interpretation of information rate*, Bell System Technical Journal **35** (1956), no. 4 917.