Particle Swarm Optimization of Bollinger Bands

Matthew Butler and Dimitar Kazakov

Faculty of Computer Science, Artificial Intelligence Group, University of York, UK
{mbutler,kazakov}@cs.york.ac.uk

Abstract. The use of technical indicators to derive stock trading signals is a foundation of financial technical analysis. Many of these indicators have several parameters which creates a difficult optimization problem given the highly non-linear and non-stationary nature of a financial time-series. This study investigates a popular financial indicator, Bollinger Bands, and the fine tuning of its parameters via particle swarm optimization under 4 different fitness functions: profitability, Sharpe ratio, Sortino ratio and accuracy. The experiment results show that the parameters optimized through PSO using the profitability fitness function produced superior out-of-sample trading results which includes transaction costs when compared to the default parameters.

Keywords: particle swarm optimization, Bollinger Bands, Sharpe ratio, Sortino ratio and parameter optimization.

1 Introduction

Technical analysis is a popular technique for modeling the stock market for price level and volatility estimation. Although these techniques can be very useful they often have several parameters which have to be decided upon which greatly influence the effectiveness of the trading signals produced. One popular indicator is Bollinger Bands (BB), as proposed by John Bollinger, which are considered to be a channel indicator, where the bands create a channel around the price movements which capture the majority of the fluctuations. In work conducted by Lento et al. [2] the authors found that trading rules developed from BBs were not profitable after transaction costs were accounted for and when generating trading rules based on several indicators the models were more reliable when BBs were excluded. The authors point out that there is a lack of attention from academia in studying BBs and that there is no theoretical explanation for using the traditional parameters settings. In Leung et al. [4] the authors compared the trading rules of two channel indicators the BBs and Moving Average Envelopes (MAE), the results indicated that the MAE tended to produce more profitable trading rules in the short-term which is the preferred time-horizon for using technical trading rules. In Lento et al. [3] the authors conclude that BBs are unable to produce profitable trading rules and that they consistently underperform relative to the buy-and-hold approach (a passive investment strategy which reflects the performance of the market as a whole). These studies all have a similar approach where the size of the window to capture the information is varied and the

M. Dorigo et al. (Eds.): ANTS 2010, LNCS 6234, pp. 504–511, 2010.

moving average is a simple moving average, as given in equation 1. These studies also share the assumption that the bands for which the channels are constructed are equal for the upper and lower limits. Given the recent empirical evidence that BBs are not as effective as other technical indicators and that there are several parameters for which no theoretical consideration is given it seems reasonable that an advanced optimization technique such as particle swarm optimization (PSO) may be appropriate to fine tune the settings. Other work [7] revealed that there was not a global set of parameters for which BBs would perform optimally for each stock and therefore promoting an active learning approach where the parameters of the BBs can be arrived upon based on historical data and which are optimized on some relevant fitness function. Once again PSO has particular strengths for performing such an optimization approach.

2 Bollinger Bands

Bollinger Bands are a technical indicator, which creates a price channel around a moving average as depicted in figure 2. These price channels can be used to identify stocks which are overbought or oversold and therefore create trading signals for buying or selling. The three main components of a BB are:

1. An N day moving average, which creates the middle band, equation 1,

$$SMA_n(t) = \frac{\sum_{i=t-N+1}^{t} P_i}{N} \tag{1}$$

2. an upper band which, is k times above the standard deviation of the middle band, and
3. a lower band, which is k times below the standard deviation of the middle band.

The default settings for using BBs are a moving average window of 20 days and a value of k equal to 2 for both the upper and lower bands. When the price of the stock is trading above the upper band, it is considered to be overbought, and conversely, an asset which is trading under the lower band is oversold. The trading rules that can be generated from using this indicator are given by equations 2–3:

$$Buy : P_n(t-1) < BB_n^{low}(t-1) \& P_n(t) > BB_n^{low}(t) \tag{2}$$

$$Sell : P_n(t-1) < BB_n^{up}(t-1) \& P_n(t) > BB_n^{up}(t) \tag{3}$$

Essentially, the above rules state that a buy signal is initialized when the price crosses the lower bound from below, and a sell signal when the price crosses the upper bound from above. In both cases the trade is closed out when the price crosses the middle band. As such, a trader will be taking long/short positions in the market; a long/short position is a trading technique which profits from increasing/declining asset prices.

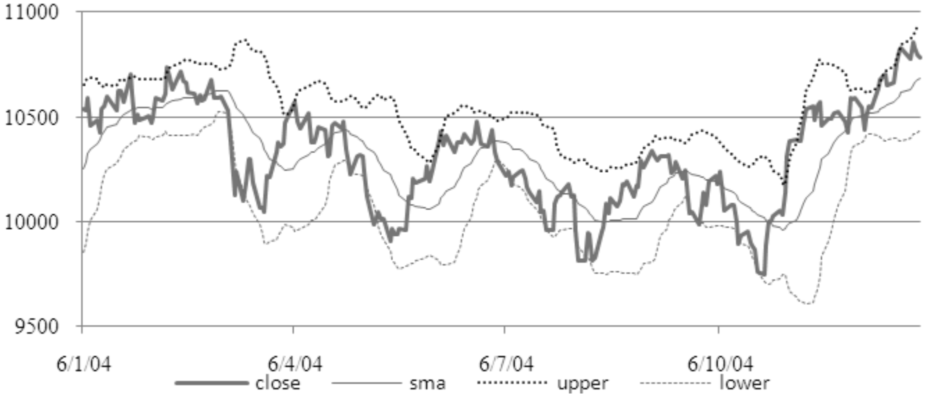

Fig. 1. A depiction of 250 days of trading for the Dow Jones Industrial Average (a market index in the USA), a 20-day moving average and the upper and lower Bollinger bands with value of k =2

3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an algorithm inspired from swarm intelligence commonly used in optimization tasks, which has had success with searching complex solution spaces, similar to the abilities of genetic algorithms (GA). PSO was chosen for this study as it has been shown to be as effective as GAs when modeling technical trading rules, as in Lee et al. [1], yet it had a much simpler implementation and arrived at a global optimum with fewer iterations. For this study the PSO was searching for the optimal values for the parameters displayed in table 1. What makes this study particular novel is that we are not making the assumption that the upper and lower bands should be equal but are allowing the PSO algorithm to search for the optimal bands independent of each other. The default settings for the type of moving average (MA) is normally a simple moving average (SMA), however it is suggested that an exponential moving average (EMA) does perform better in some markets. For this implementation we are allowing the PSO to explore this solution space where both SMA and EMA will be considered. Finally the size of the window used for calculating the moving average and the standard deviation have also been explored. The PSO algorithm was implemented with particles containing 10 dimensions with an overall swarm size of 100, the mapping of the 10-dimensional position vector to the BB parameters is provided below in figure 2.

The type of MA to use was mapped using a wrapper function which evaluated to a SMA if the particle had a value greater than or equal to 0.5 and mapped to a EMA (equation 4) if the particle had a value less than 0.5.

$$EMA_t = EMA_{t-1} + \alpha * (P_t - EMA_{t-1}) \tag{4}$$

where, α is determined by the equation $\alpha = 2/(\mathrm{N} + 1)$ where N is the size of the window and P_t is the value of the underlying financial asset at time t. Each

Table 1. The parameters that the PSO algorithm optimized. MA stands for moving average and the type of MA could be simple or exponential.

Symbol	Description
N	The size of the moving window for calculating the MA.
K_u	The value for calculating the upper band.
K_l	The value for calculating the lower band.
T	The type of MA to use.

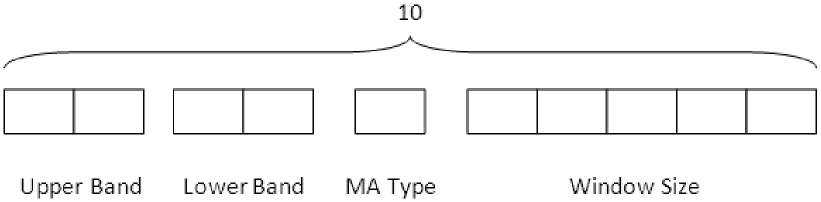

Fig. 2. The mapping of the 10-dimensional position vector to the BB parameters for each particle in the swarm

experiment was performed with 50 iterations as the stopping criterion and each particle velocity and its corresponding position in the 10-dimensional space was updated with equations 5 and 6 respectively.

$$v_{i,j} = \omega * v_{i,j} + C_1 R_1 * (local_{best\,i,j} - x_{i,j}) + C_2 R_2 * (global_{best\,j} - x_{i,j}) \quad (5)$$

Here $v_{i,j}$ is the velocity of j^{th} dimension of the i^{th} particle, c_1 and c_2 determine the influence on a particular particle by its optimal position previously visited and the optimal position obtained by the swarm as a whole, r_1 and r_2 are uniform random numbers between 0 and 1, and ω is an inertia term (see [6]) chosen between 0 and 1.

$$x_{i,j} = x_{i,j} + v_{i,j} \quad (6)$$

Here $x_{i,j}$ is the position of the j^{th} dimension of the i^{th} particle in the swarm. To encourage exploration and limit the speed with which the swarm would converge, a maximum velocity was chosen for each dimension dependent on its range of feasible mappings. In table 2 the range and maximum velocity for each parameter is displayed.

4 Fitness Functions

The overall goal of the experiment is to determine the optimal parameters for maximizing profits, it would seem obvious that training the swarm with a fitness function based on profit would be the most appropriate. However, other

Table 2. The range of feasible values for each parameter and its corresponding maximum velocity for navigating the solution space

Parameter	Range	Max Velocity
Upper Band	{4,4}	0.75
Lower Band	{4,4}	0.75
MA Type	{0,1}	0.10
Window Size	{0,500}	20

literature, Moody et al. [5], found that optimal performance was arrived at with fitness functions which have a risk to reward payoff.

The four fitness functions are described below and each contains a component for transaction costs. The profit fitness function rewarded particles which generated the highest level of profit after accounting for transaction costs without any regard for how risky the trading model was. The fitness function, shown in equation 7, is a sum over all trades (T) taken by the model:

$$fitness_i = \sum_{i=1}^{T} capital_t * \frac{(P_{1,t} - P_{0,t})}{P_{0,t}} - (\tau * capital_t) \tag{7}$$

where $fitness_i$ is the fitness of the i^{th} particle in the swarm, τ represents the transaction costs, and P_0 and P_1 are the entering and exiting price for the underlying asset. The profit for each trade is the rate of return multiplied by the capital invested minus the transaction cost which is also a function of the amount of capital invested. The Sharpe and Sortino ratios are commonly used metrics for evaluating how efficient a trading model is with the additional risk it is exposed to. In both cases a higher value indicates a more efficient use of risk. The Sharpe Ratio is shown in equation 8:

$$S = \frac{E[R - R_f]}{\sqrt{var[R - R_f]}} \tag{8}$$

where R is the return on the asset and R_f is a risk-free rate and E is the expected returns operator. The only difference in the above equation for the Sortino ratio is the denominator: where the Sharpe uses all returns from the trading model, the Sortino only considers negative returns, thus models are allowed to have a higher variance, as long as the returns are positive. Using these equations to derive the fitness function will allow for models which efficiently use risk to be assigned higher fitness with the intention that these models will perform better in the out-of-sample test periods. The fitness functions employed thus far are path dependent and therefore can be sensitive to initial conditions. To help assess the limitations of the prior approaches we introduce a fourth fitness function (equation 9) that assesses how often the trading model produces a positive gain in relation to the total number of positions it takes in the market.

$$fitness_i = \frac{\#returns > 0}{\#returns > 0 \ + \ \#returns < 0} \tag{9}$$

5 Data Description and Experiment Design

The dataset contains daily market values for the DJIA spanning 20 years from Jan 1990 to Dec 2009, where a split of 10 years for training and 5 years for testing yielded 2 separate but not fully independent test periods for the index. The four PSO models are trained separately on each of the datasets and then the optimized parameters are tested on the 5 year out-of-sample data. For comparison's sake the models are compared to a buy-and-hold approach and the default settings for the BBs using an SMA and an EMA. The default settings for SMA and EMA are a 20 day sliding window and a value of 2 for k. The transaction costs are considered to be 0.5% of the capital invested and the risk-free rate (R_f) is a constant 2%.

6 Experimental Results

The results from training are displayed in table 3; the cumulative returns are based on a $100 initial investment.

Table 3. Training results for each model and time period

	Return $		Return %		# of trades	
Model	90-99	95-04	90-99	95-04	90-99	95-04
Profit	101.17	105.26	0.011	0.052	44	42
Sharpe	-3.54	4.62	-1.03	-0.953	42	42
Sortino	-1.896	109.86	-1.01	0.098	44	48
Accuracy	22.29	32.84	-0.777	-0.671	44	44
EMA	52.62	58.63	-0.473	-0.413	112	94
SMA	30.00	43.22	-0.699	-0.567	166	164
Buy-Holds	409.12	280.91	-	-	-	-

From the training results we can see that models created from any of the BB parameter settings were inferior to the buy and hold approach. However the PSO model utilizing the profitability fitness function (PSO_{profit}) was the top performer amongst the BB models. The PSO models all produced a trading activity of around 40 transactions; a manifestation of the 40 trade minimum requirement that was imposed. Such a constraint was added because of the poor performance of the models with the most profitable being those which traded very little or not at all. The test results, reported in table 4, are more competitive between the buy and hold approach and the BB trading models, where the buy and hold approach was outperformed by PSO_{profit} between 2000 and 2004. In the second testing period the buy and hold approach was the most profitable but PSO_{profit} was a close second. The PSO models tended to trade less often than with the default settings, reducing the burden of transaction costs.

Table 4. Testing results for each model and time period

	Return $		Return %		# of trades	
Model	00-04	05-09	00-04	05-09	00-04	05-09
Profit	101.44	94.31	0.014	-0.056	16	18
Sharpe	87.37	89.94	-0.126	-0.100	20	20
Sortino	61.50	68.45	-0.384	-0.315	36	16
Accuracy	78.04	81.30	-0.219	-0.186	32	34
EMA	92.02	84.70	-0.079	-0.152	48	64
SMA	88.61	87.65	-0.113	-0.123	82	90
Buy-Holds	94.94	97.19	-	-	-	-

7 Discussion

If the primary goal of using Bollinger bands is to maximize profit then it seems obvious that a fitness function which optimizes on this metric would be the most reasonable to implement. As discussed, there is supporting literature that has found that using a risk-adjusted fitness function delivers superior out-of-sample returns. In this study this conclusion is not consistent and though these studies involve financial time-series with similar objectives (maximizing profits), the actual task of the algorithms is different and therefore may require different fitness functions. With regards to the Sharpe and Sortino ratios, they were unable to promote movement of the swarm from unfit regions of the solution space because of the calculations themselves. When all models are producing negative returns the less fit particles are being chosen to lead the swarm. From equation 8 we see that if we have two models which produce the same negative return, the

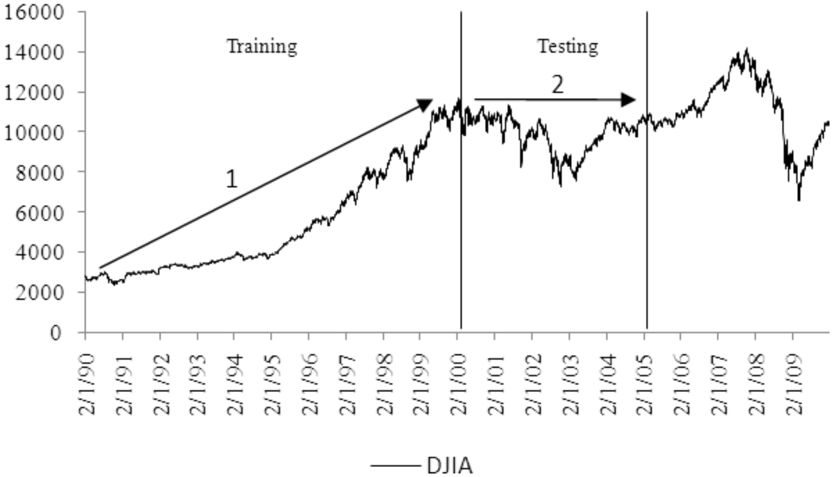

Fig. 3. A plot of the DJIA over both sets of training and testing data. Arrow 1 signifies the upward trend of the market during the first training phase and arrow 2 signifies the sideways moving market in the first testing period.

one which is more volatile and therefore less favorable will actually be ranked higher. The accuracy fitness function was generally outperformed by the other PSO methods, possibly from the lack of consideration for magnitude in the returns when assigning fitness. Although the optimization was able to outperform the default settings in the training and testing periods, it is apparent that when the market is trending, the BBs are unable to capture the excess profits. In one testing period a PSO model was able to outperform the buy and hold approach but this was in the wake of a market contraction. In figure 3 we have a graph of the DJIA over the training and testing periods.

8 Conclusions and Future Work

The main objective of this study was to investigate the short-comings of Bollinger Bands as technical indicators and if a swarm intelligence approach to optimizing its parameters would assist its effectiveness. This included an analysis of 4 different fitness functions which are commonly used in financial modeling and a comparison to the traditional use of the indicator and a market portfolio. The results to date have shown that by fine-tuning the parameters over a training period, with an appropriate fitness function, the particle swarm optimization algorithm is able to find an optimal set of parameters and that depending on the market environment is able to outperform a buy and hold approach. Future work will include expanding the study to other market indices, as well as, changing the fitness ranking of the Sharpe and Sortino ratios. Finally, some trading rules such as stop losses could be imposed to cap how much a trade is able to lose which could assist the accuracy fitness function as it would not be as susceptible to large losses during fitness assignment.

References

1. Lee, J.S., Lee, S., Chang, S., Ahn, B.H.: A comparison of ga and pso for excess return evaluation in stock markets. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005, Part II. LNCS, vol. 3562, pp. 221–230. Springer, Heidelberg (2005)
2. Lento, C., Gradojevic, N.: The profitability of technical trading rules: a combined signal approach. Journal of Applied Business Research 23(1), 13–27 (2007)
3. Lento, C., Gradojevic, N., Wright, C.: Investment information content in Bollinger Bands? Applied Financial Economics Letters 3(4), 263–267 (2007)
4. Leung, J., Chong, T.: An empirical comparison of moving average envelopes and Bollinger Bands. Applied Economics Letters 10(6), 339–341 (2003)
5. Moody, J., Wu, L., Liao, Y., Saffell, M.: Performance functions and reinforcement learning for trading systems and portfolios. Applied Financial Economics Letters 17, 441–470 (1998)
6. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, IEEE World Congress on Computional Intelligence, pp. 69–73 (1998)
7. Williams, O.: Empirical Optimization of Bollinger Bands for Profitability. Master's thesis, Simon Fraser University (2006)