

TRADING ON DUAL BEAT MODEL

Ho Ken Jom (A0075455)
Li Wenru (A0075381)
Zhang Jian (A0018269Y)

Department of Mathematics, National University of Singapore

(For a partial Fulfill of QF5206: Topics on Quantitative Finance, 2011)

I Introduction

1. Theory behind the strategy

In "*Does beta react to market conditions? Estimates of 'bull' and 'bear' betas using a nonlinear market model with an endogenous threshold parameter,*" the authors begin with a logistic smooth transition market (LSTM) model to investigate the relationship between beta risk and stock market conditions for Australian industry.

There are two possible methods to differentiate market states in previous studies. The simplest is comparing the market index to a critical threshold value. For example, Wiggins(1992) defines up (down) months as months when the (excess) market return is greater (less) than zero. However, the market index return is highly erratic and noisy, complicating attempts to identify cyclical characteristics.

An alternative (applied in the paper) uses a trend-based approach to analyze stock market conditions – in this case a twelve-month moving average of market returns (the market indicator, R*). The result can better capture long-run dependencies and trends in the data.

Figures 1 and 2 plot returns on the market index and the transition market indicator R*. As can be seen in figure 2, R* is much smoother and it abstracts from noisy and unsystematic movements in the stock market.

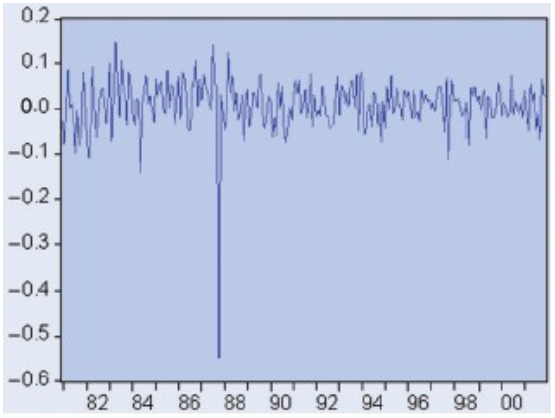


Figure 1: Return on the market index $\mathrm{R_{mt}}$

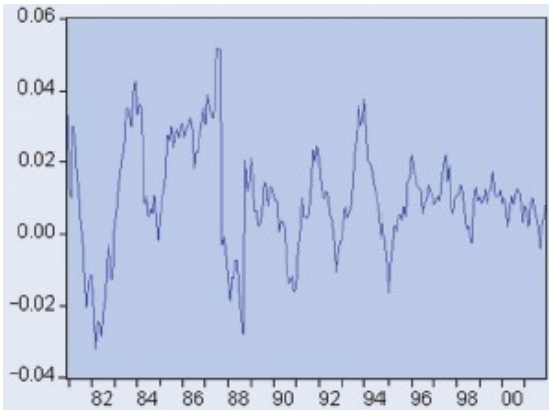


Figure 2: The transition variable (R*)

Beta is estimated using both a dual-beta market (DBM) model and logistic smooth transition market (LSTM) model in the paper.

2. Model Specification

2.1 Dual Beta Market Model

The DBM model is specified as

$$R_{it} = \alpha_i + (\alpha_i^U \times D_t) + (\beta_i R_{mt}) + (\beta_i^U \times D_t \times R_{mt}) + \varepsilon_{it}$$

D is a dummy variable defining up and down markets. When market indicator R* is bigger than threshold value c_i for industry i, it is in a "bull" market, and in a "bear" market when R* is less than c_i.

β_i^U measures the difference between the bull and bear market values of the slope coefficient, thus, the up-market value of beta is given by $\beta_i + \beta_i^U$.

2.2 Logistic Smooth Transition Market Model

The LSTM model is specified as:

$$R_{it} = \alpha_i + \beta_i R_{mt} + \left(\alpha_i^U + \beta_i^U \times R_{mt}\right) \times F(M_t) + \varepsilon_{it}$$

With

$$F(M_t) = \left(1 + \exp\left[-\gamma_i (M_t - c_i)\right]\right)^{-1},\ \gamma_i > 0,$$

F is the logistic smooth transition function, taking a value between 0 and 1. The authors set M_t=R*, the 12-month moving average of the return on the market index. c_i is threshold value for industry i.

It is worth noting that the DBM model can be considered a special case of the LSTM model, as γ_i tends towards infinity (F becomes binary in the limit).

γ_i determines the smoothness of transition between 'bearish' and 'bullish' market states - a large γ_i implies a sudden change between bear and bull market.

When R*- c_i is positive and large, F approaches 0. R_{it} is then effectively $\alpha_i + \beta_i R_{mt} + \varepsilon_{it}$ (the state is very bearish). Similarly, for R*- c_i very negative, F approaches 1. R_{it} is close to $R_{it} = \left(\alpha_i + \alpha_i^U\right) + \left(\beta_i + \beta_i^U\right) R_{mt} + \varepsilon_{it}$ (very bullish).

Given the market indicator and models, the authors use adjusted price data on the 24 industries within Australian Stock Exchange. The observations are monthly and return series are calculated as the difference of the logarithms of prices.

Code	Industry	N_i	μ_i	σ_i	S_i	κ_i	$\widehat{\beta}_i$
XAO	All ordinaries index (R_{mt})	253	0.0095	0.0576	−3.58	35.97	n/a
XGO	Gold	253	0.0013	0.1203	−0.41	7.61	1.337*
XOM	Other metals	253	0.0031	0.0918	−2.00	18.87	1.301*
XDR	Diversified resources	253	0.0092	0.0754	−1.09	10.61	1.091
XDC	Developers & contractors	253	0.0135	0.0700	−3.60	36.74	1.039
XBM	Building materials	253	0.0108	0.0585	−1.51	12.50	0.846*
XAT	Alcohol and tobacco	253	0.0169	0.0563	−2.22	20.95	0.708*
XFH	Food & household goods	253	0.0118	0.0595	−1.39	10.24	0.718*
XCE	Chemicals	253	0.0098	0.0649	−0.59	8.37	0.805*
XEG	Engineering	253	0.0066	0.0615	−0.82	6.89	0.827*
XPP	Paper & packaging	253	0.0084	0.0565	−1.04	8.02	0.726*
XRE	Retail	253	0.0131	0.0593	−2.07	21.36	0.779*
XTP	Transport	253	0.0117	0.0733	−2.28	20.96	1.017
XME	Media	253	0.0173	0.0937	−1.12	8.01	1.052
XBF	Banks	253	0.0162	0.0593	−0.92	8.73	0.774*
XIN	Insurance	253	0.0137	0.0692	−1.68	15.47	0.840*
XIF	Investment & financial services	253	0.0096	0.0544	−3.70	38.40	0.773*
XPT	Property trusts	253	0.0111	0.0359	−1.58	15.67	0.423*
XMI	Miscellaneous industrials	253	0.0004	0.1137	−7.43	84.81	1.106
XDI	Diversified industrials	253	0.0130	0.0666	−2.59	23.75	0.976
XMS	Miscellaneous services	201	0.0093	0.0523	−2.13	17.38	0.661*
XSF	Solid fuels	191	0.0028	0.0707	−1.18	9.34	0.824*
XOG	Oil and gas	191	0.0035	0.0857	−1.33	10.46	1.062
XEI	Entrepreneurial investors	191	0.0085	0.0942	−3.96	35.57	1.138
XTU	Tourism and leisure	132	0.0130	0.0483	−0.99	6.67	0.893
R_t^*	12-month moving average of XAO	253	0.0100	0.0150	−0.02	3.29	n/a

Table 1: Summary statistics

The last line provides summary statistics for transition variable R*. As expected, the standard deviation of R* is much lower than any individual index. Of particular note - it is only about one fifth that of the market index XAO in first line.

A comparison of Figure 1 and Figure 2 shows the different scales. Additionally, cyclical properties of R* can be easily observed.

Code	α	δ	β	β^U	c	γ	ESS
XGO	−0.009 (−1.541)	0.003 (0.041)	2.985 (5.942)	−1.723 (−3.267)	−0.015 (−85.047)	83	2.0581
XOM	−0.008 (−2.290)	0.051 (1.103)	1.425 (17.116)	−0.390 (−2.297)	0.015 (1132.2)	800	0.6889
XDR	−0.002 (−0.684)	0.197 (6.087)	1.272 (21.280)	−0.255 (−2.117)	0.019 (230.14)	226	0.4122
XDC	0.006 (2.362)	−0.179 (−5.454)	1.061 (9.220)	−0.161 (−1.247)	−0.004 (−56.932)	479	0.3121
XBM	0.002 (0.710)	0.096 (3.426)	0.780 (10.833)	0.144 (1.627)	−0.012 (−155.74)	126	0.2570
XAT	0.010 (4.269)	−0.148 (−4.578)	0.562 (9.757)	0.343 (2.472)	0.028 (502.34)	202	0.3599
XFH	0.004 (1.295)	−0.072 (−1.176)	0.551 (4.923)	0.245 (1.746)	−0.001 (−49.78)	763	0.4557
XCE	0.000 (0.052)	0.064 (1.917)	0.809 (13.389)	0.439 (2.945)	0.031 (1073.8)	381	0.4999
XEG	−0.004 (−1.553)	0.202 (6.336)	0.646 (3.381)	0.351 (1.762)	−0.011 (−81.658)	99	0.3542
XPP	0.001 (0.330)	0.136 (3.705)	0.836 (12.852)	−0.222 (−1.863)	0.031 (1208.9)	581	0.3509
XRE	0.008 (3.045)	0.025 (0.338)	0.945 (7.074)	−0.304 (−2.042)	−0.001 (−26.972)	177	0.3634
XTP	0.002 (0.642)	−0.117 (−3.110)	0.887 (130.137)	0.318 (2.765)	0.023 (623.2)	304	0.4745
XME	0.005 (1.036)	0.015 (0.217)	0.985 (8.002)	0.395 (1.997)	0.019 (333.7)	184	1.2636
XBF	0.009 (3.516)	0.181 (3.250)	1.006 (9.860)	−0.244 (−2.005)	0.003 (145.5)	604	0.3725
XIN	0.004 (1.203)	−0.231 (−5.075)	0.525 (6.305)	0.496 (3.706)	0.008 (312.2)	113	0.5793
XIF	0.003 (1.585)	−0.149 (−4.634)	0.438 (3.800)	0.255 (1.958)	−0.012 (−2198)	2640	0.2256
XPT	0.007 (4.260)	−0.087 (−3.861)	0.338 (8.397)	0.105 (1.453)	0.014 (3298)	2929	0.1705
XMI	−0.012 (−1.388)	−0.0581 (−0.793)	0.0289 (0.073)	1.1785 (1.998)	−0.015 (*B*)	2.47	2.1979
XDI	0.005 (2.295)	0.000 (0.000)	1.037 (15.248)	−0.203 (−2.281)	0.010 (178.7)	282	0.3135
XMS	0.004 (1.636)	−0.010 (−0.272)	0.682 (10.816)	−0.231 (−1.834)	0.030 (2613)	996	0.2096
XSF	−0.004 (−1.117)	0.128 (2.967)	1.008 (12.887)	−0.417 (−3.065)	0.018 (1018)	698	0.4191
XOG	−0.006 (−1.625)	0.081 (1.352)	2.009 (6.957)	−0.945 (−3.032)	−0.015 (*B*)	9.06	0.5033
XEI	0.000 (0.059)	−0.427 (−6.648)	0.739 (6.325)	0.318 (1.978)	0.014 (1094)	818	0.6220
XTU	0.004 (1.316)	n/a	1.011 (7.627)	−0.323 (−1.909)	0.011 (1882)	1856	0.1532

Table 2: Estimated LSTM models

Only XMI and XOG industries have a small γ which is less than 10, suggesting that most industries experience rather abrupt changes between 'bear' and 'bull' states. Recall that with high value γ, a LSTM model will behave much like a DBM model. The latter may be more appropriate, given that it uses one less parameter.

Up-market differentials in betas are recorded in the β_i^U column. 15 of these are statistically significant at the 5% level and another 6 at the 10% level. This provides good evidence that beta indeed varies between market states (as partitioned by R^* and c_i).

Of the statistically significant β_i^Us, 11 are negative, consistent with literature claims that risk in up markets is lower than down-market risk. However, in another 10 industries, the opposite is true.

Code	α	δ	β	β^U	c	T^L	RR_1	LM_1	LM_{12}	R^2	ESS
XGO	−0.010 (−1.584)	0.002 (0.032)	2.761 (6.178)	−1.489 (−3.139)	−0.0149 −0.0141	15	0.63	0.01	0.51	0.43	2.0733
XOM	−0.008 (−2.271)	0.051 (1.093)	1.425 (16.978)	−0.390 (−2.279)	0.0151 0.0153	181	0.48	0.16	0.35	0.68	0.6889
XDR	−0.002 (−0.679)	0.198 (6.071)	1.272 (21.150)	−0.255 (−2.101)	0.0184 0.0191	191	0.68	0.06	0.73	0.71	0.4122
XDC	0.006 (2.347)	−0.179 (−5.416)	1.061 (9.322)	−0.161 (−1.260)	−0.0043 −0.0041	37	0.61	0.55	0.82	0.75	0.3121
XBM	0.002 (0.711)	0.096 (3.414)	0.780 (10.907)	0.144 (1.632)	−0.0115 −0.0014	26	0.82	0.10	0.01	0.70	0.2570
XAT	0.010 (4.239)	−0.146 (−4.506)	0.564 (9.711)	0.339 (2.427)	0.0277 0.0282	218	0.55	0.02	0.03	0.55	0.3599
XFH	0.004 (1.286)	−0.072 (−1.165)	0.551 (4.885)	0.245 (1.731)	−0.0008 −0.0007	48	0.29	0.12	0.20	0.49	0.4557
XCE	0.000 (0.054)	0.064 (1.902)	0.809 (13.306)	0.439 (2.925)	0.0309 0.0312	231	0.34	0.58	0.21	0.53	0.4999
XEG	−0.004 (−1.571)	0.204 (6.244)	0.669 (3.997)	0.331 (1.868)	−0.0074 −0.0072	32	0.83	0.06	0.08	0.63	0.3543
XPP	0.001 (0.328)	0.134 (3.677)	0.836 (12.751)	−0.222 (−1.852)	0.0309 0.0312	231	0.99	0.15	0.31	0.56	0.3509
XRE	0.008 (3.022)	0.025 (0.337)	0.945 (7.021)	−0.304 (−2.027)	−0.0017 −0.0009	46	0.03	0.38	0.74	0.59	0.3634
XTP	0.002 (0.637)	−0.117 (−3.083)	0.887 (13.034)	0.318 (2.746)	0.0232 0.0238	205	0.66	0.96	0.30	0.65	0.4745
XME	0.005 (1.027)	0.015 (0.217)	0.985 (7.940)	0.395 (1.982)	0.0184 0.0191	191	0.44	0.02	0.39	0.43	1.2636
XBF	0.009 (3.486)	0.181 (3.223)	1.006 (9.784)	−0.244 (−1.987)	0.0024 0.0027	62	0.96	0.22	0.27	0.58	0.3725
XIN	0.004 (1.197)	−0.231 (−5.027)	0.525 (6.258)	0.496 (3.681)	0.0081 0.0085	113	0.79	0.41	0.21	0.52	0.5793
XIF	0.003 (1.547)	−0.149 (−4.577)	0.438 (4.177)	0.255 (2.100)	−0.0122 −0.0122	22	0.31	0.94	0.05	0.70	0.2256
XPT	0.007 (4.172)	−0.087 (−3.751)	0.338 (8.097)	0.105 (1.459)	0.0131 0.0131	164	0.26	0.83	0.29	0.43	0.1705
XMI	−0.011 (−1.401)	−0.003 (−0.029)	0.383 (1.968)	0.778 (2.623)	−0.0122 −0.0122	22	0.91	0.93	1.00	0.32	2.1988
XDI	0.005 (2.274)	−0.000 (−0.006)	1.037 (15.148)	−0.203 (−2.262)	0.0096 0.0098	129	0.81	0.65	0.07	0.72	0.3135
XMS	0.004 (1.618)	−0.010 (−0.268)	0.682 (10.720)	−0.231 (−1.820)	0.0301 0.0301	172	0.42	0.26	0.81	0.62	0.2096
XSF	−0.004 (−1.108)	0.128 (2.934)	1.008 (12.747)	−0.417 (−3.036)	0.0180 0.0181	128	0.52	0.25	0.43	0.56	0.4191
XOG	−0.006 (−1.620)	0.080 (1.374)	1.753 (5.112)	−0.688 (−1.930)	−0.0122 −0.0122	22	0.12	0.00	0.02	0.64	0.5055
XEI	0.000 (0.057)	−0.427 (−6.586)	0.739 (6.262)	0.318 (1.961)	0.0142 0.0143	120	0.86	0.03	0.03	0.63	0.6220
XTU	0.005 (1.617)	n/a	1.068 (6.380)	−0.325 (−1.756)	0.0079 0.0080	54	0.12	0.11	0.46	0.49	0.1532

Table 3: Estimated DBM models

Many of the LSTM models might be better estimated as simpler DBM models – for example, parameter estimates and standard errors are almost identical. In addition, the R^2 indicates a good fit.

T^L in table 3 counts the number of observations in the down-market regime. It can be seen that a majority of industries have less than half of their observations in the down-market regime (TL<126).

3. Paper Summary

- 'Bull' and 'bear' betas are significantly different for most industries.
- Transition between states is abrupt, supporting a dual-beta market modeling framework
- For many industries, stocks spend more time in 'bull'-market than 'bear'-market states.
- Risk in 'bull' states is not always smaller than the risk in 'bear' market states.

The first two findings form a basis for our trading strategy.

II Trading Strategy

1. Indicator Variable

The most major change is the use of a market daily rate "R" as an indicator, rather than a moving average "R*" – the opposite of the paper's approach. As backtesting shows, the strategy can be profitable, while using almost any moving average indicator will result in a loss.

This is due to, and has, several major implications on both model and strategy.

2. Market States

The market states which are divided by the threshold are no longer "bull" and "bear" markets – states which persist between five and two years respectively (in recent history). As in the paper, the market state defined by a spot rate indicator persists only on the scale of a few days. The states are referred to as "up" and "down" for simplicity.

"Up" or "down" market states identify investor sentiment, among other short-lived trends. Cases can be made for having three states, for example "up", "down" and "level" (similar to the stock analysts' buy/hold/sell ratings). Comparisons to Markov states are also apt – there could exist any number of states or even a continuous function linking market return to a β.

However, adding a state requires at least two new parameters: threshold and β. For simplicity, as well as avoid over parameterization, only two states are considered.

3. Fitting the Model

The parameters are empirically determined by solving a two-step optimization problem.

For a given threshold, the market returns are partitioned into two states, and a CAPM model is fitted for each state, given that each state has a minimum fraction of the returns (10% to 20%) and both states share the same α. This step minimizes the errors for a given threshold.

The second step involves minimizing the errors across different threshold values, providing empirically-determined estimates for c, α, β^u and β^d. Compared to the paper, values of the two βs will not be very different, however, the discontinuity that exists should be significant enough to lead to profitable trading opportunities, as shown below.

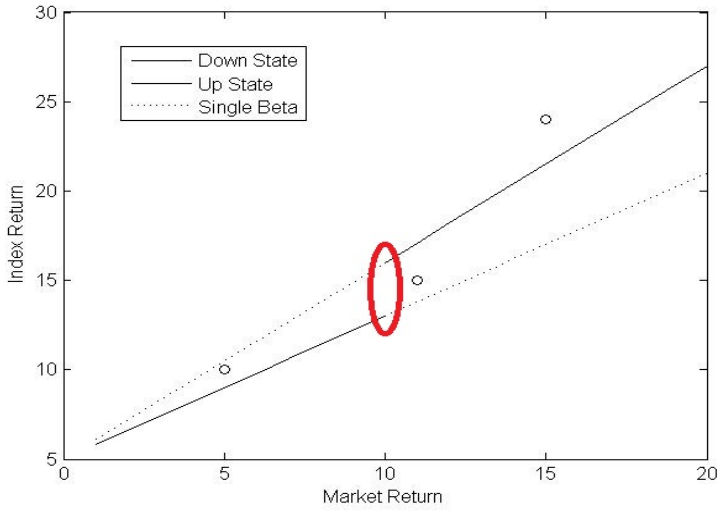


Figure 3: Possible Discontinuity (Jump) at Threshold

4. Market (In) Efficiency

The discontinuity suggested above is considered by the paper to be a sign of market efficiency, in terms of information distribution. In this context, an information asymmetry hardly exists, leading to all agents in the market receiving news and thus acting at almost the same time, leading to a sudden change.

In the context of a trading strategy, market inefficiency, specifically in discounting or forecasting, is what leads to the abrupt change – implying a profitable opportunity – which is an important economic rationale for the strategy.

5. Data Process

For a given set of fitted parameters, the strategy will follow this process when receiving new information about the market and index returns:
Compare R^m to c, obtaining the market state
Calculate "fair rate" of index return: $\alpha + (\beta^u 1_{\{up\}} + \beta^d 1_{\{down\}}) R^m$
Calculate the "deviation": $\varepsilon = R^i - \alpha + (\beta^u 1_{\{up\}} + \beta^d 1_{\{down\}}) R^m$
For current position, compare deviation with thresholds to determine new position (if any)

6. Thresholds

There are two thresholds.

The upper threshold (T^u) identifies tradable deviations. Thus, if $\varepsilon > T^u$, one would open a long position if one was not already held, and if $\varepsilon < -T^u$, enter a short if not already being held.

The lower threshold identifies situations where deviation may be dominated by noise effects. The strategy would close-out any open positions. (In some way, this internalizes a "level" state.)

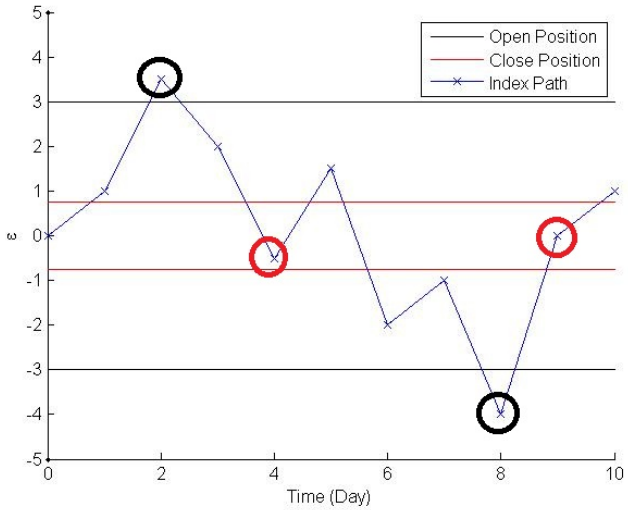


Figure 4: Possible Trading Path

The above shows a possible deviation path. The upper and lower thresholds are set at 3 and 0.75 in this case. The black-circled points indicate where the strategy would enter into a position, and red ones where it would close them out.

Backtesting has shown between 2 and 3 days between transactions (the above has 2.5 days) depending on the index as well as how aggressive thresholds are set.

III Calibration and Backtesting

1. Data Selection

Throughout the project, the market index used is the S&P500 (Yahoo Finance Symbol: ***^GSPC***), and candidate target indices scanned include

- NYSE Composite (***^NYA***)
- NASDAQ Composite (***^IXIC***)
- VANGUARD Index Trust 500 Index (***VFINK***)
- PHLX Oil Service Sector Index (***^OSX***)
- Bank of America (***BAC***) – a single stock

Data was downloaded for 01/01/2009~06/03/2011. For back testing, the data was also downloaded for 04/01/2007~06/03/2009 & 03/01/2008~06/03/2010.

Each period contain 546 points of trading data. The following plots quotes for S&P 500 and NYA during 01/01/2009~06/03/2011.

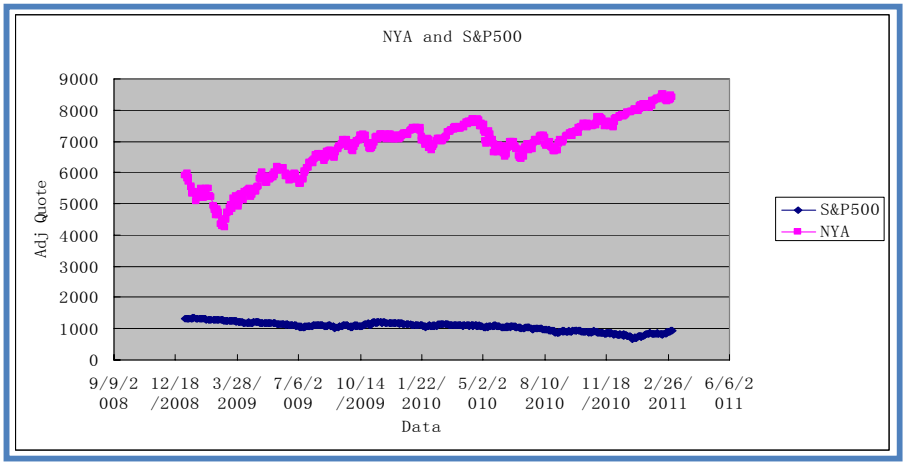


Figure 5: Index Values

We then perform the regression on the first 294 data points using both the dual beta single alpha market (DBSAM) model and single beta market (SBM) model. R^2 is shown below. It is helpful to establish that a difference exists.

A high R^2 is important as it gives confidence that the model used captures important features of the market. Strategies then have good reasons, and are more likely, to be profitable.

Target	NYA	IXIC	VFINX	OSX	BAC
R2-SBM	0.978	0.923	0.999	0.717	0.481
R2-DBSAM	0.979	0.925	0.999	0.728	0.504

Table 4: Comparison of DBSAM and SBM models

The R^2 for sector index and single BAC is relatively low, we thus decide to trade on NYA and S&P 500. This argument is also employed when determining some of the following features of our model and strategy.

1.1 Log vs Simple Return

Both log and discrete holding period returns were used. Backtesting showed that log returns performed slightly better, so log returns were used in model evaluation.

1.2 Moving Average Lags

The original paper used rolling moving average – backtesting a strategy based on such an indicator demonstrated that it was not appropriate for trading.

One reason might involve the paper's use of monthly data (only 12 points in a year). The table below shows the regression R^2 when different lag lengths were used for the indicator.

mLag	1	2	3	5	10
SBM R2	0.978	0.117	0.083	0.064	0.020
DBMSA R2	0.979	0.757	0.589	0.390	0.179

Table 5: Moving average window sizes for R*

It can be seen that the fit becomes very bad as the lag increases, thus the use of spot return instead of a moving average indicator.

1.3 Transaction Costs

While it is possible to incorporate transaction cost, due to scheduling and other constraints, it is not considered for the testing done in this report.

2. Model Estimation

The regression of SBM is straightforward using the class ***OlsRegression*** of ***AlgoQuant.***

The regression of DMSAM is tedious. First, we regress SBM, and get the corresponding alpha for SBM: α_{SBM}. We then create an array of α'

$$\boxed{\alpha' = \alpha_{SBM} + is}$$

Where i = -25, -24,...,25 and s = 0.01 throughout the project.

Next, we will create an array of partitions of market. For example, suppose we have 100 market index returns, and require that at least 30 points fall into an up or down market. Then, we will set the critical value of market index return to the 31th and 70th values in the sorted array.

Finally, we search the parameter of DBSAM in the 51 X 70 2-D array for a maximum composite R-Square. The graphs below show the outcome of SBM DBSAM regression.

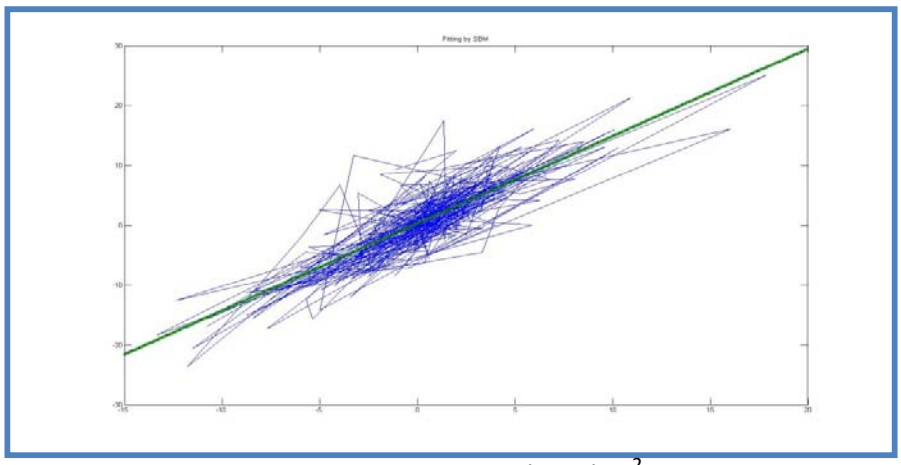


Figure 6a S&P500 and OSX (SBM), $R^2 = 0.69$
beta = 1.46. alpha = 0.175

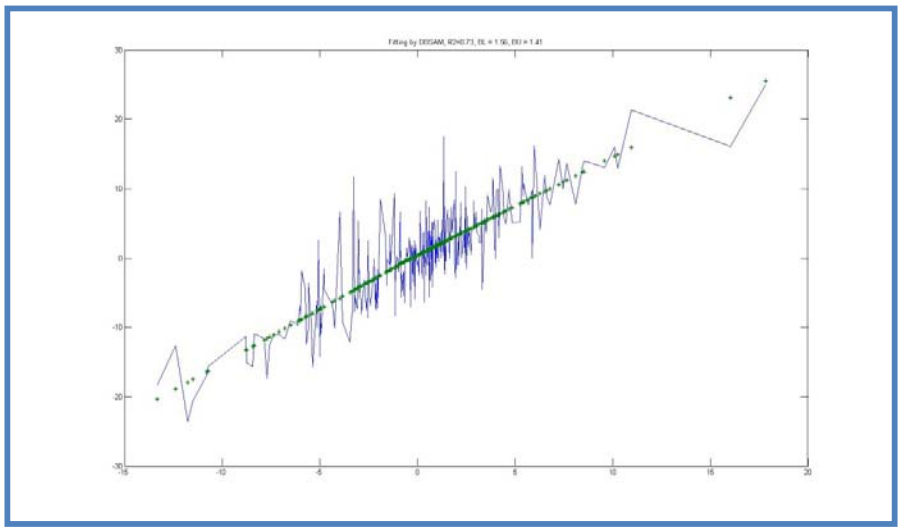


Figure 6b S&P500 and OSX (DBSAM) $R^2 = 0.76$
lower beta = 1.56, upper beta 1.44, alpha = 0.415.

Figure 6b shows that the DBSAM produces a better fit (as expected since it has one more parameter); furthermore, the up and down market betas are different; finally, the alpha of DBSAM (0.415) is significantly differ from SBM (0.175), this confirms the necessity of search alpha over a range.

3. Backtesting

In the following sections, all the data shown on the vertical axis or in the table is the annualized return, which is calculated as the ratio of the P&L against the price of target index at the time of first trading day.

For pairs trading, this may give a much small return than the actual return, since in pairs trading, the net investment could be very small, zero or negative. If we strictly follow the definition of return, then we will unable to calculate the return if the next investment is zero or negative.

4. Stationary Trading

Stationary trading, involves only model estimation for once, with model parameters remains throughout the back testing period. This is in contrast to dynamical trading.

4.1 Optimal triggering threshold

The trigger threshold is set to be a fraction multiple (not less than 1) of the mean modeling error: T^u=***THT*** *****ERR*** (See Appendix A.1 for definitions of parameters).

We have varied THT from 2 to 8. The graph below shows the performances of different THT values for SBM and DBSAM.

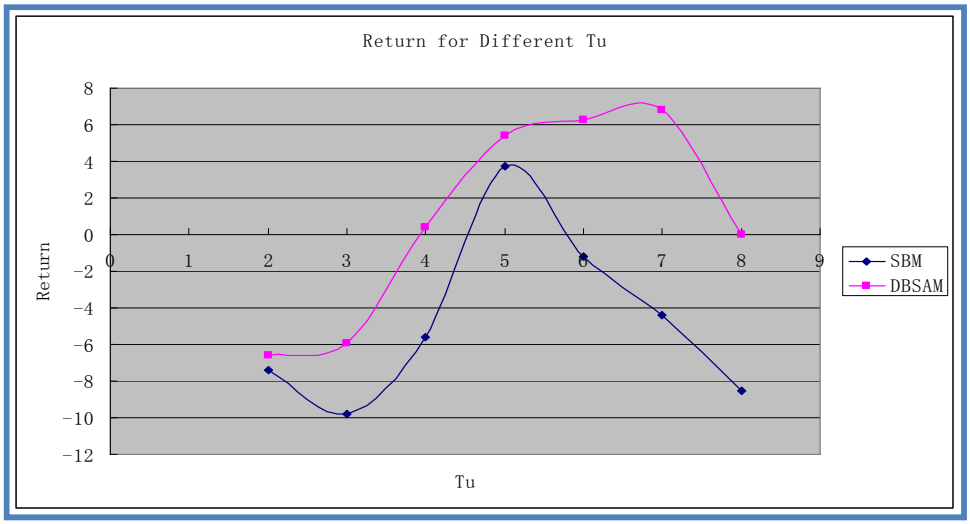


Figure 7: Optimal Position Trigger

From the graph we can see that when THT is between 5 and 6, both SBM and DBSAM can make profits, while DBSAM outperforms SBM for all THT values.

4.1 Optimal closing threshold

The trigger threshold is set to be a fraction multiple (not larger than 1) of the mean modeling error: T^d=***THCLSE*** *****ERR***

We have varied THT from 0.1 to 1. The graph below shows the performances of different THCLSE values for SBM and DBSAM.

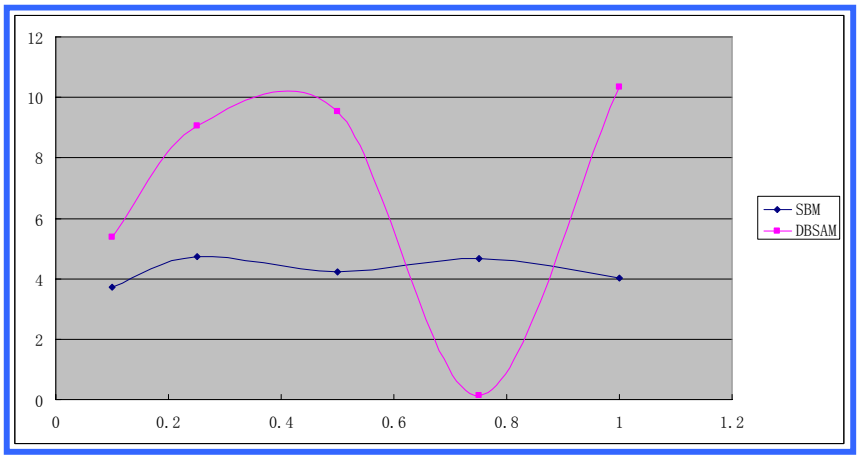


Figure 8: Optimal Position-Close Trigger

From the plot, one can see that SBM is not sensitive to the closing threshold while this is not the case for DBSAM. And further, DBSAM outperforms SBM for 4 out of 5 values of THCLSE.

4.3 Optimal market partition threshold

The market is partitioned for DMSAM using a fractional weight from 0.1 to 0.5. The results are shown below.

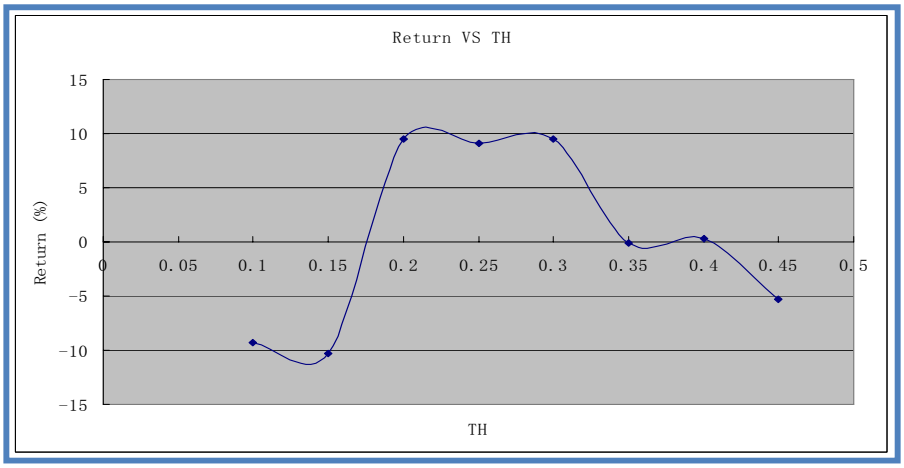


Figure 9: Optimal Market Partition Weight

Back testing suggests an optimal range for partition weight of 0.2~0.3.

4.4 Length of back testing period

In all the back tests above, the regression is conducted for the first 294 days, with trading for the left 252 days (corresponding to 1 year). Next, we try to change the length of trading and use the proceeding period for regression. For example, if the trading period is 200 days, then the regression length is 394 days.

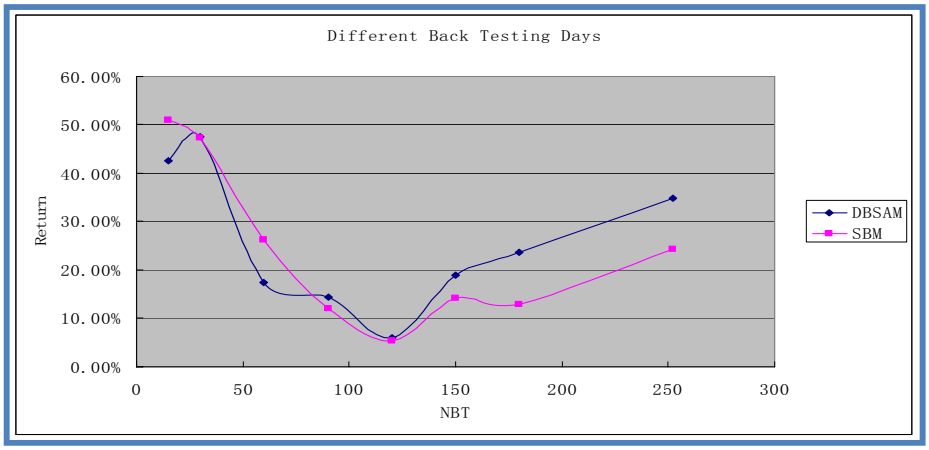


Figure 10: Varying Back Testing Period Length

It is interesting to notice that, when NBT is less than 120 days, SBM generally performs better that DBSAM, while for NBT > 120, the situation switches.

A possible explanation is that, when the length for back test is less than 120 days, the market may be relatively stable so fitting a DMSAM model will over-fit.

4.5 High frequency trading

We have also explored the performance of both SBM and DBSAM in high frequency trading. To do this, we have set THCLSE to be 0.5 and THT to be 1, which are fairly aggressive.

DBSAM generally makes more profits and that when the number of trading days increases the profits also increases.

The graph shows that for both SBM and DBSAM, the profit at NBT = 60 days, and it is extremely high relative to other points. This high value may be a fluke of the data. Otherwise, the same trend as noted above is observed.

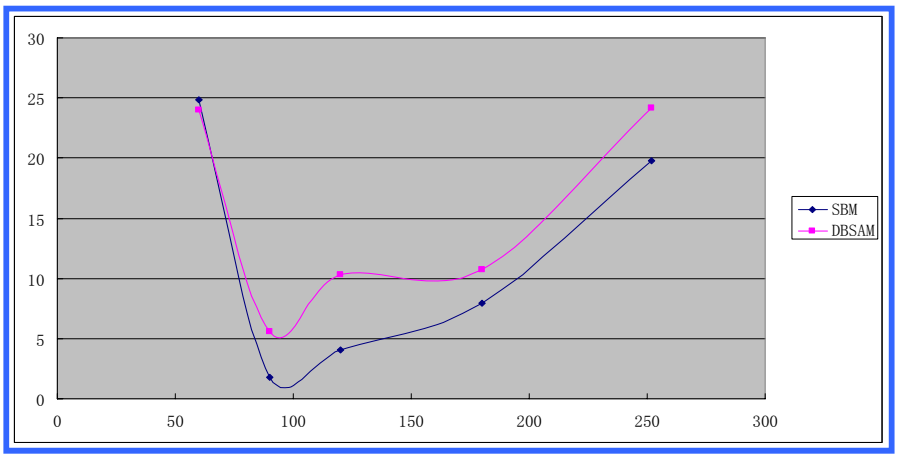


Figure 11: High Frequency with different Back Test Length

5. Dynamical Trading

This involves updating the model using new data during trading.

The DBSAM model requites intensive computing. In JAVA, each set takes about 40 minutes. To speed up the processing, a three-step approach was employed. In Java, using the YahooEOD to download the data, then printing out log returns of the market index and the target index.

The data is then feed to a CUDA program in C++, which computes the SBM and DBSAM model parameters. These parameters are then used (on JAVA) programs for trading. For more details, please see the APPENDIX. At first trial, the model is updated every week.

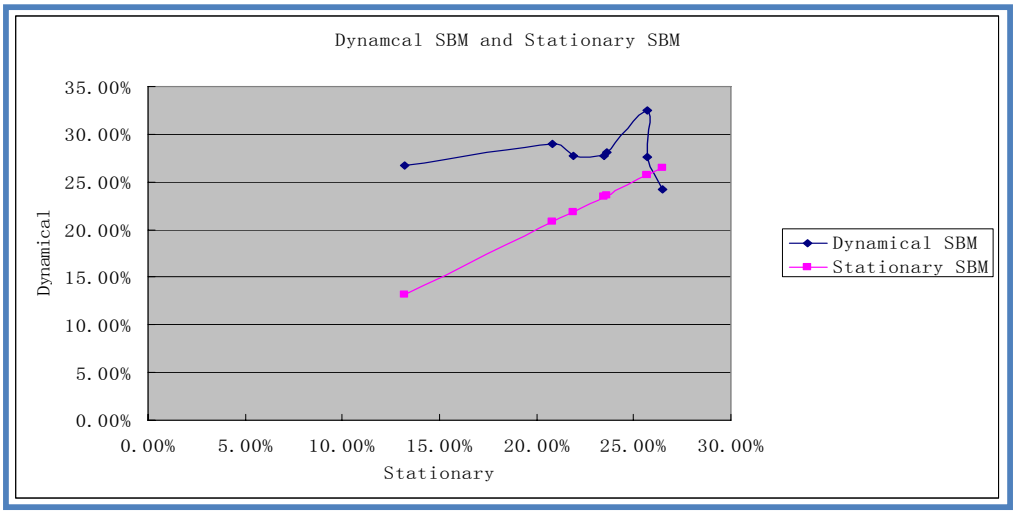


Figure 12a: Dynamical VS Static Trading for SBM, Weekly Update

For SBM, the dynamical approach produces more money in most of the cases.

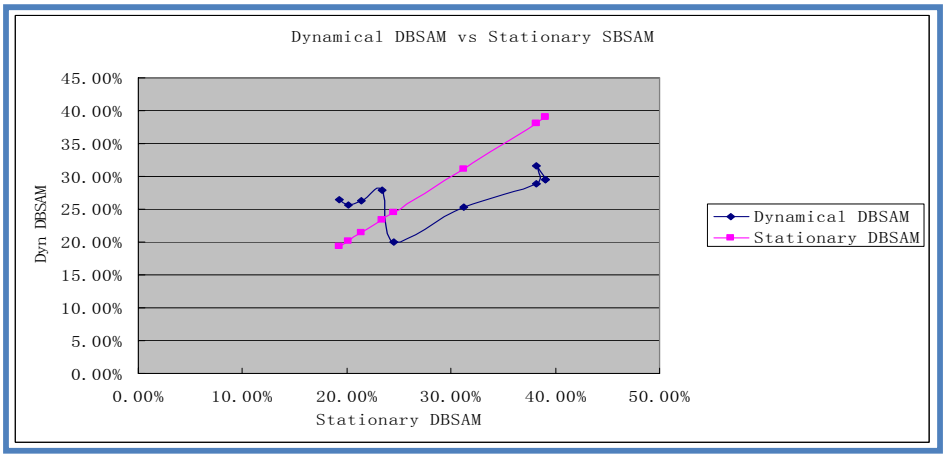


Figure 12b: Dynamical VS Static Trading for DBSAM, Weekly Update

However, for DBSAM, this is not the case. To explore the full power of dynamical trading, it is possible to update daily. However, experiments show that daily updating and weekly updating give roughly the same P&L, as shown below.

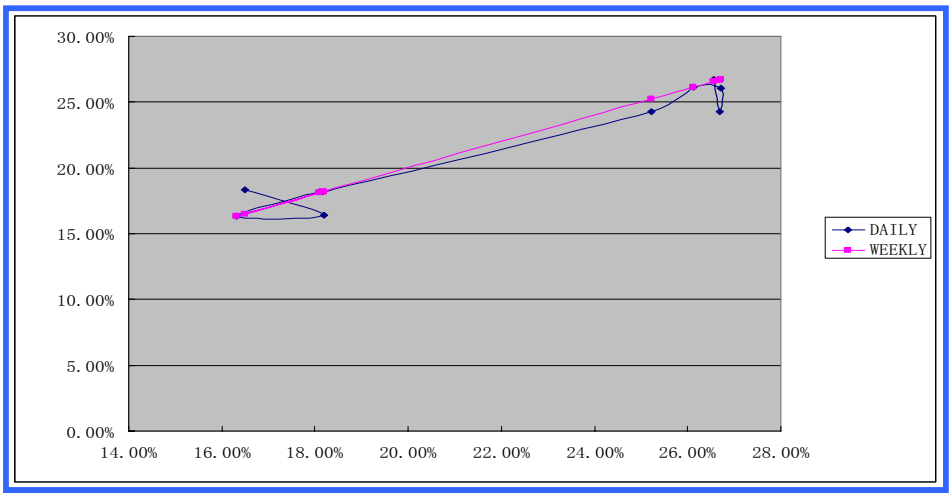


Figure 13a: SBM, Weekly Update vs Daily Update

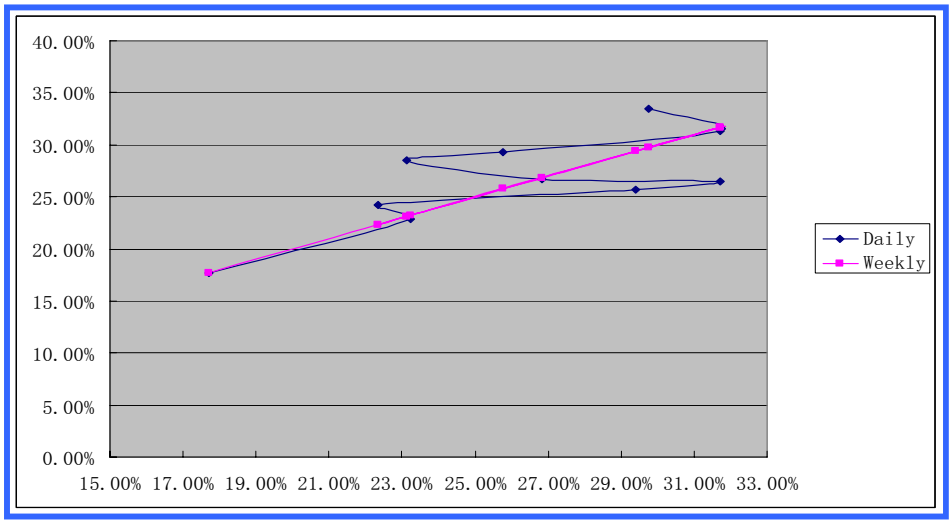


Figure 13b: DBSAM, Weekly Update vs Daily Update

For dynamical trading, since we are updating the model during trading, it is interesting to evaluate the performance using different regress length (currently is 294 days), with the hope that an optimal length of historical data will give us better performance.

In experiments, we have chosen the regression length to be 252, 180, 150, 120, 90, and 60 days, using weekly update, and then trade for 1 year end at 06/03/2011.

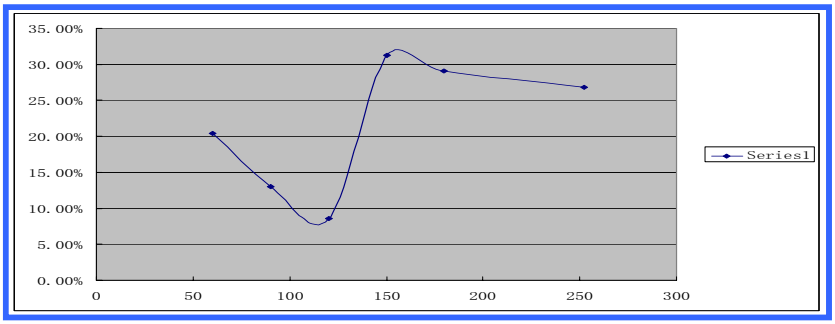


Figure 14a: DBM, Different Regression Length

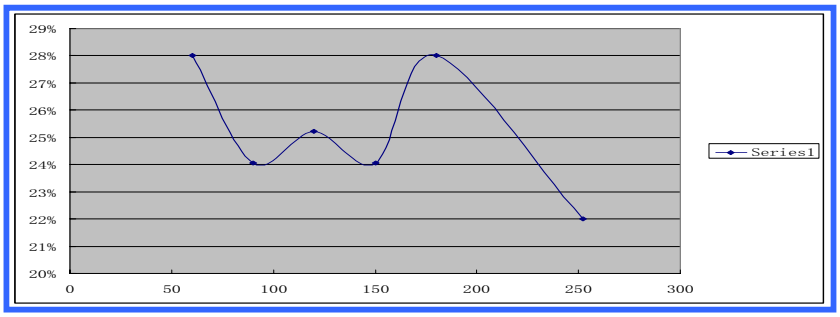


Figure 14b: SBM, Different Regression Length

When the regression length is less than 150 days, SBM outperforms DBSAM, but if above 150 days, DBSAM make more profits. Intuitively, this says that the minimum length to have a useful set of both “up” and “down” state data is about 150 days.

Both models find an optimal value at about 180 days.

6. Stress Testing

All the P&L profiles obtained above is for the data range from 01/01/09~06/03/11, which falls into a recession period starts from 2007. Although the worst time of the stock market has passed by, but the future remains volatile. That is, those preceding analysis is done in the sense that the regression is conducted in a recession period and the back test is also done in a recession period. We then set up a not so rigorously 'stress testing' by looking to the following situations.

Case I

In this case, the regression is done prior to the financial crisis, but the back testing falls into the crisis. The data is downloaded from 04/01/2007~03/01/2009, which contains 546 trading days, and the regression length is still 292 days to be consistent with previous settings.

We then back-test using a single optimal setting (TH = 0.3, THT = 6 for static trading and THT = 4 for dynamical trading*, NBT = 252 THCLSE = 0.3, with weekly updating for dynamical trading), and the outcomes are tabulated below.

Model	SBM Static	SBM Dynamic	DBSAM Static	DBSAM Dynamic
Return	0.08%	-8.1%	-18.94%	6.08%

Table 6a: Model Comparison (Returns)

Case II
Data is downloaded for 03/01/2008~06/01/2010 such that both regression and back test are in the worst times.

Model	SBM Static	SBM Dynamic	DBSAM Static	DBSAM Dynamic
Return	7.09%	1.76%	32.74%	0.10%

Table 6b: Model Comparison (Returns)

From the tables above, it seems that SBM and DBSAM Dynamic trading are the.
Of course, more testing is important before we can conclude this.

[*Testing suggests that for dynamical trading, the position triggering threshold should be slightly lower than the static counterparty (better information is available). But for consistency, they are the same in the above]

7. Dual beta dual alpha model

The dual beta single alpha model has two more parameters than the single beta single alpha model. In fact, in the original paper, there are separate alphas for each market state, but they are assumed to be equal.

We would like to examine whether this simplification is appropriate, or if more parameters can be justified.

The regression for dual beta dual alpha model (DBDAM) is much simpler than DBSAM, since we can use the class ***OlsRegression*** directly without searching any grid for alphas. After the regression, we then trade on the DBDAM for 01/01/2009~06/03/2011.

However, very surprisingly shifting the THT (position triggering threshold) from 1 to 4, the model produces large losses or at best negligible profits. If increased further, trading does not occur at all.

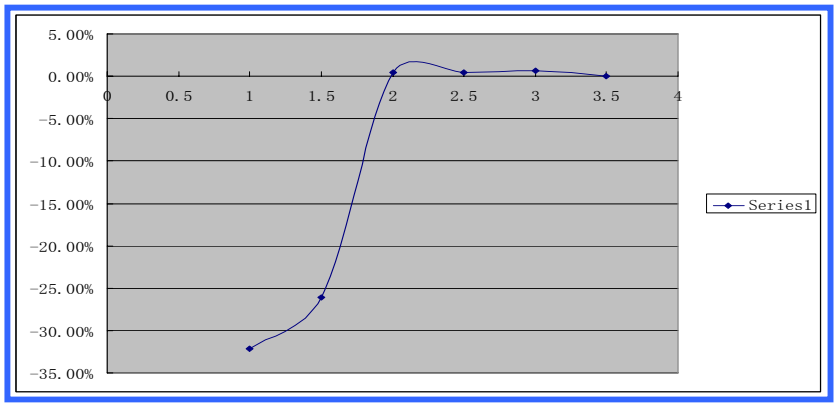


Figure 15: Varying THT for DBDAM

A possible explanation is that, the alpha corresponding to the firm specific returns, and should not be altered by the market conditions. A dual beta dual alpha model actually yields an over fitting.

8. Modified Strategy (with pairs trading)

8.1 Varying Trigger Threshold

In order to hedge the system risk, we have tried to trade on a portfolio which contains one unit of the target index and short beta units of the market index. The value of beta is determined as before and all triggering conditions work the same way defined before.

To hedge this risk, we have attempted imposing a condition such that a position is only closed when generating profits. This simple add-on is not useful for two reasons. First, it significantly reduces the number of transactions, losing profitable opportunities. Second, if the market maintains momentum, the strategy does not close its position (or open the opposite position) but instead simple continues to lose.

We then attempted a version of pairs trading. The pairs trading introduced in the lecture notes first constructs a market-neutral portfolio and then trading on the spread. Here we will modify the idea: constructing a market-neutral portfolio consisting of the market index (S&P 500) and the target index (NYA) using the proper beta (we have two values to choose), and then we use the same triggers introduced above to trade.

First, we have use the 01/01/2009~06/03/2011 data to trade for half a year, using the setting THT = 1~15, THCLSE = 0.3 and TH = 0.3). The results are shown below.

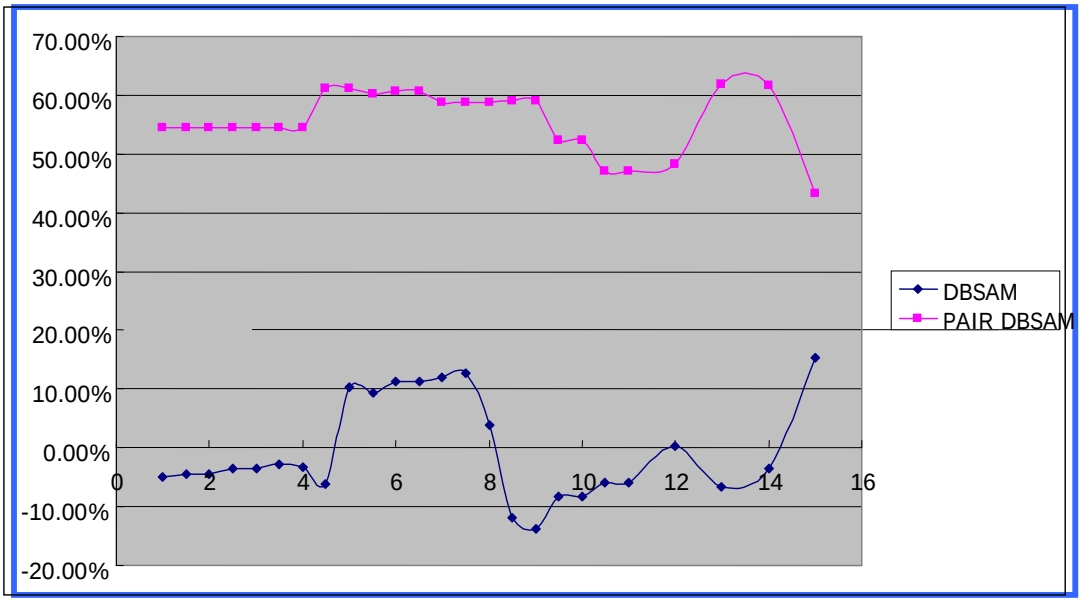


Figure 16: Pairs trading based on DBSAM (trade for 126 days)

Performance is tremendously improved by trading on the market-neutral portfolio instead of simply the target index.

Specifically, we observe that:
(i) For a very wide range of position trigger threshold which range from 1 to 15, the hedged strategy is able to produce profits, while the DBSAM model has a P&L fluctuates about the zero line, and only have a narrow range 4~7 which can make reliable profits.

(ii) The hedge portfolio yields a very high return rate between 55% & 62% (there is only one point falls to 43.30% when THT = 15); while the naked strategy is usually capped by 12% (with one exception at THT = 15).

(iii) The performance of hedged portfolio is very stable against the choosing of THT.

8.2 Different length of back testing period
We also change the regression length to be 294 days so that we can trade for a whole year, the results is shown in Figure 3.9, which shows that although now the P&L is not stable anymore (since we have now less data for regression) but the Paired Strategy definitely has higher returns.

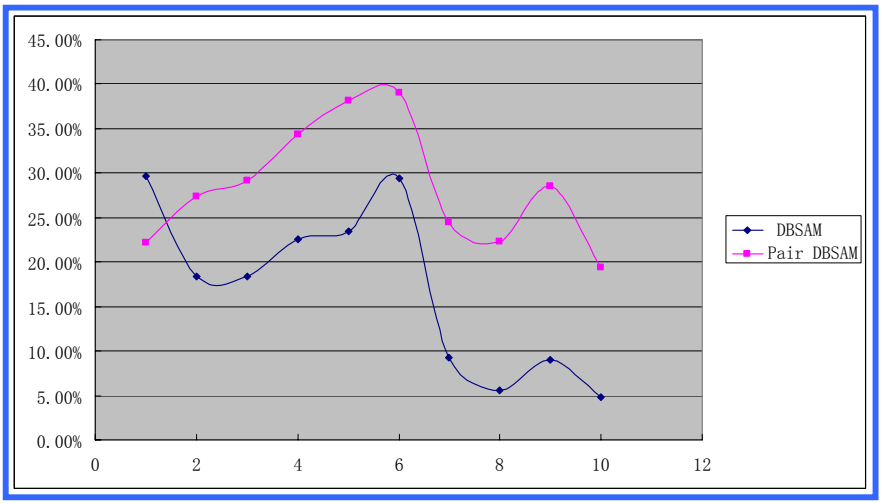


Figure 17: Pairs trading based on DBSAM (trade for 252 days)

8.3 Stress Testing for hedged portfolio

The stress testing periods was applied to the hedged portfolio, and we found that for 2007~2009 (Figure 3.10), the hedged portfolio is able to make profits for this extreme case.

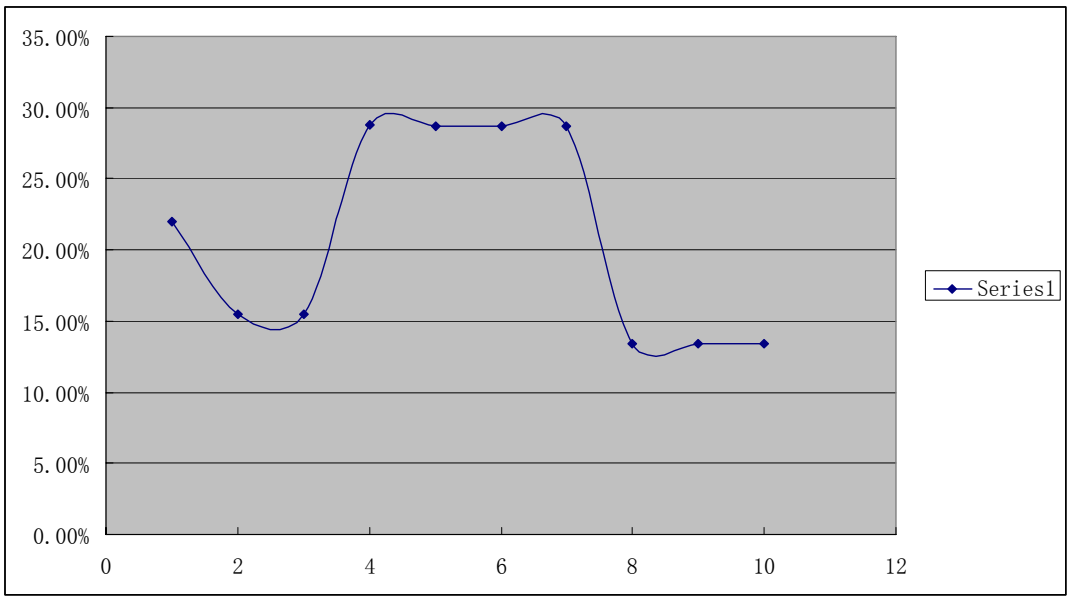


Figure 18: Stress Testing 2007~2009, Pairs Trading, Varying THT

IV Conclusion & Future Works

In this project, we regressed the data using different models (SBM, DBSAM, DBDAM) developed various trading strategies (static trading, dynamical trading and pairs trading).

Due to the limited time, we cannot fully explore the profiles of each strategy, but based on the back testing, the major conclusions are:

1. The static dual beta single alpha model performs better when the parameters are optimized.

2. Dynamical trading based on SBM generally outperforms static SBM.

3. Dynamical trading based on DBSAM not necessarily better than its static version.

4. Moving from weekly updating to daily updating not necessarily yield higher return.

5. Dual beta dual alpha model may produce over fitting.

5. The naked strategies bear system risk when the regression and trading periods fall into different market conditions.

6. The hedged portfolio has a much better performance than the index-only portfolio; and it can produce profits even in stressed situations.

Future Works

First, when trading pairs with the dual beta model, the portfolio is constructed in one of the two values/states, but only updates one period (day) later. The market state, and hence beta may change during holding period. We may need to address this issue so that the portfolio is always hedged.

Second, Unlike CAMP model, where the alpha is assumed to be zero, the market models in our report have a non-zero alpha term. Suppose we are in a long position in the target index, we then will bear a alpha risk if the alpha is significantly negative even if the system risk has been neutralized out. It may be possible to take into account account for this behavior of alpha.

Another possible improvement is to use dynamically updated model for pairs trading.

We may need to consider commissions and transaction cost for realistic P&L.

Finally, the trading based on the market-neutral portfolio looks the best among all the strategies. However, all of these results are obtained by a back testing again a single path, as testing multiple paths for all strategies was unfeasible, if ideal.

For the strategy to be reliable, the following would be appropriate next steps:

(i) work out the Sharp Ratio/Sharp-Omega using historical data or simulation using a realistic price movement model;

(ii) (ii) we also need to test the strategy for other pairs of indices.

Appendix: Computer Programs

A.1 Parameters

nTrade	number of trading days, set as 252
NBT	number of days for back testing trading
TH	threshold weight to partition market
MUF	market updating frequency, 1 for daily, 5 for weekly
THT	threshold to trigger a position
THCLSE	threshold to close
Limit	Position limit, set as +1, -1
Commission	Transaction, Commission, set as 0
LAG	lag for market index, fixed as 1
LAGI	lag for target index, fixed as 1
AS	alpha array resolution, fixed as 0.01
AR	alpha range, fixed as 0.5
Err	Average modeling err

A.2 Main Classes & C/C++ Project

A.2.1 Java

In our project, some classes in AlgoQuant has been slightly modified. To make sure that you can run all programs, I have submitted the whole Java project package. When you open the project in NetBeans, all my project file will be located in the folder "...\src\tensor\dbm", and you can also find the class testing file in the corresponding folder.

The relevant main JAVA classes are listed below.

LogData.java: this class is used to download the data and compute the log return, and then write the log return to the system output (with market index returns comes first and followed by target index return). You can modify the program to download other times periods or securities. The output can be feed to the CUDA program to pre-compute the model parameters for dynamical trading.

TradeBothStaNake.java: this class implements the static trading for naked portfolios (for both SBM and DBSAM), and output the absolute P&L, the annualized return and the number of transactions.

TradeBothDynNake.java: this is the dynamical version of ***TradeBothStaNake.java*** with weekly update. You must feed it by the weekly pre-computed model parameters (see the explanation for the CUDA project file later).
TradeBothDynNakeDaily.java: dynamical trading with daily update.
TradePair.java: trade hedged portfolio based on the static DBSAM model.

A.2.2 CUDA

The project also implements CUDA in C++ to pre-compute the model parameters. For a complete instruction for setting-up CUDA on Windows, check http://julianapena.com/2009/09/how-to-install-and-configure-cuda-on-windows/. (You need CUDA Toolkit, the SDK Code Samples and CUBLAS library).

The source code for this project is ***DBMFlog.cpp***. On the top of the source code, you will be able to set the values of parameters.

After execute the project, you need to copy all the generated .txt file into the working folder of AlgoQuant for dynamical trading.

A.3 JAVA Classes

Transaction.java This class define a simple financial Transaction
Account.java Define a trading account, while has a record of p&L
DbsaTradingNake.java Define the naked static trading based on DBSAM model
SBTradingNake.java Define the naked static trading for SBM
DynDbsaTradingNake.java Define a dynamical naked trading for DBSAM model
DynSBTradingNake.java Define a dynamical naked trading for SBM model
StaPairTrading.java Define the pairs trading for DBSAM model

A.4 Pre-Computed Data

The foloder "...\ AlgoQuant\Pre-Computed Data" contains some pre-computed data (the TH is fixed at 0.3). The sub folder "Weekly" contains pre-computations for weekly update, and "Daily" is for daily updating. To use these data. For example, if you want to simulate the weekly dynamical trading from 01/01/2009~06/03/2011, you then copy the .txt files in the folder "Weekly\2009~2011" into the AlgoQuant working directory, and run ***TradeBothDynNake.java.***

References

Woodward G., Anderson H.M., Does beta react to market conditions? Estimates of 'bull' and 'bear' betas using a nonlinear market model with an endogenous threshold parameter. Quantitative Finance, Vol. 9, No.8,December 2009, 913-924.

Wiggins, J.B., Betas in up and down markets. The Finan. Rev.,1992, 27, 107–123.