Using Support Vector Machines to Trade Aluminium on the LME.

Zac Harland – Krueger Research.

Abstract.

This paper describes and evaluates the use of support vector regression to trade the three month Aluminium futures contract on the London Metal Exchange, over the period June 1987 to November 1999. The Support Vector Machine is a machine learning method for classification and regression and is fast replacing neural networks as the tool of choice for prediction and pattern recognition tasks, primarily due to their ability to generalise well on unseen data. The algorithm is founded on ideas derived from statistical learning theory and can be understood intuitively within a geometric framework. In this paper we use support vector regression to develop a number of trading sub-models that when combined, result in a final model that exhibits above-average returns on out of sample data, thus providing some evidence that the aluminium futures price is less than efficient. Whether these inefficiencies will continue into the future is unknown.

Motivation

Is it possible to design quantitative trading models that result in above-average, risk-adjusted returns? The Efficient Markets Hypothesis (EMH) rules this out as a possibility. This is not surprising; the arguments supporting the EMH are extremely persuasive (Fama, 1965), none more so than the contention that any predictable component will be traded out of the markets by “rational” arbitragers, rendering them efficient once again.

Few would disagree with the idea that a visible discrepancy in price, for the same commodity in two markets, would quickly disappear due to the effects of arbitrage. The reality, however, is that profit opportunities, when they exist, are not as obvious as the arbitrage argument might suggest. They tend to be statistical in nature and, though they may represent a favourable bet, they are not riskless, possibly requiring infinite capital to remove completely (Zhang, 1999). Moreover, the effectiveness of “zero risk” (as opposed to statistical) arbitrage relies on the availability of a perfect (or close) substitute. This is not always the case.

If we accept that obvious, predictable components will be traded out of the markets with relative ease then we must conclude that any remaining inefficiencies, if they exist, are complex in nature to a degree that they are not easily exploited with methods used by the majority of market participants[1]. With this mind, we employ a relatively new machine learning method, support vector regression, in an attempt to extract possible regularities in the market price of aluminium - a market that is arguably less scrutinized than others such as the stock markets.

Aluminium on the LME.

The London Metal Exchange (LME) was established in 1877 and is the world's largest non-ferrous metals derivatives market with a turnover value of approximately US$2000 billion per annum. It is a 24-hour market trading through a combination of continuous inter-office dealing and open-outcry sessions at certain fixed time slots during the day. Of this, hedging represents 75-85% of turnover (Martinot et al., 2000). Aluminium began trading on the LME in 1978 though it was only in the mid-eighties that the contract became widely used when the LME price was adopted as the industry marker price (Figuerola-Ferretti & Gilbert, 2000). At the time of writing the LME price forms the effective price basis for the international base metals market.

The LME three month Aluminium Futures contract (liquidity is mainly concentrated in the three month and cash contracts) is a forward contract between buyer and seller for delivery of 25 tonnes of the metal on a specified three month "prompt" date in the future at a specified price. The majority of positions are closed before prompt by trading offsetting contracts, replacing delivery obligations with monetary differences, officially quoted in USD – though sterling, euro, mark and yen can be used for clearing purposes.

The method of trading on the LME differs to that on most standard futures exchanges partly due to the LME's close links with the physical metals industry and its status as a wholesale market (Gilbert, 1996). While initial and variation margins are called during the term of the contract, profits and losses are not realised until the contract prompt date or until it is closed out – deemed an advantage to the physical users of the exchange. Moreover, when a trade is entered it is at the current three month price, however, when exiting the trade, an adjustment has to be made to take into account the possible contango or backwardation of the contract which depends on demand, supply and interest rate factors for contracts in different delivery periods. This means that the quoted historical three month price being modelled is not what would be experienced in real-time trading. Whilst it is important to bear this in mind, it is not a serious problem as the differences will tend to even out in the long term, with long (short) positions affected adversely (favourably) in conditions of backwardation and vice versa in conditions of contango.

Support Vector Regression.

The Support Vector Machine (SVM) is a powerful machine learning method for classification and regression (see Appendix A for details) which is fast replacing neural networks as the tool of choice for prediction and pattern recognition tasks, primarily due to their ability to generalise well on unseen data. Although the SVM as a learning method has only recently gained in popularity, the underlying principles of the algorithm date back to work done by Vapnik in the early 60's (Vapnik & Chervonenkis, 1964; Vapnik & Lerner, 1963) and are based on ideas derived from statistical learning theory. This recent increase in popularity is due to advances in methods and theory which include the extension to regression from the original classification formulation. For a thorough treatment see (Vapnik, 1998; Vapnik, 1995), the tutorials(Burgess, 1998; Smola & Scholkopf, 1998) and the introduction (Cristianini & Shawe-Taylor, 2000).

SVM Regression involves a non-linear mapping of an n-dimensional input space into a high dimensional feature space. A linear regression is then performed in this feature space. SVMs use the structural risk minimisation (SRM) induction principle which differentiates the method from many other conventional learning algorithms based on empirical risk minimisation (ERM) alone, for example standard neural networks. This is equivalent to minimizing an upper bound in probability on the test set error as opposed to minimising the training set error, which should result in better generalisation. Importantly for practitioners, recently published research has shown successful application of the SVM methodology in a wide variety of fields (Barabino et al., 1999; Joachims, 1997; Mukherjee et al.,1999; Trafalis & Ince, 2000).

The method has a number of advantages over other techniques; the parameters that need to be fitted are relatively low in number and, unlike other methods such as neural networks, they do not suffer from local minima. The two main features of SVMs are their theoretical motivation from statistical learning theory and the use of kernel substitution to transform a linear method into a general non-linear method, with little added complexity.

Model Design and Methodology.

There are different approaches when it comes to deciding how much data to use when designing trading models. One view is that the market is always changing and therefore one does not want to use data too far back in history, as there is a danger that much of it will be redundant. The other approach is to use as much data as is available, reasoning that the only way to have confidence in the model's final results is if it has acceptable performance over as long a data history as possible. We subscribe to the latter approach and use the entire available data set.

To tackle the problem we invoke the *principle of divide and conquer* in that we start by attempting to build a number of trading sub-models, each using a different set of input features. These sub-models are then combined with the intention of creating a final model that is more effective than any one model used in isolation - constituting a type of *committee machine*. Transaction costs are not used as a constraint on model selection when building the sub-models; the rationale being that in some cases strong short term regularities contained within price, but not tradable in isolation due to transaction costs, might be exploitable if derived models are combined with other longer term sub-models and the use of majority voting methods. At the final stage, a *trading wrapper* - which takes into account commission, slippage and order type - is added to simulate real trading.

The data used in this study consist of the "LME provisional closing price" – the daily 5pm close of the second kerb session of the three month LME Aluminium futures contract, covering the period from 11 June 1987 to 4 Nov 1999[2][3]. Data from 5 Nov 1999 to the present date is excluded from this study as it is to be used in a final trading meta-model (that may or may not include the results from this paper) which will require further out of sample testing. This will help to alleviate the problem of “cherry picking” the best models.

The data are divided into training, validation and out of sample sets. The training period covers 2136 days from 11 June 1987 to 17 Aug 1995, the validation set 400 days from 18 Aug 1995 to 27 Feb 1997 and the out of sample from 28 Feb 1997 to 4 Nov 1999, 700 days (see Table 1).

Table 1

Set	Dates	Length
Training set	11 June 1987 to 17 Aug 1995	2136 days
Validation set	18 Aug 1995 to 27 Feb 1997	400 days
Final out of sample set	28 Feb 1997 to 4 Nov 1999	700 days

Inputs.

Finding a good representation of the data to use as inputs and outputs is very important, especially when building trading models. The objective is to find a representation that will render the signal (if one exists) more explicit and/or attenuate the noise component. The number of possible transformations of price to arrive at potential input candidates is infinite, so for the sake of simplicity we considered inputs of the form:

$$x_t = \log(price_{t-0} / price_{t-N}),\ N = 1,\ldots,10 \quad (1)$$

in addition to associated lags up to a maximum of 10 and, finally, a proprietary input based on price seasonality derived from initial exploratory data analysis. Training examples with values beyond the $\pm 3\sigma$ range were clipped and all values were then scaled to zero mean, unit variance. The number of inputs per model was kept to a maximum of six in order to limit complexity. Standard log returns were chosen as the target or dependent variable:

$$x_t = \log(price_{t+1} / price_{t-0})$$

(2)

The objective was to choose input candidates that exhibited correlation to future changes in price i.e., the target. To do so, a number of techniques to measure correlation were used including non-parametric correlation, mutual information and a technique using ANOVA also used in (Harland, 2000a; Harland, 2000b) which is similar to that originally proposed by (Burgess & Refenes, 1995). The main criteria was that any correlation exhibited by a potential input candidate had to be as constant as possible over the full length of the *training* set.

The above procedure resulted in multiple candidate SVR input sets which we restricted in number to ten. SVR was then used to build a model for each input set. The choice of kernel determines the type of the resulting learning machine. Common kernel functions include polynomial, radial basis functions (RBF) and sigmoid kernels. In this experiment we used RBF kernels which have the form:

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

(3)

where σ^2 is the width of the kernel. The SVM regression method has a number of 'tunable' parameters that need to be determined by the user: *C* a regularisation parameter, ε, and in this case the width of the RBF kernel σ^2. Table 2 shows the values that were tested.

Table 2

Parameters	Values
C	10, 100, 1000, 10000,100000
ε	0.1, 0.01, 0.001, 0.0001
σ^2	0.001, 0.01, 0.25, 0.5, 0.75

To simulate trading the following rule was used on the continuous output of the SVM:

IF SVM_OUTPUT>0 THEN LONG ELSE SHORT.

A measure of each model's performance was calculated over the validation set by dividing total daily log returns by the daily standard deviation (similar in nature to the Sharpe ratio). Those parameters that resulted in the highest value for this statistic were chosen for the final sub-model, with the proviso that the gradient of the in-sample training performance was reasonably similar to that exhibited over validation data. The above model design procedure resulted in ten sub-models. These were combined together using majority voting to arrive at the actual daily trading signal - see further details below. For the sake of brevity we provide the results for two of these sub-models in addition to the final model.

Sub-Model 1.

Figure 1 depicts the performance over the whole dataset and is based on trading one contract. The three month aluminium price is re-based to zero at the start of the period and multiplied by 25 (contract is for 25 tonnes) and represents the equity stream experienced by buying and holding one contract. The Cumulative Equity Curve (CEC) represents the profit gained/lost by following the output of the SVM following the rule; If output>0 then long else short. Figure 2 shows the performance over the out of sample data.

Figure 1

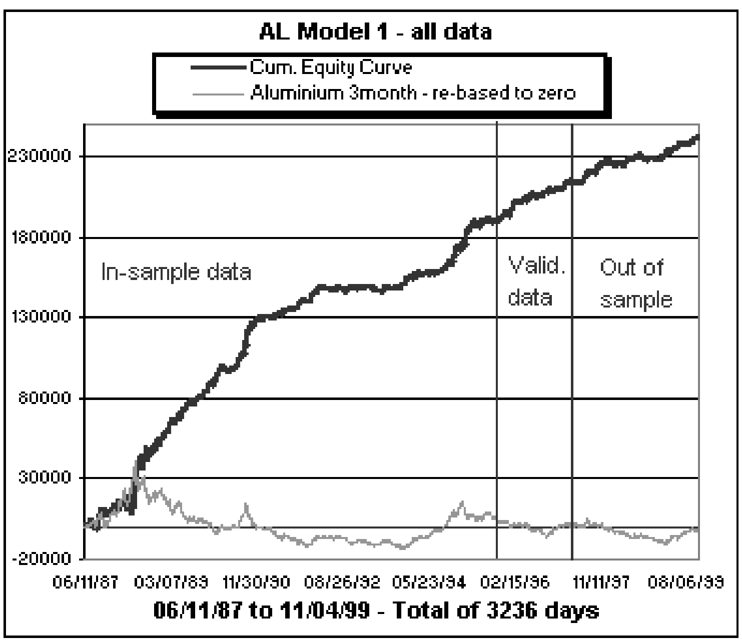


Figure 2

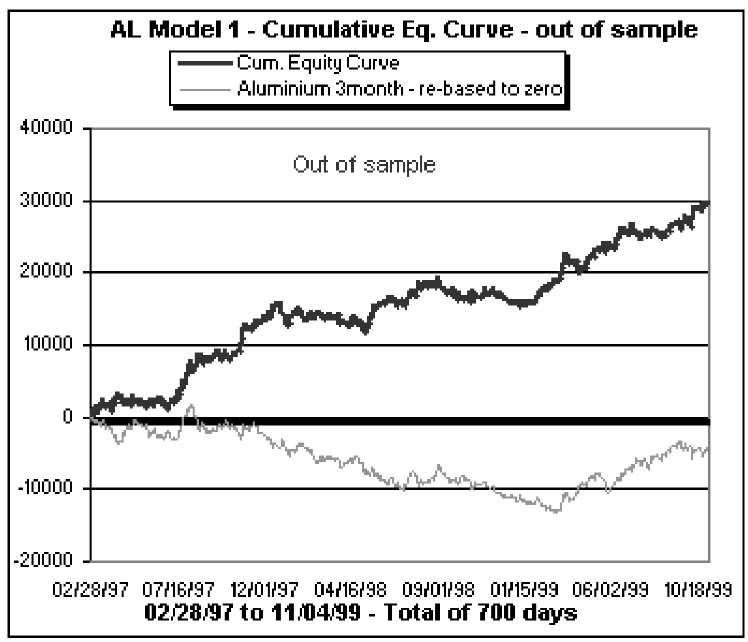


Table 3

Model 1	All Training	2nd half Training	Validation data	Out of sample data
Start Date	870611	910716	950818	970228
End Date	950817	950817	970227	991104
No. of Trades	939	469	168	300
No. of Winners	526	252	101	164
Pct. Winners	56	53	60	54
Gross Profit	$403,912	$130,287	$46,437	$73,500
Net Profit	$191,000	$56,975	$19,925	$30,525
Avg Trade	$203	$121	$118	$101
Average Winner	$767	$517	$459	$448
Avg Loser	$515	$337	$395	$315
Max. Draw	$12,950	$5,675	$4,400	$4,250
Avg. bars in Wins	3	3	3	3
Avg. bars in Losses	3	3	3	3
Avg. bars in Trades	3	3	3	3

A number of trade statistics for this sub-model can be seen in table 3. The results are also included for the second half of the in-sample training set to give a clearer picture of overall performance.

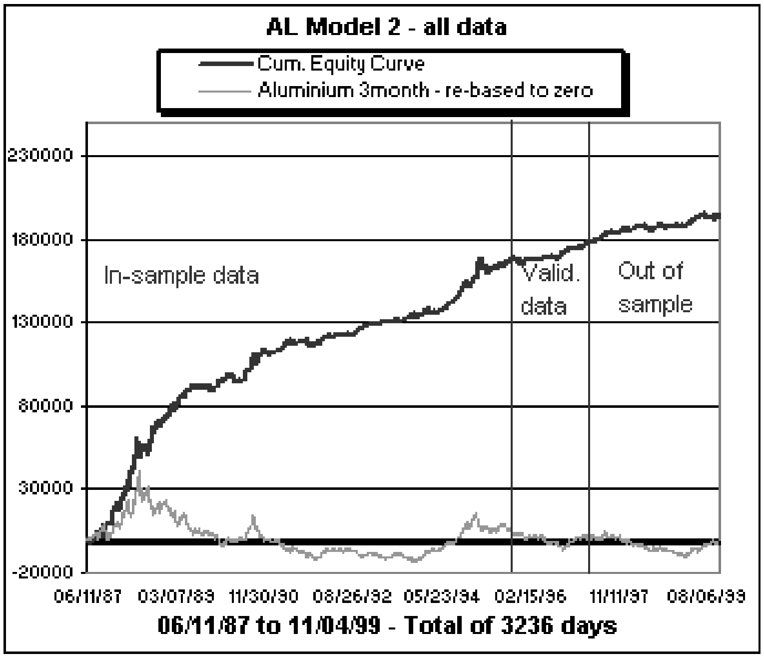
As mentioned above, no transaction costs have been included so the numbers at this stage should be seen as general statistics rather than real trading results. As expected, the training and validation results are good. Most important are out of sample results, which, in this case, exhibit a generally upward sloping CEC along with a similar winning percentage as the rest of the data. The CEC is relatively smooth over the whole data set from in-sample to out of sample, although the average trade figure decreases somewhat in the out of sample data to $101, from $118 in the validation set. If we assume transaction costs of $100 per round turn the model is not tradable in its current state, however, the results suggest that it is managing to detect some form of predictable structure in the price of aluminium and is therefore used in the final model.

Sub-Model 2.

This model uses inputs that were designed to result in longer average trade duration and higher average trade returns than the previous model. Results can be seen in Figures 3, 4 and table 4.

Figure 3

Figure 4



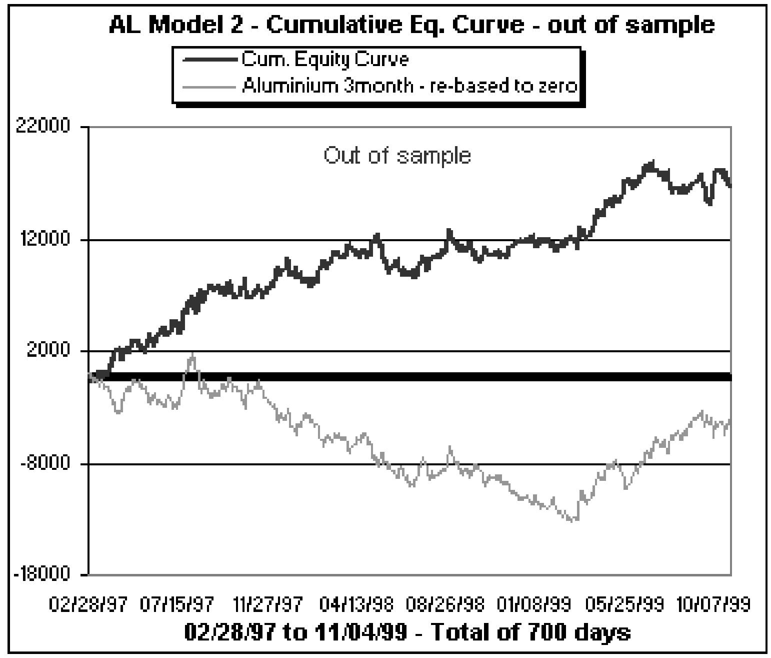


Table 4

Model 2	All Training	2nd half Training	Validation data	Out of sample data
Start Date	870611	910716	950818	970228
End Date	950817	950817	970227	991104
No. of Trades	386	164	61	108
No. of Winners	187	73	27	50
Pct. Winners	48	44	44	46
Gross Profit	$284,450	$79,400	$23,287	$35,350
Net Profit	$164,450	$46,625	$10,700	$17,600
Avg Trade	$426	$284	$175	$162
Average Winner	$1,521	$1,087	$862	$707
Avg Loser	$603	$360	$370	$306
Max. Draw	$11,750	$10,575	$4,600	$4,425
Avg. bars in Wins	8	11	9	10
Avg. bars in Losses	4	4	6	4
Avg. bars in Trades	6	7	7	7

Final Model.

The final model is the result of combining the ten sub-models, using a special rule to trade the result. The rule is explained below, the parameters of which were arrived at via optimising the Sharpe ratio over the in-sample training and validation sets subject to the constraint that the average trade was greater than $250 excluding transaction costs.

Majority Trading Rule:

1) Each SVM sub-model output is assigned 1 if >= 0.0 and -1 if <

0.0.

2) A long (short) trade is only taken if this majority is above (below) a certain (-)threshold. In this case the threshold was ± 4.

3) The result for each model on each day is then summed, producing a number representing the majority decision which can range from -10 to 10.

4) Each trade is then held until the majority decision signals a trade in the opposite direction. This results in what is commonly called a stop & reverse system and is always in the market.

Figure 5

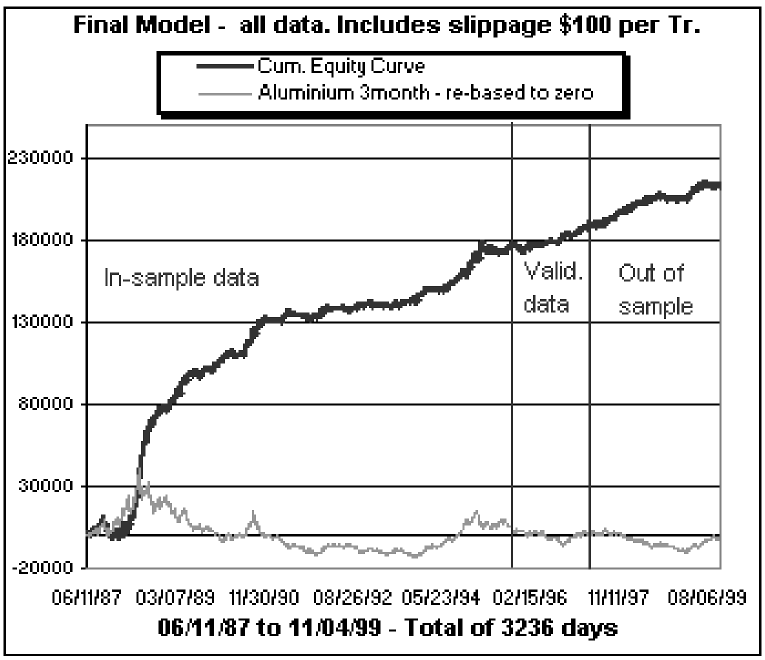
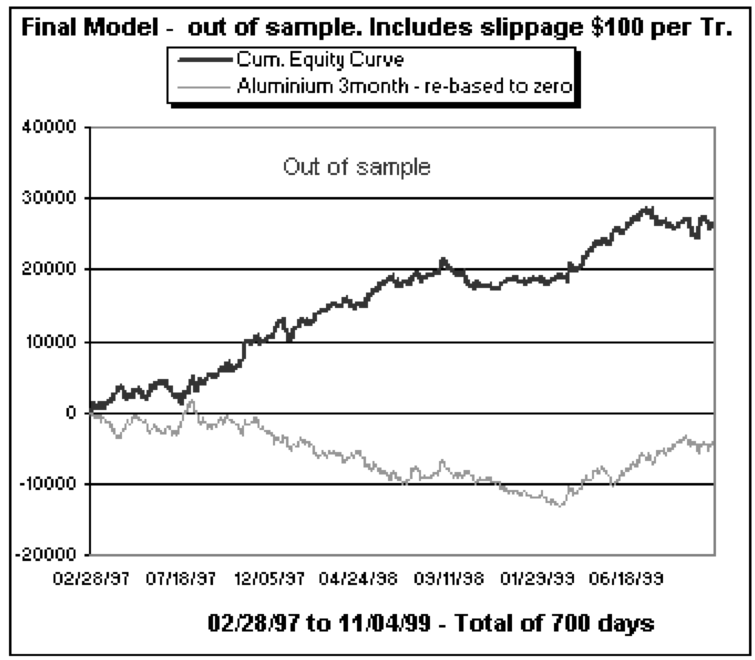


Figure 6



The results can be seen in Figures 5, 6, and in table 5. Slippage & transaction costs of $100 per trade, that is per round turn, are included.

Table 5

Final Model	All Training	2nd half Training	Validation data	Out of sample data
Start Date	870611	910716	950818	970228
End Date	950817	950817	970227	991104
No. of Trades	369	162	75	98
No. of Winners	203	76	41	58
Pct. Winners	55	46	54	59
Gross Profit	$280,450	$78,537	$29,762	$46,550
Net Profit	$172,250	$38,675	$13,125	$25,325
Sharpe Ratio	1.64	1.41	1.61	2.01
Avg Trade	$466	$238	$175	$258
Average Winner	$1,381	$1,033	$725	$802
Avg Loser	$651	$463	$489	$530
Max. Draw	$14,900	$7,175	$6,600	$4,725
Avg. bars in Wins	7	8	4	9
Avg. bars in Losses	6	6	8	5
Avg. bars in Trades	6	7	6	8

The results are encouraging; the Sharpe Ratio for the training and validation sets is just over 1.6, rising in the out of sample period to 2.0 (the Figure of 1.41 for the second half on the training data was due to a higher than average standard deviation of returns). It has an average trade of $258 and a maximum drawdown (MD) of $4725 on the out of sample data. A more sobering $14,900 MD occurs at the beginning of the training period from 19 Oct 1987 to 17 Dec 1987 - during the stock market crash of the same year. The average trade in the validation set is $175, lower than both the training and out of sample sets. At the start of the training data in Figure 5 the CEC exhibits a sharp rise and then has a relatively constant gradient from the early-nineties to the end of the data. One possible explanation is that the market was less efficient at the start of the period, though this is by no means certain. There is a flat period from 8 Mar 1995 to 18 Jun 1996, which would be difficult to trade through, however, the fact that it occurred in the training period suggests the model is not overfitting to any great extent.

No trade exit strategy has been incorporated such as stop loss exits, profit limits etc., as their addition did not result in any discernable improvement. We find this to be generally the case with trading models of this type - probably due to the model signal itself indicating the ideal time to exit a trade. Having said that the final trading system would have emergency stops placed at a distance where they would only be hit in extreme price moves - whether they would be filled at that level is another matter.

Conclusion.

An effective methodology has been developed for the application of support vector regression to trade three-month Aluminium futures on the LME. Ten sub-models were designed and then combined using a majority voting trading rule to obtain a final model that, as a first attempt, exhibits profitable performance over out of sample data. This suggests that the three month aluminium price contains inefficiencies that can be exploited using machine learning. Combining the Final Model with other models built by the author using different methods should result in a usable trading system for aluminium futures. Of course, there is no guarantee that the relationships detected over the time period analysed will continue into the future - it may be that these inefficiencies have been traded out of the market.

References.

Barabino, N., Pallavicini, M., Petrolini, A.k, Pontil, M., & Verri, A. (1999). “Support Vector Machines vs Multi-layer Perceptrons in Particle Identification.” *European Symposium on Artificial Neural Networks*, 1999, Bruges, Belgium

Burges, C.J.C. (1998). “A Tutorial on Support Vector Machines for Pattern Recognition.” *Data Mining and Knowledge Discovery*, 2(2):1-47.

Burgess, A.N. & Refenes, A.N. (1995). “Modelling Non-linear Cointegration in International Equity Index Futures.” *Neural Networks in Financial Engineering* (*Proceedings of the Third International conference on neural networks in the Capital Markets 1995)*. World Scientific.

Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*, Cambridge University Press.

Fama, E. F. (1965). "The Behavior of Stock Market Prices." *Journal of Business* 38:34-105.

Figuerola-Ferretti I. & Gilbert C.L. (2000). “Has Futures Trading Affected the Volatility of Aluminium Transactions Prices?” *AEA International Conference on Industrial Econometrics*, Luxembourg City, July 5,6 and 7, 2000.

Gilbert C.L. (1996). “Manipulation of Metals Futures: Lessons from Sumitomo.” CEPR Working Paper.

Harland Z. (2000a). “Using Nonlinear Neurogenetic Models with Profit Related Objective Functions to Trade the US T-bond Future.” In *Computational Finance 1999 (Proceedings of the Sixth International Conference on Computational Finance)*. Leonard N. Stern School of Business, New York University, January 6-8, 1999. Edited by Yaser S. Abu-Mostafa, Blake LeBaron, Andrew W. Lo, and Andreas S. Weigend, 327 - 342, MIT Press, Cambridge, MA.

Harland Z. (2000b). “Trading a 2 Year Old – the Real-Time Performance of a Neurogenetic T-bond Futures Trading System.” *The Market Technician, Journal of the Society of Technical Analysts. 38*, 14-15

Joachims, T. (1997). “Text Categorization with Support Vector Machines.” Technical Report, LS VIII Number 23, University of Dortmund.

Martinot, N., Lesourd J., Morard B. (2000). “On the Information Content of

Futures Prices - Application to LME Nonferrous Metals Futures." *AEA International Conference on Industrial Econometrics*, Luxembourg City, July 5,6 and 7, 2000.

Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J.P. & Poggio, T. (1999). "Support Vector Machine Classification of Microarray Data." AI Memo 1677, Massachusetts Institute of Technology.

Smola, A. & Scholkopf, B. (1998). "A Tutorial on Support Vector Regression." *NeuroCOLT2 Technical Report Series*, NC2-TR-1998-030.

Trafalis, T.B. & Ince, H. (2000). "Support Vector Machine for Regression and Applications to Financial Forecasting." *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00).*

Vapnik, V.N. & A.Y. Chervonenkis (1971). "On the Uniform Convergence of Relative Frequencies of Events to their Probabilities." *Theory of Probability and its Applications 16*(2), pp. 264—281.

Vapnik, V. (1998). *Statistical Learning Theory*. John Wiley & Sons.

Vapnik, V. (1995). *The Nature of Statistical Learning theory*. Spring-Verlag, New York.

Vapnik, V. & Chervonenkis, A. (1964). "A Note on One Class of Perceptrons." *Automation and Remote Control*, 25.

Vapnik, V. & Lerner, A. (1963). "Pattern Recognition Using Generalized Portrait Method." *Automation and Remote Control*, 24.

Zhang, Y. (1999). "Toward a Theory of Marginally Efficient Markets." *Physica A*, 269, 30-44.

Appendix.

The overall problem of *supervised* learning involves the selection of a function, or hypothesis, from a given hypothesis space that approximates a desired response contained within a set of *labelled* training examples. In the case of regression the labels take on real values.

Given a set of training data $S = \{(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_l, y_l)\}$ where $x_i \in X \subset \Re^N$ and the labels $y_i \in Y \subset \Re$, our objective is to find a function, or hypothesis, which minimizes the risk functional:

$$R(w) = \int L(y, f(x; w)) dP(x, y) \quad (4)$$

Where L is a loss function, $P(x, y)$ is the combined probability density of

x and y, and $f(x;w)$ is the set of hypotheses from which the learning machine can choose. *R(w)* is a measure of how accurate an hypothesis is at predicting the correct label *y* given an input *x*. In general we do not know the true distribution $P(\mathbf{x},y)$ and therefore need to estimate the risk from the data using the *empirical risk* functional:

$$R_{emp}(w) = \frac{1}{l}\sum_{i=1}^{l} L(y_i, f(x_i; w)) \qquad (5)$$

Where *l* is the number of training patterns. The function is called consistent if, asymptotically (as $l \to \infty$), the empirical risk converges to the actual risk.

Replacing the risk functional with the empirical risk functional is known as the *Empirical Risk Minimization* principle (ERM). It can been shown that minimizing the empirical risk does not necessarily result in a good hypothesis that is, minimizing equation (5) does not necessarily equate to minimizing equation (4). To use a trivial example, a function which has the values $f(x_i) = y_i$ over the training set, but the values $f(x_i) = -y_i$ on unseen data exhibits $R_{emp}(w) = 0$, but obviously does not generalize. The problem is that there exists an infinite number of hypotheses that are consistent with the training data but disagree on unseen data.

If the hypothesis space consists of a set of highly expressive functions it is said to have high a *capacity*. An hypothesis chosen from such a high capacity hypothesis space may easily fit the training set without error but may not generalize well. This is known as *over-fitting* where the learned hypothesis has fitted both the signal and the noise component of the (finite) training set.

Intuitively a “simple” hypothesis that exhibits a “good” fit is preferable to a more complex one and will generalize better on unseen data. To achieve good generalization a trade off is required between the degree to which the hypothesis fits the training set and the *capacity* of the learned hypothesis. This trade off is formalized by the Structural Risk Minimization (SRM) principal (Vapnik, 1998) which combines the minimization of the risk functional with a restriction on the capacity of the hypothesis space. Capacity is measured by the so-called Vapnik Chervonenkis (VC) Dimension (Vapnik, 1971) which is central to the underlying theory of SVMs. SRM involves formulating in probability an upper bound on the test set error and then trying to minimize this upper bound.

The main concepts of SVMs are perhaps best explained by examining them within a geometric context. Let us take the simplest case, that of a classification task with linearly separable patterns i.e., there exists a pair (w,b) such that

$$(w \cdot \mathbf{x}_i) + b \geq 1, \quad \text{for all } \mathbf{x}_i \in P$$
$$(w \cdot \mathbf{x}_i) + b \leq -1, \text{for all } \mathbf{x}_i \in N$$

(6)

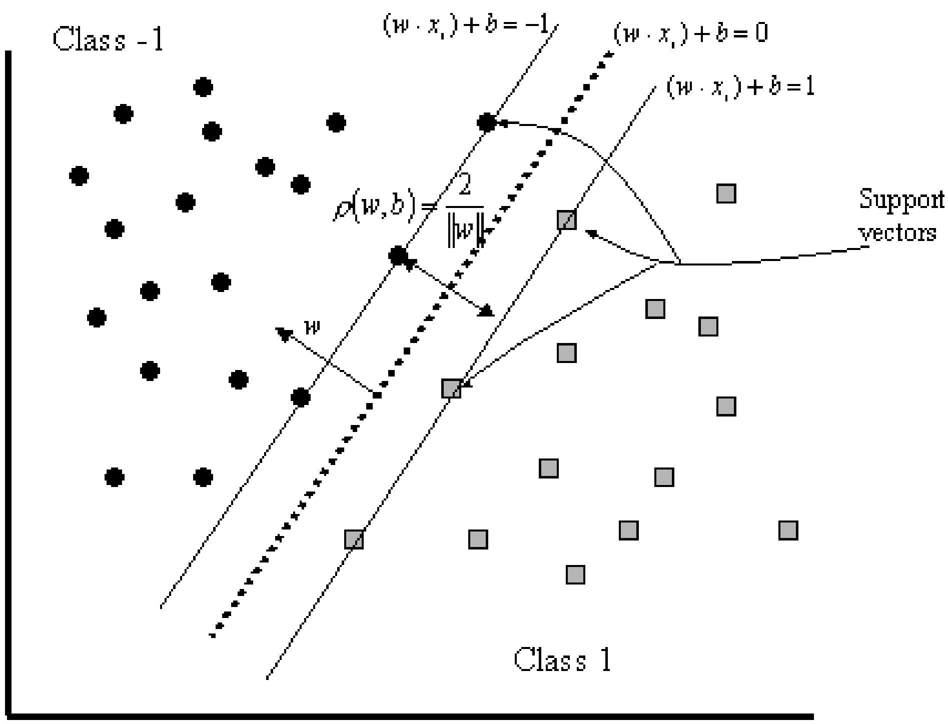
The hypothesis space is then the set of functions

$$f_{w,b}(\mathbf{x}) = \operatorname{sgn}(w \cdot \mathbf{x} + b)$$

(7)

where the weight vector w determines the orientation of the decision surface in the form of a hyperplane and the scalar b is the offset. The distance between the two closest points on either side of a separating hyperplane is known as the *margin of separation*, shown by $\rho(w,b)$, Figure 7.

Figure 7



As the data are linearly separable the classes can be separated by an infinite number of hyperplanes but only one of these will have the maximal margin of separation. In this case the objective of the SVM algorithm is to find this *optimal hyperplane* (Vapnik, 1995) which is more likely to generalise and is uniquely determined by the vectors on the margin, the so-called support vectors. (6) can be written in compact form

$$y_i[(w \cdot x_i) + b] \geq 1, \text{ for all } x_i \in P \cup N$$

(8)

w and b can be rescaled such that the points closest to the hyperplane satisfy $\min_i |(w \cdot x_i) + b| = 1$, obtaining the canonical form (Vapnik, 1998) which helps to simplify the problem. The margin of separation $\rho(w,b)$ can be shown to be equivalent to $\frac{2}{\|w\|}$. Hence the optimal hyperplane is attained by $\max \frac{2}{\|w\|}$ or equivalently, minimising the convex quadratic programming problem,

$$\min_{w,b} \phi(w) = \frac{1}{2}\|w\|^2 \quad (9)$$

$$s.t. \quad y_i[(w \cdot x_i) + b] \geq 1, \quad i = 1,\dots,l \quad (10)$$

In order to solve this Lagrange multipliers are introduced. For a thorough treatment see (Vapnik, 1998).

If the data are non-separable a number of points will lie on the wrong side of the separating hyperplane in which case we need to relax constraints (10). This is attained by introducing "slack" variables: $\xi_i \geq 0, i = 1,\dots,l$

resulting in:

$$y_i[(w \cdot x_i) + b] \geq 1 - \xi_i, \quad i = 1,\dots,l \quad (11)$$

Equation (9) becomes:

$$\min_{w,b,\xi} \phi(w) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i \quad (12)$$

To be completed in final paper.

---

[1] Fundamental and/or traditional technical analysis.

[2] In order to satisfy concerns regarding robustness we also tested the methodology using official am/pm "fixes" and gained similar results.

[3] Prior to 1988 the data is the "P.M. Unofficial" price. The start date is the earliest available from CSI data providers.