Machine Learning in FX Carry Basket Prediction

Tristan Fletcher, Fabian Redpath and Joe D'Alessandro *

Abstract–**Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Relevance Vector Machines (RVM) were used to predict daily returns for an FX carry basket. Market observable exogenous variables known to have a relationship with the basket along with lags of the basket's return were used as inputs into these methods. Combinations of these networks were used in a committee and simple trading rules based on this amalgamated output were used to predict when carry basket returns would be negative for a day and hence a trader should go short this long-biased asset. The effect of using the networks for regression to predict actual returns was compared to their use as classifiers to predict whether the following day's return would be up or down. Assuming highly conservative estimates of trading costs, over the 10.5 year (2751 trading day) rolling out of sample period investigated, improvements of 120% in MAR ratio, 110% in Sortino and 80% in Sharpe relative to the 'Always In' benchmark were found. Furthermore, the extent of the maximum draw-down was reduced by 19% and the longest draw-down period was 53% shorter.**

Keywords: Artificial Neural Network, Support Vector Machine, Relevance Vector Machine, FX Carry, Machine Learning

1 Introduction

1.1 FX Carry Basket Trading

An FX carry basket composed of a long position in high yielding currencies versus a short position in low yielding ones is a common asset for fund managers and speculative traders. Profit is realized by owning (carrying) this basket due to the difference in the interest rates between the high yielding currencies and the low yielding ones. The returns that this basket generate are subject to the risk that the difference between the yields might reduce, possibly becoming negative, and the fact that the exchange rates of the currencies might move unfavorably against the basket holder. A common basket composition is of the three highest yielding G10 currencies bought against the three lowest ones, updated daily to reflect any changes in yield rankings. This basket has a long-bias in the sense that someone holding it will tend to earn a positive return on the asset, subject to periods of negative returns (draw downs).

It would clearly be useful to an FX carry basket trader to be able to predict negative returns before they occur, so that the holder could sell out of or even go short the asset class before it would realize a loss. Several market-observable factors are known to hold a strong relationship with FX carry returns. Furthermore, the returns are known to exhibit short term persistence (i.e. auto-correlation) [1]. It is upon these two phenomena that a trader may wish to capitalize, attempting to predict when returns will be negative and hence reduce the risk of realizing poor returns for the asset over the holding period.

1.2 Machine Learning in FX Carry Basket Trading

Knowing of the relation between carry returns for any given day and both observable exogenous factors for that day and previous days' returns, it makes sense to attempt to incorporate these as inputs into a model where future returns are predicted given information available at the present. This paper outlines the use of three predictive techniques from the area of Machine Learning, representing alternative predictive models for the FX carry prediction task.

Artificial Neural Networks (ANN) have been used extensively in the general area of financial time-series prediction, with varying success e.g. [2], [3], [4] & [5] and hence represent a good starting point with the prediction problem posed here. Support Vector Machines (SVM) [6], being a more recent technique, have been used to a lesser extent e.g. [7], [8], [9] & [10] and indeed there is little evidence of their use in FX carry basket prediction and very little work on the incorporation of exogenous variables when making predictions. The more novel Relevance Vector Machine (RVM) [11] has been used even less in the financial domain e.g. [12], [13], [14] & [15] and apparently never in the area of FX carry prediction.

*(UCL: T.Fletcher@cs.ucl.ac.uk & AtMet Capital LLP: info@metricapartners.com). The authors would like to thank AtMet Capital LLP for their assistance in the research that this paper outlines.

1.3 Supervised Learning

The FX carry basket prediction problem can be expressed as attempting to find a relationship between an output y and a set of D inputs $\mathbf{x}$ where $x = \{x_1, x_2 \dots x_D\}$, i.e. $y = f(\mathbf{x})$. In Supervised Learning, the branch of Machine Learning that the techniques used here are representative of, the function f is learnt from in sample training data so that when new unseen (out of sample) data is presented, a new prediction can be made.

In this paper, y represents a future return of the carry basket, either $T = 1$ or $T = 5$ days into the future. Furthermore, both regression where $y \in \Re$ and classification where $y \in \{-1, +1\}$, i.e. the T-day return is positive or negative, are investigated. $\mathbf{x}$ is composed of five exogenous variables and L lags of y, so that:

$$\begin{aligned} y_{t+T} &= f(\mathbf{x}_t) \\ \text{where } \ \mathbf{x}_t &= \left\{x_t^1 \dots x_t^5, y_t, y_{t-1} \dots y_{t-L}\right\} \end{aligned} \tag{1}$$

1.4 Support Vector Machines

Cortes and Vapnik's Support Vector Machine [6] represents (1) in the form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(\mathbf{x}) + b \tag{2}$$

where $\phi(\mathbf{x})$ represents a non-linear mapping of $\mathbf{x}$ into a higher dimensional feature space, i.e. a basis function, and $\mathbf{w}$ and b are parameters learnt from the N instances of training data.

In classification, these parameters are found by using Quadratic Programming (QP) optimization to first find the α_i which maximize:

$$\begin{aligned} &\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \\ \text{where } \ &\alpha_i \geq 0 \ \forall_i, \ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{3}$$

The α_i are then used to find $\mathbf{w}$:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i) \tag{4}$$

The set of Support Vectors S is then found by finding the indices i where $\alpha_i > 0$. b can then be calculated:

$$b = \frac{1}{N_s} \sum_{s \in S} \left(y_s - \sum_{m \in S} \alpha_m y_m \phi(\mathbf{x}_m) \cdot \phi(\mathbf{x}_s) \right) \tag{5}$$

The mapping $\mathbf{x} \rightarrow \phi(\mathbf{x})$ is intended to make the data linearly separable in the feature space, and to this aim kernels $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ representing the Radial Basis Function:

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)}$$

and the Linear Kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \mathbf{x}_j^T$$

are used in this particular investigation.

For regression, one first needs to decide how significantly misclassifications should be treated (C) and how large the insensitive loss region inside which misclassifications are ignored should be (ϵ). One then proceeds by using QP optimization to find the $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ which maximize:

$$\begin{aligned} &\sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) t_i - \epsilon \sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) \\ &- \frac{1}{2} \sum_{i,j} (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \end{aligned}$$

subject to the constraints ($\forall_i$):

$$\begin{aligned} 0 \leq \alpha_i^+ &\leq C \\ 0 \leq \alpha_i^- &\leq C \\ \sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) &= 0 \end{aligned} \tag{6}$$

The α_i^+ and α_i^- are then used to find $\mathbf{w}$:

$$\mathbf{w} = \sum_{i=1}^{N} (\alpha_i^+ - \alpha_i^-) \phi(\mathbf{x}_i) \tag{7}$$

The set of Support Vectors S is then obtained by finding the indices i where $0 < \alpha_i < C$ and $\xi_i = 0$. b can then be calculated:

$$b = \frac{1}{N_s} \sum_{s \in S} \left(t_i - \epsilon - \sum_{m=1}^{L} (\alpha_i^+ - \alpha_i^-) \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_m) \right) \tag{8}$$

1.5 Relevance Vector Machines

Tipping's Relevance Vector Machine [11] implements a Bayesian probabilistic methodology for learning in models of the form shown in (2). A prior is introduced over the model weights governed by a set of hyperparamaters, one associated with each weight (α_i), whose most probable values are iteratively estimated from the data. If one assumes that the N target values $\mathbf{t}$ that one is attempting to predict are samples from the model subject

to Gaussian distributed noise of zero mean and variance σ^2, and that both $\boldsymbol{\alpha}$ and σ^2 have uniform distributions, then one can derive the model evidence:

$$\begin{aligned} p(\mathbf{t}|\boldsymbol{\alpha},\sigma^2) &= \int p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\boldsymbol{\alpha})d\mathbf{w} \\ &= \frac{1}{2\pi^{\frac{N}{2}}}\left|\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T\right|^{-\frac{1}{2}}\ldots \\ &\exp\left\{-\frac{\mathbf{t}^T}{2}(\sigma^2\mathbf{I}+\boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T)^{-1}\mathbf{t}\right\} \end{aligned}$$

where $\mathbf{A} = diag(\alpha_0, \alpha_1, \ldots, \alpha_N)$, $\mathbf{I}$ is the $N \times N$ identity matrix and $\boldsymbol{\Phi}$ is the $N \times D$ design matrix constructed such that the ith row represents the vector $\phi(\mathbf{x}_i)$.

This evidence can be maximized by the evidence procedure [16]:

1. Choose starting values for $\boldsymbol{\alpha}$ and β.
2. Calculate $\mathbf{m} = \beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{t}$ and $\boldsymbol{\Sigma} = (\mathbf{A} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}$ where $\beta = \sigma^{-2}$.
3. Update $\alpha_i = \frac{\gamma_i}{m_i^2}$ and $\beta = \frac{N-\sum_i \gamma_i}{\|\mathbf{t}-\boldsymbol{\Phi}\mathbf{m}\|^2}$.
4. Prune the α_i and corresponding basis functions where $\alpha_i >$ a threshold value (corresponding to w_i with zero mean).
5. Repeat (2) to (4) until a convergence criteria is met.

The hyperparameter values $\boldsymbol{\alpha}$ and β which result from the above procedure are those that maximize the marginal likelihood and hence are those used when making a new estimate of a target value t for a new input $\mathbf{x}'$:

$$t = \mathbf{m}^T\phi(\mathbf{x}') \tag{9}$$

The variance relating to the confidence in this estimate is given by:

$$\sigma^2(\mathbf{x}') = \beta^{-1} + \phi(\mathbf{x}')^T\boldsymbol{\Sigma}\phi(\mathbf{x}') \tag{10}$$

2 Experimental Design

2.1 Dataset

The total dataset comprised of target and input values from 01/01/1997 to 12/12/2008 (3118 trading days). Various combinations of in sample and out of sample periods were used for the experiments covering this dataset, but after a combination of trial and error and the consideration of the practical implications of implementing the trading system, an in sample period of one year (262 trading days) used to train the networks to make predictions for the following six months (131 days) was decided on. Logarithms of five and one day returns of this carry basket were used as target values along with the five exogenous variables and three lags of the carry basket returns as input variables.

Experiments were conducted by training the networks for one year, outputting predicted values for six months, rolling on six months so that the following out of sample started at the end of the previous in sample and recording the out of sample predictions for the 21 periods that the dataset encompassed in this manner. The time-series of predictions generated using this rolling window of predictions, which encompassed 2751 trading days from 06/01/1998 to 22/07/2008, was used as the input to various simple trading rules so that the cumulative effect of asset returns as if the FX Carry basket had been traded could be ascertained.

2.2 ANN

Neural Networks with one hidden layer using various activation functions, regularization parameters and numbers of hidden neurons were investigated. The effect of pre-processing the inputs using standard normalization methods as well as Principal Component Analysis was researched. After the activation function and a pre-processing method had been decided upon, different numbers of hidden neurons and regularization parameters were used for the several ANN used at the committee stage.

2.3 SVM and RVM

SVM and RVM using radial and linear kernels, alternative pre-processing methods and parameters for C, ϵ and σ (where relevant) were investigated. After settling on a kernel and pre-processing method, different values for C, ϵ and σ were used for the SVM and RVM when used at the committee stage.

2.4 Regression vs Classification

The ANN, SVM and RVM were used both in an attempt to predict actual one day and five day returns and also to predict whether the one/five day return was below various threshold values. It was found that the latter implementation of the networks, i.e. for classification, was much more effective. However, the ANN, SVM and RVM performed differently from each other, depending on how negatively the threshold value was set. Three alternative values for the threshold were therefore used when the networks were combined at the committee stage.

2.5 Committee

Various combinations of different implementations (i.e. parameter settings, number of hidden neurons, kernels etc) of ANN, SVM and RVM in conjunction with each

other were investigated and an optimal committee comprising of the predictions of ten networks was decided upon. These ten predictions were fed into various simple trading rules to generate trading signals, informing a trader to what extent he should be long/short the basket on any given day. It was found that in general the committee of networks was much more effective at predicting five day returns than one day returns, and it was on this basis that the optimal configuration was used.

3 Experimental Results

Conservative transaction costs of 0.04% of the position size per trade were used to estimate the returns that would have been realised for the optimal trading rule based on the network predictions over the 21 out of sample periods of which the dataset comprised. These are shown in the following table along-side the benchmark of constantly holding this long-biased asset:

Table 1: *Portfolio Metrics Comparing the Always-In Benchmark to Using the Committee Prediction*

Time Series	Always-In Benchmark	Committee Prediction
Overall Return	201%	339%
CAGR (%)	6.88%	12.36%
SD Annualized	8.4%	8.3%
SD Loss Annualized	6.9%	5.9%
Max Draw Down	-14.2%	-11.6%
Max DD Time (days)	628	295
Sharpe Ratio	0.82	1.49
MAR Ratio	0.48	1.07
Sortino Ratio	0.99	2.08

Figure 1 details the actual time series of returns using the network predictions alongside the benchmark and hence highlights the many occasions when negative returns could have been preempted and the trader would have profited by going short the basket. In this sense, the network predictions are only able to outperform the benchmark in periods when it falls significantly. This is most evident in the final two and a half year period on the graph.

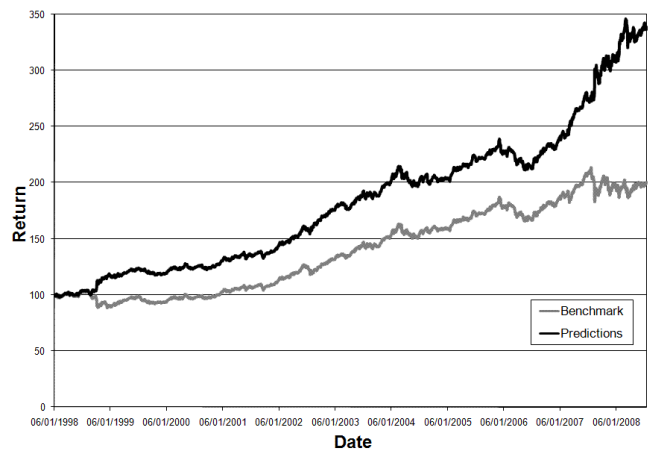

Figure 1: *FX Carry Basket Cumulative Returns for Always-In Benchmark and Committee Prediction*

4 Conclusions

Assuming conservative estimates of trading costs, over the 10.5 year (2751 trading day) rolling out of sample period investigated, improvements of 120% in MAR ratio, 110% in Sortino and 80% in Sharpe relative to the 'Always In' benchmark were found. Furthermore, the extent of the maximum draw-down was reduced by 19% and the longest draw-down period was 53% shorter.

5 Further Work

Instead of allocating each network an equal weighting when combined at the committee stage, a sensible approach would be to weight the network outputs by some estimate of their probable prediction accuracy. For the RVM this could be based on the variance σ^2 associated with each prediction - as expressed in (10). With the SVM, the margin's size can be used as a proxy for the network's out of sample accuracy - see [17]. In the case of the ANN, significant correlation was found between the in and out of sample errors for the ANN, so an out of sample error metric could be based on the in sample errors.

References

[1] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues. quantitative finance," *Quantitative Finance*, vol. 1, pp. 223–236, 2001.

[2] R. R. Trippi and E. Turban, Eds., *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance.* New York, NY, USA: McGraw-Hill, Inc., 1992.

[3] C.-M. Kuan and T. Liu, "Forecasting exchange rates using feedforward and recurrent neural networks," *Journal of Applied Econometrics*, vol. 10, pp. 347–364, 1995.

[4] S. Walczak, "An empirical analysis of data requirements for financial forecasting with neural networks," *Journal of Management Information Systems*, vol. 17, no. 4, pp. 203–222, 2001.

[5] J. Shadbolt and J. G. Taylor, Eds., *Neural networks and the financial markets: predicting, combining and portfolio optimisation.* London, UK: Springer-Verlag, 2002.

[6] C. Cortes and V. Vapnik, "Support vector networks," in *Machine Learning*, 1995, pp. 273–297.

[7] F. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," *Omega*, vol. 29, pp. 309–317, 2001.

[8] L. Cao, "Support vector machines experts for time series forecasting," *Neurocomputing*, vol. 51, pp. 321–339, 2003.

[9] K. jae Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, pp. 307–319, 2003.

[10] Y. N. Wei Huang and S.-Y. Wang, "Forecasting stock market movement direction with support vector machine," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2513–2522, 2005.

[11] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[12] T. V. Gestel, J. A. K. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. D. Moor, and J. Vandewalle, "Financial time series prediction using least squares support vector machines within the evidence framework," in *IEEE Transactions on Neural Networks*, 2001, pp. 809–821.

[13] N. Hazarika and J. G. Taylor, *Predicting bonds using the linear relevance vector machine.* Springer-Verlag, 2002, ch. 17, pp. 145–155.

[14] N. N. P. Tino and X. Yao, "Volatility forecasting with sparse bayesian kernel models," in *Proc. 4th International Conference on Computational Intelligence in Economics and Finance, Salt Lake City, UT*, 2005, pp. 1150–1153.

[15] S.-C. Huang and T.-K. Wu, "Wavelet-based relevance vector machines for stock index forecasting," in *International Joint Conference on Neural Networks (IJCNN)*, 2006, pp. 603–609.

[16] D. J. C. Mackay, "Bayesian interpolation," *Neural Computation*, vol. 4, pp. 415–447, 1992.

[17] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, 1999, pp. 61–74.