Predicting Time of Peak Foreign Exchange Rates

Charles Mulemi, Lucio Dery

0. ABSTRACT

This paper explores various machine learning models of predicting the day foreign exchange rates peak in a given window. We obtained non-trivial results which we can likely improve upon by obtaining more data, domain knowledge and trying out more complicated algorithms that account for more randomness in our data.

1. INTRODUCTION

As international students, we often have to make international transactions involving more than one monetary currency.Due to fluctuations in foreign exchange rates, it is hard to foresee the value we get for the money we send to and from home. Our motivation was to derive a model that would predict the time of the week the exchange rate would peak so that we could make those transactions. Consequently we decided to build a machine learning model that would mine from a list of previously observed econometric features and predict when the foreign exchange rate would peak on a given week. Our inputs were the historical exchange rate time series between two currencies, as well other econometric indicators. We then use a Support Vector Machine (SVM) to make predictions and output the day in the following week when the exchange rate would be at its maximum value. This paper explains the various approaches we took to build our model and the results we obtained.

2. RELATED WORK

In our research we came across interesting attempts at solving similar problems in finance. Osuna, Freund and Federico[5] worked on training support vector machines on foreign exchange time series data. Wei, Nakamori, and Wang[6] similarly used support vector machines to predict weekly movements in stock markets.Their paper points out that the complexity of the finance markets necessitated the use of more than one model.Moreover, the paper establishes preference of SVR over other training models since it minimizes structural risk as opposed to empirical error risk. Another interesting approach at predicting currency uses a neuro-fuzzy model. The model is a hybrid between neural networks and fuzzy logic. This is meant to better account with the inherent non-linearities in the variables used in such models.

2. DATA

We where able to obtain 15 years (5400 days) of Kenyan Shilling vs US dollar exchange rate from the World Bank Forex Data-bank. We also acquired statistics on the following macro-economic indicators: purchasing power parity, inflation rates, interest rates, External Debt and Balance of Goods and services for the Unites States and Kenya the same period. Our choice of these features was based on Joseph Finnerty's paper on Foreign Exchange Forecasting[1]. The exchange rate data was daily whilst the economic indicators were annual. We decided to overcome the differences in the granularity of the data sets by replicating each annual entry of an economic indicator 365 times. We split the data into two sets; 70% was used as the training set whilst the remaining 30% we used as our cross validation set. Figure 1 shows the variation of the exchange rate with time. The data varies wildly on both local and global scales, making predicting peak time in a given window a non-trivial problem.

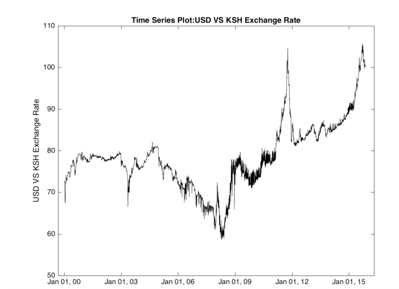

Figure 1: Time Series.

3. METHODS

In order to solve our Forecasting problem, we applied various machine learning algorithms and techniques to the problem. Before we begun using the procedures we are about to discuss, we had to decide on the best way to represent our data before feeding it into our algorithms. We structured our problem as a supervised learning problem. However, since the Time series data does not come explicitly with the desired labels we had to decide on the how to represent our data.

3.1 Sliding Window Representation

For this representation we split the Exchange Rate data into feature vectors of 7 (day) dimensions using a sliding window through the data that advanced by one day after each iteration. Our target variable for each feature vector was the Exchange Rate for the day after the end of the window. Under this representation, our goal was, given a feature vector of the exchange rates for the current week, to predict the exchange rates for the next week and output the day on which the maximum predicted exchange rate falls. We achieved this using Linear Regression and Locally Weighted Linear Regression Models.

Linear Regression

Linear regression makes a transformation from $\mathbb{R}^n \Rightarrow \mathbb{R}$ using the coefficients θ which can be learned by minimizing a cost function $J(\theta)$.
Let $x^{(i)}$ be our feature vector of Exchange Rates for the ith training example and $y^{(i)}$ be the target variable (Exchange rate of the next day) . For our problem $x^{(i)} \in \mathbb{R}^7$ and $y^{(i)} \in \mathbb{R}$
Our prediction of the Exchange rate for the next day is:

$$h_\theta = \theta^T x^{(i)}$$

and our cost function is the square error

$$J(\theta) = \frac{1}{2} \sum (h_\theta(x^{(i)}) - y^i)^2$$

We obtained θ from the normal equations

$$\theta = (X^T X)^{-1} X^T Y$$

Locally Weighted Linear Regression

This approach is similar to Linear Regression described above, differing only in terms of the cost function. For this we have the hypothesis:

$$h_\theta = \theta^\intercal x^{(i)}$$

and the cost function

$$J(\theta) = \frac{1}{2} \sum w^i (h_\theta(x^{(i)}) - y^i)^2$$

where the weights $w^{(i)}$ are

$$w^{(i)} = \exp^{\frac{-\|x - x^{(i)}\|_2}{\tau^2} - \gamma \frac{i-1}{m}}$$

Our parameters γ and τ are tuning parameters. τ captures the similarity between the feature vector we are trying to predict for and the training example $x^{(i)}$. γ on the other hand capture the closeness, in terms of date/time, between the feature vector under consideration and the training example. We again use the normal equation to solve for θ

$$\theta = (X^T W X)^{-1} X^T W Y$$

W is the diagonal matrix of w^is.

3.2 Weekly Representation

We decided on a second format for our data more suited as input to multi-class classification algorithms. In this representation, the feature vectors we derived from the Exchange Rate series were the rates over a week. Thus our 5400 data points reduced to $5404/7 = 772$ feature vectors. Our target variable for week (i), $y^{(i)}$ was set to the day of the peak exchange rate for week (i + 1). We represented days in the week as numbers such that $y^{(i)} \in [1, 7]$ with Monday corresponding to 1 and so on. We fed the above representation to the following multi-class classification algorithms.

Multi-class Support Vector Regression

Support Vector Regression was developed by Vapnik[2]. With this we try to find the optimal hyper plane that separates the data points into the required classes. Given our data of m training examples $(x^{(i)}, y^{(i)}), i = 1, 2...m$ and $y^{(i)} \in [1, 7]$ We describe a one versus all multi-class l_1 regularized SVM below. For our SVM we seek:

$$min_{w_l,\xi_i} \ \frac{1}{2}\sum_{l=1}^{7} w_l^T w_l + C\sum_{i=1}^{m} \xi_i$$

Subject to

$$w_{y_i}^T x_i - w_l^T x_i \geq e_i^l - \xi_i i = 1, ..., m$$

where

$$e_i^l = \begin{cases} 0, & \text{if } y_i = l \\ 1, & \text{otherwise} \end{cases}$$

The parameter C is our regularization parameter which we use to tweak our separating hyper-plane
Our decision function is

$$ouputClass = max_{\ l\ =\ 1,...,7} \ w_l^T x$$

Where x is our feature vector to be classified.

For SVM regression, we relied on LIBLINEAR[3] Matlab package and fitcecoc function in the Matlab statistics and Machine Learning Toolbox.

Softmax Regression

Softmax is a Generalized Linear Model (GLM) for Multi-class classification problems. It is a multi-dimensional analogue of logistic regression. Since our decision classes (days) are mutually exclusive, we chose this instead of using K-Binary classifications. Given the same data set we saw above for SVM, Softmax seeks θ which maximizes

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{7} \ 1\{y^{(i)} = j\}\log\{p(y^{(i)} = j)|x^{(i)\};\theta)}$$

Where $\theta \in \mathbb{R}^{7x7}$ and $\theta_7 = \vec{0}$ so as to avoid the need for a parameter to account for over fitting. The decision function, once θ has been determined is

$$max_j \log\{p(y^{(i)} = j)|x^{(i)};\theta)\}$$

4. RESULTS AND DISCUSSION

	Linear Regression	Locally Weighted LR	SoftMax	SVM (Linear)	SVM (Gaussian)	SVM(Poly)
Best RMSE	0.796	0.751	0.76	0.723	0.739	0.751

TABLE 1 : Model Prediction Performance RMSE

We did not perform any dimensionality reduction on our feature set. Since our feature vectors live in a relatively low dimensional space, We would not have made much gain from this both in terms of run-time improvements and inferences about the data. Our performance metrics were RMSE (Root Mean Square Error), Accuracy and Average Error.

4.1 Regressions

Our first approach to the problem of Peak Exchange Rate Time prediction was to use the Sliding window representation and perform Linear Regression. In order to predict for a feature, we used Linear Regression to generate 7 points corresponding to the next week's predicted Exchange Rates and found the day on which the peak fell. As can be seen from Table 1, we obtained a RMSE of 0.796 for predictions on our testing set. Though not a desirable

performance, the fact that this is less than 1 - $\frac{1}{7\ (days\ in\ week)} = 0.857$, assured us that the θ from Linear regression was not tending towards a random process. A plot of the confusion matrix revealed a maximum per class accuracy of roughly 30%. This occurred for classes 1 and 7. A look at the data revealed that these were the most dominant classes in our training data set and as such, a lot more of our predictions were for these classes.
After performing Linear Regression, Locally weighted Linear regression was used to try to capture more of the structure of the time series data since the LR Confusion Matrix Figure 2 suggests under fitting of the data.

As already described, the tuning parameters τ and γ were used to capture structure. τ accounts for how similar the current example is to the feature set to be predicted and γ temporal closeness of the example set and feature set respectively. As can be seen from TABLE 1 using this approach gave us a 6% reduction in the RMSE we had for Logistic Regression. The effect of the introduction of these parameters can better be understood from Figure 3. It can be noticed from Figure 3 that generally, for a fixed τ, as γ increases, the Percentage accuracy of Locally Weighted LR increases. This agrees with our intuition that points closer in time to our current feature example should give a better sense of the trend of the data around that example and thus lead to better predictions of peak day. Increasing τ also increased performance. This seemed initially counter intuitive since this meant that giving more weight to training examples more similar to the current feature set reduced performance. To rationalize this, we realized that though two features vectors might be similar, the overall trend of the data around them(increasing, decreasing or stable), which is what informs the peak day, are not correlated. Thus, giving greater weight (lowering τ) to these examples reduces performance

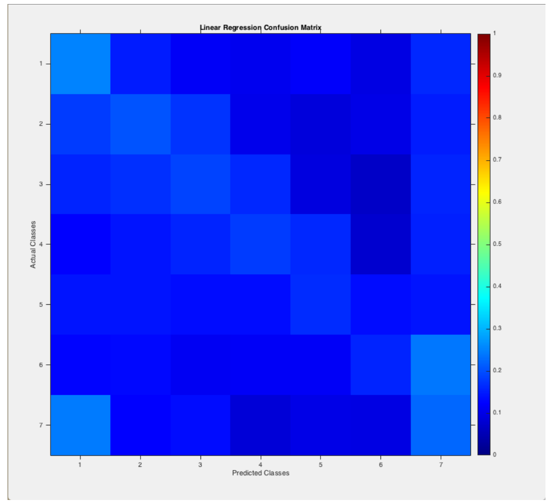

Figure 2: Linear R. Confusion Matrix

Though we had an improved accuracy, our Average Error of prediction went up from 0.779 to 1.42. This indicates that though Locally Weighted Linear Regression captured more of the trend in the data, the predicted points were generally translated vertically by a larger amount as relative to the actual values.

	Linear Regression	Locally Weighted LR
Best Average Error	0.779	1.42

TABLE 2 : Average Error of Regressions

4.2 Multi-Class Classification

Using the Weekly representation of the data already discussed, we first proceeded to perform Multi-Class Support Vector Regression. We tweaked our algorithm by vary the multiclassification technique, trying out different kernels and using l_1 regularization to try to optimize the separating hyper plane.
The techniques we explored were One vrs All, All pairs and Binary complete approaches[4].
In trying out different kernels, we tried to change the dimensionality of the space that our feature vectors live in. We tried Gaussian, Linear and Polynomial Kernels The table below shows the results for each type of kernel under each classification technique.

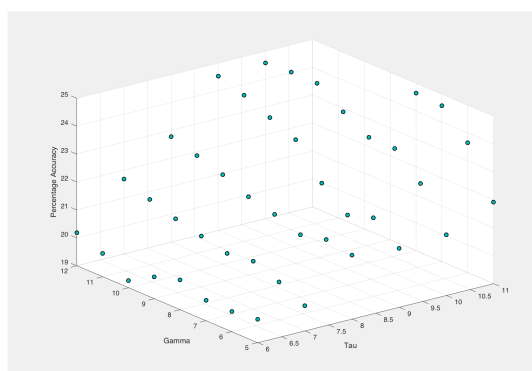

Figure 3: Tau vs Gamma vs Percent Accuracy

Kernel	Technique	Best RMSE
Linear	All Pairs	0.723
Linear	One vrs All	0.791
Linear	Binary Complete	0.734
Gaussian	One vrs All	0.751
Gaussian	All Pairs	0.739
Gaussian	Binary Complete	0.739
Polynomial	One vrs All	0.811
Polynomial	All Pairs	0.751

TABLE 3 : Best Performance SVM under l_1 regularization

The fact that the best performing kernel was the linear kernel informs us that the data does not have much hidden high-dimensional structure. The All Pairs technique consistently outperformed other techniques given the same kernel. We believe that the reason for this follows similarly from why Linear Regression performed well on classes 1, 6 and 7. Since these are the classes with the highest frequency, any binary classifier of either class against the other classes will tend to give a larger number of positive as against negative predictions. Figure 4 The confusion matrix for the Linear Kernel, All Pairs SVM evidences this hypothesis. Majority of the test samples were classified into day 1, 6 and 7. We reasoned that using ordinary down sampling technique to correct the imbalance could eliminate examples key to capturing the Time Series trend. We therefore settled with these results. After determining that a Linear, All Pairs classifier worked best, we proceed to tune this by varying the regularization parameter C. Figure 5 shows our results for this experiment. We found that tweaking the decision boundary, to make it less sensitive to outliers, by increasing the box constraint enhanced performance up till C = 5 and then performance plateaus.
Using Support Vector Regression with the techniques and tuned parameters described, we obtained a 3.7% increase in Percentage Accuracy over Locally Weighted Linear Regression.
Our final experiment was with Softmax regression using the weekly representation. With this approach, we could not get any improvement upon our previous SVR approach. We obtained a RMSE of 0.76, which, though an improvement on Linear Regression is outperformed by Locally Weighted Linear Regression.

(a) All Pairs, l_1 regularized linear SVM Confusion Matrix

(b) Box constraint, C vs Percentage Accuracy

4.3 Feature Selection

Generally, we found that including the Macro-economic indicators mentioned in the data section tended to increase the RMSE. This poor performance can be attributed to our approach to fix the mismatch in the granularity of the Time Series and the Economic Indicators. Replicating the annual values to obtain weekly and daily data introduced correlations that did previously exist in the data and which masked the Time Series trends.
Proceeding notwithstanding this limitation and using Forward Search feature selection, we found that the Macro Economic Indicators that produced the best results were Inflation Rate and Balance of Goods and Services. We had an RMSE of of 0.784 when we run forward search using our best performing SVR.

5. CONCLUSION

The nature of our problem was inherently difficult. Consistent with what we had observed from related work at finance prediction, our best performing mode was the SVM.
More approaches that we intend to explore is to use an ensemble of classifiers that work in tandem as opposed to separate prediction models. More effort can also be directed into feature selection in order to find econometric variables that have the highest correlation with Exchange Rate data and also have daily granularity. Overall, we conclude that though RMSE of 0.723 is not an exciting result, it is promising since it means we are not modeling randomness.

5. REFERENCES

[1]Finnerty, Joseph E. Foreign Exchange Forecasting and Leading Economic Indicators. N.p.: n.p., n.d. Bebr. Web. 22 Nov. 2015.

[2] V. N. Vapnik, The Nature of Statistical Learning Theory. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[3] Journal Of Machine Learning Research 9 (2008) 1871-1874, and Submitted 5/08; Published 8/08. "LIBLINEAR: A Library for Large Linear Classification." (n.d.): n. pag. Web. 11 Dec. 2015.

[4] http://www.mathworks.com/help/stats/fitcecoc.html

[5] Osuna, Edgar, Robert Freund, and Federico Girosi. "An improved training algorithm for support vector machines." Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop. IEEE, 1997.

[6] Huang, Wei, Yoshiteru Nakamori, and Shou-Yang Wang. "Forecasting stock market movement direction with support vector machine." Computers & Operations Research 32.10 (2005): 2513-2522.

[7]Lin, Chin-Shien, et al. "A new approach to modeling early warning systems for currency crises: Can a machine-learning fuzzy expert system predict the currency crises effectively?." Journal of International Money and Finance 27.7 (2008): 1098-1121.