PREDICTING THE EXCHANGE RATE EUR-LEU WITH SVM

Dumitru Ciobanu Ph. D Student
University of Craiova
Faculty of Economics and Business Administration
Craiova, Romania

Abstract: Support Vector Machine (SVM) is one of the most promising algorithms from learning machines domain. First, SVM was designed to solve classification problems but later they was adapted to deal with regression problems. In this paper I present a model that use SVM to predict the exchange rate EUR-LEU. I've used the Matlab programming language for numerical simulations.

JEL classification: C63, G17.

Key words: Support Vector Machines; regression; prediction; exchange rate.

1. INTRODUCTION

Support Vector Machines have been developed to solve the problem of classification. Problems of regression differs from classification in the sense that observations are associated with numeric values and not a label from a discrete set.

Given this difference can easily adapt the application of support vector machines to regression problems where we are dealing with numerical observations.

Thus, the regression problem can be formulated as follows:
with data

- A universe of data X,
- A sample set S, $S \subset X$,
- A target function $f: X \to \mathbb{R}$,
- A training set D, where $D = \{(x, y) | \, x \in S \text{ and } y = f(x)\}$,

we need to determine a model $\hat{f}$: $X \to \mathbb{R}$ using D so that $\hat{f}(x) \cong f(x)$ for any $x \in X$.

The basic idea of machine learning is retained, namely the determination of a model (function) that fits best with the target function for all data elements in the universe.

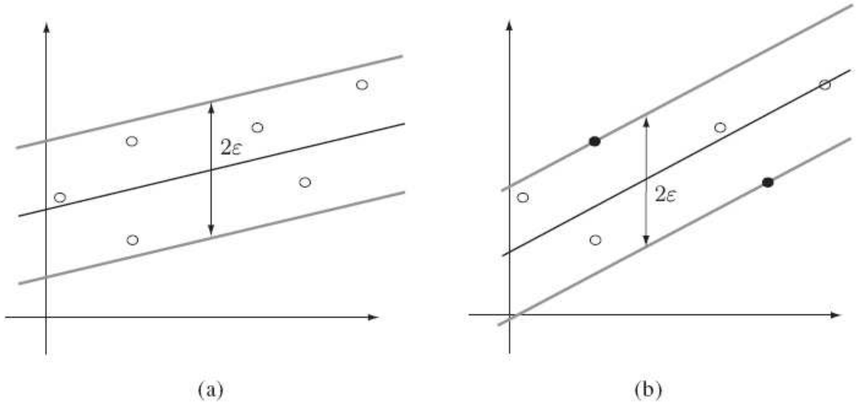
As in the case of classification X is a multidimensional real data set that is $X \subset \mathbb{R}^n$ with $n \geq 1$.

To develop support vector machine in regression context is used the maximization of the margin. The used idea is the same as for classification maximizing the margin.

It gives a hyperplane and maximizes distances from observations to it. For a regression problem where the observations of training set

$$D = \{(\overline{x}_1, y_1), (\overline{x}_2, y_2), \ldots, (\overline{x}_l, y_l)\} \subset \mathbb{R}^n \times \mathbb{R},$$

are contained in a hypertube with width 2ε and $\varepsilon > 0$ (Fig. 1), this can be interpreted as a regression model considering that there is a hyperplane positioned in the center of the hypertube that approximate observations. Usually there are several ways of positioning the hypertube with width 2ε to contain all the training observations. There is one optimal positioning of hypertube so as more observations are pushed closer to the outside of hypertube. Optimal Hypertube alignment is obtained when the distances from the observations to the central hyperplane are maximized. This is illustrated in Figure no. 1, where filled circles represent observations that act as constraints in optimization problem.



Source: (Hamel, 2009) page 197.

Figure no. 1 Linear regression modeling using hypertubes with width 2ε (a) a hypertube containing all observations, (b) optimal regression model with maximum margin.

This is very similar to the problem of maximizing the margin of decision area, and we can use the same optimization problem to determine the optimal alignment by adjusting the hyperplane with corresponding constraints.

Is solved the optimization problem

$$\min \Phi(\overline{w}, b) = \min_{\overline{w}, b} \frac{1}{2} \overline{w} \cdot \overline{w}$$

with constraints

$$y_i - \hat{f}(\overline{x}_i) \leq \varepsilon$$
$$\hat{f}(\overline{x}_i) - y_i \leq \varepsilon$$

2. Predicting the exchange rate EUR-LEU with SVM

I have downloaded the time series of the exchange rate between the euro and leu from the website of the National Bank of Romania, http://www.bnr.ro/Raport-statistic-606.aspx .

The program used to perform simulations was Matlab version 7.12.0 (R2011a).

I imported into Matlab the data of exchange rate euro-leu and I have implemented a program that uses support vector machine with time delay (autoregressive) to predict next value of the exchange rate.

I divided the observations into two sets of training and of test. For testing I kept the last 100 observations. I measured the levels of training and testing using Mean Square Error (MSE).

A successful training is that when the Mean Square Errors obtained for the two sets of training and testing are similar. From tests I noticed that the two Mean Square Errors depend on the number of observations and the number of delays used. Optimal number of observations indicated by tests was between 800 and 1100 (Fig. 2) and the number of delays between 17 and 19 (Fig. 3).

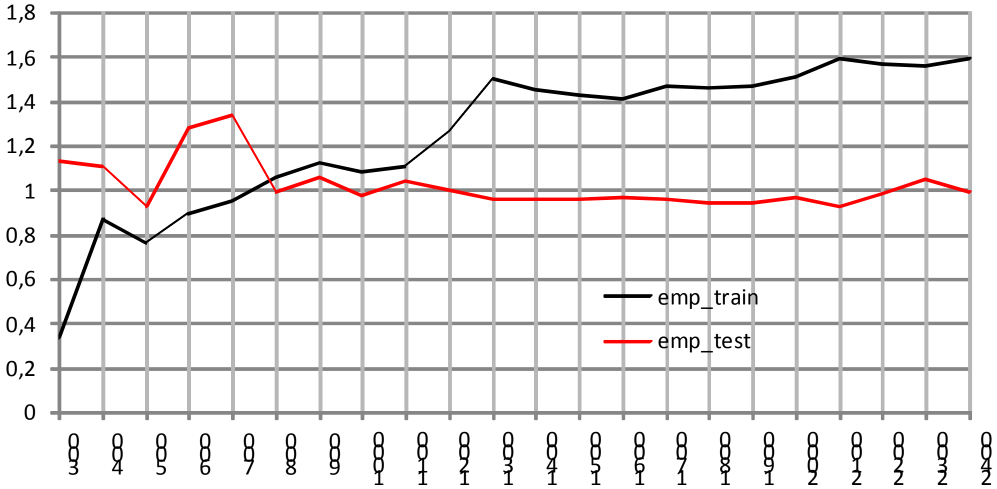


Figure no. 2 Comparative evolution of Mean Squared Error for the training and test sets when varies the number of observations used.

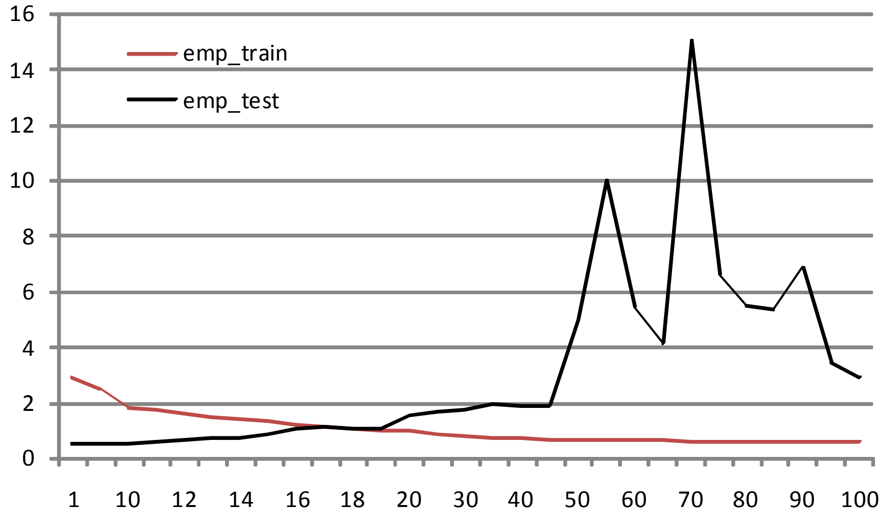


Figure no. 3 Comparative evolution of Mean Squared Error for the training and test sets when varies the number of time delays.

Figure no. 2 present the evolution of Mean Squared Error for the training and test sets when varies the number of observations used. The numbers from Ox axe represent the number of observations used and the numbers from Oy axe are the values of MSE divided by 10^{-4}. In Figure no. 3 is presented the evolution of Mean Squared Error for the training and test sets when varies the number of time delays. The numbers from Ox axe represent the number of time delay used and the numbers from Oy axe are the values of MSE divided by 10^{-4}.

To determine the model I chose to use 800 observations and 18 time delay. Of the 800 observations I used 700 for training and I kept 100 for test.

In figures 4, 5, 6 and 7 are presented the results obtained for the training set.

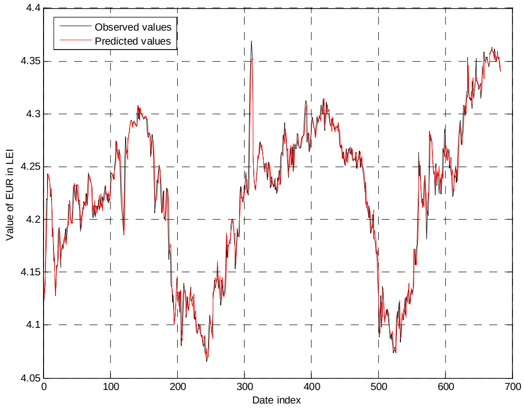


Figure no. 4 The exchange rate and predictions for training set.

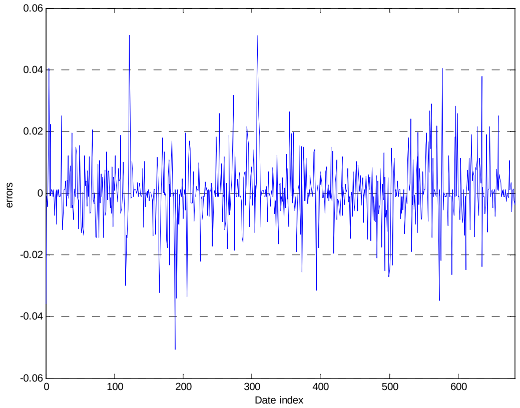


Figure no. 5 The predictions errors for training set.

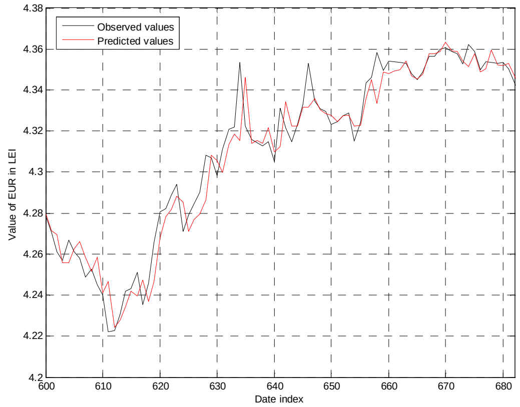


Figure no. 6 A closer look at the exchange rate and prediction for the last part of the training set.

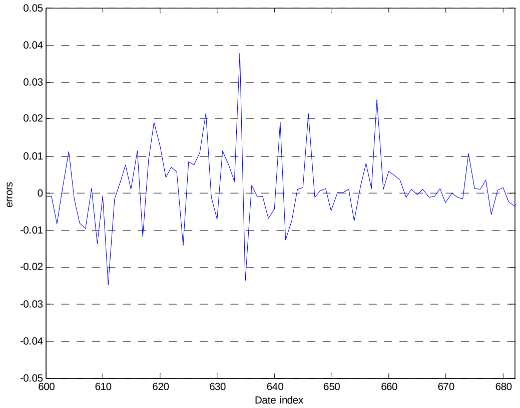


Figure no. 7 The predictions errors for the last part of the training.

To determine the parameters I used both the training and test sets. I used the mean square error to measure the effectiveness of the model. For the test set I calculated predictions with a single step, ie every time I used 18 known values to determine the prediction for the next value (Fig. 8).

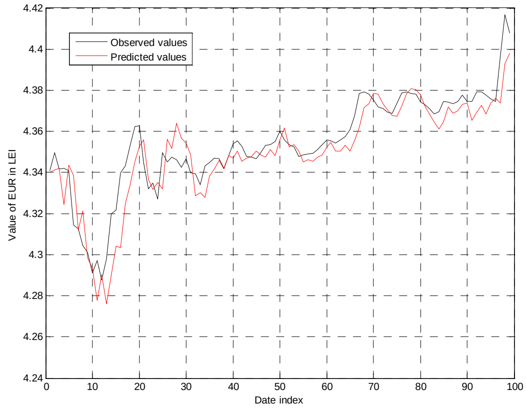


Figure no. 8 The exchange rate and prediction with one step for the test set.

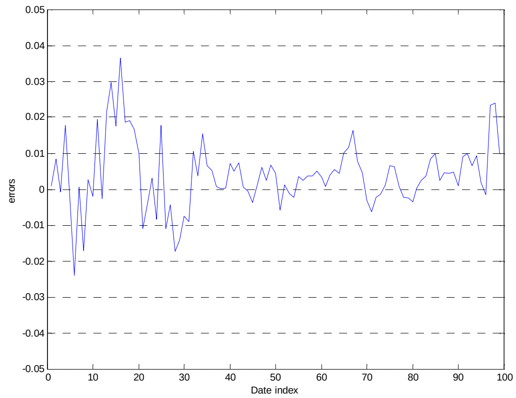


Figure no. 9 The predictions errors for test set in the case of prediction with one step.

Predictions on several steps forward are obtained using as inputs predictions obtained at previous steps. From the following two figures is observed the evolution of predictions and prediction errors for the 100 steps forward.

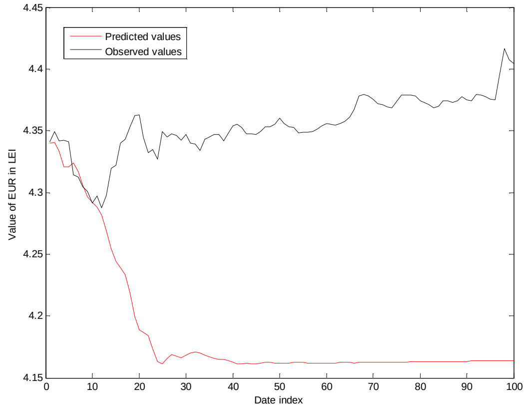


Figure no. 10 The exchange rate and prediction for 100 steps forward.

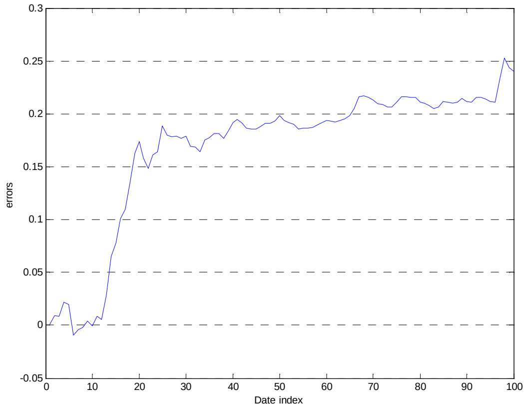


Figure no. 11 The predictions errors for test set in the case of prediction for 100 steps forward.

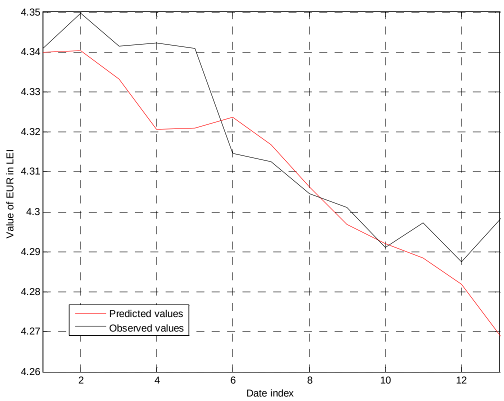


Figure no. 12 First predictions in case of 100 steps forward prediction.

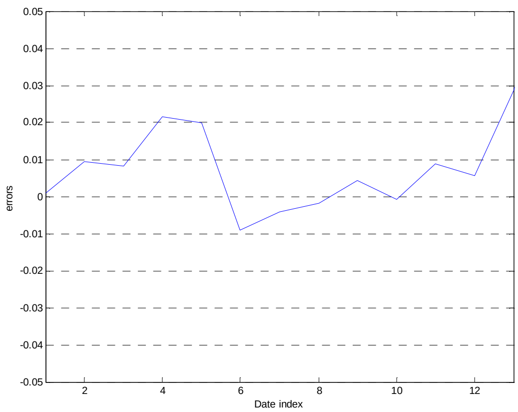


Figure no. 13 First errors for test set in the case of prediction for 100 steps forward.

If the predictions of several steps forward I got predictions with an acceptable error only for a few steps, about 10, as shown in Figure 13.

5. CONCLUSIONS

Using SVM for prediction is a good alternative for both traditional methods as well as those methods arising from computational intelligence like Neural Networks (NN). To adjust the model with data we must to determine just a few parameters and this represent a major advantage of SVM over NN.

As revealed from tests, using SVM for prediction gives the best results when we use about 1000 observations. This brings a very short time necessary to train the SVM, about 20 seconds when using a computer with Intel Cuore 2 Duo processor, who works at a maximum frequency of 2.53 GHz, and 1 GB RAM.

Such a short time necessary to train the model allows us to perform retraining whenever needed and opens the way to use SVM to obtain real-time predictions.

If multi-step ahead predictions were obtained acceptable values of errors for the first 12 predictions.

REFERENCES

1. Bennett, K. P.; Campbell, C. — Support vector machines: hype or hallelujah?, SIGKDD Explorations Newsl., Vol 2, No. 2, pp. 1–13, 2000.
2. Campbell, W. M.; Campbell, J. P.; Reynolds, D. A.; Singer, E.; Torres-Carrasquillo, P. A. — Support Vector Machines for Speaker and Language Recognition, 2006.
3. Campbell, J. Y.; Lo, A. W.; MacKinlay, A. C. — The Econometrics of Financial Markets, Princeton University Press, Princeton, NJ, 1997.
4. Cao, L.J.; Tay, F.E.H. — Support Vector Machine With Adaptive Parameters in Financial Time Series Forecasting, IEEE Transactions on Neural Networks, Vol. 14, No. 6, (2003).
5. Chapelle, O.; Sindhwani, V. — Optimization Techniques for Semi-Supervised Support Vector Machines, Journal of Machine Learning Research, No. 9, pp. 203-

Keerthi, S.S. 233, 2008.

6. Chen, S., Jeong, K., Härdle, W., Support Vector Regression Based GARCH Model with Application to Forecasting Volatility of Financial Returns, SFB 649 "Economic Risk", Humboldt-Universität zu Berlin, Berlin, 2008.
7. Ciobanu, D. On a model for predicting the exchange rate euro-leu with a NAR neural network, Proceedings of the 19th International Economic Conference – IECS 2012, Sibiu, 2012.
8. Cortes, C., Vapnik, V. Support vector networks, Machine Learning, 20(3), pp. 273-297, 1995.
9. Guggenberger, A. Another Introduction to Support Vector Machines, 2008, available online at http://mindthegap.googlecode.com/files/AnotherIntroductionSVM.pdf, accessed May 2012.
10. Gunn, S. R. Support Vector Machines for Classification and Regression, University of Southampton, 1998, available online at http://www.svms.org/tutorials/Gunn1998.pdf , accessed April 2012.
11. Hamel, L. Knowledge Discovery with Support Vector Machines, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2009.
12. Lovell, B. C., Walder, C. J. Support Vector Machines for Business Applications, Business Applications and Computational Intelligence, Publisher Idea Group, Hershey, U.S.A, pp. 267-290, 2006.
13. MacKay, D. J. C. Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks, Network: Computation in Neural Systems, no. 6, pp. 469–505, 1995.
14. Maiga, A. An Introduction of Kernel Method for Support Vector Machine, Universite de Montreal, Quebec, Canada 2011.
15. Meyer, D., Leisch, F., Hornik, K. The support vector machine under test, Neurocomputing, 55: 169–86, 2003.
16. Schölkopf, B., Smola, A. Learning with Kernels, MIT Press: Cambridge, MA, 2002.
17. Smola, J. Regression estimation with support vector learning machine, Master’s thesis, Technische Universitat Munchen, 1996.
18. Shu-Xia Lu, X.-Z. W. ”A comparison among four svm classification methods: Lsvm, nlsvm, ssvm and nsvm”, Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4277–4282, Shanghai, China, 2004.
19. Vapnik, V. Empirical Inference Science, Afterword in 1982 reprint of Estimation of Dependences Based on Empirical Data, 2006.
20. Yu, H., Yang, J., Han, J., Classifying Large Data Sets Using SVMs with Hierarchical Clusters, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 306 – 315, ACM New York, NY, USA, ISBN:1-58113-737-0, 2003.