Detection of Currency Crises by a Novel Rule Extraction Method from Support Vector Machine

Prasan Pitiranggon, Somsri Banditvilai , and Nunthika Benjathepanun

Abstract—This study attempts to obtain a set of human comprehensible fuzzy if-then rules for the detection of currency crises from Support Vector Machine (SVM). SVM is used with explanatory variables known to be associated with currency crises to detect occurrences of currency crises. Fuzzy if-then rules are then obtained from the SVM through our novel rule extraction method which is called Support Vector Space Expansion (SVSE) method in order to unveil human comprehensible patterns behind SVM black-boxed system decision. The overall results of detection of currency crises of the fuzzy if-then rules are comparable to those from the SVM, and the if-then rules obtained may be used by financial experts to try to explain patterns of related financial statuses when currency crises occur, plus the if-then rules can also be easily incorporated into a software program using any popular computer language.

Keywords—Currency Crises, Fuzzy Rule Base, Rule Extraction, Support Vector Machine.

I. Introduction

Rule extraction methods are used to obtain fuzzy if-then rules from artificial neural networks (ANN) and SVM [1], [5]. SVM [4], [12], has been shown to outperform ANN in classification in many applications [18], so the if-then rules obtained from SVM should be superior to rules obtained from ANN in many applications. Methods for SVM rule extraction can be either pedagogical or decompositional. Pedagogical techniques are those that try to relate inputs with outputs without making use of system structure, but decompositional techniques do make use of structure of the system.

We have proposed a decompositional rule extraction technique from SVM in our recently published paper [16] which we will now call it Support Vector Space Expansion (SVSE) rule extraction method. Unlike other rule extraction methods from SVM, our technique makes use of strength of firing signals of support vectors partly similar to the way the original SVM makes decision, then each support vector expands its space to cover non-support vectors with the strongest Gaussian kernel function values. SVSE also guarantees that the number of final rules is equal or less than the number of support vectors obtained by SVM. We have validated our method against SVM using 5 benchmark data sets and found that the classification power of the SVSE is comparable to SVM.

In this study, we are trying to apply SVSE to a non-benchmark data; specifically, we want to use SVSE with financial data to predict currency crises [6].

There have been studies in the field of currency crises since the 1970's. Three models have been widely accepted to explain different currency crises occurred around the world [13], [15], [17]. The first model is based on the study of Krugman in 1979; it is used to explain currency crises in some countries in Latin America in 1970's to 1980's. The second model is based on the study of Obstfeld in 1994; it is used to explain currency crises in some countries in Europe in 1992 and Mexico in 1994. The third model is used to explain currency crises in some countries in Asia in 1997 to 1998. Many indicators and indices are believed to be statistically linked to these models. Through these variables, researchers have used many techniques to come up with the most accurate detection systems. These techniques involve traditional statistical methods [17], ANN [15], and more recently SVM [2] [8].

Either ANN or SVM does not reveal clearly to human comprehension how it makes decision; this is known as black-boxed characteristic [1], [11]. There is a need to have fuzzy if-then rules for the decision in order to make human expert understand the decision, so we will extract how SVM makes each decision in the form of if-then rules, through which we will predict whether there is a currency crisis or no crisis in a particular year.

II. Background

This study uses SVM classifier with data from countries in Latin America and Asia [17] to detect currency crises during 1980 to 2002. The data used are 11 explanatory variables which have been shown in the past to be significantly associated with the occurrences of currency crises.

The 11 explanatory variables are:

1) Overvaluation of real exchange rate (OVERRER)

2) Liquid liabilities of monetary authority and commercial banks to foreign reserves (M2_FR)

3) Dummy for capital liberalization (LIBDUM_FORI)

4. Dummy for the ratio of short-term debts to foreign reserves (DUMST_FR)

5. Ratio of external debt to Gross National Income (EXD_GNI)

6. Current account as a percentage of Gross Domestic Product (CA_GDP)

P. Pitiranggon is with King Mongkut's Institute of Technology Ladkrabang, Chalongkrung Rd., Ladkrabang, Bangkok 10520, Thailand (phone: +668-1303-6484; e-mail: prasan.pitiranggon@hotmail.com).

S. Banditvilai is with King Mongkut's Institute of Technology Ladkrabang (e-mail: kbsomsri@kmitl.ac.th).

N. Benjathepanun is with King Mongkut's Institute of Technology Ladkrabang (e-mail: kbnunthi@kmitl.ac.th).

7. Domestic credit growth (CREGROW)
8. Growth rate of Gross Domestic Product (GDPGROW)
9. Lending boom as a percentage change of ratio of financial system claims on private sector relative to Gross Domestic Product over four-year period (LB)
10. Capital inflows reversal (KAREVS)
11. Dummy for regional contagion effect (CONT)

Even though SVM can detect currency crisis patterns, the way it makes decision is not obvious to human comprehension. So our novel decompositional rule extraction method, SVSE, is used to reveal human comprehensible if-then rules in this study. It uses each support vector to pair with each non-support vector to select the strongest firing signal among the support vectors and expand the coverage in input space until no more non-support vectors left to pair with. Other studies on rule extraction techniques for SVM using decompositional techniques are SVM + Prototype [14] , Tree related method [3] , and Cubes and separating hyperplane method [7], while the ones using pedagogical techniques are Iter [9] and Minerva [10].

The SVM + Prototypes algorithm is an iterative process that starts by training an SVM to obtain support vectors. It then uses a clustering algorithm to find new subsets and calculate the centroid of each cluster in low dimensional space. For each centroid, it finds the support vector located farthest from the prototype and uses the prototype as center and the support vector as vertex to create a hypercube in the input space. Then a partition test on each of the hypercubes is performed. This partition test is performed to minimize the level of overlapping between cubes for which the predicted class is different. If all subsets are processed, the algorithm converts all of the current hypercubes into rules. Ellipsoids can also be used in place of hypercubes. For another decompositional technigue, decision tree is used. This tree related method makes use of the information provided by the support vectors and the parameters associated with them. In the first stage which is a learning stage, the approach handles the rule-extraction by using labeled patterns to train an SVM and get an SVM model as a classifier with acceptable accuracy. In the second stage which is a rule generation stage, the objective is to express the concepts learned by the model in a comprehensible form. The steps are firstly select the patterns that become support vectors but discard their class label, then use the SVM model to predict the class label of those patterns, hence a special synthetic data set is generated. Finally the synthetic data set is used to train a machine learning technique with explanation capability; hence symbolic rules that represent the concepts learned by the SVM model are generated. Cubes and Separating Hyperplane method is the last decompositional technique mentioned. In this method, all input data are transformed into square observations in the interval 0 to 1. Then the method searches for a cube with one vertex on the separating hyperplane and the other located in the region below the separating hyperplane. Optimal cubes can be found from these cubes in two ways – volume maximization and point coverage maximization. The optimal cube divides the region below the separating hyperplane into two new regions – region above and on the right hand side of the cube. For an N-dimensional input space, one rule will create N new regions. Then a new optimal cube is found recursively for each new region. The algorithm stops after a predefined maximum number of iterations.

Iter is the first pedagogical method for SVM rule extraction mentioned. The main idea of the algorithm is to iteratively expand a number of hypercubes until they cover the entire input space. The algorithm starts with the creation of a user defined number of random starting cubes. These cubes correspond to points in the input space. In each iteration, the following steps are executed. Firstly, for each hypercube and for each input dimension, the algorithm calculates how far the cube can be expanded to both extremes of the dimension before it intersects with another cube; these distances are called LowerLimit and UpperLimit. Secondly, for each hypercube and for each input dimension, the algorithm calculates the size of the update. The update equals a user-specified constant, unless this size would result in overlapping cubes. If this is the case then the update is smaller such that the two blocks become adjacent. Thirdly, for each hypercube and for each input dimension, the algorithm creates two temporary cubes adjacent to the original cube along the opposite sides of each input dimension with a width of update value from the second step. For each of both cubes, the algorithm creates a number of random points lying within the cube and calculates the mean prediction for these points according to the trained continuous regression model. The difference between each of both means and the mean prediction for the original cube respectively are called LowerDiff and UpperDiff. Lastly find the global minimum over all cubes of these differences and combine the temporary cube for which the difference was minimal with its original cube. The mean prediction for this cube is updated, and all other temporary cubes are removed. Each of these cubes can then be converted into a rule of the following form:

if Var 1 $\in$ [Value1Low ,Value1High] and Var 2 $\in$ [Value2Low, Value2High]
. . . and Var M $\in$ [ValueMLow, ValueMHigh] then predict some Constant

where M is the dimension of the input space. Minerva is the other pedagogical method for SVM rule extraction mentioned. Minerva is similar to sequential covering algorithm. The covering algorithm extracts a rule set by learning one rule first, removing the input data covered by that rule, and iterating on the remainder of the data. Starting from an empty rule set, the sequential covering algorithm first looks for a rule that is highly accurate for predicting a certain class. If the accuracy of this rule is above a user-defined threshold, the rule is added to the set of already found rules, and the algorithm is repeated over the rest of the inputs that were not correctly classified by this rule. If the accuracy of the rule is below this threshold, the algorithm ends. Because the rules in the rule set can be overlapping, the rules are first sorted according to their accuracy on the training data before they are returned to the user. In Minerva, there are differences compared to the sequential covering algorithms above; the most important one is that the rules are required to be non-overlapping. Another difference is that other sequential covering algorithms stop if the performance of the rule is below a certain threshold.

For our technique, we look for unbounded support vectors which are the data points used as base locations to define separating hyperplane (Fig. 1). Then fuzzy if-then rules can be generated around support vectors based on firing strength to form our Fuzzy Rule Base (FRB) rules, e.g., IF x > c1 AND x < c2 THEN y = d. The rules are modified further by combining completely coincided ranges among the if-then conditionals using a fuzzy logic process called input scatter partitioning, and this set of rules is our final set needed.

III. Method

Data Source for Currency Crisis Study

1) International Financial Statistics CD-ROM (IFS), IMF
2) Direction of Trade Statistics Year Books and CD-ROM (DOTS), IMF
3) Balance of Payment Statistics Year Books (BOP), IMF
4) World Development Indicators CD-ROM (WDI), The World Bank
5) Global Development Finance CD-ROM (GDF), The World Bank
6) Exchange Arrangements and Exchange Restrictions Annual Report (EAER), IMF
7) Consolidated Banking Statistics (CBS), Bank for International Settlements (BIS)
8) Joint BIS-IMF-OECD-World Bank Statistics on External Debt (Joint-SED)
9) Die Falligkeitsverteilung der Internationalen Bankausleihung (FIB), BIS
10) Key indicators of developing Asian and Pacific countries, Asian Development Bank (ADB)

Countries selected for this study:

Argentina (Arg), Bolivia (Bol), Chile (Chi), Ecuador (Ecu), Mexico (Mex), Paraguay (Par), Peru (Per), Uruguay (Uru), Venezuela (Ven), India (India), Indonesia (Indo), Korea (Kor), Malaysia (Mal), Pakistan (Pak), Philippines (Phi), Singapore(Sin), Sri Lanka (Slk), and Thailand (Thai)

Data Preparation

This study follows the empirical implementation proposed by Esquivel and Larrain in 1998 [6]. The crises are required to be at least five months apart, i.e., a country could have a maximum of two crises per year. Most of the explanatory variables entered in lagged form due to the purpose of the study to interpret the empirical results as the one-period-ahead probability of a currency crisis. The explanatory variables can be classified into two types. Firstly, the stock variables, whose units are measured at one point in time and are observed every month. The variables in this first group are OVERRER, M2_FR, LIBDUM*FORI, DUMST_FR, and EXD_GNI. Secondly, the flow variables, whose units are measured per unit of time, are observed, in this case, on yearly basis. The variables in this group are CA_GDP, CREGROW, GDPGROW, LB, KAREVS, and CONT. All variables are entered in lagged form for one period except CONT which is contemporaneous. When a crisis occurs late in year t and one tries to use the explanatory variables of year t-1 to explain this crisis, sometimes many of explanatory variables change abruptly in the months before the collapse, and there was real evidence from changes in some of the explanatory variables, e.g. RER and foreign reserves, only a few months before the collapse. With respect to the stock variables if the crisis occurs late in year t (period “b” of year t), we should assume that the crisis occurred early in year t+1, and instead of taking the year-end value as usual, we take the mid-year value of year t to explain the crisis occurring in this particular period. This adjustment is made only for the stock explanatory variables. Esquivel and Larrain suggest that we should consider the characteristics of the flow variable, which indicates the change of a variable during one period in time, e.g., changing of the CA_GDP in year t. For these flow variables the year end value of year t-1 is to explain the crisis occurs in year t, even if it takes place in period “b” of year t. This adjustment should provide more accuracy for the study, which attempts to estimate the impact of the explanatory variables on the one-period-ahead probability of crisis on yearly basis.

SVM Classification

We are given an input of Q data points $\{(X_i, d_i)\}$, $i = 1,\ldots,Q$ with input data $x_i \in \Re^n$ and binary class labels $d_i \in \{-1,+1\}$, the SVM classifier satisfies the following conditions:

$$W.\Phi(X_i) + w_0 \geq +1 - \xi_i, \quad d_i = +1 \tag{1}$$

$$W.\Phi(X_i) + w_0 \leq -1 + \xi_i, \quad d_i = -1 \tag{2}$$

where W is weight vector, and the w_0 is a bias constant value; the two values are obtained from training the SVM. The function $\Phi(\bullet)$ is a non-linear function which maps the low dimensional input space into high dimensional space. The $d_i = +1$ means the output is the class we want to identify, and the $d_i = -1$ means the output is the other class. The ξ_i is a slack variable to allow misclassification. The separating hyperplane, which is the dividing line between the two classes, is represented by an equation:

$$W.\Phi(X_i) + w_0 = 0 \tag{3}$$

The margin between the two classes (Fig. 1) can be maximized by minimizing:

$$\frac{1}{2}\|W\|^2 + C\sum_{i=1}^{Q}\xi_i \tag{4}$$

subject to

$$d_i\left[W.\Phi(X_i) + w_0\right] - 1 + \xi_i \geq 0, \quad i = 1,\ldots,Q \tag{5}$$

$$\xi_i \geq 0 \tag{6}$$

Separating hyperplane $(W.\Phi(X_i) + w_0 = 0)$, which is used in main classification decision and formed from two boundaries - class 1 boundary $(W.\Phi(X_i) + w_0 = +1)$ and class 2 boundary $(W.\Phi(X_i) + w_0 = -1)$ is shown in Fig. 1. Class 1 boundary is formed from unbounded support vectors of class 1 (represented as circles in Fig. 1), and class 2 boundary is formed from unbounded support vectors of class 2 (represented as squares in Fig. 1). Margin is a distance between the two boundaries which is equal to $\frac{2}{\|W\|}$. Bounded support vectors are support vectors which are not on the class boundary but are closer to the separating hyperplane. Misclassification vector of class 1 $(W.\Phi(X_i) + w_0 \geq +1 - \xi_i)$ is a vector which is considered to be class 1 even though it is located at a distance $1 - \xi_i$ (or less) beyond separating plane into class 2 hyperspace. Misclassification vector of class 2 $(W.\Phi(X_i) + w_0 \leq -1 + \xi_i)$ is a vector which is considered to be class 2 even though it is located at a distance $1 - \xi_i$ (or less) beyond separating plane into class 1 hyperspace.

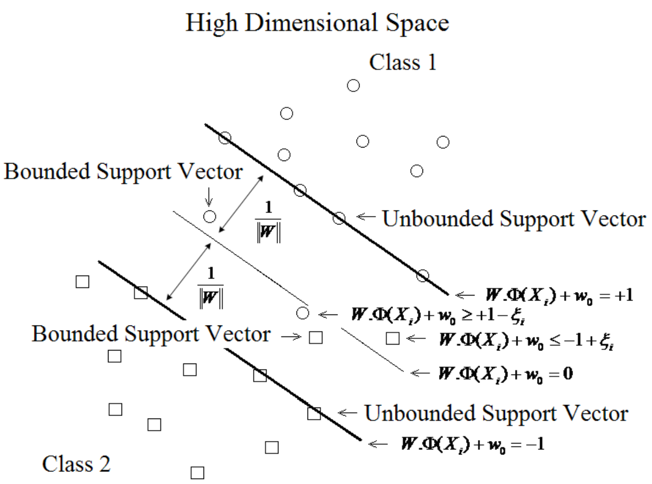

Fig. 1. Bounded and unbounded support vectors, misclassification vectors, and separating hyperplane

The part involving $\|W\|^2$ in the function maximizes the margin between the two classes in the feature space while the part involving C and ξ_i minimizes the misclassification error. The positive real constant C is a penalty parameter for misclassification. The Lagrangian with primal variables to the constraint optimization problem is given by

$$L_P(W, w_0, \Lambda, \xi, \Gamma) = \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{Q}\xi_i + \sum_{i=1}^{Q}\lambda_i\left[d_i\left[W.\Phi(X_i) + w_0\right] - 1 + \xi_i\right] - \sum_{i=1}^{Q}\gamma_i\xi_i \tag{7}$$

where $\Lambda = (\lambda_1,\ldots,\lambda_Q)^T$, $\lambda_i \geq 0$, $\Gamma = (\gamma_1,\ldots,\gamma_Q)^T$, $\gamma_i \geq 0$ are the Lagrange multiplier vectors. The solution to the optimization problem is given by the saddle point of the Lagrangian where all partial derivatives with respect to W, w_0, and ξ_i go to zero. The Karush-Kuhn-Tucker complementary conditions,

$$\lambda_i\left[d_i(W.\Phi(X_i) + w_0) - 1 + \xi_i\right] = 0, \quad i = 1,\ldots,Q \tag{8}$$

must also be satisfied.

This gives dual form of (7):

$$L_D(\Lambda) = \sum_{i=1}^{Q}\lambda_i - \frac{1}{2}\sum_{i=1}^{Q}\sum_{j=1}^{Q}\lambda_i\lambda_j d_i d_j(\Phi(X_i).\Phi(X_j)) \tag{9}$$

where $(\Phi(X_i).\Phi(X_j)) = K(X_i, X_j)$ is called a kernel function. The kernel function must satisfy Mercer's Condition which is an existence of a mapping $\Phi(X)$ and an expansion of a symmetric kernel function,

$$K(X_i, X_j) = \sum_k(\Phi_k(X_i).\Phi_k(X_j)) \tag{10}$$

iff

$$\iint K(X_i, X_j)g(X_i)g(X_j)dX_i dX_j \geq 0 \tag{11}$$

For all g(X) such that

$$\int g^2(X)dX < \infty \tag{12}$$

There are a few kernel functions which satisfy Mercer's Condition. In this study, we use Gaussian kernel because it will make the creation of equivalent fuzzy rule-based system possible, and it has been proved to satisfy Mercer's Condition.

To get support vectors, we need to maximize (9) subject to:

$$\sum_{i=1}^{Q}\lambda_i d_i = 0 \tag{13}$$

$$0 \leq \lambda_i \leq C; i = 1,\ldots,Q \tag{14}$$

Any input vector with non-zero Lagrange multiplier is a support vector.

There are two kinds of support vectors – bounded and unbounded (Fig. 1). The unbounded support vectors are the ones used for defining the separating hyperplane. These unbounded support vectors guarantee maximal margin between the two classes; in terms of calculations, they have Lagrange multipliers greater than zero but less than the penalty parameter $C(\lambda_i > 0; \lambda_i < C)$. The bounded support vectors are the ones closer to the separating hyperplane than the unbounded support vectors, so these vectors geometrically bound the two classes; they have Lagrange multipliers equal to the penalty parameter $C(\lambda_i > 0; \lambda_i = C)$.

One last parameter we need before reaching our final classifier equation is:

$$w_0 = \frac{1}{m}\left[\sum_{k=1}^{m}\left(\frac{1}{d_k} - W.X_k\right)\right] \tag{15}$$

where m = number of unbounded support vectors and

$$W = \sum_{k=1}^{N} \lambda_k d_k X_k \tag{16}$$

where N = number of all support vectors

We can now get the final classifier:

$$y(X) = sign\left(\sum_{i=1}^{N} d_i \lambda_i K(X_s.X_i) + w_0\right) \tag{17}$$

where X_s = unbounded support vector

This final equation is used in the SVM detection of currency crises. The unbounded support vectors obtained earlier are used in the rule generation step.

Rule Extraction

A. Rules Generation Based on Firing Strength

The purpose of this step is to generate preliminary rules based on the strongest firing signals associated with unbounded support vectors in high dimensional space.

In Fig. 2, all input patterns are entered into system one at a time. Gaussian kernel function as a membership function is calculated between current input and each of the support vectors, and the highest value is considered the strongest signal which will be the only one fired, and the rest will be ignored. The fired row then stores cumulative min and max value which will be replaced by new min or new max if it occurs. After all input patterns have been entered, min and max values in each row will be used as a range in conditional of each if-then rule.

Schematic diagram for implementing rule generation from unbounded support vectors found from previous step is shown in Fig. 2; X_i is input vector, and A_i is Gaussian kernel function of unbounded support vector and input vector.

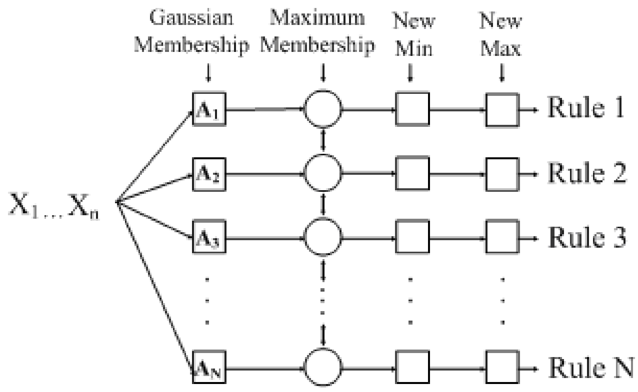

Fig. 2. Rules generation algorithm based on kernel firing strength

The final min and max values of each row are used as a range of the newly generated if-then rules. The if-then statements are in the form:

Rule 1: If ($x_{11} > a_{11-}$ AND $x_{11} < a_{11+}$) AND ($x_{12} > a_{12-}$ AND $x_{12} < a_{11+}$) AND … ($x_{1n} > a_{1n-}$ AND $x_{1n} < a_{1n+}$) THEN $y = v_1$
Rule 2: If ($x_{21} > a_{21-}$ AND $x_{21} < a_{21+}$) AND ($x_{22} > a_{22-}$ AND $x_{22} < a_{21+}$) AND … ($x_{2n} > a_{2n-}$ AND $x_{2n} < a_{2n+}$) THEN $y = v_2$
Rule 3: If ($x_{31} > a_{31-}$ AND $x_{31} < a_{31+}$) AND ($x_{32} > a_{32-}$ AND $x_{32} < a_{31+}$) AND … ($x_{3n} > a_{3n-}$ AND $x_{3n} < a_{3n+}$) THEN $y = v_3$
.
.
.
Rule N: If ($x_{N1} > a_{N1-}$ AND $x_{N1} < a_{N1+}$) AND ($x_{N2} > a_{N2-}$ AND $x_{N2} < a_{N2+}$) AND … ($x_{Nn} > a_{Nn-}$ AND $x_{Nn} < a_{Nn+}$) THEN $y = v_N$

where a_{ij-} are lower range values (cumulative min) and a_{ij+} are upper range values (cumulative max) in $\Re$. N is the total number of unbounded support vectors, and n is the dimension of input vectors.

B. Input Scatter Partitioning

The purpose of this step is to reduce generated rules and refine rule extraction in low dimensional space. We can combine many if-then statements from previous step together as long as it does not cause misclassification. Algorithm's pseudo code for input scatter partitioning is:

```
[Pre-loop condition: IF-THEN rules equal to total number of support vectors]
FOR i = 1 TO N
[N = total number of generated rules]
    IF rule i was eliminated THEN NEXT i
        DO WHILE (no class overlap from another class) or (maximum value or minimum value of the input data set reached)
            Expand ranges of IF-THEN conditional at i by a small value (less than 10% of min value of an attribute)
            IF there is class overlap GOTO END WHILE
            END IF
        END WHILE
    END IF
```

NEXT i
FOR i = 1 TO N
    DO WHILE (there are still rules to merge for this i)
        IF two ranges coincide then merge the two rules by retaining the larger ranges
        END IF
    END WHILE
NEXT i
[Post-loop condition: Number of IF-THEN rules are the same or less than rules in pre-condition]

We can use set membership symbol in place of greater than and less than signs as our final form of rules.

IF $x_{11} \in [a_{11-}, a_{11+}]$ AND $x_{12} \in [a_{12-}, a_{12+}]$ AND … $x_{1n} \in [a_{1n-}, a_{1n+}]$ THEN $y = v_1$

IV. Results

We perform classification using both SVM and our fuzzy if-then rules on Latin America compared to Asia data sets. Ten-fold cross validation technique is used in each data set to evaluate effectiveness of the classification.

Results from each data set are average number of unsigned support vectors (USV), average percent error from SVM, average number of rules from our method, and average percent error from our method, and the classification results are shown in Table I. SVM performs classification with less error in Asia data set than our method, but our method performs better in Latin America data set.

TABLE I
COMPARISON OF ERRORS FROM SVM AND SVSE RULES

Data	SVM		SVSE Rules	
	Avg USV	%Error	Avg Rules	%Error
Latin	33.90	29.47	33.90	28.02
Asia	14.70	10.63	14.70	15.46

The main reason why percent errors in SVM are different from our method is because of the misclassification vectors in the input data. SVM makes use of USV from both class 1 and class 2 to make classification decisions, but our method uses only USV from class 1. If there is more misclassification vectors in class 1 region in hyperspace, our method performs worse than SVM. But if there is more misclassification vectors in class 2 region in hyperspace, SVM performs worse than our method.

The final if-then rules obtained by our rule extraction method for Latin America entire data set, and the final if-then rules for Asia entire data set are shown in appendix.

V. Conclusion

The results of our empirical study have shown that the SVSE method can outperform SVM decisions in the Latin America data set, but under perform in the Asia data set. Percent errors of the two methods are not significantly different. It can be concluded that the results of the errors from the two methods are comparable, just like the results from 5 benchmark data sets in our previously published paper [16].

The SVSE method is shown to be a good method for rule extraction from SVM in currency crisis application and has an advantage over the decision method of SVM by revealing reasons behind the decision. And this makes it more attractive to be used in classification or prediction whenever we want to have insight into the way classification decision is made. The if-then rules obtained can be easily incorporated into a computer program using any popular computer language.

A suggestion for future study is to use data sets from other applications in order to obtain useful human comprehensible rules extracted from the pattern classification by SVM in different expert domains.

Appendix

The final form of fuzzy if-then rules is in the form:

IF $x_1 \in [a_{1-}, a_{1+}]$ AND $x_2 \in [a_{2-}, a_{2+}]$ AND … $x_{11} \in [a_{11-}, a_{11+}]$ THEN y = crisis occurs

Where x_i is one of the 11 parameters from our data set; a_{i-} is the lower range of each parameter, and a_{i+} is the upper range of each parameter.

In order to save space, we present the rules in the format:

Rule no.	$[a_{1-}, a_{1+}]$	$[a_{2-}, a_{2+}]$	$[a_{3-}, a_{3+}]$
	$[a_{4-}, a_{4+}]$	$[a_{5-}, a_{5+}]$	$[a_{6-}, a_{6+}]$
	$[a_{7-}, a_{7+}]$	$[a_{8-}, a_{8+}]$	$[a_{9-}, a_{9+}]$
	$[a_{10-}, a_{10+}]$	$[a_{11-}, a_{11+}]$	

Final fuzzy rules for currency crises for Latin America data set:

1.	[0.06, 1.06],	[4.16, 10.66],	[0.00, 12.60],
	[1, 1],	[0.01, 0.41],	[-4.08, -0.18],
	[41, 171],	[-1.85, 11.15],	[-110, 226],
	[443, 3823],	[1, 1]	
2.	[-0.61, 1.09],	[2.88, 11.38],	[0.00, 14.40],
	[1, 1],	[-0.06, 0.44],	[-5.18, -0.08],
	[106, 276],	[-12.69, 3.31],	[-89, 247],
	[-1130, 3290],	[1, 1]	
3.	[-0.61, 0.49],	[1.87, 8.87],	[0.00, 12.60],
	[1, 1],	[-0.06, 0.44],	[-4.89, -0.69],
	[150, 290],	[-10.96, 2.04],	[-85, 251],
	[1767, 5147],	[1, 1]	
4.	[-0.60, 0.58],	[4.43, 10.33],	[0.00, 9.00],
	[1, 1],	[-0.07, 0.50],	[-3.33, 3.89],
	[10, 631],	[-1.79, 8.61],	[-84, 207],
	[-4396, -1348],	[1, 1]	
5.	[-0.68, 1.02],	[5.37, 15.37],	[0.00, 14.40],
	[0, 1],	[-0.05, 0.45],	[-6.89, -0.89],
	[114, 294],	[-6.09, 11.91],	[-94, 242],

[-2030, 2650], [1, 1]
6. [-0.60, 1.10], [0.86, 56.36], [0.00, 14.40],
[0, 1], [-0.06, 0.44], [-14.60, 6.70],
[379, 7229], [-12.50, 11.50], [-126, 210],
[-15208, 26392], [0, 1]
7. [-0.60, 1.10], [1.09, 18.59], [0.00, 14.40],
[0, 1], [-0.06, 0.44], [-6.08, 6.82],
[378, 988], [-12.40, 11.60], [-120, 216],
[-9999, 5861], [0, 1]
8. [-0.65, 1.05], [1.01, 28.51], [0.00, 14.40],
[0, 1], [0.00, 0.50], [-14.68, 6.62],
[-31, 459], [-12.41, 11.59], [-101, 235],
[9126, 26286], [0, 1]
9. [-0.65, 1.05], [7.02, 17.02], [0.00, 14.40],
[1, 1], [-0.06, 0.44], [-10.36, -4.96],
[-33, 117], [-8.72, 9.28], [-83, 253],
[-2751, 1929], [1, 1]
10. [-0.30, 0.64], [1.28, 13.46], [0.00, 7.20],
[1, 1], [-0.03, 0.45], [-3.99, -0.04],
[131, 398], [-7.04, 1.44], [-97, 198],
[-1049, 1545], [1, 1]
11. [-0.65, 1.05], [0.83, 33.33], [0.00, 14.40],
[0, 1], [-0.01, 0.49], [-14.68, 6.62],
[378, 1628], [-12.20, 11.80], [-81, 255],
[-15143, 15797], [0, 1]
12. [-0.61, 1.09], [0.80, 56.30], [0.00, 14.40],
[0, 1], [-0.02, 0.48], [-14.59, 6.71],
[377, 5007], [-12.68, 12.32], [-107, 229],
[-15285, 26315], [0, 1]
13. [-0.65, 1.05], [1.01, 9.51], [0.00, 14.40],
[0, 1], [-0.05, 0.45], [-14.50, -10.90],
[-31, 149], [-7.26, 11.74], [-128, 208],
[-4647, 1593], [0, 1]
14. [-0.61, 1.09], [0.97, 8.97], [0.00, 14.40],
[0, 1], [0.00, 0.40], [-14.28, -7.68],
[-30, 160], [-3.03, 11.97], [-116, 220],
[-5744, -544], [0, 1]
15. [-0.13, 0.37], [3.87, 10.56], [3.88, 15.73],
[1, 1], [0.15, 0.49], [-8.84, 1.08],
[-5, 50], [-1.57, 5.36], [-44, 132],
[-545, 783], [1, 1]
16. [-0.69, 0.51], [1.12, 7.62], [0.00, 14.40],
[1, 1], [-0.03, 0.47], [-13.38, -9.18],
[-24, 56], [-9.15, 4.85], [-124, 212],
[-2644, 996], [1, 1]
17. [-0.66, 0.14], [2.44, 6.44], [1.54, 14.14],
[1, 1], [0.01, 0.41], [-7.69, -5.59],
[-28, 32], [5.37, 11.37], [-114, 126],
[-357, 1463], [1, 1]
18. [-0.66, 0.94], [0.70, 7.20], [-0.11, 14.29],
[1, 1], [0.02, 0.42], [-11.12, -6.92],
[-32, 88], [-4.89, 9.11], [-82, 254],
[-3282, 358], [0, 0]
19. [-0.67, 0.83], [0.72, 7.22], [-0.95, 15.25],
[0, 1], [0.01, 0.41], [2.51, 6.71],
[-31, 89], [-12.30, 3.70], [-93, 243],
[451, 5131], [0, 0]
20. [-0.66, 1.04], [5.25, 34.25], [-0.46, 15.74],
[0, 1], [-0.02, 0.48], [-14.59, 2.21],
[-27, 343], [-12.23, 11.77], [-89, 247],
[-15093, -7813], [0, 1]
21. [-0.68, 1.02], [3.26, 56.26], [0.00, 14.40],
[0, 1], [-0.01, 0.49], [-14.50, 6.80],
[-25, 685], [-12.63, 11.37], [-88, 248],
[9118, 26278], [0, 1]
22. [-0.68, 1.02], [0.69, 24.69], [0.00, 14.40],
[0, 1], [0.02, 0.42], [-14.55, 3.45],
[-27, 383], [-12.58, 11.42], [-100, 236],
[9133, 26293], [0, 1]
23. [-0.63, 1.07], [1.02, 41.02], [0.00, 14.40],
[0, 1], [0.02, 0.42], [-14.65, 6.65],
[-33, 687], [-12.17, 11.83], [-114, 222],
[9114, 26274], [0, 1]
24. [-0.45, 0.75], [0.77, 4.27], [0.00, 10.80],
[0, 0], [-0.06, 0.44], [-7.53, -4.53],
[-24, 66], [6.36, 11.36], [-82, 254],
[-1385, 1216], [0, 0]
25. [-0.35, 0.65], [0.76, 4.76], [-0.31, 15.89],
[0, 0], [0.03, 0.43], [-5.32, -2.32],
[-30, 70], [-7.02, 0.98], [-90, 198],
[-989, 1091], [1, 1]
26. [-0.28, 0.10], [0.91, 3.67], [0.00, 3.37],
[0, 0], [-0.06, 0.15], [-7.05, -3.18],
[6, 44], [-1.00, 7.35], [-28, 47],
[-238, 555], [1, 1]
27. [-0.69, 1.11], [3.86, 21.36], [0.00, 14.40],
[0, 1], [-0.03, 0.47], [-10.50, 0.00],
[367, 717], [-11.70, 9.30], [-106, 230],
[-4467, 4633], [0, 1]
28. [-0.69, 1.01], [0.77, 56.27], [0.00, 14.40],
[0, 1], [-0.04, 0.46], [-14.65, 6.65],
[383, 3393], [-11.70, 12.30], [-97, 239],
[-15320, 26280], [0, 1]
29. [-0.63, 1.07], [0.95, 55.95], [0.00, 14.40],
[0, 1], [-0.04, 0.46], [-14.80, 6.80],
[375, 7235], [-12.14, 11.86], [-99, 237],
[-15310, 26290], [0, 1]
30. [-0.64, 1.06], [21.90, 55.90], [-0.30, 14.10],
[0, 1], [-0.01, 0.49], [-14.53, 6.77],
[-31, 809], [-11.76, 12.24], [-125, 211],
[-15254, 17506], [0, 1]
31. [-0.32, 0.28], [1.74, 4.74], [-1.76, 7.24],
[0, 0], [-0.06, 0.44], [-3.53, -1.73],
[-22, 38], [-5.39, -1.39], [0, 192],
[-247, 793], [1, 1]
32. [-0.11, 0.45], [1.50, 7.73], [7.39, 11.26],
[1, 1], [0.02, 0.46], [1.28, 6.55],
[-3, 83], [-4.76, 2.10], [-83, 63],
[-48, 2448], [1, 1]
33. [-0.45, 0.13], [2.15, 5.01], [0.00, 3.60],
[1, 1], [0.06, 0.46], [-5.47, -3.25],
[-24, 42], [3.07, 7.59], [-70, 66],
[-1736, 338], [1, 1]
34. [-0.09, 0.57], [1.48, 5.53], [0.00, 3.60],
[1, 1], [-0.03, 0.50], [-10.22, -4.24],
[30, 166], [4.84, 9.00], [-97, 55],

	[120, 2400],	[1, 1]	
35.	[-0.13, 0.47],	[0.85, 3.85],	[0.00, 5.40],
	[0, 0],	[-0.05, 0.45],	[-4.22, -2.42],
	[-3, 57],	[-2.75, 3.25],	[-118, 122],
	[210, 1510],	[0, 0]	
36.	[0.02, 0.60],	[1.16, 2.35],	[0.00, 3.79],
	[0, 0],	[-0.05, 0.21],	[-4.65, 2.90],
	[17, 168],	[1.17, 4.97],	[-105, 63],
	[-3313, 20],	[1, 1]	

Final fuzzy rules for currency crises for Asia data set:

1.	[-0.30, 0.30],	[51.91, 88.41],	[0.00, 9.00],
	[0, 1],	[-0.09, 3.81],	[-8.47, 15.83],
	[-16, 53],	[-8.19, 8.81],	[-102, 137],
	[-3775, 20924],	[0, 1]	
2.	[-0.31, 0.29],	[0.56, 9.56],	[-1.00, 8.00],
	[1, 1],	[-0.05, 0.95],	[-8.34, -2.94],
	[-13, 46],	[-7.90, 9.10],	[-69, 122],
	[-3761, -1681],	[1, 1]	
3.	[-0.22, 0.28],	[0.92, 7.92],	[-1.66, 9.14],
	[1, 1],	[-0.02, 0.78],	[-6.63, -2.43],
	[-12, 47],	[-1.04, 8.96],	[-82, 157],
	[-3955 ,-835],	[1, 1]	
4.	[-0.27, 0.23],	[0.71, 13.21],	[0.00, 9.00],
	[0, 1],	[0.00, 1.50],	[-6.17, 1.63],
	[-11, 48],	[-8.30, 8.70],	[-74, 117],
	[8070, 14830],	[0, 1]	
5.	[-0.32, 0.14],	[2.95, 22.78],	[0.00, 1.80],
	[0, 0],	[0.01, 0.19],	[-2.40, 4.43],
	[10, 29],	[0.79, 6.50],	[-60, 81],
	[-3697, 995],	[1, 1]	
6.	[-0.29, 0.21],	[2.40, 11.40],	[0.00, 9.00],
	[1, 1],	[-0.06, 1.04],	[-8.57, -3.77],
	[-14, 55],	[-1.94, 9.06],	[-90, 149],
	[-4004, -1144],	[0, 0]	
7.	[-0.26, 0.24],	[0.71, 51.71],	[0.00, 9.00],
	[0, 1],	[0.00, 3.80],	[-8.48, 15.82],
	[-16, 53],	[-7.99, 9.01],	[-75, 116],
	[9719, 33119],	[0, 1]	
8.	[0.01, 0.23],	[3.69, 27.27],	[0.00, 1.80],
	[0, 0],	[0.01, 0.29],	[-5.93, -2.64],
	[8, 39],	[2.55, 7.32],	[5, 48],
	[131, 7279],	[1, 1]	
9.	[-0.25, 0.25],	[9.07, 20.07],	[0.00, 9.00],
	[0, 1],	[-0.04, 1.16],	[-8.34, -5.04],
	[-18, 51],	[-8.13, 8.87],	[-92, 147],
	[636, 6096],	[0, 1]	
10.	[-0.27, 0.23],	[0.93, 10.93],	[0.00, 9.00],
	[1, 1],	[0.00, 1.10],	[-2.82, 2.88],
	[-13, 46],	[-7.31, 2.69],	[-89, 150],
	[-1886, 3053],	[1, 1]	
11.	[-0.30, 0.29],	[8.92, 16.44],	[0.00, 7.20],
	[1, 1],	[-0.05, 0.48],	[-6.98, -3.84],
	[-12, 50],	[0.04, 8.75],	[-65, 150],
	[-1483, 439],	[1, 1]	
12.	[-0.21, 0.29],	[3.43, 7.43],	[0.00, 7.20],
	[0, 0],	[-0.07, 0.43],	[-5.80, -3.40],
	[-18, 51],	[0.39, 8.39],	[-82, 157],
	[-2633, -1073],	[0, 0]	
13.	[-0.29, 0.21],	[0.95, 9.95],	[0.00, 9.00],
	[0, 1],	[-0.04, 1.06],	[-8.28, -2.28],
	[-10, 49],	[-4.81, 9.19],	[-77, 114],
	[2439, 7379],	[1, 1]	
14.	[-0.27, 0.30],	[1.02, 5.09],	[-1.66, 9.72],
	[1, 1],	[2.27, 3.74],	[-2.12, 15.63],
	[-8, 49],	[-7.44, 8.51],	[-75, 118],
	[-938, 5272],	[1, 1]	
15.	[-0.32, 0.28],	[10.77, 20.77],	[0.00, 9.00],
	[1, 1],	[-0.08, 1.02],	[-8.62, -2.92],
	[-14, 55],	[-6.70, 9.30],	[-105, 134],
	[-2667, 2272],	[0, 0]	
16.	[-0.27, 0.23],	[0.86, 7.86],	[0.00, 9.00],
	[1, 1],	[-0.03, 1.07],	[-8.37, -5.97],
	[-16, 54],	[-1.10, 8.90],	[-105, 134],
	[602, 4242],	[0, 0]	

REFERENCES

[1] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-Based Systems*, vol. 8, no. 6, pp. 373–389, 1995.

[2] I. Arciniegas. SVM Sensitivity Analysis: An Application to Currency Crises Aftermaths. IEEE Transactions on Systems, Man, and Cybernetics – Part A: *Systems and Humans*, Vol. 34, No. 3, pp. 387-398, 2004.

[3] N. Barakat and J. Diederich, "Eclectic Rule-Extraction from Support Vector Machines", *International Journal of Computational Intelligence*, vol. 2, no. 1, pp. 59 – 62, 2005.

[4] C. Cortes and V. N. Vapnik, "Support Vector Networks, *Machine Learning*," vol. 20, pp. 273-297, 1995.

[5] J. Diederich (Ed.), "Rule Extraction from Support Vector Machines," *Studies in Computational Intelligence*, vol. 80, Springer-Verlag, Berlin, Heidelberg, pp. 3-30, 2008.

[6] G. Esquivel and F. Larrain, Explaining Currency Crisis, pp. 1-42, 1998. Available: http://www.cid.harvard.edu/hiid/646.pdf

[7] G. Fung, S. Sandilya, and R. B. Rao, "Rule extraction from linear support vector machines," In *Proceedings of the 11th ACM SIGKDD international Conference on Knowledge Discovery in Data Mining*, pp. 32–40, 2005.

[8] X. Hui and J. Sun. An Application of Support Vector Machine to Companies' Financial Distress Prediction, V. Torra et al. (Eds.): *MDAI 2006*, LNAI 3885, pp. 274 – 282, 2006.

[9] J. Huysmans, B. Baesens, and J. Vanthienen, "ITER: an algorithm for predictive regression rule extraction," in *8th International Conference on Data Warehousing and Knowledge Discovery*, Springer Verlag, lncs 4081, pp. 270-279, 2006.

[10] J. Huysmans, R. Setiono, B. Baesens, and J. Vanthienen, "Minerva: Sequential Covering for Rule Extraction," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 38, no. 2, pp. 299 – 309, 2008.

[11] E. Kolman and M. Margaliot, "Are artificial neural networks white boxes?" *IEEE Trans. Neural Networks*, vol. 16, no. 4, pp. 844-852, 2005.

[12] S. Kumar, *Neural Networks: A Classroom Approach*. McGraw-Hill, International Edition, 2005, pp. 273-304.

[13] C. Lin, H. Khan, Y. Wang, and R. Chang. A New Approach to Modeling Early Warning Systems for Currency Crises: can a machine-learning fuzzy expert system predict the currency crises effectively? *CIRJE-F-411*, 2006.

[14] H. Nunez, C. Angulo, and A. Catala, "Rule Extraction Based on Support and Prototype Vectors," in: *Studies in Computational Intelligence*, vol. 80, Springer-Verlag, Berlin, Heidelberg, pp. 109-134, 2008.

[15] T. Peltonen. Are Emerging Market Currency Crises Predictable? A Test. *European Central Bank Working Paper*, No. 571, 2006.

[16] P. Pitiranggon, N. Benjathepanun, S. Banditvilai, and V. Boonjing, "Fuzzy Rules Generation and Extraction from Support Vector Machine Based on Kernel Function Firing Signals," *International Journal of Engineering and Applied Science*, vol. 6, no. 4, pp. 244-251, 2010.
[17] S. Thearpiriyakij, J. Breitung. An Empirical Approach to Currency Crises: Latin America and Asia. *Diplomabeit, Rheinisch-Friedrich-Wilhelms-Universitat, Bonn*, Jun 2005.
[18] V. N. Vapnik, *Statistical Learning Theory*. John Wiley&Sons, 1998, pp. 375-520.