Development of Intelligent Hybrid Learning System Using Clustering And Knowledge-Based Neural Networks for Economic Forecasting : First Phase

Zamzarina Che Mat @ Mohd Shukor and Mohd Noor Md Sap

Information System Department
Faculty of Computer Science and Information System
University Technology of Malaysia, K.B. 791
81310 Skudai, Johor, Malaysia
Tel. 60-07-5532419, Fax. 60-07-5565044.
mc023013@siswa.utm.my, mohdnoor@fsksm.utm.my

ABSTRACT

The economic forecasting environment is currently undergoing drastic changes and has a complex and challenging task. Practically, people design a database application or use a statistical package to conduct the analysis on the data. Former approach can be done on the online data, but it must be developed after stating the goal of analysis, which means it only possible for a limited and specific purpose. Whereas the statistical approach must be done for the offline data, however it can lead to the missing pattern and undiscovered knowledge from the available data (Shan, C., 1998). For the effort to extract implicit, previously unknown, hidden and potentially useful information from raw data in an automatic fashion, leads us to the usage of data mining technique that receives big attention from the researchers recently. This paper proposed the issues of joint clustering and knowledge-based neural networks techniques as the application for point forecast decision-making. Future prediction (e.g., political condition, corporation factors, macro economy factors, and psychological factors of investors) perform an important rule in Stock Exchange, so in our prediction model we will be able to predict results more precisely. We proposed K-Means clustering algorithm that is based on multidimensional scaling, joined with neural knowledge based technique algorithm for supporting the learning module to generate interesting clusters that will generate interesting rules for extracting knowledge from stock exchange databases efficiently and accurately.

Keywords

Data Mining, Hybrid Learning, Clustering, Neural Networks, Knowledge Based

1.0 INTRODUCTION

Making predictions and building trading models are central goals for financial institutions as an investor or financial manager. The difficulty in forecasting time series such as economic or financial data is usually attributed to the limitation of many conventional forecasting models, but could encourage many researchers to develop the more predictable forecasting models. Artificial intelligence such as neural network and other learning algorithms techniques have been recognized as useful forecasting models than the conventional statistical forecasting models.

Data mining can be defined as process of searching patterns and relationships from large amount of data in databases using sophisticated data analysis tools and techniques to build models that may be used to make valid predictions. Jun and Keng (1999) defines data mining as searching for valuable information in large volumes of data. It is the essential process of extraction of implicit previously unknown and potentially useful information such as knowledge, rules, constraints and regularities from data stored in repositories using pattern recognition technologies. The result of the extracted knowledge can be applied in information management, decision process, query processing and many other application (Yongjian,F., 1996).

Existing clustering algorithms for sequence and structure datasets operate on the object's similarity space. Algorithm such as feature-based and similarity-based clustering algorithm are quite limiting as they cannot scale to very large datasets, cannot be used to provide a description as to why a set of objects was assigned to the same cluster that is native to the object's features and cannot be used to find clusters that have conserved features. Furthermore it cannot provide a description as to why the objects assigned to the same cluster that is native to the object's feature. The only way to overcome this shortcoming is to :

i. Develop scalable and computationally algorithms for large sequence and structure datasets that operates directly on the object's native representation.

ii. Develop clustering algorithms that can provide concise explanation on the characteristics of the objects that were assigned to each cluster.

In previous research, a multitude of promising forecasting methods for predicting stock price from numeric data have been developed. These methods include statistics, ARIMA (Auto Regression Integrated Moving Average), Box-Jenkins, stochastic and neural networks (Clarence N.W, T., 1993; Kazuhiro, K., 1995; Wuthrich, B., et. al, 1998; Alexandra, I.C., and Toshio, O., 1998; Sheng-Chai, C., 1999). Duffie (Chengyi, S., et. al, 1996) built stochastic models of stock market based on stochastic differential equation, but they have some disadvantages as follow:

i. there is a function used in the models that represent the influence on stock prices of various factors including corporation factors, macro economy factors, political factors and psychological factors of investors. The function is very difficult to decide or cannot be decided at all;

ii. the models cannot be used for prediction.

Our study focuses on the new integration model of clustering method and neural network knowledge based technique to gain more meaningful time series forecasting data for the efficient and effective learning.

The next section proposed the general architecture of the hybrid system. The third section and fourth section describe clusters mining and Neural Network Knowledge Based respectively. The final section discuss conclusion.

2.0 ECONOMIC FORECASTING GENERAL ARCHITECTURE

The general architecture of our hybrid learning system is represented in figure 1. The main process architecture is clusters generation that consists of two steps; training and running. The training step is conducted for generating the neural network knowledge based of clustering, i.e. the basic structure of neural networks knowledge based while the running step is used for creating the target output, i.e. the action part, of interesting clusters. In running steps, neural network knowledge based is used for supporting the module to generate learned complete data and frequent clusters that will help to generate interesting rules.

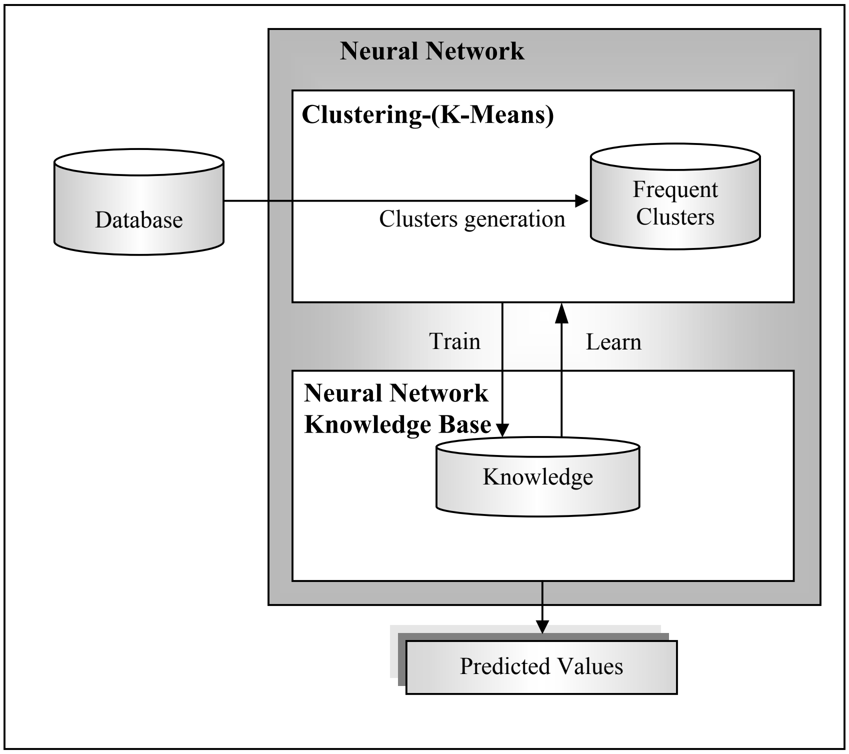

Figure 1 : The General Architecture of Hybrid Learning System for Economic Forecasting

The clusters generation follows these steps :

i. generate clusters of item sets from stock exchange database. The item sets are combined into one cluster when the support and interestingness of pair item sets are greater than or equal to the minimum support and interestingness,

ii. evaluate the generated clusters to identify whether the generated clusters interesting or not, and

iii. generate rules from each cluster

Every rule generated by the clustering algorithm is then stalled by neural knowledge based in knowledge databases as set of knowledge and finally prediction is made from the set of rules in the knowledge database.

3.0 MINING CLUSTERS USING MULTIDIMENSIONAL SCALING

Clustering is an important and fundamental area in data mining. In data mining, efforts have focused on finding methods for efficient and effective cluster analysis in databases.

Different types of data were used for data representation and data analysis in databases. Examples of data types attributes used in data analysis are interval-scaled, ratio-scaled, ordinal, binary and nominal variables. In the clustering algorithm, we assume that objects in databases are represented by the same set of attributes from stock exchange databases :

$$A_1, A_2, \ldots, A_p \qquad (1)$$

where p is the number of attributes. Each attribute, $\boldsymbol{A_i}$, describes a domain of values denoted by $\boldsymbol{DOM(A_i)}$. An object $\boldsymbol{X}$ is represented as a conjunction of attribute-value pairs:

$$[A_1 = x_1] \wedge \ldots \wedge [A_p = x_p] \qquad (2)$$

where

$$x_i \in DOM(A_i) \text{ for } 1 \le i \le \boldsymbol{p}. \qquad (3)$$

We represent ***X*** as a vector :

$$[x_1, x_2, \ldots, x_p] \qquad (4)$$

3.1 The degree of similarity

The distance measure between two categorical objects has been defined by the total mismatches of the attribute categories of the two objects. The dissimilarity of two attribute values is 0 or 1 based on whether the attribute values is match. The degree of similarity between attribute values is define as follows :

Definition 1 : The degree of similarity between two attribute values is the size of the analogy reflecting the implicit relationships of the two attribute values. Let

$$v_i \in DOM(A_k),\ v_j \in DOM(A_k), \qquad (5)$$

where $k = 1, \ldots, p$ be the two attribute values. The degree of similarity $\boldsymbol{s_{ij}}$ is defined as:

$$s_{ij} = Sim\ (v_i, v_j) \qquad (6)$$

Sim (v_i, v_j) is an arbitrary function representing the size of analogy of v_i and v_j.

There are a few properties for the degree of similarity. Usually, it is normalized so it is in interval [0, 1], thus the range is $\mathbf{0 \le s_{ij} \le 1}$. The self similarity is scaled to be unity**,**

$$\mathbf{s_{ii} = 1} \text{ and } \mathbf{s_{ij} = s_{ji}}. \qquad (7)$$

Here, a distance means the discrepancy between two size of unlikeness between the two attribute values in an attribute and the dissimilarity means the size of unlikeness between the two attribute values in an attribute. If the number of values in an attribute $|\boldsymbol{DOM(A_i)}|$ is m, the similarity matrix is m x m symmetrical matrix of inter-values. Similarity can be converted into dissimilarity without losing the generality with simple transformation :

$$Dissimilarity\ (\delta_{ij}) = 1 - similarity\ (s_{ij}) \quad (8)$$

3.2 The Clustering Algorithm – Modified k-means

The center of C_i for a data set can be represented by a midpoint and a mean of attributes.

$$m_i = \left[\left(\frac{\sum_{Xn \in Ci} X_n (x_{11})}{|C_i|}, \frac{\sum_{Xn \in Ci} X_n (x_{12})}{|C_i|} \right), \ldots, \left(\frac{\sum_{Xn \in Ci} X_n (x_{p1})}{|C_i|}, \frac{\sum_{Xn \in Ci} X_n (x_{p2})}{|C_i|} \right) \right]$$

$$= [m_{i1}, \ldots, m_{ip}] \qquad (9)$$

From the definition of the distance between two objects, we can compute the distance between objects and the center of cluster Ci for a data set.

$$|X_n - m_i| = \left[\gamma \sum_{j=1}^{P} \sum_{l=1}^{2} (X_n(x_{jl}) - m_{ijl})^2 \right]^{\frac{1}{2}} \quad (10)$$

where γ is weight. Next, the error criterion function is computed as :

$$E = \sum_{i=1}^{K} \sum_{X_n \in C_i} |\ X_n - m_i|^2 \qquad (11)$$

By using the center, distance, and the error formulate above, we can implement the k- means algorithm as below:

$X_n \in C_i$

K

$i=1 \quad X_n \in C_i$

4.0 NEURAL NETWORK KNOWLEDGE BASED

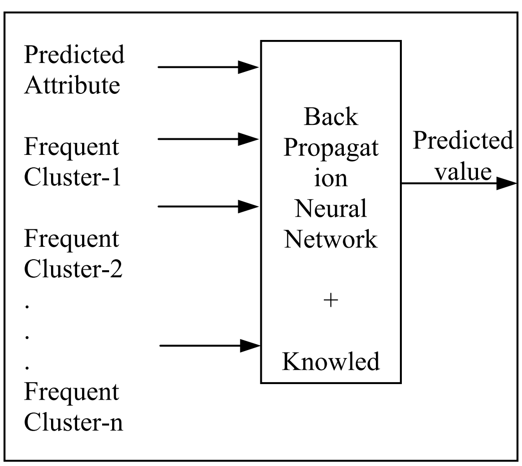

Figure 2 : The Neural Network Knowledge Based Prediction Model

In this module, two intelligent techniques, i.e. neural network and knowledge based are combined. It is used in both training and running steps. In the training steps it used for creating the neural network knowledge based of clustering. In the running step, it is used for supporting the module in order to generate learned complete cluster and interesting rules.

The basic steps of these approach are (i) compile/encode the available theoretical knowledge into an adequate Artificial Neural Network (ANN), (ii) use sets examples of data from stock exchange database that had been clustered to train the network, hence introduce additional knowledge into ANN, (iii) extract the refined theory under symbolic form; to be reinserted into the ANN, the cycle being repeated, until some stopping criteria are satisfied.

The simplest way to implement discrete rules is in the form of binary logical functions. An n-variable binary logical function

$$f: \mathbf{B}^n \rightarrow \mathbf{B},\ n \in \mathbf{N},\ \mathbf{B} = \{F,T\};\ y = f(x_1, x_2,) \quad (12)$$

attaches one of the truth values F-false or T-true to its output (dependent, consequent) variable y, for each combination of the truth values of the input (independent, antecedent) variables $x_1, x_2, \ldots, x_n$.Usually, numeric coding is used for the binary logical values true (T) and false (F): 0 and 1 (unipolar representation), or –1 and 1 (bipolar representation), respectively.

There are two main approaches to knowledge extraction: (1) decompositional (structural analysis) methods, which assign each unit of ANN a prepositional variable, and establish the logical links between these variables; (2) pedagogical (input-output mapping) methods, which treat the network as a black-box, without analyzing its internal structure. In our hybrid learning system we focused on the decompositional approach which puts in correspondence the ANN sub-symbolic structure.

This approach shed light into the neural network black box by combining symbolic, rule-based reasoning with neural learning. Its form a three-link chain in which symbolic knowledge, in the form of propositional rules, is revised and corrected using neural networks. Thus, the approach makes possible the use of neural networks as the underlying algorithm of a rule-refinement system.

The first link of the chain is to insert knowledge, which need be neither complete nor correct, into a neural network. This step changes the representation of the rules from symbolic to neurally-based, thereby allowing refinements to be made at a finer grain. Networks created using this algorithm will be referred as Knowledge-based Neural Networks (KNNs).

The second link of the chain is to train the KNN using a set of classified training examples and standard neural learning methods. In so doing, the rules upon which the KNN are based are corrected so that they are consistent with the training examples.

The final link is to extract rules from trained KNNs. This is an extremely difficult task for arbitrarily-configured networks, but it is somewhat less daunting for KNNs due to their initial comprehensibility. This method efficiently extract rules from the networks. Significantly, when evaluated in terms of the ability to correctly classify examples not seen during training, the method produces sets of rules that are superior to the networks from which they came. Moreover, the extracted rules are superior to the rules resulting from rule-refinement methods that act directly on the rules rather than their re-representation as a neural network.

4.1 The Knowledge Based Aftificial Neural Network Algorithm

The neural networks we use in this hybrid learning system is based on "feedforward" neural network (Towell G.G and Shavlik J.W., 1993). The equation below defined a *logistic activation function* :

$$NetInput_i = \sum_{j \in \{\text{Connected\_Units}\}} Weight_{ji} * Activation_j \quad (13)$$

$$Activation_i = \frac{1}{1 + e^{-(\text{NetInput}_i - \text{Bias}_i)}} \quad (14)$$

This method consist two largely independent algorithms : (i) a *rules- to- network translator* and (ii) a *refiner* (i.e. a neural learning algorithm).

Briefly the rules-to-networks translation is accomplished by establishing a mapping between a rule set and a neural network. The translation specifies the features that are probably relevant to making a correct decision. The second major step is to refine the network using standard neural learning algorithms and a set of classified clusters training examples. At the completion of this step, the trained network can be used as a classifier that is likely to be more accurate than those derived by other machine learning methods. . Below is rules-to-network algorithm that proposed by Towell G.G and Shavlik J.W. (1993):

.

Rules-to- network algorithm :

1. *Rewriting* – transform the set of rules into a format that clarifies its hierarchical structure

    If (there is > 1 rule for
    consequent)
    Then
    every rule for this consequent
    with > 1 antecedent is
    rewritten as 2 rules.

2. *Mapping* – establishes a mapping between a transformed set of rules and neural network.
    Creates network that have 1- to- 1 correspondence with elements of the rule set.
    (weights specified by the rule set, and the biases corresponding to consequents are set.)

3. *Numbering* – numbers units in the nets by their 'level'
    (This number is a necessary precursor to several of the following steps.)

4. *Adding hidden units* – adds hidden unit to the nets.
    Giving nets the ability to learn derived features not specified in the initial rule set.
    (this step is optional)

5. *Adding input units* – augment the nets with input feature (cluster rule sets) which a domain expert believe are relevant.

6. *Adding links* – Adds links with weight = 0 to the network using the numbering of units established in step 4.

    Links are added to connect each unit numbered $n - 1$ to each unit numbered n.

7. *Perturbing* – perturb all the weights in the network by adding a small random number to each weight.

This method use *cross-entropy error* function that suggested by Hinton (Towell G.G and Shavlik J.W., 1993). The *cross-entropy* function interprets the training signal and networks outputs as conditional probabilities. It also attempts to minimize the difference between these probabilities.

$$Error = -\sum[(1 - d_i) * log_2(1-\alpha_i) + (d_i * log_2(\alpha_i))] \quad (15)$$

where α_i is the activation of output unit i, d_i is the desired activation for unit i, and n is the number of output units.

$$regulizer = \lambda \sum_{i \in \omega} \frac{(\omega i - \omega_{initi})^2}{1 + (\omega_i - \omega_{initi})^2} \quad (16)$$

The λ term in above equation controls the tradeoff between the ability of the network to learn the training set and distance from the initial rules. The principal effect of the addition of this regularization term is to increase the interpretability of trained nets by encouraging networks to leave their original weights unchanged.

4.2 Rule Extraction (M-of-N Algorithm)

In real world problems, it is common to have sets of variables playing equivalent roles in the analyzed system, so that there is an inherent symmetry of the rules to be extracted. This symmetry can greatly reduce the complexity of the extraction procedure. Moreover, imposing from start the symmetry restrictions predicted by the domain theory, simpler ANNs and learning algorithms result. The *M-of-N* method is based on the idealised relation $\boldsymbol{v_{i\ border} = (2M - N)w + b_i}$, where the bias is chosen $\boldsymbol{b_i = (N - 2\ M_{border} + 1)\ w}$, so that:

$$v_{i_{border}} = \left[2\left(M - M_{border}\right) + 1\right] w = \begin{cases} > 0, & \text{when } M > M_{border} \\ < 0, & \text{when } M < M_{border} \end{cases} \quad (17)$$

The steps of the **M-of-N** algorithm are:

(M-of-N Algorithm)

1. For each output and hidden neuron *i*

   Form clusters of similarly weighted links

   Replace the individual weights of all the links in a group with the average value for the group.

   (This step is not necessary if symmetry is imposed from the start of the training, using a modified Back Propagation).

2. Eliminate insignificant groups (that can not change the *true/false* output state).

3. Keeping the weights constant, use Back Propagation to readjust the biases

4. For each neuron considered at (1)

   Construct a single rule of the form

   If *(at-least-Mborder-of-N antecedents are satisfied)*
   then *(statement attached to unit i)*

5. Whenever possible, combine rules to reduce the overall number.

Clustering of every links can efficiently be done by specialized algorithms, but if the existing DT gives the necessary hints, it is simpler to impose symmetry in the Back Propagation algorithm.

5. CONCLUSION

This intelligent hybrid method is proposed to generate predictive value efficiently and accurately. The most suitable, effective and efficient method are identified and observed based on the result of mining the knowledge pattern. From this point, the enhancement will be held on the chosen method to deliver better performance result in term of the accuracy of detected knowledge pattern.

The combination of neural knowledge based with clustering algorithm in the processing phase is to generate interesting rules. For certain problems and types of knowledge, when computing the internal activation potential, the additive aggregated input can be replaced with an multiplicative aggregated input, resulting in *second order units* which can better represent the knowledge (Berkhin e.t. al, 2002). The advantages of applying neural network knowledge based system with clustering algorithm in this module are: (i) it can eliminate superfluous attribute in the system formed by example cases and (ii) it can obtain reduced training examples for the neural network with the significant input.

REFERENCES

Alexandra, I.C. and Toshio, O.(1998). *Energy function construction and implementation for stock exchange prediction NNs*. Second International Conference Based Intelligent Electronic System.

Berkhin, P. and Becher, J. (2002). *Learning Simple Relations: Theory and Applications.* In Proceedings of the 2nd SIAM ICDM, 420-436, Arlington, VA.

Chengyi, S. et al. (1996). *Adaptive clustering of stock prices data using cascaded competitive learning neural networks*.

Clarence, N.W. (1993). *Trading a NYSE stock with a simple artificial neural network based financial trading system*". IEEE.

Guha, S. et al. (1998). *CURE: An efficient clustering algorithm for large databases.* In Proc. Of 1998 ACM-SIGMOD Int.Conf. on Management of Data.

Guha, S. et al. (1999). *ROCK: a robust clustering algorithm for categorical attributes.* In Proc. Of the 15th Int'l Conf. On Data Engineering.

Han, J. et al. (1996). *DBMiner: A system for data mining in relational databases and data warehouse*. Proc 1996 International Conference on Data Mining and Knowledge Discovery, Portland, Oregon.

Hudli, A.V., Palakal, M., " A Neural Network Based Expert System Model*", Proceedings of the 1991 IEEE International Conference on Tool For AI*, San Jose, Nov 1991.

Kazuhiro, K. (1995). *Neural multivariate prediction using event knowledge and selective presentation learning*. IEEE.
MacKay, D. J. C. (1992). *A practical Bayesian framework for backprop networks.* Neural Comp., 4:448--472.

Olson, C. (1995). *Parallel algorithms for hierarchical clustering.* Parallel Computing, 21, 1313-1325, 1995.

Olson, C. (1995). *Parallel algorithms for hierarchical clustering.* Parallel Computing, 21, 1313-1325.

Shan, C. (1998). *Statistical approach to predictive modeling in large databases*. Simon Fraser University: MSc Thesis.

Sheng-Chai, C. et al. (1999). *A forecasting approach for stock index future using Grey theory and neural network*. IEEE.

Towell, G.G. and Shavlik, J.W. (1991). *The extraction of Refined Rules from Knowledge-Based Neural Network*. Machine Learning Research Croup Working Paper 91-4.

Towell, G.G. and Shavlik, J.W. (1993). *Knowledge Based Artificial Neural Networks*. Artificial Intelligence, Accepted 9/93.

Utsugi, A. (1996). *Topology selection for self-organizing maps.* Network: Computation in Neural *Systems*, 7:727--740.

Utsugi, A. (1997), *Hyperparameter selection for self-organizing maps.* Neural Comp, 9:623--635.

Wuthrich, B. et al. (1998). *Daily stock market forecast from textual web data*. IEEE.

Zaiane, R. (1999). *Introduction to data mining*. CMPUT690 Principles of Knowledge Discovery in Databases.

Zhao, Y. and Karypis, G. (2002). *Comparison of agglomerative and partitional document clustering algorithms.* In SIAM(2002) workshop on Clustering High-dimensional Data and Its Applications.

Zhao, Y. and Karypis, G. (2000) *Clustering in life sciences.* Department of computer of science, university of Minnesota, Minneapolis, MN 55455.