Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2000 Proceedings

Americas Conference on Information Systems (AMCIS)

2000

Comparing the Effects of Cognative Style, Subjective Emotion, and Physiological Phenomenon on the Accuracy of Intuitive Time-Series Forecasting Using an SONN: A Proposal

Hyeon Joon Yoo
Sangmyung University, hjyoo@smuc.sangmyung.ac.kr

Byoungho Song
Sangmyung University, bhsong@pine.sangmyung.ac.kr

Hung Kook Park
Sangmyung University, parkh@pine.sangmyung.ac.kr

Russell L. Pimnel
University of Alabama - Tuscaloosa, rpimmel@coe.eng.ua.edu

Comparing the Effects of Cognitive Style, Subjective Emotion, and Physiological Phenomenon on the Accuracy of Intuitive Time-Series Forecasting Using an SONN: A Proposal

Hyeon-Joong Yoo
Division of Information and Telecommunications, Sangmyung University, Korea
hjyoo@smuc.sangmyung.ac.kr

Byoungho Song
Department of Software Science, Sangmyung University, Korea
bhsong@pine.sangmyung.ac.kr

Hung Kook Park
Division of Information and Telecommunications, Sangmyung University, Korea
parkh@pine.sangmyung.ac.kr

Russell L. Pimmel
Electrical Engineering Department University of Alabama, Tuscaloosa, USA, rpimmel@coe.eng.ua.edu

Abstract

Self-organizing neural network (SONN) is known to be able to extract features in input samples [Kohonen, 1995]. By updating not only the weight vector of the winning neuron in the self-organizing layer but also those of its neighboring neurons, neighboring neurons would eventually become to respond similarly to a specific input vector. Then the distribution of winning neurons for a class may be distinguished from those for other classes. Luttrell proposed a SONN which can inherently use the correlation between input vectors of separate clusters and he called it self-supervised adaptive neural network [Luttrell, 1992].

In this report, we propose the use of the self-supervised adaptive algorithm in analyzing the correlation between cognitive style and the accuracy of intuitive time-series forecasting, and suggest a way to compare the relative degree of correlation between each of cognitive style, subjective emotion and physiological phenomenon and the accuracy of intuitive time-series forecasting.

Category: Technical
Keywords: emotion engineering, self-organizing neural network, self-supervised adaptive neural network, determinant analysis, intuitive forecasting, scatter matrix, J-measure

Introduction

We often need to make judgmental time-series forecastings such as estimating stock exchange indices, weather forecasting, and so on. In this report, we propose a non-traditional method to analyze the effect of cognitive style of decision-makers on the accuracy of intuitive time-series forecasting. We use the self-supervised adaptive algorithm [Luttrell, 1992] to find out any correlation between them.

Self-Supervised Adaptive Neural Network

Self-organizing neural network, a competitive network, extracts features in input samples by usually projecting input vectors from a space of higher dimensions into a space of lower dimensions [Kohonen, 1995]. Self-organizing algorithm updates not only the weight vector of the winning neuron in the self-organizing layer but also those of its neighboring neurons. With this training scheme, neighboring neurons would become to respond similarly to a specific input vector.

When the training is completed successfully, we may expect that the weight vectors of neighboring neurons constitute prototypes for a certain class. That means the distribution of winning neurons for a class may be distinguished from the distributions of winning neurons' groups of other classes.

Luttrell proposed self-supervised adaptive neural network [Luttrell, 1992], which is a SONN and inherently uses the correlation between input vectors of separate clusters. The self-supervised adaptive algorithm achieves the ability by updating the weight vectors of neurons in a cluster using the information of the training status of the other clusters. He uses the information to determine the shape of the neighborhood function. In the self-supervised adaptive algorithm the neighborhood function is not necessarily symmetric, which constitutes the most important difference from the conventional SONNs. And the degree of correlation between input vectors of separate clusters affect the performance of the network [Luttrell, 1992]. Actually with the self-supervised adaptive algorithm, we could obtain better forecasting performance in power load forecasting problem than other much more complicated models [Yoo et al., 1999].

Experiments

We first evaluated cognitive styles of 29 students, and measured their forecasting error. Then we added 48 students to get enough number of students for each cognitive style. Hence, the total number of subjects was 77.

Data Generation

A. Cognitive Styles

We used a test to categorize students into four different cognitive styles, i.e., Analytic (A), Behavioral (B), Conceptual (C), and Directive (D). The number of subjects in each group was 25 (32.5 %), 17 (22.0 %), 23 (29.9 %), and 12 (15.6 %), respectively.

We use four-dimensional vector consisting of the four cognitive styles as the input vector to a neuron cluster. When a student belongs to a specific cognitive style, we assign 0.8 to the corresponding element of the input vector and 0.2 to the other elements.

B. Forecasting Error

We showed the students some time-series field data and measured the mean absolute percent error (MAPE) of subjects' forecasting results.

The correlation between cognitive style and the forecasting error is explored using the following method.

The Structure of the Self-Supervised Adaptive Neural Network

We use two clusters of neurons in the self-supervised layer and deploy neurons in one dimension for each cluster. The cognitive styles form the input vector to the first cluster, and the forecasting error becomes the input to the other cluster.

Hence, we use four-dimensional input vector for the first cluster. Each input element corresponds to A, B, C, and D cognitive styles, respectively. When the student belong to class A, 0.8 is assigned to the first element and 0.2 to the rest. The input vector to the second cluster has one element, which is the forecasting error. However, the number of neurons for each cluster is the same and large enough (at least two times the number of classes for the input vectors for the first cluster).

During the training, we accumulate and store the output (i.e., feature value) of each neuron for each pattern.

Discriminant Analysis

After the training, we use the histograms of the accumulated feature values of neurons to obtain the scatter matrices for discriminant analysis of statistics [Fukunaga, 1990] to eventually figure out the relative correlation degrees between the cognitive styles and the error of intuitive time-series forecasting.

During the training we accumulate feature values of each neuron for each pattern in the array

$$F_{p,g,n}$$

where the subscripts p, g and n are indices of patterns, clusters, and neurons, respectively. Then we store the accumulated feature values of neurons for each class (or cognitive style) in the array

$$FC^{c}_{g,n} = \sum_{p=1}^{NP(c)} F_{p,g,n}$$

for patterns in class *c* (3.1)

where *NP(c)* = number of patterns in class *c*. The average of the histogram values of neurons for each class is stored in the array

$$FCM^{c}_{g,n} = \frac{1}{NP(c)} FC^{c}_{g,n}$$

for patterns in class *c* (3.2)

The two arrays in Eqs. 3.1 and 3.2 reflect the sensitivity of neurons to each class.

To find out how differently the neurons respond to different classes by using the mean feature values, we compute the discriminant array shown in Eq. 3.3.

$$FD^{c1,c2}_{g} = \sum_{n=1}^{N} | FCM^{c1}_{g,n} - FCM^{c2}_{g,n} |$$

where *c1* and *c2* are the indices for classes and N stands for the number of neurons in each cluster. This matrix is zero-diagonal and symmetric. The average of discriminant feature over clusters is stored in the array

$$FDM^{c1,c2} = \frac{1}{NG} \sum_{g=1}^{NG} FD^{c1,c2}_{g} \qquad (3.4)$$

where NG stands for the number of neuron clusters in the network.
The average of discriminant feature over clusters and classes is stored in the array

$$FDM^{c} = \frac{1}{NC}\sum_{c2=1}^{NC} FDM^{c,c2} \tag{3.5}$$

which shows the distinctness of each class. The average of discriminant feature over class and comparing class is computed and stored in the array

$$FDM_{g} = \frac{1}{NC^{2}}\sum_{c1=1}^{NC}\sum_{c2=1}^{NC} FD_{g}^{c1,c2} \tag{3.6}$$

which shows the performance of each cluster.

For class c, we can compute

$$E\{(\mathbf{X}-\mathbf{M}_c)(\mathbf{X}-\mathbf{M}_c)^T \mid \omega_c\} = \sum_{p=1}^{P_c} (\mathbf{x}_p-\mathbf{M}_c)(\mathbf{x}_p-\mathbf{M}_c)^T \tag{3.7}$$

where $\mathbf{x}_p$, $\mathbf{M}_c$, and P_c stand for the feature vector, class mean feature vector, and the number of feature vectors in class c, respectively. In a cluster, the diagonal elements show the squared distance between feature vectors

$$F_{p,g,n}$$

and class mean feature vectors

$$FCM_{g,n}^{c}.$$

We assign an array for this matrix as

$$FS_{g,n1,n2}^{c}.$$

To show the scatter of samples around their class expected vector, we use the measure

$$S_w = \sum_{c=1}^{NC} p(\omega_c)\, E\{(\mathbf{X}-\mathbf{M}_c)(\mathbf{X}-\mathbf{M}_c)^T \mid \omega_c\}$$

$$= \frac{1}{NP}\sum_{c=1}^{NC}\sum_{p=1}^{NP(c)} (\mathbf{x}_p-\mathbf{M}_c)(\mathbf{x}_p-\mathbf{M}_c)^T \tag{3.8}$$

where NP stands for the total number of patterns, $NP(c)$ stands for the number of patterns in class c and NC stands for the number of classes. This measure corresponds to the average of the array feature scatter over classes. We assign an array for this within-class feature scatter matrix as

$$FSW_{g,n1,n2}$$

We use a measure for the scatter of class mean feature vectors as in Eq. 3.9.

$$S_b = \sum_{c=1}^{NC} p(\omega_c)\,(\mathbf{M}_c-\mathbf{M})(\mathbf{M}_c-\mathbf{M})^T$$

where

$$\mathbf{M} = E\{\mathbf{X}\} = \sum_{c=1}^{NC} p(\omega_c)\,\mathbf{M}_c. \tag{3.9}$$

We declare an array to store the results of the formula (between-class feature scatter) as

$$FSB_{g,n1,n2}.$$

Finally, we use a J-measure

$$tr(S_w^{-1} S_b) \tag{3.10}$$

to formulate criteria for class separability. It is larger when the between-class scatter is larger or the within-class scatter is smaller.

We can repeat the simulation with subjective emotions and physical phenomena instead of the cognitive styles. Then, by using the results of Eq. 3.10 from the three different simulations, we can compare the correlation degrees between each of the three parameters and the forecasting error.

We expect that using the self-supervised adaptive neural network is advantageous over using the conventional self-organizing neural network in figuring out the correlation degrees between input vectors of separate clusters, since the self-supervised adaptive network can inherently use the correlation between them, and its performance is proportional to the degree of the correlation [Luttrell, 1992].

Acknowledgements

This research was supported by 1999 research funds from Korea Research Foundation (KRF-99-042-C00141).

References

T. Kohonen, *Self-Organizing Maps*, Springer, 1995.

Luttrell S. P., "Self-supervised adaptive networks", *IEE Proceedings-F*, vol. 139, pp. 371-377, 1992.

H-J Yoo and R. Pimmel, “Short term load forecasting using a self-supervised adaptive neural network”, *IEEE Trans. on Power Systems*, vol. 14, pp. 779-784, 1999.

K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, 1990.