Control chart pattern recognition using semi-supervised learning

Miin-Shen Yang* and Jenn-Hwai Yang
Department of Applied Mathematics
Chung Yuan Christian University
Chung-Li 32023
TAIWAN

Abstract: - This paper presents a semi-supervised learning algorithm for a control chart pattern recognition system. A learning neural network is trained with labeled control chart patterns based on unsupervised learning. We then use the classification method based on a statistical correlation coefficient approach to test patterns. We find that the proposed semi-supervised learning algorithm is effective according to numerical comparisons.

Key-Words: Control chart; Pattern recognition; Semi-supervised learning; Labeled pattern; Recognition rate.

1 Introduction

There are growing interests by researchers in control chart pattern recognition. Although Shewhart control charts [15] are the most popular charts that were widely used in industry to detect abnormal process behaviors, these control charts do not provide pattern-related information because they focus only on the latest plotted data points. In general, there are six unnatural patterns in control charts: upward trend, downward trend, upward shift, downward shift, cycle and systematic (see [8]). These patterns present the long-term behavior of a process. To examine this long-term trend in the process over time, control chart pattern recognition has the capability to detect unnatural patterns. Since neural networks have been successfully used to achieve human-like performance in speech and image recognition, they have been widely applied in varieties of pattern recognition. In recent years, one branch of researches in control chart pattern recognition using the neural network approach gets big growing.

Neural networks are generally classified into two categories: supervised and unsupervised. Supervised learning uses a “teacher” in learning stages to guide what behavior will response for certain impulse. Back-propagation and learning vector quantization are supervised. The unsupervised learning is to learn and update one or more weights that have more similarity to input data. The self-organizing feature map and adaptive resonance theory (ART) networks are known to be this type.

The supervised neural networks had been widely used for control chart pattern recognition (see [5], [6], [10], [11], [13], [14], [16]). Although the supervised neural learning techniques give good recognition accuracy, they have limitations because these techniques are lack of adaptiveness without retraining and suffer a slow learning process (see [1]). Thus, Al-Ghanim [1] first presented an unsupervised learning neural network on control charts based on ART networks by feeding some different kinds of unnatural patterns with different disturbance levels, for examples, different shift quantity of shift pattern, different slope values of trend pattern, etc. However, it could only detect unnatural pattern behavior. But it could not identify what an unnatural pattern will occur. This is because unsupervised learning could not label the final output neurons to what an unnatural pattern belongs.

Although to monitor an out-of-control signal in X-bar chart and to detect irregular unnatural patterns are important in production process, to identify unnatural patterns can help us to improve the process. Labeling a pattern to the output neurons is necessary. In this paper, we present a semi-supervised learning algorithm for a control chart pattern recognition system. A learning neural network is trained with labeled control chart patterns based on an unsupervised learning. In this case, we can retain the essence of unsupervised learning scheme, but also label the output neurons to a certain unnatural pattern. According to numerical comparisons, the proposed semi-supervised learning algorithm is effective for control chart pattern recognition.

2 A semi-supervised learning algorithm

In recent years, there are many studies about applying neural learning networks to control chart pattern recognition. Most of them use various kinds of supervised learning networks to generate prototypes for patterns where these prototypes are presented for identifying control charts. But the optimal amount of prototypes for each pattern is difficult to be decided. On the other hand, some unsupervised competitive learning networks with applications to pattern recognition usually need to give a priori number of output neurons. The purpose in a learning (or training) stage is to update winner neuron weights to achieve its stability. To stabilize the network, a common approach is to set up a learning rate so that it is always decreasing in time. But these learning rules usually suffer from a stability and plasticity dilemma problem [9]. ART neural networks were proposed to solve this stability and plasticity dilemma (see [3], [4]) and have good results in clustering. Although it presents a good learning mechanism, unsupervised learning, such as ART, makes the network not suitable for control chart pattern recognition. This is because an unsupervised learning network cannot label the output neurons. In control chart pattern recognition, it is necessary to label output neuron weights so that we can clearly indicate what kind of abnormal control chart pattern is presented. We therefore propose this semi-supervised learning algorithm. The idea is that we use the unsupervised competitive learning rules, but the labeled data are used for learning (or training). This is why we call it a semi-supervised learning algorithm. We need to point out that our semi-supervised learning method is different from those partially supervised [2], [12] or semi-supervised [7] clustering where both of labeled and unlabeled data are used in the clustering algorithms.

The proposed method is based on the statistical correlation coefficient as a similarity measure. We mention that the statistical correlation coefficient was used on control chart pattern recognition with good results by Yang and Yang [17] where it was used in the identifying (or classification) stage by the following equation:

$$r=\frac{\frac{1}{n-1}\sum(x_i-\bar{X})(y_i-\bar{Y})}{\sqrt{\frac{1}{n-1}\sum(x_i-\bar{X})^2}\sqrt{\frac{1}{n-1}\sum(y_i-\bar{Y})^2}}=\frac{\sum(x_i-\bar{X})(y_i-\bar{Y})}{\sqrt{\sum(x_i-\bar{X})^2}\sqrt{\sum(y_i-\bar{Y})^2}} \quad (1)$$

The higher correlation coefficient between two pattern vectors shows higher similarity. However, Yang and Yang [17] simply use the sample average as an only prototype for each control chart pattern in the training stage. Here, we use a semi-supervised learning to find more prototypes for presenting each control chart pattern in the training stage.

To label all trained prototypes we should have labeled training samples for each pattern. These training data sets can be provided by experiences or simulation. Here we have six different unnatural patterns named as upward shift, downward shift, upward trend (or increasing trend), downward trend (or decreasing trend), cycle and systematic patterns. All of them can be divided into normal and disturbance parts. In general, we normalize all data points of control chart so that the pattern $n(t)$ can follow a standard normal distribution in a regular situation. The pattern sample generators are defined as follows:

(a) Upward and downward shift patterns

$$x(t)=n(t)+u\times d \quad (2)$$

$$u=\begin{cases}0 & \text{before shifting,}\\ 1 & \text{after shifting,}\end{cases}$$

where d is the shift quantity randomly taken from 1 to 2.5 for upward shift and from –1 to –2.5 for downward shift.

(b) Upward and downward trend patterns

$$x(t)=n(t)\pm d\times t \quad (3)$$

where d is the trend slope randomly selected from 0.05 to 0.12 for upward trend and from -0.05 to -0.12 for downward trend.

(c) Cyclic pattern

$$x(t)=n(t)+d\times\sin(\frac{2\pi t}{\Omega}) \quad (4)$$

where d is the amplitude randomly selected from 0.5 to 2.5 and Ω is the cycle length taken as Ω =8 here.

(d) Systematic pattern

$$x(t)=n(t)+(-1)^t\times d \quad (5)$$

where d is the amplitude randomly selected from 0.5 to 2.5.

Similar to ART, we do not fix the number of prototypes. Using a threshold value named "vigilance parameter" to determine whether the similarity is enough or not. If the correlation coefficient reaches the vigilance parameter, we then update the weights. On the contrary, a new neuron will be activated. The vigilance parameter in this stage, called h_1, needs to be determined before training. In the traditional competitive learning network, there is only one winner neuron to update its weight, called winner-take-all. But such approach easily causes one neuron to win too often and others not to have an opportunity to learn. If the degree of similarity between input data and neuron weight is greater than h_1, all neuron weights are allowed to be updated as follows:

$$W_j(t) = W_j(t-1) + \lambda_j(t)(X_t - W_j(t-1)) \qquad (6)$$

where X_t is the input data at time t and $\lambda_j(t)$ is the learning rate for the neuron j that is decreasing monotonically. The meaning of (6) is that the neuron weight will be updated to be closer to the input data but maintain in the original status for the others. The learning rate is defined for all neurons as follows:

$$\lambda_j(t) = \frac{1}{t_j} \qquad (7)$$

where t is update times for the neuron j. Thus, the proposed learning algorithm can be created as follows:

The Semi-supervised learning algorithm

Step 1 : Choose the threshold h_1, the pattern length n and the pattern training sample number N.

Step 2 : Select a pattern sample generator from (2) ~ (5). Generate N pattern vectors $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N$ with a pattern length n and different disturbance level d for each pattern.

Step 3 : Set $W_1(1) = X_1$, c=1, t_1=1 and t_j=0, for j=2,3,…,N

DO t = 2 to N

Input $\boldsymbol{X}_t$ and set I = 0

DO j=1 to c

Evaluate the correlation coefficient γ_j between $\boldsymbol{X}_t$ and the neuron weight $W_j(t-1)$

IF $\gamma_j > h_1$ THEN $t_j = t_j + 1$

$$W_j(t) = W_j(t-1) + \frac{1}{t_j}(X_t - W_j(t-1)),$$

I=I+1

ELSE $W_j(t) = W_j(t-1)$

END DO

IF I=0 THEN c=c+1 and $W_c(t) = X_t$

END DO

Step 4 : Output all activated neuron weights W_j and regard them as the prototypes representing this current pattern.

Step 5 : Change another pattern generator and repeat steps 2 and 3 until all pattern prototypes are generated.

We know that supervised learning always penalizes the winner neuron weights if they have incorrect output labels, but rewards the winner neuron weights when they have correct output labels according to the input labeled training data. Although the proposed algorithm uses labeled input training data, the unsupervised learning equation (6) is used to update the neuron weights so that we called it a semi-supervised learning. All of control chart patterns are learned to have several different numbers of prototypes for presenting each control chart pattern.

3 Experimental results and comparisons

The pattern length n is also an argument for such prototype generating technique and it is always depending on different classification approaches. We adopt here n=30 and h_1=0.5 where similar discussion was in Yang and Yang [17]. We take N=300 of the training samples for each pattern. The proposed semi-supervised learning algorithm in this paper has a major improvement, especially in the cases that the training data were collected by experiences. In Yang and Yang [17], they only condense all samples of each pattern to one prototype by taking its average. However, there are maybe one or more different clusters between these samples. In that case it is better to have more prototypes as representatives for the pattern. Table 1 shows the different number of activated neurons with different h_1 after a training algorithm is finished. It is clearly that larger h_1 will activate more neurons.

Table 1. Nubmer of activated neurons

	u.s.	d.s.	u.t.	d.t.	cyc.	sys.
$h_1 = 0.3$	1	1	3	3	2	4
$h_1 = 0.4$	2	2	4	4	5	5
$h_1 = 0.5$	6	4	10	8	9	9
$h_1 = 0.6$	12	9	22	19	15	11

We combined all trained neuron weights as prototypes for testing and all the weights have been labeled because each pattern is training individually. It is also necessary to use equations (2) ~ (5) to generate samples for the testing stages. A threshold, h_2, is created for qualifying the winner whether it matches enough or not. If the similarity measure is smaller than h_2, the winner is not similar enough. We will classify it as a normal pattern. The mechanism will further help us to identify these normal conditions and continue until an unnatural pattern is recognized. If a normal pattern is presented, then a false alarm will occur (i.e. type I error) when it is recognized as an unnatural pattern. Otherwise, a type II error is used to measure the capability of classification for unnatural patterns. Clearly, a larger threshold h_2 will decrease a type I error but increase a type II error. We take h_2=0.5 here that is similar to Yang and Yang [17]. Thus, a classification algorithm is created as follows:

The Classification Algorithm

Step 1: Choose a threshold h_2.

Step 2: A processing data sequence containing recent *n* points is regarded as the pattern size to be recognized.

Step 3: Input the data sequence to the recognizer and calculate its statistical correlation coefficient using (1) with all labeled neuron weights.

Step 4: Choose the maximum value among all outputs to determine which pattern is a winner and then we can classify it as the winner label. If the maximum value is smaller than the threshold h_2, we classify it as a normal pattern.

We can see that Yang and Yang [17] simply took a sample average as the prototype for each control chart pattern in the training stage so that only one prototype is evaluated for one abnormal pattern. However, our proposed semi-supervised learning will generate several different numbers of prototypes for each abnormal pattern (see Table 1). Based on the same testing samples, we compare these two methods according to the correct classification rates. Correct classification rates are counted by the numbers of correct classification over the numbers of testing samples. The correct classification rates of 200 testing samples for two methods are shown in Table 2,. We find that the proposed semi-supervised learning algorithm presents a quite better than Yang and Yang for the shift patterns, but a little worse than Yang and Yang for the trend patterns, and a little better than Yang and Yang for cyclic and systematic patterns. Overall, the proposed method actually presents better results than Yang and Yang [17].

Table 2. Correct classification rates for two methods

	Upward Shift	Downward Shift	Upward Trend	Downward Trend	Cyclic	Systematic	Normal
Our proposed semi-supervised learning	97.5	98	92.5	94.5	97	96.5	100
Yang & Yang (2005)	94.5	93.5	94	94.5	96	96	100

References:

[1] A. Al-Ghanim, 1997, An unsupervised learning neural algorithm for identifying process behavior on control charts and a comparison with supervised learning approaches. Computers and Industrial Engineering, 32 (3), 627-639.

[2] A.M. Bensaid, L.O. Hall, J.C. Bezdek and L.P. Clarke, 1996, Partially supervised clustering for image segmentation, Pattern Recognition, 29(5), 859-871.

[3] G.A. Carpenter and S. Grossberg, 1987, A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, Image Process, 37, 54-115.

[4] G.A. Carpenter and S. Grossberg, 1988, The ART1 of adaptive pattern recognition by a self-organization neural network. Computer, 21, 77-88.

[5] S. I. Chang and C.A. Aw, 1996, A neural fuzzy control chart for detecting and classifying process mean shifts. International Journal of Production Research, 34, 2265-2278.

[6] C.S. Cheng, 1997, A neural network approach for the analysis of control chart patterns. International Journal of Production Research, 35, 667-697.

[7] B. Gabrys and L. Petrakieva, 2004, Combining labelled and unlabelled data in the design of pattern classification systems. International Journal of Approximate Reasoning, 35, 251-273.

[8] E.E. Grant and R.S. Leavenworth, 1996, Statistical quality control. New York: McGraw-Hill.

[9] S. Grossberg, 1976, Adaptive pattern classification and universal recoding, I: parallel development and coding of neural feature detectors. Biol. Cybernet., 23, 121-134.

[10] R.S. Guh and J.D.T. Tannock, 1999, Recognition of control chart concurrent patterns using a neural network approach. International Journal of Production Reseaarch, 37, 1743-1765.

[11] H. Hwarng and N. Hubele, 1993, Backpropagation pattern recognizers for X-bar control charts: Methodology and performance, Computers and Industrial Enginnering, 24 (2), 219-235.

[12] W. Pedrycz and J. Waletzky, 1997, Fuzzy clustering with partial supervision, IEEE Trans. Systems, Man, and Cybernetics-Part B, 27(5), 787-795.

[13] M.B. Perry, J.K. Spoerre and T. Velasco, 2001, Control chart pattern recognition using back propagation artificial neural networks. International Journal of production Research, 39, 3399-3418.

[14] D.T. Pham and E. Oztemel, 1994, Control chart pattern recognition using learning vector quantization networks. International Journal of Production Research, 32, 721-729.

[15] W.A. Shewhart, 1931, Economic control quality manufactured products. New York: Van Nostrand.

[16] M.S. Yang and J.H. Yang, 2002, A fuzzy-soft learning vector quantization for control chart pattern recognition. International Journal of Production research, 40, 2721-2731.

[17] J.H. Yang and M.S. Yang, 2005, A control chart pattern recognition using statistical correlation coefficient method. Computers & Industrial Engineering, 48, 205-221.