Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition

D. T. Pham, Sameh Otri, A. Ghanbarzadeh, E. Koç
Manufacturing Engineering Centre, Cardiff University, Queen's Buildings, The Parade, Newport Rd, Cardiff CF24 3AA , Wales, UK

Abstract

Control charts are employed in manufacturing industry for statistical process control (SPC). It is possible to detect incipient problems and prevent a process from going out of control by identifying the type of patterns displayed by the control charts. Various techniques have been applied to this control chart pattern recognition task.

This paper presents the use of Learning Vector Quantisation (LVQ) networks for recognising patterns in control charts. The LVQ networks were trained, not by applying standard training algorithms, but by employing a new optimisation algorithm developed by the authors. The algorithm, called the Bees Algorithm, is inspired by the food foraging behaviour of honey bees.

The paper first describes the Bees Algorithm and explains how the algorithm is employed to train LVQ networks. It then discusses the recognition of control chart patterns by LVQ networks optimised using the Bees Algorithm.

Keywords:
Bees Algorithm, Neural Networks, Pattern Recognition, Control Charts, LVQ.

1. Introduction

Statistical Process Control (SPC) is a tool for improving the quality of processes [1]. SPC employs statistical means such as control charts to show how consistently a process is performing and whether it should be adjusted. SPC control charts enable a manufacturing engineer to compare the actual performance of a process with customer specifications and provide a process capability index to guide and assess quality improvement efforts. By means of simple rules, it is possible to determine if a process is out of control and needs corrective action. However, it is possible to detect incipient problems and prevent the process from going out of control by identifying the type of patterns displayed by the control charts. Various techniques have been applied to this control chart pattern recognition task [2, 3].

This paper presents the use of Learning Vector Quantisation (LVQ) networks for recognising patterns in control charts in order to determine if the process being monitored is operating normally or if it shows gradual changes (trends), sudden changes (shifts) or periodic changes (cycles) (Figure 1). The LVQ networks were trained, not by applying standard training algorithms, but by employing a new optimisation algorithm developed by the authors. The algorithm, called the Bees Algorithm, is inspired by the behaviour of honey bees [4].

The paper is organised as follows. Section 2 outlines the Bees Algorithm. Section 3 explains the LVQ network and both the standard LVQ training method and the training procedure based on the Bees Algorithm. Section 4 presents the results of control chart pattern recognition experiments using standard LVQ and Bees-Algorithm-trained LVQ networks.

2. The Bees Algorithm

2.1. Bees in nature

A colony of honey bees can be seen as a diffuse creature which can extend itself over long distances in multiple directions in order to exploit a large number of food sources at the same time [5, 6]. In principle, flower patches with plentiful amounts of nectar or pollen that can be collected with less effort should be visited by more bees, whereas patches with less nectar or pollen should receive fewer bees [6, 7].

The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees search randomly from one patch to another. During the harvesting season, a colony continues its exploration, keeping a percentage of the population as scout bees [5].

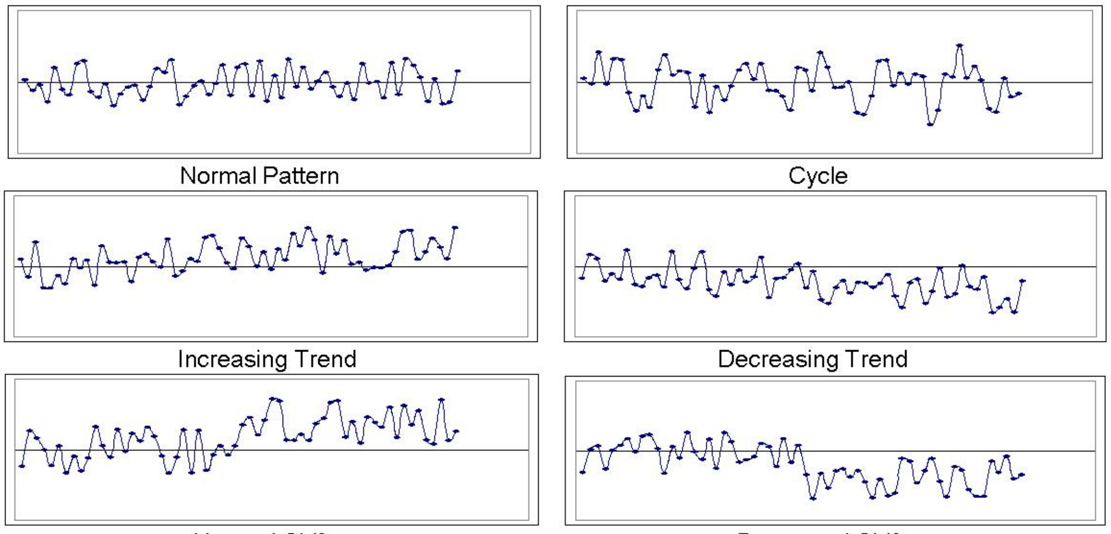

Figure 1. Main types of control chart patterns

When they return to the hive, those scout bees that found a patch which is rated above a certain threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the "waggle dance" [6]. This dance is essential for colony communication, and contains three vital pieces of information regarding flower patches: the direction in which it will be found, its distance from the hive and its quality rating (or fitness) [5, 8]. This information guides the bees to find the flower patches precisely, without using guides or maps. Each individual's knowledge of the outside environment is gleaned solely from the waggle dance. This dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it [8]. After waggle dancing on the dance floor, the dancer bee (i.e. the scout bee) goes back to the flower patch with follower bees that were waiting inside the hive. The number of follower bees assigned to a patch depends on the overall quality of the patch. This allows the colony to gather food quickly and efficiently.

While harvesting from a patch, the bees monitor its food level. This is necessary to decide upon the next waggle dance when they return to the hive [6]. If the patch is still good enough as a food source and then it will be advertised in the waggle dance and more bees will be recruited to that source.

2.2. Proposed Bees Algorithm

As mentioned above, the Bees Algorithm is inspired by the natural foraging behaviour of honey bees to find the optimal solution [4]. Figure 2 shows the pseudo code for the algorithm in its simple form.

The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of elite bees (e), number of patches selected out of n visited points (m), number of bees recruited for patches visited by "elite bees" (nep), number of bees recruited for the other (m-e) selected patches (nsp), size of patches (ngh) and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitnesses of the points visited by the scout bees are evaluated in step 2.

```
1. Initialise population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met)
   //Forming new population.
4. Select elite bees and elite sites for
   neighbourhood search.
5. Select other sites for neighbourhood search.
6. Recruit bees around selected sites and evaluate
   fitnesses.
7. Select the fittest bee from each site.
8. Assign remaining bees to search randomly and
   evaluate their fitnesses.
9. End While.
```

Figure 2. Pseudo code of the Bees Algorithm

In step 4, bees that have the highest fitnesses are chosen as "elite bees". Then, in steps 5 – 7, the algorithm conducts searches in the neighbourhood of the elite bees and of the other selected bees. The latter can be chosen directly according to the fitnesses associated with the points they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighbourhood of the elite bees which represent more promising solutions are made more detailed by recruiting more bees to follow elite bees then other selected bees. Also within scouting, differential recruitment is one of the key operations of the Bees Algorithm. Both scouting and differential recruitment are used in nature.

However, in step 7, for each site only one bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction. This restriction is introduced here to reduce the number of points to be explored. In step 8, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population – representatives from each selected patch and other scout bees assigned to conduct random searches.

3. Standard LVQ Network

3.1. Network structure and standard learning method

The LVQ neural network was developed by Kohonen [9] and has been successfully used for many classification problems. Figure 3 shows an LVQ network, which consists of three layers of neurons: an input layer (buffer), a hidden layer and an output layer. The network is fully connected between the input and hidden layers and partially connected between the hidden and output layers, with each output neuron linked to a different cluster of hidden neurons (also known as Kohonen neurons). The weights of the connections between the hidden and output neurons are fixed at 1. The weights of the input to hidden neuron connections form the components of "reference" vectors, with one reference vector assigned to each hidden neuron. When an input vector is supplied to the network for recognition, the hidden neuron whose reference vector is closest in terms of Euclidean distance to the input vector is said to win the competition against all the other hidden neurons to have its output set to "1". All other hidden neurons are forced to produce a "0". The output neuron connected to the cluster of hidden neurons that contains the winning neuron also emits a "1" and all other output neurons, a "0". The output neuron that produces a "1" gives the class of the input vector, each output neuron being dedicated to a different class. For example, the outputs in the case of the control chart patterns shown in Figure 1 are as given in Table 1.

Table 1. Representation of the output categories

Pattern	Class	Outputs					
		1	2	3	4	5	6
Normal	1	**1**	0	0	0	0	0
Increasing trends	2	0	**1**	0	0	0	0
Decreasing trends	3	0	0	**1**	0	0	0
Upwards shifts	4	0	0	0	**1**	0	0
Downwards shifts	5	0	0	0	0	**1**	0
Cyclic	6	0	0	0	0	0	**1**

The learning method is supervised [10] and based on "competitive" learning, in which neurons compete to have their weights updated. During learning, the neurons in the hidden layer compete amongst themselves in order to find the winning neuron whose weight vector is most similar to the input vector [11]. The winning neuron gives the class of the input vector, as in the recognition phase. Only the winning neuron will modify its weights using a positive or negative reinforcement learning formula, depending on whether the class indicated by the winning neuron is correct or not. If the winning neuron belongs to the same class as the input vector (the classification is correct), it will be allowed to update its weights and move slightly closer to the input vector (positive reinforcement). On the contrary, if the class of the winning neuron is different from the input vector class (the classification is not correct), it will be made to move slightly further from the input vector (negative reinforcement).

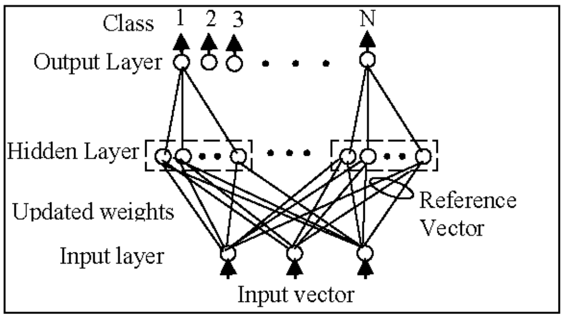

Figure 3. Topology of an LVQ network

3.2. LVQ network training procedure

The training of an LVQ network can be regarded as the minimisation of an error function. The error function defines the total difference between the actual output and the desired output of the network over a set of training patterns [12]. Training proceeds by presenting to the network a pattern of known class taken randomly from the training set. If the class of the pattern is correctly identified by the network, the error component associated with that pattern is null. If the pattern is incorrectly identified, the error component is set to 1. The procedure is repeated for all the patterns in the training set and the error components for all the patterns are summed to yield the value of the error function for an LVQ network with a given set of reference vectors.

In terms of the Bees Algorithm, each bee represents an LVQ network with a particular set of reference vectors. The aim of the algorithm is to find the bee with the set of reference vectors producing the smallest value of the error function.

The LVQ network training procedure using the Bees Algorithm thus comprises the following steps:

1. Generate an initial population of bees.
2. Apply the training data set to determine the value of the error function associated with each bee.
3. Based on the error value obtained in step 2, create a new population of bees comprising the best bees in the selected neighbourhoods and randomly placed scout bees.
4. Stop if the value of the error function has fallen below a predetermined threshold.
5. Else, return to step 2.

4. Control Chart Pattern Recognition Experiments

4.1. Control chart patterns

Each pattern used in the experiments was a time series comprising 60 points. The value $\overline{y}(t)$ at each point t was normalised to fall in the range [0, +1] according to the following equation [12]:

$$\overline{y}(t) = \frac{y(t) - y_{\min}}{y_{\max} - y_{\min}}$$

where

$\overline{y}(t)$ = scaled pattern value (in the range 0 to 1)

$y_{\min}$ = minimum allowed value (taken as 35)

$y_{\max}$ = maximum allowed value (taken as 125)

4.2. Training and test data

A total of 1500 patterns, 250 patterns in each of the six classes, were generated using the following equations:

1. Normal patterns:

$$y(t) = \mu + r(t)\,\sigma$$

2. Cyclic patterns:

$$y(t) = \mu + r(t)\,\sigma + a\sin(2\pi t / T)$$

3. Increasing or decreasing trends:

$$y(t) = \mu + r(t)\,\sigma \pm g\,t$$

4. Upwards or downwards shifts:

$$y(t) = \mu + r(t)\,\sigma \pm k\,s$$

where

μ = mean value of the process variable being monitored (taken as 80 in this work)

σ = standard deviation of the process (taken as 5)

a = amplitude of cyclic variations (taken as 15 or less)

g = magnitude of the gradient of the trend (taken as being in the range 0.2 to 0.5)

k = parameter determining the shift position (= 0 before the shift position; = 1 at the shift position and thereafter)

r = normally distributed random number (between – 3 and +3)

s = magnitude of the shift (taken as being in the range 7.5 to 20)

t = discrete time at which the pattern is sampled (taken as being within the range 0 to 59)

T = period of a cycle (taken as being in the range 4 to 12 sampling interval)

$y(t)$ = sample value at time t

1002 patterns (167 in each class) were used for training an LVQ classifier and 498 patterns (83 in each class) were employed for testing the training classifier.

4.3. LVQ network configuration

The configuration adopted for the LVQ networks was as follows:

- Number of input neurons: 60 (one for each point in the input pattern).
- Number of output neurons: 6 (one for each pattern).
- Number of input neurons: 36 (6 for each cluster).

Therefore, each bee would define a set of thirty 60-dimensional reference vectors.

4.4. Bees Algorithm parameters

Table 2 shows the values of the parameters adopted for the Bees Algorithm. The values were decided empirically.

In addition, the algorithm was initialised with all weight values set randomly within the range 0 to 0.1.

Table 2. The parameters of the Bees Algorithm

Bees Algorithm parameters	Symbol	Value/ range
Population	n	200
Number of selected sites	m	20
Number of elite site	e	1
Patch size	ngh	0.01
Number bees around elite points	nep	20
Number of bees around other selected points	nsp	10

4.5. Results

Table 3 presents the classification results obtained for ten separate runs of the Bees Algorithm.

Table 3. LVQ classification results

Number of runs	Train accuracy	Test accuracy
1	94.24%	93.39%
2	97.10%	96.05%
3	96.20%	95.51%
4	96.93%	96.56%
5	97.92%	95.10%
6	95.71%	94.36%
7	96.17%	95.63%
8	97.03%	96.44%
9	97.89%	95.65%
10	96.40%	96.03%
Max	97.92%	96.56%
Min	94.24%	93.39%
Mean	96.56%	95.47%

A typical plot of how classification accuracy evolves during training is shown in Figure 4.

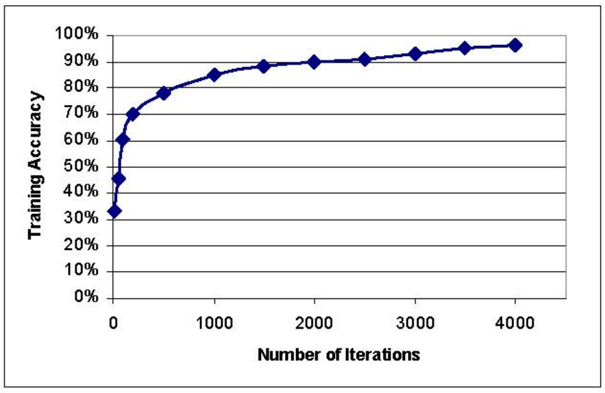

Figure 4. Effects of number of iterations on the classification accuracy

For comparison, the average for the ten runs is given in Table 4 against the average classification results for an LVQ network trained using the standard LVQ algorithm [13].

Table 4. Results of different pattern recognisers

Pattern recogniser	Learning accuracy	Test accuracy
LVQ (stand.)	95.18%	92.31%
LVQ (Bees)	96.56%	95.47%

5. Conclusion

This paper has described the use of the Bees Algorithm to train the LVQ neural network for control chart pattern recognition. Despite the high dimensionality of the problem – each bee represented 2160 parameters that had to be determined – the algorithm still succeeded to train more accurate classifiers than that produced by the standard LVQ training algorithm. Future work will be directed at investigating classifiers based on other neural network paradigms and comparing their performances.

6. Acknowledgements

The research described in this paper was performed as part of the Objective 1 SUPERMAN project, the EPSRC Innovative Manufacturing Research Centre Project and the EC FP6 Innovative Production Machines and Systems (I*PROMS) Network of Excellence.

7. References

[1] Montgomery, D. C. (2000), "Introduction to Statistical Quality Control", 4th ed., Wiley, New York, NY.
[2] Pham, D T and Chan, A B (2001) "Unsupervised Adaptive Resonance Theory Neural Networks for Control Chart Pattern Recognition", Proceedings of Institution of Mechanical Engineers, Volume 215, Part B, pp 59-67.
[3] Pham, D T and Oztemel, E (1995) "An Integrated Neural Network and Expert System Tool for Statistical Process Control", Proceedings of Institution of Mechanical Engineers, Vol. 209, Part B, pp 91-97.
[4] Pham, D.T., S. Otri, E. Koc, A. Ghanbarzadeh, S. Rahim, and M. Zaidi, (2005). Technical Note: "The Bees Algorithm"., Manufacturing Engineering Centre, Cardiff University: Cardiff, UK.
[5] Frisch, K.v., Bees: (1976), "Their Vision, Chemical Senses and Language". Revised Edition ed., Ithaca, N.Y.: Cornell University Press.
[6] Seeley, T.D., (1996), "The Wisdom of the Hive: The Social Physiology of Honey Bee Colonies", Cambridge, Massachusetts: Harvard University Press.
[7] Bonabeau, E., M. Dorigo, and G. Theraulaz, (1999) "Swarm Intelligence: from Natural to Artificial Systems", New York: Oxford University Press.
[8] Camazine, S., J.-L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraula, and E. Bonabeau, (2003), "Self-

Organization in Biological Systems" Princeton: Princeton University Press.
[9] Kohonen, T., (1989). "Self-Organising and Associative Memory" (3rd ed.). Springer-Verlag ,Berlin
[10] Jain, A. K. and Dubes, R. C. (1988), "Algorithms for Clustering Data". Prentice Hall, Englewood Cliffs, NJ.
Maass, W., Bishop C. M., (2001). "Pulsed Neural Networks". The MIT Press, London, ISBN: 0-262-63221-7.
[11] Kohonen, T., (1990). "The Self-Organising Map". Procs. IEEE, Vol. 78, No. 9, pp. 1464-1480.
[12] Pham D.T. and Oztemel E. (1992), "Control Chart Recognition Using Neural Networks". Journal of Systems Engineering. pp. 256-262,.
[13] Pham, D. T. and Oztemel, E., (1994). "Control Chart Pattern Recognition Using Learning Vector Quantisation Networks". International Journal of Production Research, Vol. 32, No. 3, pp. 721-729.