

Control Chart Patterns Recognition based on DAG-SVM

Zhongbao Xiao^{1,2, a*}, Shuhong Chen^{1, b}

¹Shenyang Institute of Automation, Chinese Academy of Sciences,

Shenyang 110016 , China

²University of Chinese Academy of Sciences , Beijing 100049 , China

^axiaozhongbao@sia.cn, ^bchensh@sia.cn

Keywords: DAG,SVM,PSO

Abstract. Statistical process control charts have been widely utilized in manufacturing processes for determining whether a process is run in its intended mode or in the presence of unnatural patterns, it's a multi-class classifier problem. Effective approaches to recognize control chart patterns is essential for a manufacturing process to maintain high-quality products. This paper we use the Directed Acyclic Graph(DAG)tree learning architecture ,which combines many two-class classifiers together to solve the multi-class classifier problem. For each node we chose the support vector machine(SVM)using a particle swarm optimization(PSO) algorithm to optimize the parameter of the SVM kernel function. Here the PSO not only takes the kernel function parameters as variables but also the feature vector of the SVM to optimize .Simulation results show the propose algorithm achieves a high recognition accuracy and solve the unable recognition area.

Introduction

Control charts are widely used in modern industrial and service organizations in recent years. Various kinds of control chart have been developed according to different quality attributes and control targets. Monitoring process fluctuation with control charts was first proposed by shewhart in 1924. It is believed that the process fluctuation involves abnormal changes due to assignable causes and normal changes due to unassignable causes. Therefore automatically recognizing control chart patterns (CCPs) is an essential issue for identifying the process fluctuation effectively. CCPs can exhibit six common types of pattern: normal(NOR),cyclic(CYC),increasing trend(IT),decreasing trend(DT),upward shift(US),downward shift(DS). Fig.1 shows these six patterns [1].

Haghtalab and Xanthopoulos proposed an unsupervised consensus learning approach in CCPs recognizing. A hybrid approach by integrating wavelet transform and improved particle swarm optimization-based support vector machine (P-SVM) for on-line recognition of concurrent CCPs was developed by Du, Shichang in [5]. Hachicha did a survey of CCPs recognition literature in 2012 and found the Artificial Neural Network(ANN) was the most used method. Vannee and Ebrahimzadeh studied an efficient genetic algorithm for selecting of appropriate parameters of the classifier and confirmed a high accuracy.

As mentioned above, much effort has been done to develop a high accuracy method to solve the CCPs recognizing problem. In this study we improve the PSO algorithm which combine the feature vector and kernel function parameter as a dimension of search space and give a fix value of each feature between zero and one, the rest step is the same as the standard algorithm. In the decision part, we use DAG instead of One against One(OAO) or One against All(OAA),this solves the

unable separate area the OAO or OAA may present, at the same time the accuracy achieves high .

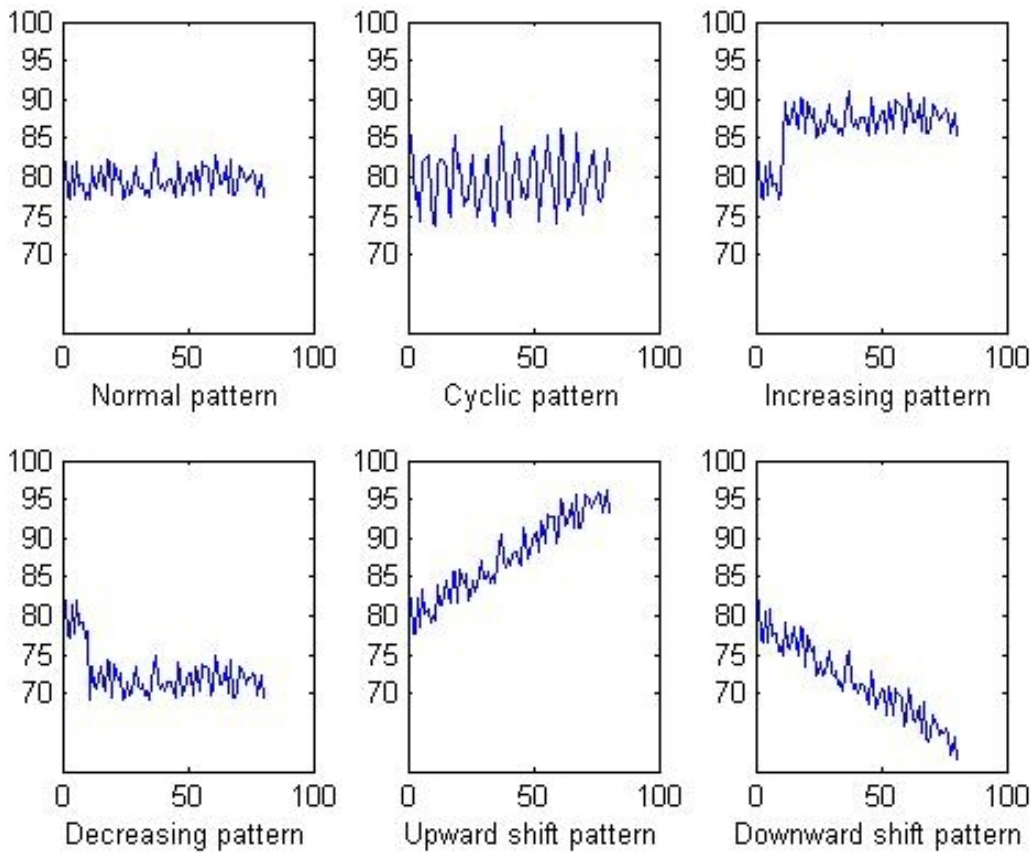


Fig.1. six patterns

SVM Preliminary

SVM is developed from solving the linear separable problem which tries to find a hyperplane separating the example data with maximal margin and least error. This tries to make sure the empirical risk and trust risk are minimum so that the real risk is minimum.

For binary problem, we have training data points:

$$(x_i, y_i); x_i \in \mathbb{R}^{n \times 1}, y_i \in \{1, -1\}, i=1, 2, 3, \dots, l$$

Assume the points are linear separable, there is a hyperplane.

$$\omega^T x + b = 0; \omega \in \mathbb{R}^{n \times 1}, x \in \mathbb{R}^{n \times 1}, b \in \mathbb{R}$$

Subject to

$$y_i (\omega^T x_i + b) \geq 1; i=1, 2, \dots, l \quad (1)$$

To maximize the distance between the data points and the hyperplanes, we have

$$L_P = \min \frac{(\|\omega\|)^2}{2} \quad (2)$$

$$\text{S.T. } y_i (\omega^T x_i + b) \geq 1; i=1, 2, \dots, l \quad (3)$$

The above is an optimization problem with a convex quadratic objective and only linear constraints. Its solution gives us the optimal margin classifier.

When we construct the Lagrangian for our optimization problem we have:

$$L_P = \frac{1}{2\|\omega\|^2} - \sum_{i=1}^l a_i [y_i (\omega^T x_i + b) - 1]; a_i \geq 0, i=1, 2, \dots, l \quad (4)$$

Note that a_i is Lagrange multiplier.

Let's find the dual form of the problem.

$$L_D = \max \left(\sum_{i=1}^l a_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l a_i a_j y_i y_j x_i^T x_j \right) \quad (5)$$

$$\text{S.T. } \sum_{i=1}^l a_i y_i = 0; \quad (6)$$

According to Karush-Kuhn-Tucher(KKT)conditions, we get:

$$y_i (\omega^T x_i + b) = 1; \text{subject to } \forall a_i \neq 0, \text{ these points satisfied this equation are called support vectors in this}$$

$$\text{problem. The optimal result } b^* = y_i - \omega^T x_i \quad (7)$$

the optimal decision function is then given by

$$f(x) = \text{sgn} \left(\sum_{i=1}^l y_i a_i^* x^T x_i + b^* \right) \quad (8)$$

where a_i^* is the solvion of the dual problem.

For input points with a high noise level, an SVM using soft margins can be expressed as follows with the introduction of the non-negative slack variables

$$y_i (\omega^T x_i + b) \geq 1 - \varepsilon_i ; i=1,2,\dots,l \quad (9)$$

to obtain the separating hyperplanes, we should solve the following quadratic objective.

$$\emptyset = \min \frac{1}{2 \|\omega\|^2} + C \sum_{i=1}^l \varepsilon_i \quad (10)$$

$$C > 0$$

$$\text{S.T. } y_i (\omega^T x_i + b) \geq 1 - \varepsilon_i ; i=1,2,\dots,l ; \varepsilon_i > 0 \quad (11)$$

where C is the penalty parameter ,which controls the trade-off between the complexity of the decision function and the number of training examples that have been misclassified.

For nonlinear separable case, we may solve the problem in a high dimension space by a kernel function. We use the radial basis function(RBF)kernel which is the commonly used.

$$\text{RBF kernel function: } K(\vec{x}_i, \vec{y}_i) = e^{(-\gamma \|\vec{x}_i - \vec{y}_i\|)}$$

Where γ is the parameter of kernel function.

After we use the RBF kernel function, our dual problem becomes:

$$L_D = \max \left[\sum_{i=1}^l a_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l a_i a_j y_i y_j K(\vec{x}_i, \vec{y}_i) \right] \quad (12)$$

$$\text{S.T. } \sum_{i=1}^l a_i y_i = 0 \quad (13)$$

$$0 \leq a_i \leq C ; i=1,2,\dots,l \quad (14)$$

Solving the above problem we get our decision function:

$$f(x) = \text{sgn} \left(\sum_{i=1}^l y_i a_i^* K(\vec{x}_i, \vec{y}_i) + b^* \right) \quad (15)$$

The performance is controlled by the C and γ , which are called hyperparameters.

Multi-class SVM

The SVM is essentially a binary classifier, however, the CCPs recognizing is a multi-classifier problem. The followings are the widely used methods to extend binary SVM to multi-class problem: one-against-one(OAO),one-against-all(OAA),directed acyclic graph(DAG), error correcting output coding(ECOC),data-driven error correcting output coding(DECOC).It's believed that when there are few patterns it's a perfect choice of OAO to build multi-class classifier. By doing this we can achieve high accuracy and effective .however, it has a fatal problem it will appear a unable separable area, so in this paper we chose the DAG, which overcome that drawback. Fig.2 shows the typical schematic diagram of DAG, we use the same parameter as the OAO classifier to make it simple.

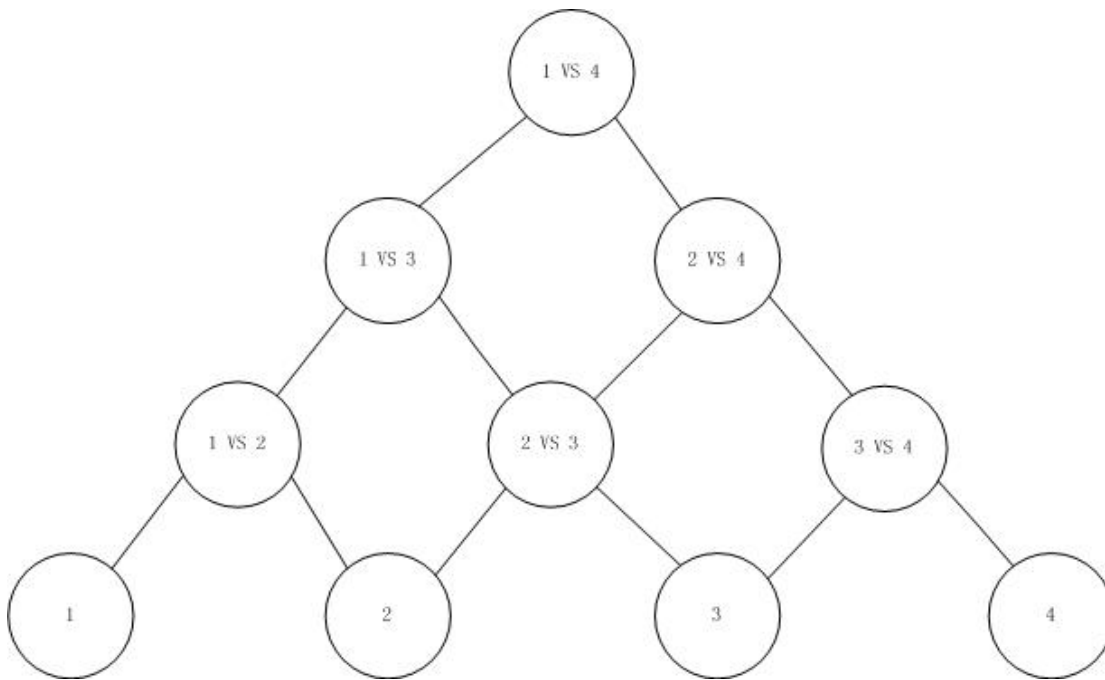


Fig.2. schematic diagram of DAG

Particle Swarm Optimization

PSO is a member of the intelligent optimization algorithms. PSO is simple in concept, few in parameters, and easy in implementation. We can describe the PSO algorithm in following way:

Assuming in the n dimensions space, there is a group(X) made up of m particles(x_i).

$X = \{x_1, x_2, \dots, x_m\}$; where $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}^T$; $i=1, 2, \dots, m$. For each particle, its speed is

$v_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}^T$, the best previous position of the i th is $P_i = \{P_{i1}, P_{i2}, \dots, P_{in}\}^T$, the best particle among all particles is expressed as:

$$P_g = \{P_{g1}, P_{g2}, \dots, P_{gn}\}^T$$

The speed and position of every particle is dynamically adjusted according to the equation (16,17)

$$v_{is}^{(t+1)} = \omega * v_{is}^t + c_1 r_1 (p_{is}^{(t)} - x_{is}^{(t)}) + c_2 r_2 (p_{gs}^{(t)} - x_{is}^{(t)}) \quad (16)$$

$$x_{is}^{(t+1)} = x_{is}^{(t)} + v_{is}^{(t+1)} \quad (17)$$

Where s is the dimension, c_1 and c_2 are positive constants, r_1 and r_2 are two random functions in the range $[0,1]$, and ω is the inertial weight.

PSO-SVM

As we have discussed before, the performance of the decision function is determined by the parameter c and r , but on the other hand one of the SVM classifier problem is selecting the number of features. So in the next, we combine the feature and c and r together to optimize.

Reference [1] shows the flowchart of the PSO-based parameter determination and feature selection approach for the SVM classifier. Particle representation. The particle is comprised of three parts: the input features, C and r , when the RBF kernel function is selected Fig.3 illustrates the representation of a particle with dimension $n+2$, where n is the total number of input features of a data set. The value of the n variables ranges between 0 and 1. If the value of a variable is less than or equal to 0.5, then its corresponding feature is not chosen. Conversely, if the value of a variable is greater than 0.5, then its corresponding feature is chosen [1].

Simulation

We use the equations (18-23) to produce our data points for each pattern.

$$(i) \quad \text{Normal pattern: } y(t) = \mu + r(t) \times \sigma \quad (18)$$

$$(ii) \quad \text{Cyclic pattern: } y(t) = \mu + r(t) \times \sigma + a \times \sin \frac{2\pi t}{T} \quad (19)$$

$$(iii) \quad \text{Increasing pattern: } y(t) = \mu + r(t) \times \sigma + g \times t \quad (20)$$

$$(iv) \quad \text{Decreasing pattern: } y(t) = \mu + r(t) \times \sigma - g \times t \quad (21)$$

$$(v) \quad \text{Upward shift pattern: } y(t) = \mu + r(t) \times \sigma + b \times s \quad (22)$$

$$(vi) \quad \text{Downward shift pattern: } y(t) = \mu + r(t) \times \sigma - b \times s \quad (23)$$

Where $\mu=80, \sigma=5$, $r(t)$ is a normal distribution; a is less than 15 and above 0, T is between 4 and 12 sampling intervals; g sets between 0.2 and 0.4; s ranges 7.5 and 20; b is zero before the shift and b equals one at the shift thereafter.

Features

According to [1], we chose 12 features which include 8 shape features and 4 statistical features. We produce 258 data points for each pattern, 129 for training and the rest for testing. The simulation parameter is set as below.

$$C \in [1, 1000]; \gamma \in [0, 1]$$

Swarm size: 20

constants: $c_1 = c_2 = 2$

maximum number of iterations (t_{\max}):

ω ranges $[0.9 \quad 12]$ and is linearly decreasing according to the following equation.

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} * t$$

Conclusions

At first experiment, we test the OAO class-binarization strategy and max-win principle. The result is showing in table 1. Here we have 6 patterns and construct 15 classifiers according to OAO strategy. Obviously, every pattern is separated in its own field, however when we analysis the table carefully we could find in row 2th, 3th, 5th the maximum one is so close to other one in the same row. In other words, it's not robust or it's easy to make wrong and the reason why it could be this is

because we use the max-win principle. In order to avoid this problem, in our second experiment, we change our principle to DAG tree, which can solve the unable area problem radically. We construct our DAG tree from bottom based on the difference between each pattern in table 1. That is to say we chose pattern 1 and 5 as the both ends and the biggest difference from them as their neighbourhood, then repeat this step. Our final DAG tree is showing in Fig. 3.

Table 1 result of OAO vote

Principle		NOR	CYC	IT	DT	US	DS
OAO	NOR	5	1.2868	2.6899	1.3488	1.9380	2.7364
	CYC	3.8925	4.9302	1.9845	1.1783	1.0155	2.0620
	IT	1	0.0233	4.9767	3.7829	2.4884	2.7287
	DT	0.9690	0.0388	2.9147	4.9767	2.8062	3.2946
	US	0.9767	0.0233	3.9535	2.0545	5	2.9922
	DS	1.0078	0.0155	2.9767	3.0465	2.9612	4.9922

We run this new architecture with the same SVM classifiers as previous simulation. Table 2 shows the obtained result in the form of pattern confusing matrix. By doing this, we can see it achieves a high accuracy, what's more, it can never appear the unable separable area.

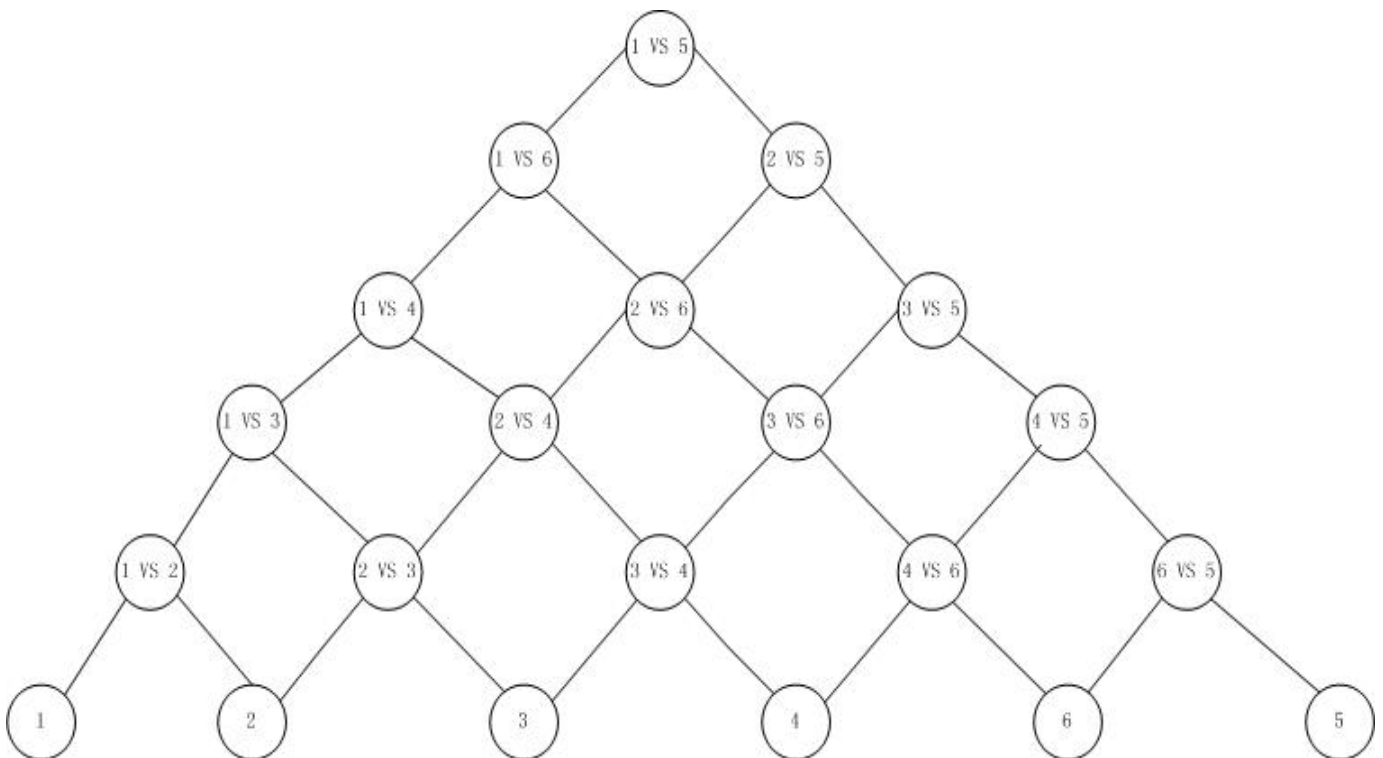


Fig.3. DAG tree

Table 2 pattern confusing matrix

Algorithm	Accuracy		NOR	CYC	IT	DT	US	DS
DAG-SVM	127.67/129	NOR	129	0	0	0	0	0
		CYC	1	128	0	0	0	0
		IT	0	0	126	3	0	0
		DT	0	0	0	126	0	3
		US	0	0	0	0	129	0
		DS	0	0	0	1	0	128

At present, we build our DAG tree referencing to our previous result. It's still difficult to present a new approach for every DAG tree constructing. The future plane is to find an efficient way to construct the DAG tree with high accuracy.

Reference

- [1] Vahid, Ranaee*, Ata, Ebrahimzadeh. Application of the PSO-SVM mode for recognition of control chart patterns[J]. elsevier, 2010, (49): 577-586.
- [2] H. Yi, X.F. Song, B. Jiang, Structure selection for DAG-SVM based on misclassification cost minimization, *Int. J. Innov. Comput. Inf. Control* 7 (2011) 5133–5143.
- [3] WANG Tong, GAO Xianwen, ZHAI Yujia etc. Improved Detection and Isolation of Sensor Fault Based on PSO-LSSVM Prediction[J]. *Information and control*, 2014,43(2): 146-151.
- [4] Vapnik V. *The nature of statistical learning theory*[M]. New York, USA: Springer, 1995: 1-104.
- [5] Du, S.,Huang, D., & Lv, J.(2013). Recognition of concurrent control chart patterns using wavelet transform decomposition and multiclass support vector machines. *Computers & Industrial Engineering*, 66(4), 683–695.
- [6] B. Fei, J. Liu, Binary tree of svm: a new fast multiclass training and classification algorithm, *IEEE Trans. Neural Netw.* 17 (3) (2006) 696–704