# SPC Control Chart Classification using Support Vector Machine and Label Spreading

Hai T. Mai and Young-chan Kim
Department of Computer Engineering
Hanbat National University
125 DongseoDaero, Yuseong-gu,Daejeon 305-719, ROK
hai.maituan@gmail.com, yckim@hanbat.ac.kr

**Abstract**

*StatisticalProcess Control (SPC) charts also known as Shewhartcharts have been used widely in many processes. However, with so many control charts in manufacturing processes, process engineers find the difficulty to interpret/classify control charts into the specific categories. Several attempts have been made to classify control charts automatically but there is still a room for higher recognition accuracy. This paper demonstrates how effectively support vector machines as supervised learning can be used in SPC Control Chart classification. However, in reality, labeled data always is limited and expensive whereas unlabeled data is obviously cheap and available. In this study, we also present a new approach to classify control chart using Label Spreading as semi-supervised learning for unlabeled data.*

## 1. Introduction

In manufacturing processes, statistical process control plays a very important role to not only improve product quality but also effectively monitor process performance. In fact, no process is truly stable; the process variance is the result from both natural/inherent variation (common cause) and unnatural variation (assignable cause). A final objective of SPC is to quickly detect the occurrence of assignable causes of process shifts in order to investigate process and take corrective actions before many products are manufactured.

In statistical process control, one of the most powerful tools that use for problem-solving is control chart which is useful to achieve process stability and improve capability through the reduction of variability. The eventual goal of statistical process control is the elimination of variability in process. It may not be possible to eliminate completely variability, but the control chart is an effective tool in reducing variability as much as possible. By doing that, it brings the abnormal process to back to normal process that has only inherent variation.

## 2. Control Chart

Control charts can be classified into two general types. Control charts for central tendecy and variability are collectively called ***variables control charts.***

The X Chart is the most widely used chart for controlling central tendecy , whereas charts based on either the sample range or sample standard deviation are used to control process variability.

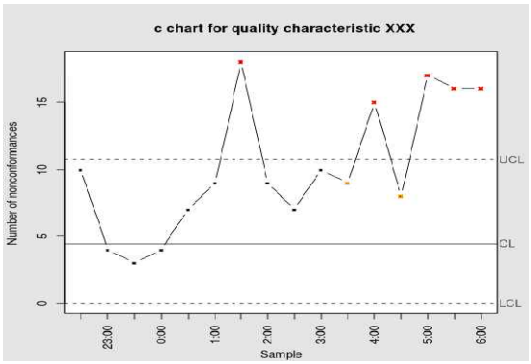


Figure 1 Control chart

Many quality charactiristics are not measured on a continuous scale or even a quatitative scale. Therefore, we can judge product as conforming or nonconforming based on it has specific attributes or the number of defects that appearing on product. Control charts support such quality characteristics are called ***attributes control charts***

Control charts have had a long history of use in U.S and all over the world. There are at least five reasons for their popularity:

- Control charts are a proven technique for improving productivity
- Control charts are effective in defect prevention
- Control charts prevent unnecessary process adjustment
- Control charts provide diagnostic information
- Control charts provide information about process capability

However, control charts can be used improperly without enough knowledge or historical data to judge. Furthermore, control charts themselves have no pattern-related information that indicates how abnormal they are; due to lack of control chart pattern recognition ability.

Control chart patterns have six types of pattern: Normal (NR), Cyclic (CC), upward trend (UT), downward trend (DT), upward shift (US), downward shift (DS). All patterns are except Normal show that the process does not work correctly and requires the corrective actions to be taken.

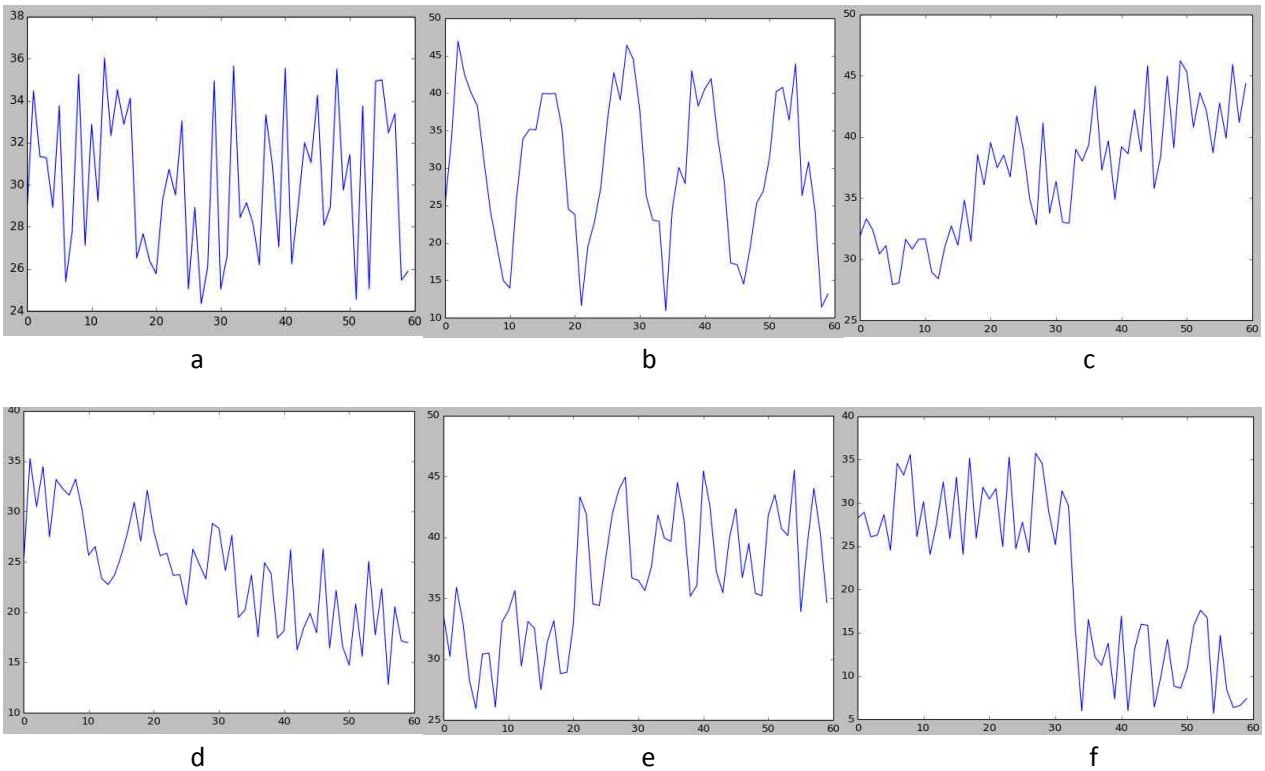


Figure 2 (a) Normal pattern, (b) Cyclic pattern, (c) Upward trend pattern, (d) Downward trend pattern
(e) Upward shift pattern, (f) Downward trend pattern

Generally speaking, control chart patterns can be classified into four types because of we can group downward trend and upward trend into trend patterns; downward shift and upward shift into shift patterns. As a result, they are pretty much similar to other researchers. Whenever anomaly patterns come, the numerous practitioners can address their corresponding assignable causes to the below [6]:

- Trend patterns: A trend can be defined as a continuous movement in either positive or negative direction. Possible causes are tool wear, operator fatigue, equipment deterioration, and so on.

- Shift patterns: A shift can be defined as a sudden change above or below the average of the process. This change may be caused by an alternation in process setting, replacement of raw materials, minor failure of machine parts, or introduction of new workers, and so forth.

- Cyclic patterns: Cyclic behaviors can be observed by a series of peaks and troughs occurring in the process. Typical causes are the periodic rotation of operators, systematic environmental changes or fluctuation in the production equipment.

- Systematic patterns: The characteristic of systematic patterns is that a point-to-point fluctuation has systematically occurred. It means a low point is always followed by a high point and vice versa. Possible causes include difference between test sets and difference between production lines where product is sampled in rotation.

## 3. Previous works

Many attempt shave been made to solve control chart pattern recognition problems.

From 1992, some researchers started building control chart interpretation system [4] that used expert system to make transparent interpretation, however, expert system build on the human knowledge so that sometimes the inferences are not always consistent. Furthermore, expert system based on rules that require a lot of efforts to setup many rules, because of this, it can make the inference processing slow.

Nevertheless, most of researchers prefer neural networks which have better performance than that of expert systems and statistical classification. Therefore, many researchers used artificial neural network (ANN) with back propagation algorithm. Back propagation network has some strong points such as good performance in recognizing control chart patterns; still it has many inherent disadvantages such as slow training speed and local optimum [10].

Moreover, decision trees algorithm has been used by some researchers to classify control chart patterns. Although decision trees is simple to understand and interpret as well as the trees can be easily visualized, but it also has some weak points such as decision tree learners can create over-complex trees that do not generalize the data well (over-fitting) or it's unstable and can output a completely different tree just because of small variations in the data.

Most of the existing techniques used supervised learning algorithm to train the model but in the real manufacturing processes, such labeled data is limited, expensive and requires experts to put the label on it. Conversely, unlabeled data is cheap and always available and easy to get with very large amount of data without expert's judgments on it.

In this paper, we propose two approaches to classify SPC control chart. One is multi-class support vector machine with OAO (one against one) and another is label spreading as semi-supervised learning, the model can be trained using labeled data as a small portion and unlabeled as a large portion of training dataset.

## 4. Classifier

The main objective of control chart pattern recognition is how to choose the best classifier with highest accuracy rate which is appropriate for manufacturing processes. We have proposed a multi-class SVM (OAO) as supervised learning and label propagation as semi-supervised learning.

### 4.1 Support Vector Machine

The original SVM algorithm was invented by Vladimir N. Vapnik and the current standard incarnation (soft margin) was proposed by Corinna Cortes and Vapnik in 1993 and published in 1995. SVMs are among the best (and many believe and indeed the best) supervised learning algorithm and always in the list of top 10 machine learning algorithms. Non-linear kernel-based algorithm such as Gaussian Process etc… which have the significant limitation is that the kernel function $k(x_n,x_m)$ must be evaluated for all possible pairs $x_n$ and $x_m$ of training points, which can be computationally infeasible during training and can lead to excessive times when making predictions for new data points. Conversely, SVM is ***sparse solution*** so the predictions for new points depend only on the kernel function evaluated at a subset of the training data points.

#### A. Binary SVM:

Basically, binary SVM constructs a separated hyper-plane that maximizes the margin between two data points which belong to the two separated classes. Given training data and its corresponding labels $(x_n,y_n)$,n= 1… N, $x \in R^d$, $y \in \{-1,+1\}$, suppose training set can be separated by the hyper-plane $w^T+b=0$ where w is weight vector and b is bias, then the hyper-plane maximizes the margin by minimizing the function:

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

subject to $y_i(w^T x_i+b) \geq 1$ for all $x_i$ i =1,2..N.

There is a tradeoff between maximum margin and the number of misclassified data points. To get the better generalization and controlling of number of misclassified data points, slack variable is introduced to allow a number of misclassified data, then the function to be minimized:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \varepsilon_i$$

subject to $y_i(w^T x_i+b) \geq 1-\varepsilon_i$ for all $x_i$i =1,2..N.

In case of nonlinear separable cases, SVM was proposed that the original finite-dimensional space be mapped into a much higher- or infinite-dimensional space, presumably, making the linear separation easier in that space using kernel function $K(x_i,x_j)$. There are some popular kernel functions that are using for SVMs: Linear, Radius basic function (RBF), Polynomial. It's possible to use other kernel functions as long as they satisfy Mercer's condition. The performance of SVM is based on hyper-parameters(C,K).

| Support Vector Machine Type | Kernel $k(x_i,x_j)$ |
|---|---|
| Polynomial(homogeneous) | $(x_i.x_j)^d$ |
| Polynomial(inhomogeneous) | $(x_i.x_j+1)^d$ |
| Radial basic function | $\exp(-\gamma\|x_i-x_j\|^2)$ |

#### B. Multiclass SVM:

Fundamentally, two widely methods which are used to extend binary SVMs to Multiclass SVMs, are OAO (One Against One) and OAA (One Against All). Jonathan Milgram together with Mohamed Cheritet and Robert Sabourin had taken a research on which one is the

better method with SVM[12]. The answer will depend on what the problem is! Definitely, it is not reasonable to claim that one strategy is always better that the other but according to the specific constraints, the number of classes, and the number of training samples, it will be one or the other of two strategies that will be more appropriate to solve the classification problem.

In SPC control chart recognition, number of classes can be extended more by process engineers and in the reality of manufacturing or production; the number of training samples will be very large (for example: nuclear power plant or oil refinery plant). In such cases, the training time can be problematic with OAA and the accuracy of OAA and OAO are not significant different. In this paper, we use OAO in our classification so it will works well even in the large datasets.

### 4.2 Label Propagation

#### A. Label propagation

Label propagation algorithm (LP) [16] is a semi-supervised learning algorithm which is based on graph and shows good results when the amount of labeled data is low with respect to the supervised options. It works as a form of propagation on a graph, where a label of node propagates to its neighbor nodes according to their proximity. The labeled data work like sources which transmit labels though unlabeled data. All nodes are covered with data points for both unlabeled and labeled data in a full connected graph. The edge between nodes $i, j$ ( $i, j = 1,2 \dots n$) which represents their similarity, and its weight is calculated as:

$$w_{ij} = exp\left(-\frac{\|x_i - x_j\|^2}{\alpha^2}\right) \qquad (1)$$

where$\alpha$ is a bandwidth hyper-parameter. So, the sample similarity matrix is$[W]_{n\times n}$, where n is the number of samples for a given training set. LP algorithm also defines a $n \times n$ probabilistic transition matrix$[P]_{n\times n}$:

$$P_{ij} = P(i \to j) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \qquad (2)$$

where $P_{ij}$ is the probability of transition from node $i$ to node $j$. Define a $l \times c$(c: number of classes and $l$ is the number of labeled data) matrix $f_L$and a$u \times c$ ($u$ is the number of unlabeled data) matrix $f_U$, whose element $f_{ij}$ as:

$$f_{ij} = \begin{cases} 1, & j = c, x_i \epsilon L \\ 0, & j \neq c, x_i \epsilon L \\ \epsilon\,[0,1], & x_i \epsilon U \end{cases} \qquad (3)$$

$$f_X = \begin{bmatrix} f_L \\ f_U \end{bmatrix} \qquad (4)$$

LP algorithm propagates the label of node from its neighbor in form of

$$f_X^{(i)} = P \times f_X^{(i-1)} \qquad (5)$$

Label propagation's implementation goes as follow:

1. Set loop variable i=0, initialize $f_U^{(i)} = 0$ ;
2. Compute probabilistic transition matrix $P$ , according to (2);
3. Set $i = i + 1$, and compute $f_U^{(i)}$, according to (5);
4. Repeat from step (2) to step (3) until $f_U^{(i)} = f_U^{(i-1)}$ reaches;
5. Label the unlabeled data $x_j$ using its label $y_j = \arg max_k f_{jk}$ , where $\arg max_k f_{jk}$ is the final label of $x_j$.

#### B. Label Spreading

Label propagation and label spreading [17] both work by constructing a similarity graph over all items in the input dataset. But they differ in modification to the similarity matrix that graph and the clamping effect on the label distributions. By using clamping algorithm can change the weight of the labeled data that effect to the

inference. Label propagation uses directly the raw similarity matrix constructed from the data with no modifications. Dissimilarity, Label Spreading minimizes a loss function that has regularization properties; it leads to more robust to noise. Label Spreading iterate on a modified version of the original graph and normalizes the edge weights by computing the normalized graph Laplacian matrix [17]. Label Spreading is similar to label propagation, except it uses affinity matrix based on normalized graph Laplacian and soft clamping across the labels. Both label propagation and label spreading often use two kernel methods.

| Kernel Type | Formula |
|---|---|
| k-nearest neighbor | $1[x' \epsilon\ kNN(x)],\ \mathrm{k} \geq 1$ |
| Radial basic function | $\exp(-\gamma \left\| x_i - x_j \right\|^2),\ \gamma > 0$ |

Generally speaking, RBF computation requires a fully connected graph in a dense matrix which may large combined with the cost of performing full matrix multiplication calculation for each iteration can take quite long running times. KNN kernel produces a memory-friendly matrix that can reduce a lot running times.

Because of the advantages of label spreading, in this paper we will use label spreading to classify SPC control chart patterns.

## 5. The proposed approach

Based on the published papers, there still exist some significant issues in designing SPC control chart pattern recognition, which if solved appropriately; it can improve not only the accuracy but also the way of training control chart pattern recognizers.

The proposed approach consists of two phases:

- Use effectively multiclass Support Vector Machine (one against one) as supervised learning techniques to classify all six types of control chart with high accuracy.
- Use Label Propagation (Label Spreading) as semi-supervised learning techniques to classify all six types of control chart with both labeled and unlabeled data.

In order to train and evaluate the performance and quality of proposed system, we have used the dataset from practical real world data [19].The dataset is basically separated into three groups training dataset, cross validation dataset and testing dataset. This dataset contains 600 samples of control charts. The number of control charts of training & cross validation group will be removed their labels gradually from 10% of data to 90% of data.

## 6. The proposal of SVM for control chart

### 6.1 SVM Design

There are many kernels that are suitable for SVM including user-defined kernel as long as it satisfies the Mercer's condition. For the sake of accuracy with large data, in this paper we have chosen SVM (one against one) with RBF kernel. For each three groups of dataset (training, validation, testing), the parameter C and gamma will be examined to get the best options among the ranges as:

$$C \in [0.006: 1]\ ;\ \gamma \in [0.0001: 0.1]$$

In this paper, we have used 30% data for training, 30% for cross validation and 40% data for testing.

### 6.2 Training SVM Model

Basically, we loop all C and $\gamma$ and try to find down the best couple of C and $\gamma$ that not only fit the training dataset but also giving the minimum error rate with cross validation dataset.

$$C_{min} = 1;\ \gamma_{min} = 0.001$$

### 6.3 SVM Validation

In order to validate how these parameters work with new dataset, we have to test this model with all 40% testing dataset (240 samples). The result has had a higher accuracy and others classifier, it has only 1 error out of 240 samples while doing with all testing dataset. The accuracy of 99.83% has been reached.

**Table 1**
Confusion matrix of recognizer with 99.83% accuracy

| | Nor | Cycle | Up-T | D-T | Up-S | D-S |
|---|---|---|---|---|---|---|
| Nor | 40 | | | | | |
| Cycle | | 40 | | | | |
| Up-T | | | 40 | | | |
| D-T | | | | 39 | | 1 |
| Up-S | | | | | 40 | |
| D-S | | | | | | 40 |

## 7. The proposal of Label Spreading for control chart

### 7.1 Label Spreading Design

In this section, we will use label propagation (label spreading) to classify all 6 types of SPC charts. Label spreading has two popular kernel functions to propagate data from labeled data to unlabeled data. One of them is RBF (Radial Basic Function), in its formula; we can see it has only one parameter is gamma($\gamma$). Another method to propagate data is KNN (k nearest neighbor), in formula, we can see it has only one parameter is k. In training phases, we have to optimize gamma and k to make model procedure the minimum error rate on testing data.

Entire dataset will be divided into small groups of datasets. Some labels of data will be removed in order to train our model in semi-supervised mode.

The percentage of unlabeled data in training dataset will be increased from 10% to90% of entire dataset so we can see how model perform with these datasets.

### 7.2 Training Label Spreading with RBF

As we mentioned before, label spreading with RBF kernel has two important parameters are gamma($\gamma$) and camping factor alpha($\propto$).

$$\propto \in [0.01: 0.5];\ \gamma \in [0.001: 1]$$

After carefully testing with data, we have the best alpha for all dataset is $\propto = 0.2$ and gamma $\gamma$ will change according to the percentage of unlabeled data in dataset. Label spreading with RBF training is quite similar to other RBF training machine learning techniques. Firstly, we train with training dataset, then validate parameters by cross validation and get the best parameters which giving the minimize error rate on not only training dataset but also the cross validation dataset. After getting the best parameters, we have to re-train label spreading model with the selected parameters and all 600 samples including unlabeled data. The result table as below:

**Table 2**
Gamma and alpha parameters with increasing of percentage of unlabeled data in training dataset.

| | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% |
|---|---|---|---|---|---|---|---|---|---|
| $\propto$ | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| $\gamma$ | 0.09 | 0.3 | 0.09 | 0.09 | 0.06 | 0.03 | 0.09 | 0.09 | 0.09 |

### 7.3 Label Spreading with RBF validation

As we can see in the figure, the accuracy is maintained quite well despite of the increasing of unlabeled data in training dataset. But if so many unlabeled data come into training dataset, around 70% of training data are unlabeled, the error rate start increasing dramatically.

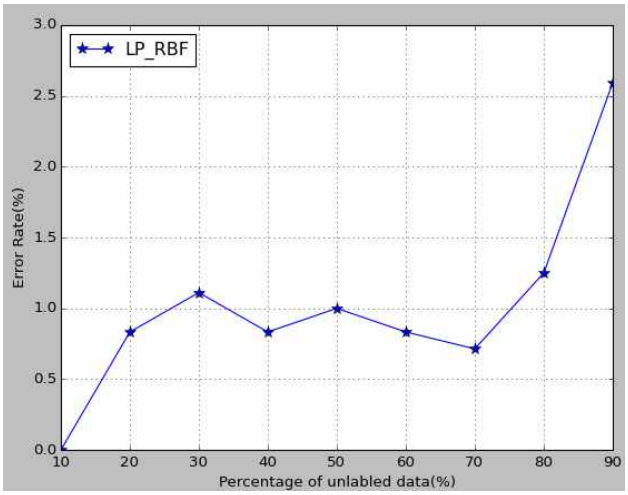


Figure 3 Error Rate on Label Spreading with RBF kernel

### 7.4 Training Label Spreading with KNN

In label spreading with KNN (k-nearest neighbor), we must optimize two parameters are camping factor alpha($\propto$) and k (number of neighbor using to classify). Because of the ability to learn unlabeled data in label spreading KNN, we always give all 600 samples for training but removed some labels from them.

$$\propto \in [0.01:1];\ k \in [2:10]$$

**Table 3**
K and alpha parameters with increasing of percentage of unlabeled data in training dataset.

| | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% |
|---|---|---|---|---|---|---|---|---|---|
| $k$ | 3 | 3 | 3 | 3 | 5 | 5 | 4 | 6 | 6 |
| $\propto$ | 1 | 0.2 | 0.6 | 0.6 | 0.1 | 0.003 | 0.009 | 0.009 | 0.2 |

### 7.5 Label Spreading with KNN validation

From the error rate plot, we can see the accuracy is low (3.2%) and maintained from 30% to 60% of unlabeled data. When more unlabeled data come into play, the percentage of unlabeled data is bigger than 60%, error rate jumps up very fast.

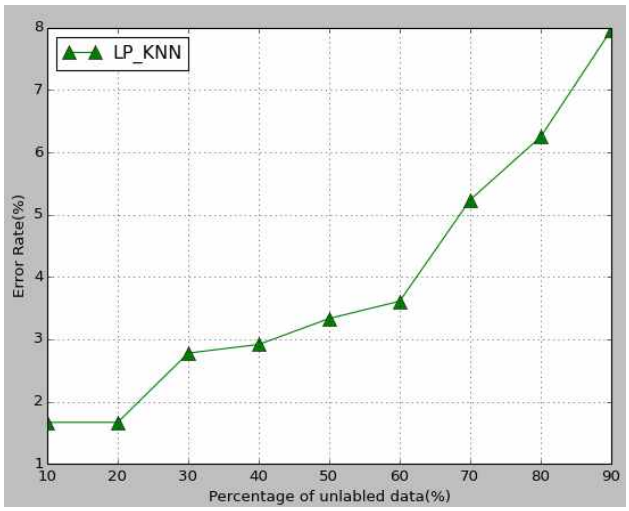


Figure 4 Error Rate on Label Spreading with KNN kernel

## 8 Interpreting results

From the results Label Spreading using RBF and KNN, what we should do if we want to know how they are different from SVM's result. Thus, we also need to train SVM in the datasets that used to train RBF and KNN of Label spreading, so we can have a good comparison of how SVM and label spreading work in different training data. The result of comparison can easily be interpreted by following plot:

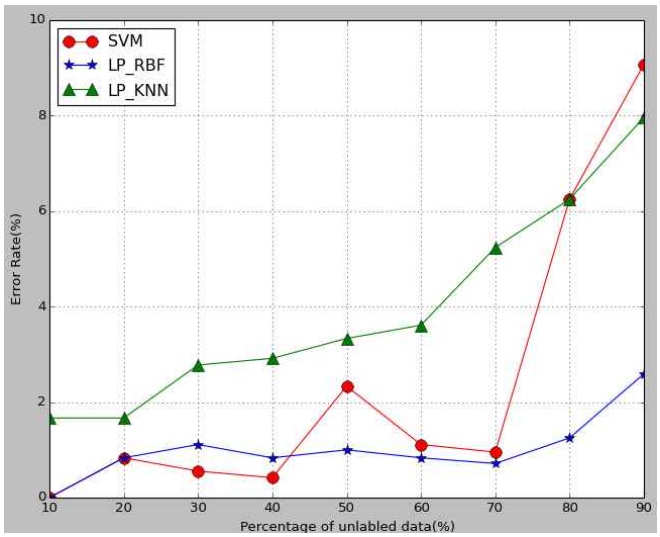


Figure 5 Comparison between SVM and label Spreading

## 9 Conclusion

Control Chart Pattern Recognition (CCPR) is very important tools to help process engineers know how well processes work. By using the corresponding assignable causes [6], process engineers can have a very good suggestion about what are the causes of variation. It can be the equipment deterioration or processes just are operated by new worker.

This paper clearly demonstrates how well SVM can work in supervised-learning and how we can train SVM with real practical datasets. This paper also proposes new approach in semi-supervised learning for recognition control chart patterns. With many labeled data, researchers should choose SVM as a good candidate to classify control chart but with small number of labeled data then Label Spreading with RBF is a quite good option.

However, RBF function is not simple computation with very large datasets, in this case, KNN outperforms RBF and SVM because of its computations are simple and only memory-friendly matrixes are needed to compute.

All SVM, label spreading implementation results reported here are developed in python and based on sklearn[18] a machine learning library in Python.

Future work of this paper can be deep learning in pre-training phases and we final can use SVM or back-propagation as a final classification layer to classify the control chart patterns.

## Reference


[1] Douglas C.Montgomery, *Introduction to Statistical Quality Control*, USA: Willey, 2005.

[2] Christopher M.Bishop, *Pattern Recognition and Machine Learning*, Berkeley, USA: Springer, 2006.

[3] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-supervised learning*, Cambridge, MA,USA:MIT Press, 2006.

[4] Mark Shewhart“Interpreting Statictical Process Control (SPC) Charts using machine learning and expert system techniques”presented at Aerospace and Electronics Conference, Dayton, OH, USA, 1992.

[5] RatnaBabuChinnam and VinayS.Kumar“Using Support Vector Machines for Recognizing Shifts in Correlated Manufacturing Processes” presented at Neural Networks, 2001. Proceedings.IJCNN '01.International Joint Conference, Washington, DC, USA, 2001.

[6] Cheng, C.S., “A neural network approach for the analysis of control chart patterns”.Int. J. Prod. Res., 1997, 35, 667–697

[7] Wang, Chih-Hsuan, Guo, Ruey-Shan, Chiang, Ming-Huang and Wong, Jehn-Yih (2008) “Decision tree based control chart pattern recognition”, International Journal of Production Research, 46:17, 4889 — 49017

[8] A. Ebrahimzadeh, V. Ranaee “Recognition of Control Chart Patterns Using Genetic Algorithm and Support Vector Machine” presented at Networked Digital Technologies, 2009. NDT '09. First International Conference, Ostrava, 2009

[9] Qinghong Le, XinghaiGao“A New ANN Model and Its Application in Pattern Recognition of Control Charts” presented at Intelligent Control and Automation, 2004. WCICA 2004.Fifth World Congress, 2004.

[10]JalelAddeh, A. Ebrahimzadeh, V. Ranaee“Control chart Pattern Recognition using adaptive back-propagation artificial neural networks and efficient features” presented at Control, Instrumentation and Automation (ICCIA), 2011 2nd International Conference, Shiraz, Iran.

[11] Alice E. Smith“X-Bar and R Control Chart Interpretation Using Neural Computing”, Int. J. Prod. Res., 1993

[12] J. Milgram and M. Cherier and R. Sabourin, "One-Against-One or One-Against-All: Which One is Better for Hanwriting Recognition with SVMs?," in 10th International Workshop on Frontiers in Handwriting Recognition, La Baule, France, October 2006.

[13] Miin-Shen Yang , Jenn-Hwai Yang “Control chart pattern recognition using semi-supervised learning”proceeding ACS'07 Proceedings of the 7th Conference on 7th WSEAS International Conference on Applied Computer Science - Volume 7Pages 272-276, Wisconsin, USA, 2007.

[14] Fei Wang feiw, Changshui Zhang“Label Propagation Through Linear Neighborhoods” Knowledge and Data Engineering, IEEE Transactions on Volume:20 , Issue: 1 , Jan. 2008.

[15] Yuan-Dong LAN, Hui-Fang DENG, "Use Label Propagation with Active Learning Algorithm for Data Classification", JCIT: , Vol. 7, No. 7, pp. 104 ~ 113, 2012.

[16] Xiaojin Zhu, ZoubinGhahramani“Learning from Labeled and Unlabeled Data with Label Propagation” CMU CALD tech report CMU-CALD-02-107, 2002.

[17]D. Zhou, O. Bousquet, T.N. Lal, J. Weston and B. Schölkopf. Learning with Local and Global Consistency. Advances in Neural Information Processing Systems (NIPS) 16, 321-328. (Eds.) S. Thrun, L. Saul and B. Schölkopf, MIT Press, Cambridge, MA, 2004

[18] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[19]https://archive.ics.uci.edu/ml/datasets/Synthetic+Control+Chart+Time+Series