A Practical Guide to Weighted Support Vector Machine Toolbox for Control Chart Pattern Recognition

Talayeh Razzaghi, Petros Xanthopoulos
Department of Industrial Engineering and Management Systems
University of Central Florida
Orlando, Florida

The Support vector machines (SVM) is a popular supervised learning algorithm proposed first by Vapnik (2000). The weighted support vector machine (WSVM) is a SVM adaptation to the cost sensitive learning framework. This supervised learning technique has been successfully applicable for imbalanced classification. In this guide, we propose a simple procedure which obtains reasonable results for control chart pattern recognition (CCPR). Although in this guide, WSVM has been proposed for automated process monitoring and early fault diagnosis purposes, but it is applicable for other fields with minor changes.

For the evaluation of WSVM algorithm, we have developed a toolbox in MATLAB, termed WSVMToolbox. It features:

- Implementations of SVM (Vapnik, 2000) and WSVM (Veropoulos et al., 1999) techniques for time series data
- A guide to generate simulated data for different abnormal patterns, routines to pre-process data sets
- An experiment result format and functions for calculation of Sensitivity, Specificity, Accuracy, and G-mean for imbalanced classification

We note that experiments on both SVM and WSVM are conducted with LIBSVM-3.12 and LIBSVM-weights-3.12 (Chang & Lin, 2011). The whole script is developed in MATLAB and LIBSVM is interfaced in it. Therefore before the implementation of this toolbox, we suggest the user to download LIBSVM from http://www.csie.ntu.edu.tw/ cjlin/libsvm.

1 Proposed Procedure

The WSVMToolbox conducts the following procedure:

- Generate data in the format of time series with specific window lengths
- Perform data preprocessing
- Use the RBF kernel (For reasons see Xanthopoulos & Razzaghi (2014))

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma \geq 0. \tag{1}$$

- Use model selection techniques to find the best parameter C and γ
- Implement the optimal or near optimal parameters for C and γ to train the training dataset
- Report G-mean, Sensitivity, Specificity, and Accuracy for test dataset

We explain this procedure in a more detail in the next sections.

1.1 Data Generation

Data based on different normal and abnormal patterns are generated using GenData.m and GenDataMulti.m functions for binary and multiclass classification (for the mathematical models see Xanthopoulos & Razzaghi (2014)). For binary classification, the user should first select the abnormal data type and set the input parameters. We provide a table for abnormal parameter type as following,

Table 1: Abnomal pattern types symbols used in WSVMToolbox

Abnormal Pattern	**Symbol**
Up trend	1
Down trend	2
Up Shift	3
Down shift	4
Cyclic	5
Systematic	6
Stratification	7

Other input parameters are imbalanced ratio(r), window length(w), parameter of abnormal pattern (t) for binary classification. The imbalanced ratio (r) in the code is determined as,

Table 2: Imbalanced ratio symbols used in WSVMToolbox

r	**imbalanced ratio**
0	%50
5	%55
10	%60
15	%65
20	%70
25	%75
30	%80
35	%85
40	%90
45	%95

The input parameters for muticlass classification is slightly different from binary classification. They consist of all abnormal parameters, the size of minority(n) and majority(m) class. The abnormal parameters are given using muticlass.mat. Furthermore, we can select window w and abnormal parameter values from Table 3. All input parameters should be given in Main.m file for both binary and muticlass classification.

Table 3: Summary of parameter range for computational experiments (Xanthopoulos & Razzaghi, 2014)

Name	**Symbol**	**Range**
Window length	w	$[10, 100]$
Process mean (all patterns)	μ	0
Standard deviation of normal process	σ	1
Slope (Up/Down trend pattern)	λ	$[0.005\sigma, 0.605\sigma]$
Shift (Up/Down shift pattern)	ω	$[0.005\sigma, 1.805\sigma]$
Standard deviation (Stratification pattern)	$\acute{\varepsilon_t}$	$[0.005\sigma, 0.8\sigma]$
Cyclic parameter (Cyclic pattern)	α	$[0.005\sigma, 1.805\sigma]$
Systematic parameter (Systematic pattern)	k	$[0.005\sigma, 1.805\sigma]$

1.2 Data Preprocessing

Data preprocessing and scaling before applying any data mining algorithm is the key step. Scaling makes all features in the same numeric ranges. We suggest normalize all data prior to classification, so that they have zero mean and unitary standard deviation (zscore() function in MATLAB is used).

This step can be found in the first line of wsvmmodel.m and wsvmmodelmulti.m for binary and muticlass classification respectively.

1.3 Model Selection

The SVM and WSVM algorithms have certain parameters that need to be tuned during the training phase: C and γ (RBF kernel). For this, we use the "grid search" model selection using cross-validation. We use exponentially growing sequences of C and γ to identify good parameters, such as C $\in \{2^{-5}, 2^{-4},...,2^{15}\}$, $\gamma \in \{2^{-15}, 2^{-13}, ..., 2^{3}\}$. These parameter sequence sets are also suggested in Chang & Lin (2011).

1.4 WSVM Training and Testing

The learning process is conducted in 10-fold cross validation loop using wsvmmodel.m (binary) and wsvmmodelmulti.m (multiclass classification). For cross validation purposes, 90% of the data is used for training and the rest 10% is used for testing. The output for binary classification is sensitivity, specificity, accuracy, and G-mean. For multi-class classification, the output is the accuracy and confusion matrix table. The diagonal elements of confusion matrix table show the accurate classification percentage.

In order to run the main.m, the user should put all files including LIBSVM and LIBSVM-weights .mex files in the same folder as WSVMToolbox.

2 Citation

The users of WSVMToolbox are encouraged to use the following information to cite it:

Xanthopoulos, P., & Razzaghi, T. (2014). A weighted support vector machine method for control chart pattern recognition. Computers & Industrial Engineering, 70, 134-149.

References

Chang, C., & Lin, C. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 27.

Vapnik, V. (2000). *The nature of statistical learning theory*. Springer-Verlag New York Inc.

Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the international joint conference on artificial intelligence*, vol. 1999, (pp. 55–60). Citeseer.

Xanthopoulos, P., & Razzaghi, T. (2014). A weighted support vector machine method for control chart pattern recognition. *Computers & Industrial Engineering*, *70*, 134–149.