

Support Vector Machine for Classify Dynamic Human/Vehicle Shapes

Razali M.T¹ and Adznan Jantan¹

¹ Dept. of Mechatronic & Robotic Engineering
Faculty of Electrical & Electronic Engineering
Universiti Tun Hussein Onn Malaysia
86400 Parit Raja Batu Pahat, Johor

² Dept. of Computer & Communication Engineering
Faculty of Engineering
Universiti Putra Malaysia
43400 Serdang, Selangor

¹mdrazali@uthm.edu.my, ²adznan@eng.upm.edu.my

Abstract

Currently Support Vector Machines (SVM) became subject of interest because of its ability to give high classification performance in a wide area of application. Most of the classifier model especially based on supervised learning involve complicated learning model and yet the performance sometimes worst. This paper proposes a SVM model to classify between human and vehicle shapes in various pose. SVM classify data by first construct a decision surface that maximizes the margin between the data. For testing new data, SVM will calculate the sign signifying where this new data reside in the constructed decision surface. The developed model will be used to classify an outdoor scene of human and vehicle shapes in dynamic pose. Results of the experiments showed a satisfied performance with the proposed approach.

1. INTRODUCTION

Object recognition and classification is the sequence of steps that must be performed after appropriate features have been detected. Based on the detected features in an image, one must formulate hypotheses about possible objects in the image. These hypotheses must be verified using models of objects. Not all object recognition techniques require strong hypothesis formation and verification steps. Most recognition strategies have evolved to combine these steps in varying amounts. In the following, a few basic recognition strategies used for recognition objects in different situations are reviewed.

One of the earliest approaches to object recognition is based on template matching. This method based on the concept of finding the similarity between two entities such as points, curves, or shapes

of the same type. In template matching, a template typically a 2D shape of the pattern to be recognized is available. The pattern to be recognized is matched against the stored template while taking into account all allowable pose such as translation and rotation and scale changes. The similarity measurement is normally defined as a metric distance. Various distance measures have been exploited in object include city block distance, Euclidean distance, cosine distance, histogram intersection distance, χ^2 statistics distance, quadratic distance, and Mahalanobis distance [1].

Nearest Neighbourhood (NN) classifier [2] is based on the fact that if two elements are close in their representation space, they probably belong to the same class. The nearest neighbour classifier is useful when the number of training samples is small. Each form is represented by a feature vector which represents variance of motion and variance of compactness in a multi-dimensional representation space. In order to identify the class of a form, the form class of its nearest points in the representation space is significant information. A strict rule is to impose that the nearest neighbours belongs to the same class to take a decision. This rule reduces the number of errors, but it leads to many rejects. It can be smoothed by weighting the voting of each neighbour according to its rank or distance.

Neural networks can be viewed as a massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. The main characteristics of neural networks are that they have the ability to learn complex nonlinear input-output relationships using sequential training procedures and adapt themselves to the data. During training, many instances of the objects to be further recognized are input to the net. If the training set is carefully selected to represent all objects encountered later during the recognition phase, then

the net may learn the classification boundaries in its feature space. The most commonly used family of neural networks for recognizing system is the feed-forward network, which includes Multilayer Perceptron (MLP) networks [3].

Support vector machine (SVM) are receiving increasing attention these day and have achieved very good accuracy in pattern recognition. In a binary classification, SVM try to find a hyperplane to separate the data into two classes. In the case in which all the data are well separated, the margin is defined as two times the distance between the hyperplane with the largest margin, which provides good generalization ability based on Vapnik's VC dimension theory [4]. To increase the classification ability, SVMs first map the data into a higher dimension feature space with kernel function, then use a hyperplane in that feature space to separate the data. For testing a new unseen dataset, SVM calculate the distance of this sample from the decision surface, and produced a sign signifying which side of the surface they reside. SVMs have demonstrated good generalization performance in face recognition [5], plankton recognition [6], gene classification [7], Chinese character classification [8], and text categorization [9].

Recognition techniques are mandatory to add to monitoring system to understand the behaviour of dynamic events in complex scenarios. Template matching and nearest neighbour method capable to classify the object with high accuracy if the data is separable, but if the data is in non separable case, neural network and support vector machine (SVM) perhaps the best method for this purpose. SVM is a new pattern recognition technique based on simple and elegance theory. It simplicity for training a highly non separable data compared to neural networks make this method receive increasing intention nowadays. It's good highly accuracy in text recognition, face detection, and surveillance system make it worthwhile to explore the ability of this approach to be implemented in the system.

The paper is organized as follows. Section 2 presents a fundamental of SVM and how to implement it during training and testing session. Section 3 describes in detail how to determine the best kernel and capacity control values. The result is presented in section 4 and summary and future researches are presented in section 5.

2. FUNDAMENTAL OF SVM

The SVM learning algorithm is based on finding optimal separating hyperplane for the two classes of data. The support vector classifier is based on the class of hyperplanes:

$$\langle w \cdot x \rangle + b = 0 \tag{1}$$

Corresponding to the decision functions

$$f(x) = \text{sign}(\langle w \cdot x \rangle + b) \tag{2}$$

Where w is the normal vector that perpendicular to the hyperplane, x is the input data set, $f(x)$ is the output data set and b is the bias values.

In Figure 1, the optimal hyperplane (solid line) is defined as the one the maximal margin of separation between the two classes. It can be uniquely constructed by solving a constrained quadratic optimization problem. The output from this solution is the support vectors (SV) (inside dot circle near dash line) that will be used for classification.

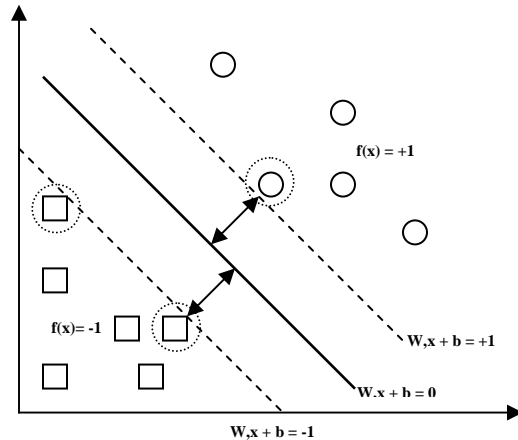


Figure 1: SVM classification example for separable two classes.

To increase the classification ability for non separating data, SVM map the data into a higher dimension feature space with kernel function $K(x,y)$ where the data has much more possibility to be separated, then use a hyperplane in this feature space to separate the data as shown in Figure 2.

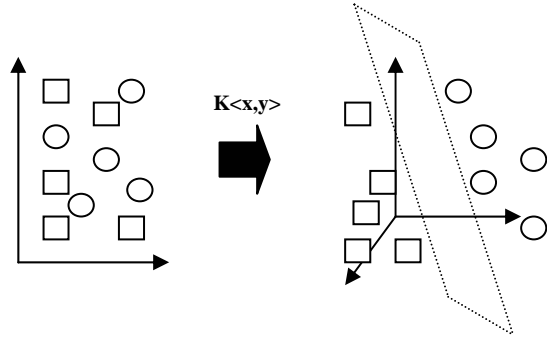


Figure 2: SVM map data into higher dimension feature space in which the data have a much greater chance of being linearly separable.

SVM construct the maximum hyperplane by solve optimization problem defined by:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^N \alpha_i \quad (3)$$

With constraint:

$$\sum_{i=1}^N \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C \quad i = 1, \dots, N$$

Where

$K(x_i, x_j)$ = Kernel function.
 α_i, α_j = Lagrange multipliers
 y_i, y_j = Training output sample
 x_i, x_j = Training input sample
 N = Number of training sample
 C = Capacity control

The optimization processes will produce a Lagrange multipliers (α) and bias (b) values that will be used for classification. For predicting output ($Y_{predict}$) using new data input (X_{test}), equation below will be used :

$$Y_{predict} = \sum_{i=1}^N \sum_{j=1}^M \alpha y_{train} K(x_{test}, x_{train}) + b \quad (4)$$

Where

$K(x_{test}, x_{train})$ = Kernel function.
 α = alpha value for SV
 y_{train} = Training output value
 x_{train} = Training input value
 x_{test} = Testing input value
 $Y_{predict}$ = Predicted output value
 N = Number of training sample
 M = Data dimension

3. KERNEL AND CAPACITY CONTROL (C) SETTING

In order to get a good performance of the SVM classifier, there are two points that must be considered carefully which are kernel function and capacity control (C). Kernel function $K(x_i, x_j)$ is used to map the nonlinear input into a higher dimensional feature space, in which the learning capability of the machine is significantly increased. This kernel function will enable the operations to be performed in the input space rather than in the potentially high dimensional feature space. Hence, the inner product does not need to be calculated in the feature space. In this paper three common kernel functions which are Linear, Polynomial and Gaussian Radial Basis Function (GRBF) were tested. These kernel function capabilities were compared and the one that gives the higher accuracy in training and testing will be used in the real time system.

The detail algorithm for each kernel is shown below:

1. Linear Kernel

$$K(x, x') = \langle x, x' \rangle \quad (5)$$

2. Polynomial Kernel

$$K(x, x') = (\langle x, x' \rangle + 1)^d \quad (6)$$

3. Gaussian Radial Basis Function Kernel

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2d^2}\right) \quad (7)$$

Choosing a suitable parameter for capacity control (C) is crucial for good performance during training. Generally, C controls between how hard to have a large margin and how hard to avoid training error. As C getting large, the solution with minimum training error is acquired and increment in C beyond this point will have no effect. This point cannot be overemphasized because training SVM with an excessively large value of C tends to increase the required training time by several orders of magnitude. On the other hand, if C is too small, it will manifest itself in a solution with no unbounded support vectors, and therefore no ability to determine the bias (b) term [10].

For simplification, the step for classification using SVM can be summarized as below:

1. Prepare the pattern of input-output matrix.
2. Choose the best kernel function.
3. Select the parameter of the kernel function and the capacity control (C).
4. Execute the training algorithm using quadratic programming optimization to obtain the values of α .
5. Classified unseen using α value and the support vectors.

4. RESULT

For real time mode application, the selected kernel function must attain a high performance in training and testing with a minimum number of support vector (SV). Figure 3 up to Figure 8 show the performance of training and testing with number of support vectors, for several kernel functions that based on training and testing data. The training data acquired from MIT database [11] consists of 386 images (212 of human images and 174 of vehicle images), while the testing data consists of 114 images taken from the video sequences of outdoor scene in [12]. The input feature consist of the selected best five HU moment values in [12] whereas the output classes is assigned to +1 for human and -1 for vehicle.

From simulations, value of C that is greater than 100 produced a static percentage performance (training and testing) and number of support vector (SV). Thus, only a range of C that lies from 1 to 100 has been analysed in detail. Also, only the polynomial function with a power of 2 and 3 is considered. As for Gaussian Radial Basis Function (GRBF), apart from C the width parameter (d) is also tested simultaneously. For this reason, a cross intersection between these two parameters (C and $d \in \{0.1 \text{ to } 1\}$) consist of 1000 combinations, equipped to the kernel function have been analyzed.

Result for Linear Kernel Analysis

Figure 3 shows the result for linear kernel analysis. From the graph, linear kernel function shows that with the increment of C , the performance for both training and testing fluctuate between minimum of 54.92%(training) and 76.32%(testing) to maximum of 99.48%(training) and 84.21%(testing) , while the number of support vectors decreased constantly. From the graph analysis, $C = 63$ gave the highest performance of training (99.48 %) and testing

(84.21%) with lowest magnitude of support vector capacity (3.11%).

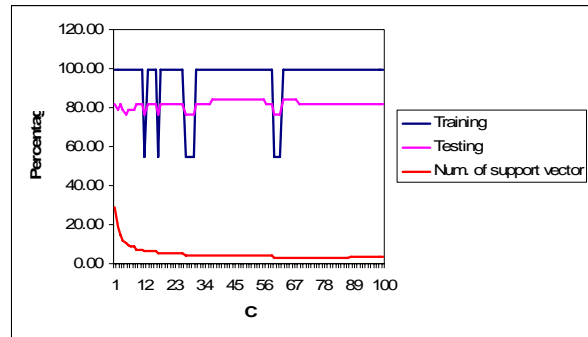


Figure 3. Percentage performance for testing, training and number of SV versus C for linear kernel.

Result for Polynomial Kernel Analysis

Figure 4 and 5 show the analysis result for polynomial power of two and three respectively. For polynomial kernel of power of two and three, the performance for training achieved 100% as the C value reached above 74 (polynomial 2) and 34 (polynomial 3). The analyzed testing performance and capacity of support vectors beyond these two C values gave 86.84% of testing performance with 2.59% of support vectors for $C=76$ (polynomial 2), and 84.21% of testing performance with 2.59% of support vectors for $C=34$ (polynomial 3).

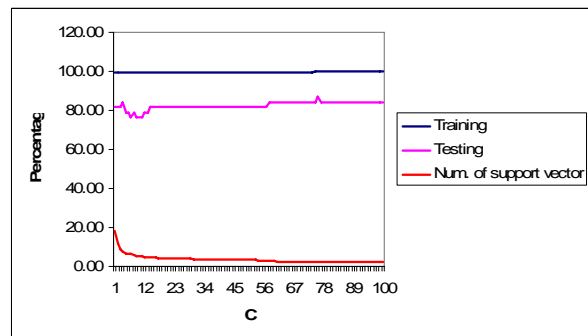


Figure 4. Percentage performance for testing, training and number of SV versus C for polynomial power of 2.

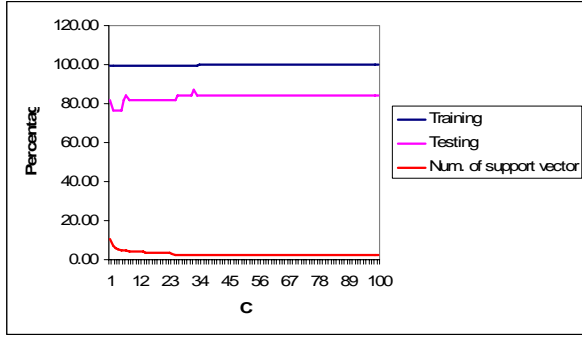


Figure 5. Percentage performance for testing, training and number of SV versus C for polynomial power of 3.

Result for GRBF Kernel Analysis

Figure 6, 7, and 8 show the analysis result for Gaussian Radial Basis Function (GRBF) for training, testing, and number of support vector (SV) respectively. GRBF shows that the highest performance in training and testing can be achieved when the combination of small value of α and C were supplied, as α increased, the bigger C value need to be selected in order to give the best performance. From Figures, it showed that d range from 0.8 to 1 does not achieved 100% performance in training, so it has been discarded. For the rest of the width parameters (0.1 to 0.7), 100% performances in training and 86.84 performances in testing is achieved. The highest of SV capacity obtain by these parameters is 3.63% while the lowest one is 2.07%.

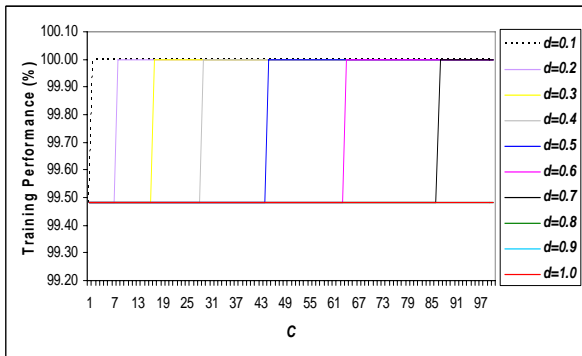


Figure 6. Training performance versus C for GRBF width (d) of 0.1 to 1.

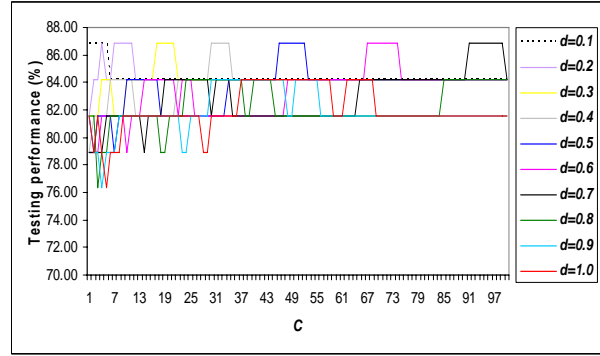


Figure 7. Testing performance versus C for GRBF width of 0.1 to 1.

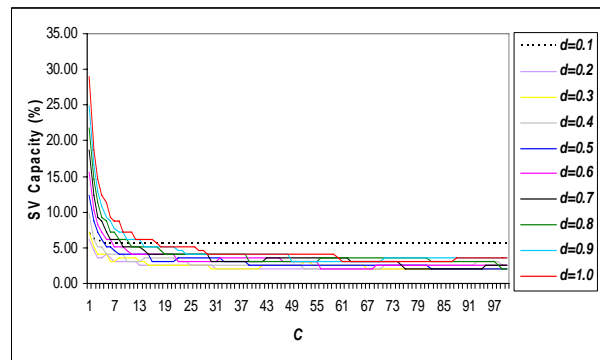


Figure 8. Support vector (SV) capacity versus C for GRBF width of 0.1 to 1.

The results for the best cross intersection parameters between C and d from Figure 3 to 8 can be summarized in Table 1 below.

Table 1. Performance for several kernel functions for classifies training and testing data.

Kernel	d	C	Training (%)	Testing (%)	SV Capacity (%)
Linear	-	63	99.48	84.21	3.11
Poly.	-	76	100	86.84	2.59
2	-	34	100	84.21	2.59
Poly.	0.1	3	100	86.84	3.63
3	0.2	8	100	86.84	3.63
GRBF	0.3	17	100	86.84	2.59
	0.4	33	100	86.84	3.11
	0.5	46	100	86.84	2.59
	0.6	67	100	86.84	2.07
	0.7	91	100	86.84	2.07
	0.8	85	99.48	84.21	3.11
	0.9	50	99.48	84.21	3.11
	1	62	99.48	84.21	3.11

From Table 1, it can be clearly seen that the entire selected kernel parameter excluding linear kernel and *GRBF* with d larger than 0.8, gave a 100% performance in training and 86.84% performance in testing. However, as for the support vector capacity, *GRBF* with $d=0.6$ ($C=67$) and $d=0.7$ ($C=91$) gave the lowest capacity (2.07%) though it posses the same performances as others. However, since $d=0.6$ gave a larger margin (0.147696) than $d=0.7$ (0.126733), thus, it will be used in the real time mode.

5. CONCLUSIONS

In this paper, a system to classify between human and vehicle using Support Vector Machine (SVM) classifier is discussed under dynamic pose. From the analyses, SVM training session via Gaussian Radial Basis Function (GRBF) with $C=67$ and $d=0.6$ gave the highest performance in training and testing, with only need eight of support vector (SV). The proposed SVM model gave a good result in 100% during training and 86.84% during testing.

In future, the system can be improve by extend the classification framework to other object classes like groups of people or sub classes like cars, vans, trucks, and motorcycles. A hierarchical multi class SVM probably best for this purpose by training the data using one versus all method.

REFERENCES

1. Dengsheng Zhang, 2002. Image Retrieval Based on Shape, Ph.D Thesis, Monash University.
2. Zhou Q and Aggarwal J. Tracking and Classifying Moving Objects from Video. *In Proc. 2nd IEEE Int'l Workshop on Performance Evaluation of Tracking in Surveillance* Dec 9 2001. Kauai, Hawaii.
3. Daniel Toth and Til Aach. Detection and Recognition of Moving Objects Using Statistical Motion Detection and Fourier Descriptors. *12th International Conference on Image Analysis and Processing* 2003. Washington, DC.
4. Burges, CJC. A tutorial on support vector machines for pattern recognition. *Data Mining Knowledge Discovery* 1998; 29(2):121-167.
5. Osuna E, Freund R, and Girosi F. Training support vector machines: An application to face detection. *Proc. Computer Vision and Pattern Recognition* 1997; Puerto Rico:130-136.
6. Lou T, Kramer K, Goldgof D, Hall L, Sampson S, Remsen A, Hopkins T. Learning to recognize plankton. *IEEE International Conference on Systems, Man & Cybernetics* October 2003; Washington, D.C: 888-893.
7. Zhang Qizhong and Zhejiang. Gene Selection and Classification Based on Expression Data. *In Proc. IEEE/ICME International Conference on Complex Medical Engineering (CME 2007)* May 23-27 2007. Beijing
8. Liang Zhao and Takagi N. An Application of Support Vector Machine to Chinese Character Classification Problem. *In Proc. Of IEEE International Conference on System, Man and Cybernetics (ISIS 2007)* October 7-10 2007. Montreal.
9. Dumais, ST, Platt J, Heckerman D, and Sahami M. Inductive learning algorithms and representations for text categorization. *Proc. ACM-Conf. Information and Knowledge Management (CIKM98)* Nov 1998. Bethesda, Maryland; 148-155.53
10. Rifkin, RM.,2002. Everything Old is New Again: A Fresh Look at Historical Approaches in Machine Learning, Ph.D Thesis, Massachusetts Institute of Technology, Boston USA.
11. <http://cbcl.mit.edu/software-datasets/>
12. Razali M.T and Adznan B.J. Detection and Classification of Moving Objects for Smart Vision Sensor. *In Proc. of 2nd IEEE International Conference on Information & Communication Technologies: From Theory to Application (ICCTA -06)* April 24-28 2006. Umayyad Palace, Damascus Syria.