Insect Detection using SVM Techniques of Image Processing

Dr. Suresh M B
Professor & Head, Dept. of ISE,
East West Institute of Technology
Bangalore, INDIA

Aakash
Undergraduate Student, Department of ISE,
East West Institute of Technology
Bangalore, INDIA

Abstract:**A Multi-class Support Vector Machine classifier is presented in this paper, along with its application to insect detection and classification. In the machine learning group, Support Vector Machines (SVM) are a well-known approach for binary classification problems. MCSVMs (Multi-class SVMs) are typically generated by combining multiple binary SVMsThe aim of this research is to demonstrate: first, the robustness of various types of kernels for Multi-class SVM classifiers, second, a comparison of different Multi-class SVM constructing methods, such as One-Against-One and One-Against-All, and finally, a comparison of Multi-class SVM classifier accuracy to Adaboost and Decision Tree. One-Against-All Support Vector Machines (OAASVM) outperform One-Against-One Support Vector Machines (OAOSVM) with polynomial kernels, according to simulation performance. On insect datasets from the UCI machine learning dataset, OAASVM outperforms Adaboost and Decision Tree classifiers in terms of accuracy.**

Keywords: Support vector machine; SVM Classification; Boosting; Multi-class SVM;

I. INTRODUCTION

Plant diseases have wreaked havoc, as they can significantly reduce the quality and quantity of agricultural products. It causes a lot of harm. Crop loss due to pests in India is estimated to be between 10 and 30% per year, with insect pests accounting for 26% of the total.

Vapnik [1] suggested Support Vector Machines (SVMs), which are a group of similar supervised learning methods for classification, regression, and ranking. SVMs are classification prediction tools that employ Machine Learning theory as a principled and extremely reliable approach for maximizing detection and classification predictive accuracy. SVMs are techniques that use a hypothesis space of linear separators in a high-dimensional feature space and are trained with an optimization theory learning algorithm that allows a learning bias derived from statistical learning theory [1, 2].To design separating hyperplanes for classification problems, the SVM technique was developed. SVM was generalized in the 1990s for constructing nonlinear separating functions and approximating real-valued functions. Text categorization, character recognition, bioinformatics, bankruptcy prediction, spam categorization, and other applications of SVMs are only a few examples [5].

The remainder of the paper is laid out as follows. The classification of SVMs is introduced in section II, along with a formulation of a One-class SVM. Section III shows how binary SVMs are used to solve the Multi-class SVMs problem using various methods. The experimental results are presented in Section IV to demonstrate the superiority of Multi-class SVM over other Decision Tree classifiers, and the conclusion is presented in Section IV.

II. METHODOLOGY

The concept of decision hyperplanes, which define decision boundaries in input space or high-dimensional feature space, underpins SVM classification. From a collection of named training datasets, SVM creates linear functions (hyperplanes in either input or feature space). This hyperplane will attempt to separate positive and negative samples. The linear separator is usually designed to have the greatest distance between the hyperplane and the nearest negative and positive samples. This, intuitively, leads to accurate classification for training data that is similar to, but not identical to, testing data.

During the training process, SVM uses a data matrix as input data and marks of sample as belonging to a specific class (positive) or not (negative) (negative). Each sample in the matrix is treated by SVM as a row in an input space or high-dimensional feature space, with the number of attributes indicating the space's dimensionality. The best hyperplane that separates each positive and negative training sample is determined by the SVM learning algorithm. The qualified SVM can be used to make predictions about test samples (which are new to the class).

SVM solves nonlinear problems by transforming an n-dimensional input space into a high-dimensional function space. Finally, a linear classifier is built in this high-dimensional feature space that functions as a nonlinear classifier in input space. The following [1, 3, 4, 6, 7, 8] introduce the majority of mathematical principles as context materials for designing Multi-class SVM.

A. *Linear Separable SVMs Classifier*

Consider the issue of binary classification with N training samples (data). Each sample is represented by a tuple (xi, yi) and (I = 1, 2,..., N), with xi=(xi1, xi2,..., xin) corresponding to the ith sample's attribute collection. Traditionally, yi -1, 1 is used as the class label. A linear classifier's (separator's) decision boundary can be written as follows:

$$\boldsymbol{w}\mathbf{T}\boldsymbol{x}+ b = 0, \tag{1}$$

where w denotes the weight vector and b denotes the bias expression. Although there are numerous linear separators, the purpose of SVM design is to establish a decision boundary that is as far away from any data point as possible. The margin of the classifier is determined by the distance between the decision boundary and the nearest data point. Because of this design technique, the decision boundary for an SVM is completely defined by a (usually small) portion of the data points that define the separator's location. Help vectors are the names given to these points. The margin and support vectors for a two-class problem are shown in Fig 1.

If the training data can be separated linearly, there is a pair (w, b) that says:

$$\boldsymbol{w}\mathbf{T}\boldsymbol{x}i + b \geq 1 \text{ if } yi = 1, \qquad (2)$$

$$\boldsymbol{w}\mathbf{T}\boldsymbol{x}i + b \leq -1 \text{ if } yi = -1, \qquad (3)$$

The linear classifier is defined as:

$$f(\boldsymbol{x}) = \text{sign}\,(\boldsymbol{w}\mathbf{T}\boldsymbol{x} + b). \qquad (4)$$

The functional margin of the ith sample xi with respect to a hyperplane (w, b) is defined as follows for a given dataset and decision hyperplane:

$$©_i = yi(\boldsymbol{w}\mathbf{T}\boldsymbol{x}i + b). \qquad (5)$$

The functional margin of a decision boundary dataset is then twice that of any sample in the dataset with the smallest functional margin (the factor of 2 comes from computing across the total width of the margin, as in Fig. 1). The shortest distance between a point and a hyperplane is considered to be perpendicular to the plane, and thus parallel to w. w/‖w‖ is a unit vector in this direction. The geometric margin, as shown in Fig. 1 by, is the maximum width of the band that can be equipped for separating the support vectors of the two groups. The distance between any xi sample and the separator equals:

$$\bar{\gamma} = y_i\,(\boldsymbol{w}^{\mathbf{T}}\boldsymbol{x}_i\ +\ b)/\ \|\boldsymbol{w}\|\ =\ 1/\|\boldsymbol{w}\| \qquad (6)$$

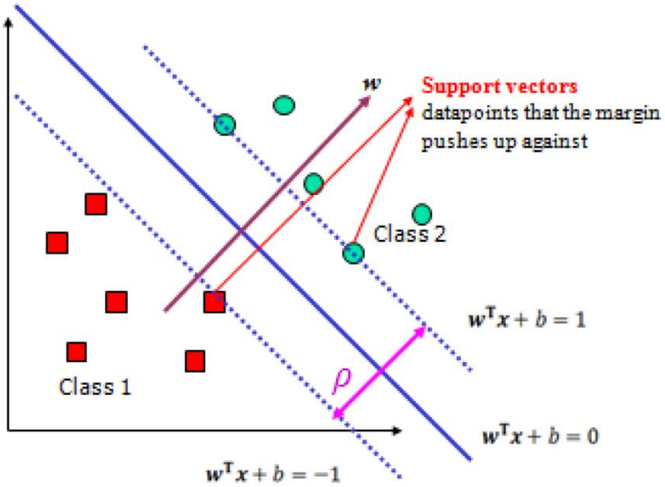

Fig 1: Decision Boundary and Margin of SVM

The aim of designing a linear separator is to optimize the geometric margin (6) in order to find the best w and b, which can be expressed as follows:

$\rho = 2\,/\|\boldsymbol{w}\|$ is maximized,

For all (x_i, y_i), $i=1, \ldots, N$:s.t.$y_i(\mathbf{w^T x}_i + b) \geq 1$. (7)

It's a convex quadratic optimization problem with linear constraints and a quadratic function. Quadratic optimization problems are a common type of mathematical optimization problem, and there are numerous algorithms for solving them. Normal quadratic programming (QP)libraries can be used to create SVM classifiers. The problem described above can be rephrased as a minimization:

$\boldsymbol{\Phi}(\boldsymbol{w}) = \|\boldsymbol{w}\| = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$is minimized,

For all $(\boldsymbol{x}_i, y_i)$, $i=1,\ldots, N$: s.t.$y_i(\mathbf{w^T x}_i + b) \geq 1$. (8)

Constructing a dual problem inwhich a Lagrange multiplier I is connected to each inequality constraint (yi (wTxi + b) ≥ 1) in the primal problem is part of the solution:

Find α1, …, αN such that it maximises with respect to under the following conditions:

$$\sum_{i=1}^{N} y_i\alpha_i = 0 \quad \text{s. t.} \quad \alpha_i \leq 0 \qquad \text{for} \quad i = 1,2, \ldots,N. \qquad (9)$$

For I = 1, 2..., N, the solution of the dual problem I must satisfy the condition αi{yi (wTxi - b) -1)} = 0 for i =1, 2, …, N.

The following is the solution to the primal:

$$\text{N}w = \sum \alpha_i y_i x_i \qquad (10)$$

For data samples that are not support vectors, the majority of I are zero in the solution above. Any non-zero I indicates that the corresponding xi is a support vector. As a result, the classification function is as follows:

$$f(\text{x}) = \Sigma\alpha_i y_i \text{x}^{\text{T}}\text{x} + \text{b}. \qquad (11)$$

III. MULTI-CLASS SVMS (MCSVM)

A. One-Against-All Support vector SVMs Classifier

M binary SVM classifiers are built in the case of M-class problems (M >2). The ith SVM is equipped to mark samples in the ith class as positive samples while the rest are classified as negative samples. A test sample is collected from all M SVMs and labelled according to the maximum performance among the M classifiers in the recognition stage.

B. One-Against-One Support Vector Machine(OAOSVM)

OAOSVM creates binary classifiers for M (M -1)/2 classes using all binary pair-wise combinations of the M classes. Of classifier is qualified by using positive examples from the first class and negative examples from the second class. The Max Wins algorithm is used to combine these classifiers. It chooses the next class based on the votes of the majority of the classifiers. The number of examples used to train each of the OAOSVM classifiers is smaller, with examples from only two of the M classes taken into account.

IV. EXPERIMENTAL RESULTS

A. Effect of Kernel Vector Machine

The kernel function used to convert data from input space to a higher dimensional feature space has a significant impact on SVM efficiency. Except for satisfactory simulation study results, there are no definite rules for this collection.

The results of MCSVM with the three kernel functions are shown in Table I. In our experiment, the degree of the polynomial kernel is 3, and the trade-off (between slack variables and regularization parameter) is set to C = 60. This number (60) was arrived at after a series of tests with various values. Despite the higher precision of the polynomial kernel (see Table I), the calculation time for classifying the samples in the RBF kernel is half that of the polynomial.

TABLE I. COMPARISION OF THREE DIFFERENT KERNELS

Kernel	c	Accuracy		Computation time	
		Training	Test	Training	Test
Linear	10	93.8%	92.5%	58.8s	52.9s
	40	94.1%	93.8%	92.2s	93.3s
	60	97.2%	96.4%	40.74s	40.21s
	80	93.4%	92.9%	123.9s	134.6s
Polynomial	10	95.4%	94.2%	416.7s	413.6s
	40	96.6%	95.1%	656.5s	620s
	60	99.1%	96.9%	139.54s	134.5s
	80	97.3%	95.2%	711.6s	783.7s
RBF	10	92.4%	93.1%	152.9s	155.8s
	40	93.1%	92.2%	132.3s	139.1s
	60	95.5%	96.1%	50.48s	79.49s
	80	92.3%	92%	186s	187.7s

B. Effect of Classifier Selection

We compared the classification results of the MCSVM and two other classifiers, Decision Tree and Adaboost, to check the efficacy and robustness of the MCSVM classification approach. Table III summarizes the findings. The accuracy of MCSVM is clearly superior to that of other classifiers, as shown in Table II.

TABLE II.AVERAGE CLLASIFICATION ACURACY RESULT OFTHREE ALGORITHMS

Classifier	Accuracy		Computation time	
	Training	Test	Training	Test
MCSVM	99.1%	96.9%	13.82s	14.6s
Decision Tree	95%	94.1%	0.11s	0.22s
Adaboost	96.4%	95.5%	0.29s	0.39s

CONCLUSION

The specifics of the Multi-class SVM architecture were discussed in this paper, as well as experimented hypothyroid dataset classification results. Using the MCSVM classifier with various types of kernels, comparative results for hypothyroid classification have been given. The experimental results show that MCSVM outperforms Adaboost and Decision Tree in terms of efficiency and accuracy. It has also been shown that the accuracy of OAASVM with polynomial kernel is superior to that of other methods.

REFERENCES

[1] C.H.Chen, Pattern Recognition and Computer Vision, World Scientific, USA, 1993.

[2] F.J. Leong, A.S. Leong, “Digital Imaging Applications in Anatomic Pathology”, Advances in Anatomic Pathology”, vol. 10, pp. 88–95, March 2003.

[3] P. Arena, A.Basil, M. Bucolo, L. Fortuna, “Image processing for medical diagnosis using CNN”, Nuclear Instruments and Methods in Physics Research, vol. A 497, pp. 174–178, January 2003.

[4] Camargo, J.S. Smith, “An image-processing based algorithm to automatically identify plant disease visual symptoms”, Biosystems Engineering, vol. 102, pp. 9–21, January 2009.

[5] Anatomy of a color histogram - Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society C

[6] “Seamless Mosaicing of Image-Based Texture Maps”, Victor Lempitsky, Denis Ivanov, Department of Mathematics and Mechanics ,Moscow State University

[7] Roland Br´emond and Josselin Petit and Jean-Philippe Tarel,” Saliency Maps of High Dynamic Range Images,” Universi´e Paris Est, LEPSiS, INRETS LCPC

[8] Bruning, B., Berger, B., Lewis, M., Liu, H., Garnett, T., 2020. Approaches, applications, and future directions for hyperspectral vegetation studies: an emphasis on yieldlimiting factors in wheat. Plant Phenome J.

[9] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[10] V. Vapnik, “The natural of statistical learning theory,” Springer, New York, 1995.

[11] V. Vapnik, B.Boser., and I. Guyon, “A training algorithm for optimal margin classifiers,” Conference on Computational Learning Theory 15th (COLT 1992), 1992, pp. 144-152.

[12] V. Kumar, M. Steinbach, and P. N. Tan, “Introduction to data mining,” Pearson, Addison Wesley, . London, 2006.