

H.G. Chew, C.C. Lim, and R.E. Bogner. Dual-nu Support Vector Machines and applications in multi-class image recognition. In *Proceedings of the 6th International Conference on Optimization: Techniques and Applications (ICOTA6 2004)*, CD-ROM, Ballarat, Australia, 2004.

Correspondence:

Hong-Gunn Chew
Department of Electrical and Electronic Engineering
The University of Adelaide
Adelaide
SA 5005
Australia
hgchew@eleceng.adelaide.edu.au

-blank page-

Dual-nu Support Vector Machines and applications in multi-class image recognition

H.G. Chew^{†‡}, C.C. Lim[†], and R.E. Bogner^{†‡}

{hgchew, cclim, bogner}@eleceng.adelaide.edu.au

[†] *School of Electrical
and Electronic Engineering
The University of Adelaide
Adelaide
SA 5005 Australia*

[‡] *Cooperative Research Centre for
Sensor Signal and Information Processing
SPRI Building Technology Park
Mawson Lakes Boulevard
Mawson Lakes SA 5095 Australia*

Abstract

Dual-nu Support Vector Machine (SVM) is an effective method in pattern recognition and target detection. It offers competitive performance in detection and computation with traditional classifiers. In this paper, we show that the Dual-nu SVM is capable of achieving classification performance for binary classification no worse than other types of Support Vector Machines, including C-SVM and nu-SVM. We investigate the use of Dual-nu SVM in multi-class image recognition using the winner-takes-all rejection strategy. Performance of Dual-nu SVM on a 60,000-element training set and 10,000-element test set handwritten digit recognition problem is analysed.

1 Introduction

The Support Vector Machine (SVM) is a supervised learning paradigm that has found important applications in image classifications [11, 4]. It implements structural risk minimisation, which is a learning principle that attempts to minimise the error and the complexity of the decision function [15, 1]. The SVM minimises classification errors by maximising the margin between the two classes in a feature space induced by a kernel, and minimising complexity by using least training points to support the decision hyperplane. Hence, the complexity of the SVM solution is proportional to the number of support vectors.

Training an SVM from the implementation point of view is equivalent to solving a linearly constrained quadratic programming problem. Its objective function consists of the width of the margin $2/\|\mathbf{w}\|$ and an error penalty term, and is constrained by a box constraint and an equality constraint. The setting of the error penalty in the objective function is based on trial and error, which requires additional time consuming training. In many applications, prior knowledge such as the detection rate required is available. It is advantageous to incorporate such prior knowledge into SVMs to give improved performance in generalisation and computation. One such SVM is the ν -SVM [14]. It provides a bound on the selection of the error penalty and reduces the need to test many different error penalty values to find the optimal one. The incorporation of prior knowledge can be pursued further for training dataset with uneven class size, commonly found in target detection applications and multi-class image recognition problems. Dual- ν SVM [2] is designed to not only reduce the complexity of error penalty selection, but also to match performance in detection and computation with other types of SVMs and other traditional classifiers.

The purpose of this paper is twofold. We show analytically that indeed Dual- ν SVM (2ν -SVM) is capable of achieving no worse classification performance for binary classification than other types of support vector machines, including C -SVM and ν -SVM, while reducing the computational requirements. The problem of multi-class image recognition using 2ν -SVM is investigated. We use the winner-takes-all rejection strategy as it is simple to implement and yet effective in classification performance. Due to the one-against-rest training dataset, the use of dual error parameters in 2ν -SVM provides better control in the training. Performance of 2ν -SVM on a 60,000-element training set and 10,000-element test set handwritten digit recognition problem is analysed.

2 Support Vector Machine Formulation

The Support Vector Machine is an implementation of structured risk minimisation [15, 1]. The SVM is first trained with a dataset with each data point having one of two classification labels: positive (+1) and negative (-1). The training dataset can therefore be divided into the positive class and the negative class. The original SVM formulation (which is commonly termed C -SVM) requires an error penalty C parameter to train the machine. This parameter is essentially set by trial and error for each problem, and requires additional time consuming training. The parameter selection process involves training multiple SVMs with different error parameters until an optimal classifier is obtained. The introduction of ν -SVM partially overcomes the unintuitive selection of the error parameter by replacing the parameter with ν . The parameter ν is in effect the bounds on the classification error, and can be set based on the error rate of the first iteration 2ν -SVM. The optimal ν -SVM can be found after two or three iterations while C -SVM normally requires many more iterations.

Both C -SVM and ν -SVM utilise a single error parameter to weigh the costs of errors with the width of the decision margin. In cases where the numbers of training data for each class of a problem are different (e.g. positive-negative class ratio of 1:2, 10:1), the decision boundary would be biased towards the class with less training data. The result is a classifier that makes more classification errors in that class. A more general formulation for both types of SVM have been introduced with class biasing: $2C$ -SVM [4] and 2ν -SVM [2]. We will briefly discuss these two types of SVMs in this section, and the relationship between these SVMs in the following section.

2.1 Dual- C Support Vector Machines

The formulation of C -SVM was first introduced in [1]. The original C -SVM formulation includes a single error parameter C as a regularisation factor between the width of the margin and the total distance of each error from the margin. The change to two error parameters, one for each class, improves the capability of the SVM to be able to incorporate classification biasing, with only a simple change in the formulation. Thus, $2C$ -SVM [4] introduces C_+ and C_- as the error parameters for the positive and negative classes respectively. $2C$ -SVM is a more general formulation than C -SVM, and we can return to C -SVM by setting $C_+ = C_- = C$.

Consider a set of l data vectors $\{\mathbf{x}_i, y_i\}$, with $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$, $i = 1, \dots, l$, where \mathbf{x}_i is the i -th data vector that belongs to a binary class y_i . We seek the hyperplane that best separates the two classes with the widest margin while minimising the cost of errors governed by the error parameters $C_+, C_- > 0$. The maximal margin hyperplane problem is formulated in the following primal problem:

Problem (P_{2C}).

$$\min_{\mathbf{w}, b, \xi_i} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \sum_i C_i \xi_i \right\} \quad (1)$$

subject to

$$y_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad (2)$$

$$\xi_i \geq 0, \quad (3)$$

where

$$C_i = \begin{cases} C_+, & y_i = +1 \\ C_-, & y_i = -1 \end{cases}. \quad (4)$$

The function $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^n$ is a mapping function from the data space to the feature space to provide generalisation for the decision function that may be a non-linear function of the training data. The vector $w \in \mathbb{R}^n$ is the normal vector of the hyperplane in the feature space, b is the offset of the decision hyperplane, and ξ_i are slack variables to relax the constraint for non-separable problems.

The problem is equivalent to maximising the margin $2/\|\mathbf{w}\|$, while minimising the cost of the errors $\sum C_i \xi_i$. The margins are defined by $\mathbf{w} \cdot \Phi(\mathbf{x}) + b = \pm 1$.

The $2C$ -SVM training problem can be formulated as a Wolfe dual Lagrangian problem [4]. The Wolfe dual Lagrangian is explained by [6]. The present problem involves:

Problem (D_{2C}).

$$\max_{\{\alpha_i\}} \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right\} \quad (5)$$

subject to

$$0 \leq \alpha_i \leq C_i, \quad (6)$$

$$\sum_i \alpha_i y_i = 0, \quad (7)$$

where α_i are the Lagrange multipliers, and $K(\cdot, \cdot)$ is the kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \quad (8)$$

The resulting decision variables α_i define the decision hyperplane that separates the feature space into the positive and negative classes. The decision function is

$$f(x) = \text{sgn} \left(\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right). \quad (9)$$

The decision variables or Lagrange multipliers α_i can be thought of as the weights to the training vectors that support the decision hyperplane. As such, the corresponding training vectors are termed as follows:

Remark 2.1. *Training data vectors, \mathbf{x}_i , with corresponding decision variables $\alpha_i > 0$ are termed support vectors (SVs), and support vectors with $\alpha_i = C_i$ are additionally termed bounded support vectors (BSVs).*

The number of SVs and BSVs for a problem forms the basis for error parameter selection in 2ν -SVM.

2.2 Dual- ν Support Vector Machines

The formulation of ν -SVM was first introduced in [14] to simplify the selection of the error parameter. The error parameter was changed from C , which ranges from 0 to infinity, to ν , which ranges between 0 and 1. The parameter ν sets the bounds on the number of support vectors as well as bounded support vectors, such that (ratio of BSV) $\leq \nu \leq$ (ratio of SV). The parameter C varies greatly in different classification problems, requiring many iterations to determine the optimal parameter, while we have found that ν can be set at 0.1 in most cases for the first iteration. However, ν -SVM again only has one error parameter, and can be extended to dual errors to allow more flexibility in the training process. Dual- ν also overcomes the limitation and restriction of ν -SVM when the training class sizes are not similar [7].

In the Dual- ν formulation [3], we introduce ν_+ and ν_- (to replace C_+ and C_- in $2C$ -SVM) as the error parameters of training for the positive and negative classes respectively.

Again consider a set of l data vectors $\{\mathbf{x}_i, y_i\}$, with $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$, $i = 1, \dots, l$, where \mathbf{x}_i is the i -th data vector that belongs to a binary class y_i . The 2ν -SVM primal formulation, with the error parameters $0 \leq \nu_+, \nu_- \leq 1$, is:

Problem ($P_{2\nu}$).

$$\min_{\mathbf{w}, b, \rho, \xi_i} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 - \sum_i C_i (\nu \rho - \xi_i) \right\}, \quad (10)$$

subject to

$$y_i (\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq \rho - \xi_i, \quad (11)$$

$$\xi_i \geq 0, \quad (12)$$

$$\rho \geq 0, \quad (13)$$

where

$$C_i = \begin{cases} C_+, & y_i = +1 \\ C_-, & y_i = -1 \end{cases}, \quad (14)$$

with

$$\nu = \frac{2\nu_+\nu_-}{\nu_+ + \nu_-}, \quad (15)$$

$$C_+ = \left[l_+ \left(1 + \frac{\nu_+}{\nu_-} \right) \right]^{-1} = \frac{\nu}{2l_+\nu_+}, \quad (16)$$

$$C_- = \left[l_- \left(1 + \frac{\nu_-}{\nu_+} \right) \right]^{-1} = \frac{\nu}{2l_-\nu_-}. \quad (17)$$

ρ is the position of the margins as defined by $\mathbf{w} \cdot \mathbf{x} + b = \pm\rho$, and l_+ and l_- are the numbers of training points for the positive and negative classes respectively. The problem is now equivalent to maximising the margin $2/\|\mathbf{w}\|$, while minimising the position of the margins $\pm\rho$ and the cost of the errors $C_i\xi_i$, where \mathbf{w} is the normal vector and b is the bias, describing the hyperplane, and ξ_i is the slack variable for classification errors.

Remark 2.2. The original ν -SVM formulation by [14] can be derived from 2ν -SVM by letting $\nu_+ = \frac{\nu_{ori}l_-}{2l_+}$ and $\nu_- = \frac{\nu_{ori}l_+}{2l_-}$ where ν_{ori} is the error parameter of ν -SVM. If the training class size is balanced, that is $l_+ = l_-$, it follows that $\nu_+ = \nu_- = \nu_{ori}$, which shows the similarity of the two formulations.

The 2ν -SVM training problem ($P_{2\nu}$) can be formulated as a Wolfe dual Lagrangian problem [2]:

Problem ($D_{2\nu}$).

$$\max_{\{\alpha_i\}} \left\{ -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right\} \quad (18)$$

subject to

$$0 \leq \alpha_i \leq C_i, \quad (19)$$

$$\sum_i \alpha_i y_i = 0, \quad (20)$$

$$\sum_i \alpha_i \geq \nu, \quad (21)$$

where $i, j \in 1, \dots, l$, α_i are the Lagrange multipliers, and $K(\cdot, \cdot)$ is the kernel function (8).

In solving the 2ν -SVM problem, constraint (21) can be simplified from an inequality to an equality as follows:

Proposition 2.3. *The optimal solution of Problem ($D_{2\nu}$) results in*

$$\sum_i \alpha_i = \nu. \quad (22)$$

Proof. It can be seen that $\sum_i \alpha_i > \nu$ cannot form the optimal solution as the objective function can be maximised further by decreasing α_i . \square

Remark 2.4. *A similar equality result as Proposition 2.3 exists in ν -SVM, and is discussed in [14].*

Remark 2.5. *It can be seen in Problem ($P_{2\nu}$) that we have made $\sum_i C_i = 1$ as a result of normalising the solution and simplifying the formulation. The sum can be found from the definitions (16) and (17) as well as (15):*

$$\sum_i C_i = l_+ C_+ + l_- C_- = \frac{\nu}{2\nu_+} + \frac{\nu}{2\nu_-} = 1. \quad (23)$$

3 Relationship between 2ν -SVM and $2C$ -SVM

The newer 2ν -SVM formulation is not as widely used as the more proven $2C$ -SVM. However, the use of 2ν -SVM will result in the same classifier as with $2C$ -SVM, without the difficulty in selecting the error parameter. The differences in error parameters between 2ν -SVM and $2C$ -SVM are indeed not without relations. The two types of machines can result in the same optimal solution to a problem with the proper setting of the corresponding error parameters as we will show below. With 2ν -SVMs, the easier selection of ν s simplifies the error parameters search, as compared to $2C$ -SVMs, and thus can result in better performing SVMs.

Note that in this section, we shall denote the variables to the optimal solution of a $2C$ -SVM with the superscript C , and of a 2ν -SVM with the superscript ν .

3.1 Relating 2ν to $2C$

An optimal solution to 2ν -SVM has a corresponding optimal solution in $2C$ -SVM.

Proposition 3.1. *If $\{\mathbf{w}^\nu, b^\nu, \xi_i^\nu, \rho^\nu\}$ with the corresponding $\{\alpha_i^\nu\}$ is an optimal solution to a 2ν -SVM given the error parameters ν_+ and ν_- , then $\{\mathbf{w}^C, b^C, \xi_i^C\}$ where $\mathbf{w}^C = \mathbf{w}^\nu/\rho^\nu$, $b^C = b^\nu/\rho^\nu$, $\xi_i^C = \xi_i^\nu/\rho^\nu$ with $\{\alpha_i^C\} = \{\alpha_i^\nu/\rho^\nu\}$ is an optimal solution to the corresponding $2C$ -SVM, with error parameters*

$$\begin{aligned} C_+ &= \left[\rho^\nu l_+ \left(1 + \frac{\nu_+}{\nu_-} \right) \right]^{-1}, \\ C_- &= \left[\rho^\nu l_- \left(1 + \frac{\nu_-}{\nu_+} \right) \right]^{-1}. \end{aligned} \quad (24)$$

Proof. Consider the primal formulation of 2ν -SVM where the optimal solution $\{\mathbf{w}^\nu, b^\nu, \xi_i^\nu, \rho^\nu\}$ minimises the objective function (10). Lemma 3.2 given below states that the solution is also the optimiser of

$$\min_{\{\mathbf{w}, b, \xi_i, \rho\}} \frac{1}{2} \|\mathbf{w}\|^2 + \sum_i C_i^\nu \xi_i \quad (25)$$

subject to $\nu\rho = \nu\rho^\nu$, where C_i^ν is given by C_+ and C_- using Equation (14). The last constraint becomes $\rho = \rho^\nu$ and removes ρ as an optimising variable. However, the $2C$ -SVM formulation requires the margins to lie at ± 1 , or $\rho = 1$. We can change the feature space by dividing by ρ^ν , and have $\mathbf{w}' = \mathbf{w}/\rho^\nu$, $b' = b/\rho^\nu$, $\xi_i' = \xi_i/\rho^\nu$ and $C_i^C = C_i^\nu/\rho^\nu$, to get

$$\min_{\{\mathbf{w}', b', \xi_i'\}} \frac{1}{2} \|\mathbf{w}'\|^2 + \sum_i C_i^C \xi_i' \quad (26)$$

subject to

$$y_i[\mathbf{w}' \cdot \Phi(\mathbf{x}_i) + b'] \geq 1 - \xi_i', \quad (27)$$

$$\xi_i' \geq 0, \quad (28)$$

$$\rho/\rho^\nu = 1. \quad (29)$$

This is exactly the $2C$ -SVM Primal Problem, and thus the $2C$ -SVM solution is $\{\mathbf{w}^C, b^C, \xi_i^C\}$ where $\mathbf{w}^C = \mathbf{w}^\nu/\rho^\nu$, $b^C = b^\nu/\rho^\nu$, $\xi_i^C = \xi_i^\nu/\rho^\nu$. Note that C_i^ν , and thus Equations (15)–(17), are also divided by ρ^ν to give the $2C$ -SVM error parameters C_+ and C_- . The normal of the hyperplane \mathbf{w} is the combination of all the vectors weighted by α_i [2]. Since \mathbf{w} is scaled by ρ^ν , both C_i^ν and α_i^ν are also be scaled by ρ^ν . The Dual $2C$ -SVM solution is thus $\{\alpha_i^C\} = \{\alpha_i^\nu/\rho^\nu\}$. \square

Lemma 3.2. *Consider the optimisation problem of the form*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \{a(\mathbf{x}) + b(\mathbf{x})\} \\ \text{subject to} \quad & g(\mathbf{x}) \geq 0, \quad h(\mathbf{x}) = 0. \end{aligned} \quad (30)$$

If \mathbf{x}^ is a feasible optimiser of (30), $\mathbf{y} = \mathbf{x}^*$ is also a feasible optimiser of*

$$\begin{aligned} \min_{\mathbf{y}} \quad & \{b(\mathbf{y})\} \\ \text{subject to} \quad & g(\mathbf{y}) \geq 0, \quad h(\mathbf{y}) = 0, \\ & a(\mathbf{y}) = a(\mathbf{x}^*). \end{aligned} \quad (31)$$

Proof. Let $\hat{\mathbf{y}}$ be the optimiser of (31), such that $b(\hat{\mathbf{y}}) < b(\mathbf{x}^*)$, and $a(\hat{\mathbf{y}}) = a(\mathbf{x}^*)$. Therefore

$$a(\hat{\mathbf{y}}) + b(\hat{\mathbf{y}}) < a(\mathbf{x}^*) + b(\mathbf{x}^*),$$

which contradicts the initial condition that \mathbf{x}^* is the optimiser of (30). Thus $\mathbf{y} = \mathbf{x}^*$ is also a feasible minimiser of $b(\mathbf{x}')$ in (31). \square

3.2 Relating $2C$ to 2ν

Similarly, an optimal solution to $2C$ -SVM has a corresponding optimal solution in 2ν -SVM.

Proposition 3.3. *If $\{\mathbf{w}^C, b^C, \xi_i^C\}$ with the corresponding $\{\alpha_i^C\}$ is an optimal solution to a $2C$ -SVM given the error parameters C_+ and C_- , then $\{\mathbf{w}^\nu, b^\nu, \xi_i^\nu, \rho^\nu\}$ where $\rho^\nu = (l_+C_+ + l_-C_-)^{-1}$, and $\mathbf{w}^\nu = \rho^\nu \mathbf{w}^C$, $b^\nu = \rho^\nu b^C$, $\xi_i^\nu = \rho^\nu \xi_i^C$ with $\{\alpha_i^\nu\} = \{\rho^\nu \alpha_i^C\}$ is an optimal solution to the corresponding 2ν -SVM, with error parameters*

$$\begin{aligned}\nu_+ &= \frac{\sum_i \alpha_i^C}{2C_+ l_+}, \\ \nu_- &= \frac{\sum_i \alpha_i^C}{2C_- l_-}.\end{aligned}\tag{32}$$

Proof. Consider the dual formulation of $2C$ -SVM where the optimal solution $\{\mathbf{w}^C, b^C, \xi_i^C\}$ maximises the objective function (18). Lemma 3.4 given below states that the solution is also the optimiser of

$$\max_{\{\alpha_i\}} \left\{ -\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right\}\tag{33}$$

subject to $\sum_i \alpha_i = \sum_i \alpha_i^C$, where C_i^C is given by C_+ and C_- using Equation (14). The last constraint becomes equal to the new ν after some scaling. However, the 2ν -SVM formulation requires the sum of C_i to be equal to one (Remark 2.5). This requirement is met by dividing the Dual space by $\sum_i C_i^C = l_+C_+ + l_-C_-$. If we let $\rho^\nu = (l_+C_+ + l_-C_-)^{-1}$ and have $\alpha_i^\nu = \rho^\nu \alpha_i$, $C_i^\nu = \rho^\nu C_i^C$ and $\nu = \rho^\nu \sum_i \alpha_i^C$, we get

$$\max_{\{\alpha_i^\nu\}} \left\{ -\frac{1}{2} \sum_{i,j} \alpha_i^\nu \alpha_j^\nu y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right\}\tag{34}$$

subject to

$$0 \leq \alpha_i^\nu \leq C_i^\nu,\tag{35}$$

$$\sum_i \alpha_i^\nu y_i = 0,\tag{36}$$

$$\sum_i \alpha_i^\nu = \nu.\tag{37}$$

This is exactly the 2ν -SVM Dual Problem, and thus the 2ν -SVM solution is $\{\alpha_i^\nu\} = \{\rho^\nu \alpha_i^C\}$. Returning to the Primal variables, the normal \mathbf{w} is the combination of all the vectors weighted by α_i [2]. The transformation from $2C$ -SVM to 2ν -SVM scaled α_i by ρ^ν , the normal \mathbf{w} should be similarly scaled. The same argument follows for the other optimising variables. The 2ν -SVM error parameters are calculated from C_i^ν and ν using Equations (15)–(17). \square

Lemma 3.4. *Consider the optimisation problem of the form*

$$\begin{aligned}\max_{\mathbf{x}} & \quad \{a(\mathbf{x}) + b(\mathbf{x})\} \\ \text{subject to} & \quad g(\mathbf{x}) \geq 0, \quad h(\mathbf{x}) = 0.\end{aligned}\tag{38}$$

If \mathbf{x}^ is a feasible optimiser of (38), $\mathbf{y} = \mathbf{x}^*$ is also a feasible optimiser of*

$$\begin{aligned}\max_{\mathbf{y}} & \quad \{b(\mathbf{y})\} \\ \text{subject to} & \quad g(\mathbf{y}) \geq 0, \quad h(\mathbf{y}) = 0, \\ & \quad a(\mathbf{y}) = a(\mathbf{x}^*).\end{aligned}\tag{39}$$

Proof. The proof is obtained from Lemma 3.2 by minimising $[-a(\mathbf{x}) - b(\mathbf{x})]$ and $[-b(\mathbf{x})]$ for the two objective functions. □

With Propositions 3.1 and 3.3, we have shown that if an optimal solution exists in one formulation of SVMs, a corresponding optimal solution also exists in the other formulation. Therefore, one formulation can do no worse than the other formulation, provided the correct error parameters have been chosen. However, the search for the optimal error parameters for a problem is difficult and time consuming. 2ν -SVM improves on the search by providing a more intuitive error parameter model, thus resulting in simpler search and selection, and reduce overall training times.

Remark 3.5. *The decision functions for 2C-SVM (f_{2C}) and 2ν -SVM ($f_{2\nu}$) are related as well, with*

$$f_{2C}(\mathbf{x}) = f_{2\nu}(\mathbf{x})/\rho^\nu. \quad (40)$$

4 Using 2ν -SVMs for Multi-Class Problems

The SVM is a binary classifier. There are strategies that use binary classifiers to classify multiple classes like for multi-class classification or recognition. The strategies involve using multiple binary classifiers to form one multi-class classifier. A few of these strategies [13, Chapter 7.6] are one-against-one, one-against-rest and error correcting code. We shall briefly describe each of these strategies.

- The one-against-one (or pairwise) strategy splits the n -class problem into $(n-1)n/2$ binary class subproblems by creating a binary classifier for every single pair of classes. In testing, a test data point belongs to the class that has been classified into the most, in all the binary classifiers.
- The one-against-rest (or winner-takes-all) strategy takes each class and trains a classifier against the rest of the classes. This requires n binary classifiers. This strategy uses a slightly different decision function of the SVM without the $sgn()$ function, as the distance from the decision hyperplane is used. A test data point belongs to the class whose classifier decision is most positive in value.
- The error correcting code strategy relies on the principles of error correcting code to relieve the cost of misclassification of each binary classifier. As with data error correcting codes, a misclassification in one binary classifier does not necessarily result in a misclassification in the end. Using more binary classifiers to build the overall classifier would produce higher performance when individual binary classifiers have high misclassifications. This strategy is more complex as it requires more analysis into the optimal code to use, while not necessarily improving much on classification performance especially when the problem does not have high misclassification rates. The code uses at least $\log_2 n$ binary classifiers, and more classifiers are required as the misclassification rates of individual classifiers are found to be high.

We have found that the one-against-rest strategy provides comparable classification performance while being easy to implement. However, as the training set for each binary classifier involves one class (usually set as positive class) against the rest of the classes, the ratio of data points in each class would be $1 : (n-1)$. The uneven ratio causes biasing effect that can be corrected using dual error parameter SVMs like 2ν -SVM.

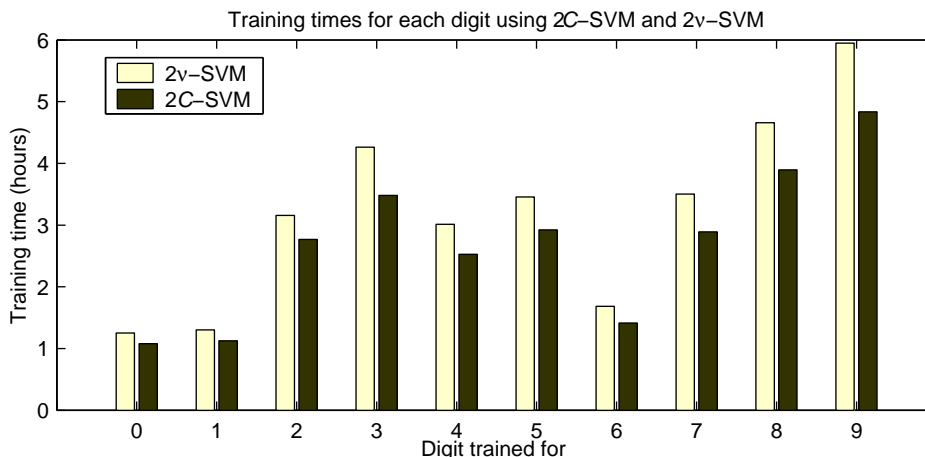


Figure 1: Computation times of $2C$ -SVM and 2ν -SVM for one error parameter value

There is currently ongoing research into multi-class SVMs with multi-class objective functions such as [10, 8]. These SVM formulations still follow the same principle as C -SVM, and would benefit with the change to multiple ν error parameters.

5 Results

We have selected the one-against-rest strategy for its simple implementation and good classification performance. The strategy’s use of unbalanced training class sizes can easily be handled with 2ν -SVM.

The MNIST handwritten digit recognition dataset [9] is the primary source we use for comparisons between $2C$ -SVM and 2ν -SVM. The dataset is widely used in pattern recognition research and has many results published that can be compared with. The dataset has ten handwritten digits (0–9) digitised into 28×28 -pixel images, in 60,000 training images and 10,000 test images.

The purpose of this exercise is to compare the results obtained by [12] using C -SVM, with the newer 2ν -SVM. It was expected that 2ν -SVM would provide an easier error parameter selection, as compared to $2C$ -SVM, and hopefully improve the classification performance as well. Thus we used the parameters in [12] as our starting point, with the radial basis function kernel of width 15.0 and error parameter $C = 10$.

The training were performed on a Pentium 4 2.2GHz PC with 1GB of RAM, using the algorithms in [5] for $2C$ -SVMs and 2ν -SVMs. Figure 1 shows the computational time required for training each SVM to classify one digit against the rest. It shows that 2ν -SVM takes about 20% more time to train, compared to $2C$ -SVM. The figure, however, does not show the process and time required to determine the optimal error parameter to use.

Figure 2 shows the classification performance for different values of the error parameters. The process to search for the optimal C to use normally involves searching from a large value and decreasing C until the optimal classification performance is found. Figure 2a shows four trainings iterations, $C = 10^3, 10^2, 10$ and 1, are required before $C = 10$ is found to be the best compromise between classification performance and generalisation. The error parameter C does not have an upper limit and the range to use varies from problem to problem, therefore it is difficult to determine the right value to start searching from. We could have started from $C = 10^5$ and taken 2 more trainings iterations, or we could have started from $C = 10$ and taken only 2 iterations. The starting search value for

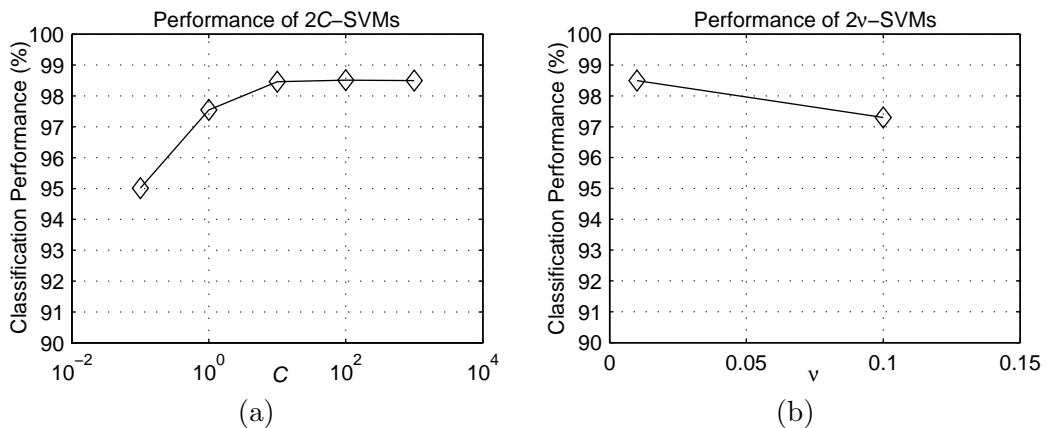


Figure 2: Classification performance for (a) C -SVM and (b) 2ν -SVM for various values of the error parameters C and ν

Table 1: Classification performance comparison

SVM	Classification Performance for Digit (%)										Overall
	0	1	2	3	4	5	6	7	8	9	
$2C$ -SVM $C_+=C_-=10$	99.3	98.4	98.4	98.4	98.4	99.0	97.7	98.6	97.2	99.3	98.5
2ν -SVM $\nu_+=\nu_-=0.01$	99.3	98.4	98.3	98.5	98.3	98.7	98.0	98.5	97.4	99.2	98.5

2ν -SVM is easier to define and is valid for most applications. We start the search from $\nu = 0.1$ downwards. From the first training with $\nu = 0.1$, we found that the training error was 0%, and the test error was about 2%. Thus we trained using $\nu = 0.01$, which is the optimal error parameter to use (Figure 2b). Therefore, only two iterations were required: $\nu = 0.1$ and 0.01.

Table 1 shows the test performance using $2C$ -SVM and 2ν -SVM, both having similar performance. Overall, 2ν -SVMs are easier to train as the optimal error parameter can be found with fewer number of training iterations compared to $2C$ -SVMs, while providing similar classification performances. The total training time required for 2ν -SVM is therefore less than for $2C$ -SVM.

6 Conclusion

The SVM binary classifier compares well with traditional classifiers as well as newer classifiers [9]. We have shown that 2ν -SVM is easier to use, and results in competitive classification performance in both binary and multi-class classification. 2ν -SVM allows easier selection of the error parameter ν s, while can do no worse than $2C$ -SVM in classification performance.

The relationship between the solutions of 2ν -SVM and $2C$ -SVM have been derived, and shows that the two formulations can and do result in the same solution. The relationship allows us to use 2ν -SVM with its simpler error parameter ν while having the same performance as $2C$ -SVM.

References

- [1] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.
- [2] H.G. Chew, R.E. Bogner, and C.C. Lim. Dual-nu support vector machine with error rate and training size biasing. In *Proceedings of the 26th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, pages 1269–1272, Salt Lake City, Utah, USA, 2001. IEEE, Piscataway, NJ, USA.
- [3] H.G. Chew, R.E. Bogner, and C.C. Lim. On initialising nu-support vector machine training. In *Proceedings of the 5th International Conference on Optimisation: Techniques and Applications (ICOTA 2001)*, pages 1740–1747, Hong Kong, 2001.
- [4] H.G. Chew, D.J. Crisp, R.E. Bogner, and C.C. Lim. Target detection in radar imagery using support vector machines with training size biasing. In *Proceedings of the Sixth International Conference on Control, Automation, Robotics and Vision (ICARCV 2000)*, Singapore, 2000.
- [5] H.G. Chew, C.C. Lim, and R.E. Bogner. An implementation of training dual-nu support vector machines. In L.Q. Qi, K.L. Teo, and X.Q. Yang, editors, *Optimization and Control with Applications*. Kluwer, 2004.
- [6] E.K.P. Chong and S.H. Zák. *An Introduction to Optimization*. Discrete Mathematics and Optimization. Wiley-Interscience Series, USA, 1996.
- [7] D.J. Crisp and C.J.C. Burges. A geometric interpretation of ν -svm classifiers. *Advances in Neural Information Processing Systems*, 12:244–251, 2000.
- [8] C.W. Hsu and C.J. Lin. A comparison on methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, 2002.
- [9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [10] Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines. In *Proceedings of the 33rd Symposium on the Interface, CA, USA, 2001*.
- [11] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proceedings of CVPR'97, Puerto Rico, 1997*.
- [12] B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, Munich, 1997.
- [13] B. Schölkopf and A.J. Smola. *Learning with Kernels — Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. MIT Press, USA, 2002.
- [14] B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.
- [15] V.N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer Verlag, New York, USA, 1982. Original edition in Russian: Nauka, Moscow, 1979.