Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **200X**

Website: http://AIMsciences.org

pp. **X–XX**

ON REGULARISATION PARAMETER TRANSFORMATION OF SUPPORT VECTOR MACHINES

Hong-Gunn Chew
Cheng-Chew Lim

School of Electrical and Electronic Engineering
The University of Adelaide
SA 5005
AUSTRALIA

(Communicated by the associate editor name)

Abstract. The Dual-nu Support Vector Machine (SVM) is an effective method in pattern recognition and target detection. It improves on the Dual-C SVM, and offers competitive performance in detection and computation with traditional classifiers. We show that the regularisation parameters Dual-nu and Dual-C can be set such that the same SVM solution is obtained. We present the process of determining the related parameters of one form from the solution of a trained SVM of the other form, and test the relationship with a digit recognition problem. The link between the Dual-nu and Dual-C parameters allows users to use Dual-nu for ease of training, and to switch between the two forms readily.

1. **Introduction.** The Support Vector Machine (SVM) implements structural risk minimisation which is a learning principle that attempts to minimise the error and the complexity of the decision function [1, 17]. The supervised learning paradigm has been used with many applications in image classifications [3, 10]. The SVM learns from a two-class training set by maximising the width of a 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.

Training an SVM is formulated as 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 optimisation problem is large and can be solved using numerical methods such as those in [4, 8, 12, 16, 18, 19]. The setting of the error penalty in the objective function is based on repeated trial, although there are automated algorithms [13], which still requires additional time consuming training. Prior knowledge in many applications such as the detection rate required is available. Such prior knowledge can be incorporating into SVMs to give improved generalisation and computation performance.

The ν-SVM [15] is one such formulation that provides a bound on the selection of the error penalty and reduces the need to test different error penalty values to find the optimal one. The incorporation of prior knowledge can be pursued further for

2000 *Mathematics Subject Classification.* Primary: 68T10; Secondary:90C20.
Key words and phrases. Support Vector Machine, Pattern recognition, Quadratic optimisation.

training dataset with uneven class size, commonly found in target detection applications and multi-class image recognition problems. Dual-ν SVM is an effectively way to incorporate prior knowledge [2, 4]. It is designed to match performance in detection and computation with other types of SVMs and other traditional classifiers, while retaining ν-SVM's reduced error penalty selection complexity.

This paper highlights three main points. First, we introduce the Dual-C and Dual-ν SVM formulations in Section 2. The Dual-C SVM is a proven classifier for a wide range of applications [**?**, **?**, 10] and is the class biasing extension of the original C-SVM, while the Dual-ν SVM is the extension of ν-SVM. Second, we show analytically in Section 3 that there is a relationship between the solutions of Dual-ν SVM and Dual-C SVM. That means the results of one SVM can be transformed into a solution of the other SVM, with identical decision functions. Last, an experiment using the benchmark pattern recognition dataset (MNIST) in Section 4 demonstrates transformation between the Dual-ν SVM solution and the Dual-C SVM solution. The experiment also shows the simpler error penalty selection requirements while achieving equal or better classification performance for binary classification than the Dual-C SVM.

The transformation demonstrates the ability of the new Dual-ν SVM formulation to obtain the same optimum solutions as Dual-C SVM while reducing the computational requirements.

2. Support Vector Machine Formulation.

The Support Vector Machine is trained with a dataset with each data point having one of two classification labels: positive (+1) and negative (−1). The C-SVM and ν-SVM formulations both utilise a single error parameter during training to weigh the costs of errors with the width of the decision margin. A common phenomenon in pattern recognition where the numbers of training data points for each class are different, 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 each type of SVM has been introduced with class biasing: Dual-C SVM (denoted as 2C-SVM) [3] and Dual-nu SVM (denoted as 2ν-SVM) [4]. A separate error parameter for each classification label allows the resulting SVM to be biased to one class, or to correct an existing training dataset bias, as documented in [3] for 2C-SVM and in [2, 7] for 2ν-SVM. 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 original C-SVM formulation [1] uses 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. A simple change in the formulation to two error parameters, one for each class, improves the capability of the SVM to be able to incorporate classification biasing. The 2C-SVM formulation [3] introduces C_+ and C_- as the error parameters for the positive and negative classes respectively. 2C-SVM, being a more general formulation, can reduce to C-SVM by setting $C_+ = C_- = C$.

Consider a set of l data vectors $\{\mathbf{x}_i, y_i\}$, with $\mathbf{x}_i \in \mathrm{R}^d$, $y_i \in \{+1, -1\}$, $i = 1, \ldots, l$, 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})**.**

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

subject to

$$\begin{aligned} y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b) &\geq 1-\xi_i\,, \\ \xi_i &\geq 0\,, \end{aligned}$$

where

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

The mapping function $\Phi : \mathrm{R}^d \to \mathrm{R}^n$ moves 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 $\mathbf{w} \in \mathrm{R}^n$ and the bias $b \in \mathrm{R}$ describes the hyperplane with $\mathbf{w}\cdot\Phi(\mathbf{x})+b = 0$ in the feature space, and $\xi_i \in \mathrm{R}$ 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 is convex. It can be formulated as a Wolfe dual Lagrangian problem [3, 5], expressed as

Problem (D_{2C})**.**

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

subject to

$$\begin{aligned} 0 \leq \alpha_i &\leq C_i\,, \\ \sum_i \alpha_i y_i &= 0\,, \end{aligned}$$

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

$$K(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j). \tag{1}$$

The resulting decision variables α_i define the decision hyperplane that separates the feature space into the positive and negative classes. The decision function thus determines the positive or negative side of the hyperplane that the data point lies on, and is given by

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

The Lagrange multipliers α_i can be thought of as the weights to the training vectors that support the decision hyperplane. Therefore, the corresponding training vectors are termed in the following remark.

Remark 1. Training data vectors, $\mathbf{x}_i$, with corresponding decision variables $\alpha_i > 0$ are termed support vectors, and support vectors with $\alpha_i = C_i$ are additionally termed bounded support vectors. In addition, only bounded support vectors can have $\xi_i > 0$ [14].

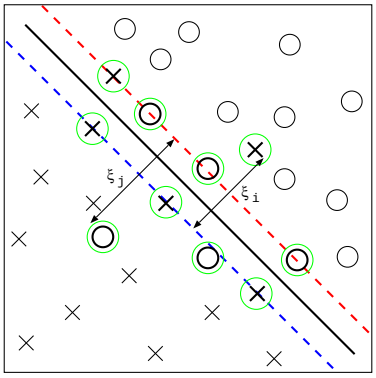

FIGURE 1. Support vectors (circled) of a SVM solution of two classes ($\times$ and $\circ$)

Figure 1 shows an example of a two-dimension SVM solution. In the figure, there are a total of ten support vectors (five $\times$ and five $\circ$) as indicated by the circular highlight. Of these, there are 4 bounded support vectors (two from each class) that have crossed their associated margins.

The number of support vectors and bounded support vectors 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 [15] was developed to simplify the selection of the error parameter. The error parameter was changed from $C \in (0, \infty)$ to $\nu \in (0, 1)$. The parameter ν sets the bounds on the number of support vectors as well as bounded support vectors, such that

$$\text{(ratio of Bounded Support Vectors)} \leq \nu \leq \text{(ratio of Support Vectors)}.$$

The parameter C varies greatly in different classification problems, requiring many iterations to find a suitable value. In contrast, we have found that ν can be set at 0.1 in most cases for the first iteration. However, ν-SVM has only one error parameter, and its training range becomes limited when the training class sizes are different [6]. The training range to produce a feasible SVM is limited by a training set that is non-separable (lower bound) or by an unbalanced training set (upper bound).

The extension to dual errors in Dual-ν allows more flexibility in the training process, and also overcomes the limitation and restriction of ν-SVM. The Extended ν-SVM of Perez-Cruz *et al.* [11] extends the range of the error parameter ν but does not removes the effects of biasing. The new 2ν-SVM removes the restriction of the unbalanced training set, as the data in each class is now weighted separately. Therefore, the range of the 2ν-SVM error parameters is only limited with a lower bound by a non-separable training set and the lower bound reveals the minimum number of training errors of the set.

We introduce ν_+ and ν_- in the Dual-ν formulation [4] as the error parameters of training for the positive and negative classes. The subscript $\pm$ is used to denote both the $+$ and $-$ subscripts of the corresponding variable. That is, $\nu_\pm$ means both ν_+ and ν_-.

Consider a set of l data vectors $\{\mathbf{x}_i, y_i\}$, with $\mathbf{x}_i \in \mathrm{R}^d$, $y_i \in \{+1, -1\}$, $i = 1, \ldots, l$, where $\mathbf{x}_i$ is the i-th data vector that belongs to a binary class y_i. With the error parameters $0 \leq \nu_{\pm} \leq 1$, the 2ν-SVM primal formulation takes the form of:

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

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

subject to

$$\begin{aligned} y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i) + b) &\geq \rho - \xi_i\,, \\ \xi_i &\geq 0\,, \\ \rho &\geq 0\,, \end{aligned}$$

where

$$C_i = \begin{cases} C_+, & y_i = +1 \\ C_-, & y_i = -1 \end{cases}, \tag{3}$$

with

$$\nu = \frac{2\nu_+\nu_-}{\nu_+ + \nu_-}\,, \tag{4}$$

$$C_+ = \left[l_+ \left(1 + \frac{\nu_+}{\nu_-} \right) \right]^{-1} = \frac{\nu}{2l_+\nu_+}\,, \tag{5}$$

$$C_- = \left[l_- \left(1 + \frac{\nu_-}{\nu_+} \right) \right]^{-1} = \frac{\nu}{2l_-\nu_-}\,. \tag{6}$$

The position of the margins, ρ, is 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 $C_i\xi_i$. The hyperplane is defined by the normal vector, $\mathbf{w}$, and the bias, b, and ξ_i is the slack variable for classification errors, as in the case of $2C$-SVM.

Remark 2. The ν-SVM formulation by [15] can be derived from 2ν-SVM by letting $\nu_+ = \frac{\nu_s l}{2l_+}$ and $\nu_- = \frac{\nu_s l}{2l_-}$ where ν_s is the error parameter of ν-SVM. If the training class size is balanced, that is $l_+ = l_-$, it follows that $\nu_+ = \nu_- = \nu_s$, which shows the similarity of the two formulations.

Remark 3. 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 (5) and (6) as well as (4):

$$\sum_i C_i = l_+C_+ + l_-C_- = \frac{\nu}{2\nu_+} + \frac{\nu}{2\nu_-} = 1\,.$$

The 2ν-SVM training problem $(P_{2\nu})$ is a convex function. It can be formulated as a Wolfe dual Lagrangian problem [2], as

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

$$\max_{\{\alpha_i\}} \left\{ -\frac{1}{2} \sum_{i,j} \alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \right\} \tag{7}$$

subject to

$$0 \leq \alpha_i \leq C_i\,,$$

$$\sum_i \alpha_i y_i = 0\,, \tag{8}$$

$$\sum_i \alpha_i \geq \nu\,, \tag{9}$$

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

In solving the 2ν-SVM problem, constraint (9) can be simplified from an inequality to an equality as follows:

Lemma 2.1. *The optimal solution of Problem* $(D_{2\nu})$ *results in*

$$\sum_i \alpha_i = \nu\,.$$

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. □

Note that a similar equality result as Lemma 2.1 exists in ν-SVM, and is discussed in [15].

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

The differences in error parameters between 2ν-SVM and $2C$-SVM are indeed not without relations. We proceed to show that for a classification problem, both SVMs can result in the same optimal solution with the proper setting of the corresponding error parameters. The easier selection of $\nu_\pm$ with 2ν-SVMs simplifies the error parameters search, as compared to $2C$-SVMs, and thus can result in better performing SVMs.

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

$$\begin{aligned} C_+ &= \left[\rho^\nu l_+ \left(1 + \tfrac{\nu_+}{\nu_-}\right)\right]^{-1}, \\ C_- &= \left[\rho^\nu l_- \left(1 + \tfrac{\nu_-}{\nu_+}\right)\right]^{-1}. \end{aligned} \tag{10}$$

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 (2). Lemma 3.1 given below states that the solution is also the optimiser of

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

subject to $\nu\rho = \nu\rho^\nu$, where C_i^ν is given by C_+ and C_- using Equation (3). 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

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

subject to

$$\begin{aligned} y_i\left(\mathbf{w}' \cdot \Phi\left(\mathbf{x}_i\right) + b'\right) &\geq 1 - \xi_i', \\ \xi_i' &\geq 0, \\ \rho/\rho^\nu &= 1. \end{aligned}$$

This is the same as the Primal Problem (P_{2C}), and therefore 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 (4)–(6), 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 [4]. 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 Problem (D_{2C}) solution is thus $\{\alpha_i^C\} = \{\alpha_i^\nu/\rho^\nu\}$. □

Lemma 3.1. *If $\mathbf{x}^*$ is a feasible optimal solution of*

$$\begin{aligned} \min_{\mathbf{x}} \quad & a(\mathbf{x}) + b(\mathbf{x}) \\ \text{subject to} \quad & g(\mathbf{x}) \geq 0,\ h(\mathbf{x}) = 0, \end{aligned} \tag{11}$$

then, $\mathbf{y}^* = \mathbf{x}^*$ is also a feasible optimal solution of

$$\begin{aligned} \min_{\mathbf{y}} \quad & b(\mathbf{y}) \\ \text{subject to} \quad & g(\mathbf{y}) \geq 0,\ h(\mathbf{y}) = 0, \\ & a(\mathbf{y}) = a(\mathbf{x}^*). \end{aligned} \tag{12}$$

Proof. Let $\hat{\mathbf{y}}$ be the optimiser of (12), 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 (11). Thus $\mathbf{y}^* = \mathbf{x}^*$ is also a feasible minimiser of $b(\mathbf{y})$ in (12). □

Proposition 1 shows that the $2C$-SVM solution is scaled from the 2ν-SVM solution by the derived margin position ρ^ν. Indeed, the error parameters of $2C$-SVM are scaled versions of the 2ν-SVM.

Remark 4. Given the 2ν-SVM solution, the error parameters (10) of the corresponding $2C$-SVM are

$$\begin{aligned} C_+ &= C_+^\nu/\rho^\nu, \\ C_- &= C_-^\nu/\rho^\nu \end{aligned} \tag{13}$$

where C_+^ν, C_-^ν are the variable limits as defined by Equations (5) and (6), and ρ^ν is the margin position of the 2ν-SVM solution.

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

Proposition 2. *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*

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

Proof. Consider the dual formulation of $2C$-SVM where the optimal solution $\{\alpha_i^C\}$ maximises the objective function (7). Lemma 3.2 given below states that the solution is also the optimiser of

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

subject to $\sum_i \alpha_i = \sum_i \alpha_i^C$, where C_i^C is given by C_+ and C_- using Equation (3). The last constraint becomes equal to the new ν after some scaling. However, the 2ν-SVM formulation requires $\sum C_i = 1$ (Remark 3). This requirement is met by dividing the Dual space by $\sum_i C_i^C = l_+C_+ + l_-C_-$. With $\rho^\nu = (l_+C_+ + l_-C_-)^{-1}$ and thus $\alpha_i' = \rho^\nu \alpha_i$, $C_i^\nu = \rho^\nu C_i^C$ and $\nu = \rho^\nu \sum_i \alpha_i^C$, we get

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

subject to

$$\begin{aligned} 0 \le \alpha_i' &\le C_i^\nu, \\ \sum_i \alpha_i' y_i &= 0, \\ \sum_i \alpha_i' &= \nu. \end{aligned}$$

The above optimisation problem is precisely 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 [4]. 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 (4)–(6). □

Lemma 3.2. *If $\mathbf{x}^*$ is a feasible optimal solution of*

$$\begin{aligned} \max_{\mathbf{x}} \quad & a(\mathbf{x}) + b(\mathbf{x}) \\ \text{subject to} \quad & g(\mathbf{x}) \ge 0,\ h(\mathbf{x}) = 0, \end{aligned} \tag{14}$$

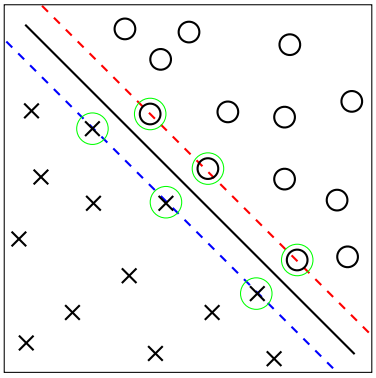

FIGURE 2. A separable dataset

then, $\mathbf{y} = \mathbf{x}^*$ *is also a feasible optimal solution of*

$$\begin{aligned} \max_{\mathbf{y}} \quad & b(\mathbf{y}) \\ \text{subject to} \quad & g(\mathbf{y}) \geq 0,\ h(\mathbf{y}) = 0, \\ & a(\mathbf{y}) = a(\mathbf{x}^*). \end{aligned} \tag{15}$$

Proof. The proof is obtained from Lemma 3.1 by minimising $[-a(\mathbf{x}) - b(\mathbf{x})]$ and $[-b(\mathbf{x})]$ for the two objective functions.

□

There is an interesting observation from Proposition 2 when we have a separable dataset giving a solution with no bounded support vectors. A separable dataset has data points that can be separated by a hyperplane in the feature space. Figure 2 shows an example of a separable two-dimension dataset. There are no bounded support vectors when there are no data points that cross the margin, therefore $\alpha_i < C_i, \forall i$.

The parameters $\nu_\pm$ are inversely proportional to the parameters $C_\pm$ from Equation (2), as $\sum \alpha_i^C$ will not change with increasing $C_\pm$ as long as $\alpha_i < C_i, \forall i$, as is the case for a separable dataset. This properly does not only applies to separable problems, but generally to all problems for a wide range of parameters values.

Remark 5. The parameters $\nu_\pm$ increases while the corresponding parameters $C_\pm$ decreases for any given problem.

Similar to Remark 4, the transformation from $2C$-SVM to 2ν-SVM involves the scaling by the variable ρ^ν. If we consider the $\{C_+^\nu,\ C_-^\nu\}$ parameters required for optimising 2ν-SVM, it would appear that the regularisation parameters do not require the solution of the $2C$-SVM, but only the supplied error parameters C_+ and C_-. Indeed this is correct, but there is another variable ν that is required for the optimisation of 2ν-SVM, and that variable requires the optimisation variables from the solution of the $2C$-SVM.

Remark 6. Given the $2C$-SVM solution, the variable limits in Equations (4)–(6) of the corresponding 2ν-SVM are

$$\begin{aligned} \nu &= \rho^\nu \textstyle\sum_i \alpha_i^C\,, \\ C_+^\nu &= \rho^\nu C_+\,, \\ C_-^\nu &= \rho^\nu C_- \end{aligned} \tag{16}$$

where $\{\alpha_i^C\}$ is the solution of $2C$-SVM, with $\rho^\nu = (l_+C_+ + l_-C_-)^{-1}$.

From Remark 4 and Remark 6, it is evident that the corresponding solutions of $2C$-SVM and 2ν-SVM are related by ρ^ν. In addition, the respective decision functions are also related.

Remark 7. The decision functions for $2C$-SVM (f_{2C}) and 2ν-SVM ($f_{2\nu}$) are related with

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

We have shown with Proposition 1 and Proposition 2 that if an optimal solution exists in one formulation of SVMs, a corresponding optimal solution also exists in the other formulation. Therefore, with the correct error parameters being chosen, one formulation can perform equally as well as the other formulation. However, the search in $2C$-SVM for the optimal error parameters $C_\pm$ for a problem is often difficult and time consuming due to the wide search range of $C_\pm \in (0, \infty)$. 2ν-SVM provides a more intuitive error parameter model that improves on the parameter search, and thus results in simpler search and selection, and shorter overall training times.

4. Practical Results.

In order to compare the results obtained using 2ν-SVM, and the results obtained using $2C$-SVM with the transformation of the parameters from νs to Cs, we will use the results of $2C$-SVM to transform the parameters Cs back to νs to compare that with the original results.

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 as a benchmark. The dataset has ten handwritten digits (0–9) digitised into 28×28-pixel images, in 60,000 training images and 10,000 test images.

We select the one-against-rest (or winner-takes-all) strategy for its simple implementation and excellent classification performance [14]. In our experiment, we classify handwritten images of 10 digits. The one-against-rest strategy takes each class and trains a classifier against the rest of the classes. This requires ten binary classifiers, one for each digit to identify it against the other digits. The strategy's use of unbalanced training class sizes can easily be handled with 2ν-SVM and $2C$-SVM.

4.1. Comparing Classifiers.

The main purpose is to compare the performance of $2C$-SVM and 2ν-SVM with different error parameters. The parameters $C_\pm \in (0, \infty)$ of $2C$-SVM does not have an upper limit, and the optimal value to choose varies from problem to problem. 2ν-SVM, on the other hand, is governed by $\nu_\pm \in (0, 1)$ of a limited range. The starting value of $\nu_\pm = 0.1$ is found to be a good starting value through extensive testing with different datasets and problems.

We use the MINST dataset to train both 2ν-SVM and $2C$-SVM with varying parameter values using the radial basis function kernel of width 15. Table 1 shows the classification performances of the SVMs. The $2C$-SVM results clearly shows that the number of trials needed to find the best performance depends on the starting parameter value. Since there is no upper limit to the parameters $C_\pm$, it is impossible to provide a general guide of where to start from. The resulting effect is the need to complete more iterative trials of different parameter values before the optimal one is found.

TABLE 1. Classification performance comparison

	Classification Performance for Digit (%)										
SVM	0	1	2	3	4	5	6	7	8	9	Overall
$C_+ = C_-$	$2C$-SVM										
0.01	91.4	88.7	92.1	87.0	89.1	94.3	87.8	87.5	91.1	92.5	**89.9**
0.1	95.5	97.2	95.3	94.8	93.6	95.5	94.4	94.7	95.0	93.8	**95.0**
1	97.6	98.4	97.6	97.4	97.5	98.2	97.8	97.2	96.7	97.0	**97.6**
10	98.5	99.4	98.4	98.4	98.3	98.5	99.0	98.1	97.9	98.1	**98.5**
100	98.5	99.3	98.5	98.5	98.3	98.3	99.0	98.6	98.3	97.7	**98.5**
1000	98.5	99.2	98.5	98.5	98.3	98.3	99.0	98.6	98.3	97.7	**98.5**
$\nu_+ = \nu_-$	2ν-SVM										
0.20	96.1	97.4	95.9	94.8	93.2	96.1	94.9	96.1	94.1	93.1	**95.2**
0.10	97.2	98.2	97.4	97.4	97.0	97.6	97.2	97.0	96.8	96.7	**97.3**
0.05	97.5	98.6	98.3	98.3	98.2	98.6	98.3	97.8	97.6	97.7	**98.1**
0.02	97.9	98.9	98.3	98.8	98.4	99.0	98.7	98.2	98.2	98.0	**98.4**
0.01	98.4	98.9	98.5	98.4	98.3	98.5	98.7	98.4	97.9	98.5	**98.5**
0.001	98.5	99.3	98.6	98.5	98.3	98.3	99.0	98.7	98.2	97.8	**98.5**

The 2ν-SVM starting point of $\nu_\pm = 0.1$ requires at least 10% of training vectors to be support vectors. In most problems, this requirement results in a well performing classifier, with the classifier not over-fitting (too few support vectors) or over-generalising (too many support vectors) to the training dataset.

We can see from Table 1 that for this hand written digit dataset, the performance of 2ν-SVM ranges between 95.2% and 98.5%, while $2C$-SVM ranges between 89.9% and 98.5%. Choosing $C_\pm = 0.01$ as the starting value will result in a longer iterative search for the optimal value of $C_\pm = 10$. The strength in 2ν-SVM over $2C$-SVM is the need for fewer iterations to select the optimal parameter value, as starting from $\nu_\pm = 0.1$ will always result in a well performing classifier.

4.2. **Verify Transformation.** Proposition 1 and Proposition 2 define the transformation of the error parameters between 2ν-SVM and $2C$-SVM for a particular dataset. The results in the previous section shows that 2ν-SVM provided the best performance with $\nu_\pm = 0.01$. We will train a set of 2ν-SVMs (one for each digit) using the parameters in the previous section, and transform their solutions into the parameters for $2C$-SVMs. The 2ν-SVM solution and the $2C$-SVM solution can be compared by checking the Lagrange multipliers $\{\alpha_i\}$, with Proposition 1 stating that the resulting multipliers should be $\{\alpha_i^C\} = \{\alpha_i^\nu/\rho^\nu\}$.

The $2C$-SVM solution is transformed back into the parameters for 2ν-SVM to verify Proposition 2. The multipliers should again be $\{\alpha_i^\nu\} = \{\rho^\nu \alpha_i^C\}$. We can also compare this final solution with the initial 2ν-SVM solution.

Table 2 shows the results of the transformation from 2ν-SVM to $2C$-SVM (top section), and then back to 2ν-SVM (bottom section). The $2C$-SVM parameters $\{C_+, C_-\}$ transformed from 2ν-SVM has the approximate ratio of $9:1$. If we have $\nu_+ = \nu_-$, Equation (10) gives the only difference between C_+ and C_- as l_+ and l_-. That is, the ratio of $C_+ : C_-$ is the inverse ratio of the training class sizes, which in our dataset is about $1:9$. This agrees with the strategy proposed in [3] to correct unbalanced training class sizes biasing. The numerical method for training the SVMs induces a small numerical error that is dependent on the termination threshold used. Thus, the $2C$-SVM solution is expected have an insignificantly

TABLE 2. Parameter transformation from 2ν to $2C$ and back to 2ν, starting from $\nu_{\pm} = 0.01$

		Digit									
	Parameter	0	1	2	3	4	5	6	7	8	9
2ν to $2C$	C_+	23.6	28.1	70.1	88.4	77.7	90.1	37.2	85.6	105.6	161.1
	C_-	2.58	3.55	7.73	10.06	8.38	8.95	4.07	9.98	11.41	17.73
	ave error to 2ν ($\times 10^{-6}$)	6.2	5.9	12.3	14.7	11.4	12.5	0.7	10.3	16.3	14.3
2ν to $2C$ to 2ν	$\Delta\nu_+$ ($\times 10^{-3}$%)	- 5	- 5	-12	-12	-10	-12	- 7	-10	-15	-16
	$\Delta\nu_-$ ($\times 10^{-3}$%)	- 5	- 5	-12	-12	-10	-12	- 7	-10	-15	-16
	ave error to $2C$ ($\times 10^{-6}$)	6.7	6.5	13.0	14.3	12.0	12.9	7.2	11.2	16.2	14.4
	ave error to 2ν ($\times 10^{-6}$)	3.0	3.6	6.5	10.5	6.0	8.2	3.7	6.3	11.1	8.1

small difference to the 2ν-SVM solution. The error tabled shows that we have achieved a similar solution.

The second 2ν-SVM solution that was transformed from the $2C$-SVM solution has a similar set of parameters as the initial value of $\nu_+ = \nu_- = 0.01$. The biggest difference was for Digit 9 where it is a mere 0.015%. This set of parameters and the low error between the Lagrange multipliers verifies that the transformation from $2C$-SVM to 2ν-SVM works as proposed.

5. **Conclusion.** We have derived the relationship between the solutions of 2ν-SVM and $2C$-SVM to show that the two formulations can and do result in the same solution.

The relationship allows us to use 2ν-SVM with its simpler error parameters $\nu_{\pm}$ while having the same performance as $2C$-SVM. It can provide the user with a reasonable set of parameters for $2C$-SVM to use, by training with 2ν-SVM first and then transforming results to the $2C$-SVM parameters. This method removes the need to search for the values for $C_{\pm}$, which is problem dependent.

The transformation shows that the 2ν-SVM and the $2C$-SVM both produce the same solution, and that any solution obtained by one formulation can be obtained by the other formulation. The 2ν-SVM formulation provides an intuitive parameter selection while having similar computational load, and thus should provide users with easier and faster classification optimisation than $2C$-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, 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.

[4] 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*. Springer, 2005.
[5] E.K.P. Chong and S.H. Zȧk. *An Introduction to Optimization*. Wiley-Interscience Series, USA, 2nd edition, 2004.
[6] D.J. Crisp and C.J.C. Burges. A geometric interpretation of ν-svm classifiers. *Advances in Neural Information Processing Systems*, **12** (2000), 244–251.
[7] M.A. Davenport, R.G. Baraniuk, and C.D. Scott. Controlling false alarms with support vector machines. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, Toulouse, France, 2006.
[8] S.C. Fang, D.Y. Gao, R.L. Sheu, and S.Y. Wu. Canonical dual approach for solving 0-1 quadratic programming problems. *Journal of Industrial and Management Optimization*, **4** (2008), 125–142.
[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86** (1998), 2278–2324.
[10] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proceedings of CVPR'97, Puerto Rico*, 1997.
[11] F. Perez-Cruz, J. Weston, D.J.L. Hermann, and B. Schölkopf. Extension of the ν-svm range for classification. In J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Models and Applications*, **190** (2003), pages 179–196.
[12] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.
[13] K. Schittkowski. Optimal parameter selection in support vector machines. *Journal of Industrial and Management Optimization*, **1** (2005), 465–476.
[14] B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, Munich, 1997.
[15] B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, **12** (2000), 1207–1245.
[16] K.L. Teo, V. Rehbock, and L.S. Jennings. A new computational algorithm for functional inequality constrained optimization problems. *Automatica*, **29** (1993), 789–792.
[17] V.N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer Verlag, New York, USA, 1982. Original edition in Russian: Nauka, Moscow, 1979.
[18] Z.B. Wang, S.C. Fang, D.Y. Gao, and W.X. Xing. Global extremal conditions for multi-integer quadratic programming. *Journal of Industral and Management Optimization*, **4** (2008), 213–225.
[19] Z. Wei, L. Qi, and J.R. Birge. A new method for nonsmooth convex optimization. *Journal of Inequalities and Applications*, **2** (1998), 157–179.

Received March 2008; revised September 2008.

E-mail address: hgchew@eleceng.adelaide.edu.au
E-mail address: cclim@eleceng.adelaide.edu.au