DUAL ν-SUPPORT VECTOR MACHINE WITH ERROR RATE AND TRAINING SIZE BIASING

Hong-Gunn Chew [†] [*] Robert E. Bogner [†] [*] Cheng-Chew Lim [*]
email: {hgchew, bogner, cclim}@eleceng.adelaide.edu.au

[†] Corporative Research Centre for Sensor Signal and Information Processing,
SPRI Building, Technology Park, Mawson Lakes Boulevard,
Mawson Lakes, SA 5095, Australia

[*] Department of Electrical and Electronic Engineering,
Adelaide University, Adelaide, SA 5005, Australia

ABSTRACT

Support Vector Machines (SVMs) have been successfully applied to classification problems. The difficulty in selecting the most effective error penalty has been partly resolved with ν-SVM. However, the use of uneven training class sizes, which occurs frequently with target detection problems, results in machines with biases towards the class with the larger training set. We propose an extended ν-SVM to counter the effects of the unbalanced training class sizes. The resulting Dual ν-SVM provides the facility to counter these effects, as well as to adjust the error penalties of each class separately. The parameter ν of each class provides a lower bound to the fraction of support vector of that class, and the upper bound to the fraction of bounded support vector of that class. These bounds allow the control on the error rates allowed for each class, and enable the training of machines with specific error rate requirements.

1 INTRODUCTION

Support Vector Machine (SVM) is a classification paradigm based on statistical learning [1][3][6]. It is relatively new in the pattern recognition area and has only been researched extensively for the past few years. One major advantage of SVMs over more traditional classifiers is that pre-processing, or feature extraction, of the data is not essential before training or classifying. This removes a large variable in the search for the best performing classifier.

The training of an SVM requires the setting of an error penalty for training vectors that lie beyond the margins that provide a generalised specification of the decision regions. However, the choice of error penalty is not intuitive and largely depends on the problem at hand. The error penalty is usually determined iteratively by choosing an arbitrarily small value, and adjusting it from the resulting SVM, and retraining until the required performance is obtained from the SVM.

The formulation of ν-SVM by Schölkopf *et al.* [5] removes the error penalty factor, and replaces it with a new parameter ν. With ν limiting the maximum number bounded support vectors, as well as the minimum number of total support vectors, the selection of ν is more intuitive. In this paper, we show that when training sets with uneven class sizes are used, the resulting ν-SVM is undesirably biased towards the larger class.

We introduce a modification of ν-SVM with two νs, one for each class, that we termed Dual ν-SVM. The use of two νs allows the adjustment of bounds of support vectors for each class separately. This adjustment can counter the effects of uneven training class sizes, and allows the flexibility of specifying a different error rate for each class. In classification problems, target detection in particular, it is essential to have the ability to vary the error rate for each class to suit the situation as there is always a compromise between error rates, performance and cost. With the modified Dual ν-SVM, the error rates can be easily chosen without the usual iterative steps required with the original SVM. This means fewer machines need to be trained, and thus less time is required to obtain the final SVM. For large training sets, the computational time for each training is long, and with fewer SVMs to train, time saving is significant.

By having the salient feature of setting the lower bound on the number of support vectors, the resulting SVM is able to generalise better. This implies that the machine will perform well, not only on the training data set.

2 ν-SVM

The original SVM algorithm, proposed by Vapnik [6], seeks the hyperplane that best separates two classes of data vectors.

Consider a set of l data vectors

$\{\mathbf{x}_i, y_i\}, \quad i = 1, \ldots, l, \quad y_i \in \{-1,1\}, \quad \mathbf{x}_i \in \Re^d$

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.

More specifically, we want to find the hyperplane

$\mathbf{w} \cdot \mathbf{x} + b = 0$

subject to the constraints

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

$\xi_i \geq 0$

to minimise

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \ . \tag{1}$$

This is equivalent to maximising the margin $2/\|\mathbf{w}\|$, while minimising the cost of the errors $C(\Sigma\ \xi_i)$, where $\mathbf{w}$ is the normal vector and b is the bias, describing the hyperplane, and ξ_i is the slack variable.

Schölkopf *et al.* [5] proposed ν-SVM by incorporating a change from C in the original SVM algorithm with ν. The optimisation problem is minimised, with respect to $\mathbf{w}$, b and ξ_i, taking the form

$$\min\left\{\frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{l}\sum_i \xi_i\right\} \tag{2}$$

subject to

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

$\xi_i \geq 0$

$\rho \geq 0$. (3)

This changes the width of the margin to $2\rho/\|\mathbf{w}\|$, which is to be maximised while minimising the margin errors, and ρ is the position of the margin.

The primal Lagrangian formulation is

$$\begin{aligned}\min\Big\{ \ L_P \equiv \ & \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{l}\sum_i \xi_i \\ & - \sum_i \alpha_i \left(y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - \rho + \xi_i\right) \\ & - \sum_i \mu_i \xi_i - \delta\rho\Big\}\end{aligned} \tag{4}$$

with Lagrangian multipliers $\alpha_i, \mu_i, \delta \geq 0$.

In its dual Lagrangian form, we have

$$\max\left\{L_D \equiv -\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)\right\} \tag{5}$$

subject to

$$0 \leq \alpha_i \leq \frac{1}{l} \tag{6}$$

$$\sum_i \alpha_i y_i = 0 \tag{7}$$

$$\sum_i \alpha_i \geq \nu \tag{8}$$

where $K(\cdot,\cdot)$ is the kernel function to map the data to another space. With a trained SVM, ***support vectors*** (SVs) are data vectors with $\alpha_i > 0$, and ***bounded support vectors*** (BSVs) are support vectors with $\alpha_i = 1/l$ and $\xi_i > 0$. The resulting decision function is

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

2.1 Characteristics of ν

When we consider a trained ν-SVM, the width of the margin would, in most cases, not be zero; that is $\rho > 0$. By the Karush-Kuhn-Tucker (KKT) conditions [4], if $\rho > 0$, then $\delta = 0$. This means that constraint (8) reduces to an equality condition

$$\sum_i \alpha_i = \nu \ . \tag{10}$$

Since the BSVs have $\alpha_i = 1/l$, the contribution of BSVs to α_i is N_{BSV}/l, where N_{BSV} is the number of BSVs. In other words,

$$\frac{N_{\mathrm{BSV}}}{l} \leq \nu \ . \tag{11}$$

Since SVs have a maximum $\alpha_i = 1/l$, there is a minimum number of SVs, N_{SV}, to contribute to α_i

$$\frac{N_{\mathrm{SV}}}{l} \geq \nu \tag{12}$$

That is, ν is the fraction of data vectors limiting the maximum number of BSVs and the minimum number of SVs.

These bounds seem to perform well, but how the bounds hold for each class should be investigated.

2.2 Effects of ν in each class

We can see from constraint (7) that

$$\sum_i \alpha_i y_i = \sum_{i:y_i=+1} \alpha_i - \sum_{i:y_i=-1} \alpha_i = 0$$

$$\therefore \sum_{i:y_i=+1} \alpha_i = \sum_{i:y_i=-1} \alpha_i \ . \tag{13}$$

This leads to (10) becoming

$$\begin{aligned}\sum_i \alpha_i \ &= \sum_{i:y_i=+1} \alpha_i + \sum_{i:y_i=-1} \alpha_i \\ &= 2\sum_{i:y_i=+1} \alpha_i = 2\sum_{i:y_i=-1} \alpha_i = \nu\end{aligned} \tag{14}$$

Applying (14) it to the bounds (11) and (12), we obtain

$$\frac{2N_{\mathrm{BSV+}}}{l} \leq \nu \leq \frac{2N_{\mathrm{SV+}}}{l} \tag{15}$$

$$\frac{2N_{\mathrm{BSV-}}}{l} \leq \nu \leq \frac{2N_{\mathrm{SV-}}}{l} \tag{16}$$

where $N_{\mathrm{BSV+}}$ ($N_{\mathrm{BSV-}}$) is the number of bounded support vectors in the positive (negative) class, and $N_{\mathrm{SV+}}$ ($N_{\mathrm{SV-}}$) is the number of support vectors in the positive (negative) class.

Let $R_{+/-}$ be the ratio between the positive class size l_+, and negative class size l_-. Multiplying (15) by $1/l_+$, and substituting

$$l = l_+\left(1 + \frac{1}{R_{+/-}}\right)$$

yields

$$\frac{N_{\mathrm{BSV}+}}{l_+} \le \frac{\nu}{2}\left(1+\frac{1}{R_{+/-}}\right) \le \frac{N_{\mathrm{SV}+}}{l_+} \tag{17}$$

and similarly for the negative vectors

$$\frac{N_{\mathrm{BSV}-}}{l_-} \le \frac{\nu}{2}\left(1+R_{+/-}\right) \le \frac{N_{\mathrm{SV}-}}{l_-}. \tag{18}$$

With a class size ratio of other than one, the bounds for each class are different, and there is a bias towards less BSVs in the class with a larger training size. This translates to the SVM having fewer training errors, with $\xi_i \ge \rho$, in that class.

This biasing behaviour is usually not desirable, particularly in the case of target detection, where there is usually a lack of target data vectors. When there is a lack of target data vectors, the resulting SVM has a high target misdetection rate, and this is opposite the desired SVM with low target misdetection rate and low false detection rate. The error penalties need to be different for each class to counter the biasing behaviour.

3 DUAL *ν*-SVM

We build on the optimisation problem by introducing the error penalties back into the problem. Schölkopf *et al.* have shown that *ν*-SVM results in functionally the same machine as

$$\min\left\{\frac{1}{2}\|\mathbf{w}\|^2 - C\left(\nu\rho - \sum_i \xi_i\right)\right\} \tag{19}$$

where C is the error penalty.

We propose to incorporate an error penalty to each vector as

$$\min\left\{\frac{1}{2}\|\mathbf{w}\|^2 - \sum_i C_i\left(\nu\rho - \xi_i\right)\right\} \tag{20}$$

subject to the same constrains

$$y_i\,(\mathbf{x}_i \cdot \mathbf{w} + b) \ge \rho - \xi_i,$$
$$\xi_i \ge 0,\ \rho \ge 0. \tag{21}$$

By replacing C_i with the error penalty C_+ for positive vectors ($y_i = +1$), and the error penalty C_- for negative vectors ($y_i = -1$), the error penalties for each class can be controlled individually.

In the primal Lagrangian formulation, we have

$$\begin{aligned} L_P \equiv\ & \frac{1}{2}\|\mathbf{w}\|^2 - \sum_i C_i\left(\nu\rho - \xi_i\right) \\ & - \sum_i \alpha_i\left(y_i\left(\mathbf{x}_i \cdot \mathbf{w} + b\right) - \rho + \xi_i\right) \\ & - \sum_i \mu_i \xi_i - \delta\rho \end{aligned} \tag{22}$$

with $\alpha_i, \mu_i, \delta \ge 0$. We can determine the equivalent dual Lagrangian to maximise with respect to α_i, μ_i, δ, by equating the corresponding partial derivatives to 0; *viz*

$$\frac{\partial L_P}{\partial \mathbf{w}_\upsilon} = \mathbf{w}_\upsilon - \sum_i \alpha_i y_i \mathbf{x}_{i\upsilon} = 0$$

$$\frac{\partial L_P}{\partial b} = -\sum_i \alpha_i y_i = 0$$

$$\frac{\partial L_P}{\partial \xi_i} = C_i - \alpha_i - \mu_i = 0$$

$$\frac{\partial L_P}{\partial \rho} = -\nu\sum_i C_i + \sum_i \alpha_i - \delta = 0 \tag{23}$$

which results in

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i + \mu_i = C_i$$

$$\sum_i \alpha_i = \nu\sum_i C_i + \delta. \tag{24}$$

Substituting (24) into (20), we obtain the same dual Lagrangian formulation

$$\max\left\{L_D \equiv -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right)\right\} \tag{25}$$

subject to

$$0 \le \alpha_i \le C_i \tag{26}$$

$$\sum_i \alpha_i y_i = 0 \tag{27}$$

$$\sum_i \alpha_i \ge \nu\sum_i C_i \tag{28}$$

with the decision function being

$$f(\mathbf{z}) = \mathrm{sgn}\left(\sum_i \alpha_i y_i K\left(\mathbf{x}_i, \mathbf{z}\right) + b\right). \tag{29}$$

Since b does no appear in the dual Lagrangian, we compute it from the resulting SVM by using two unbounded support vectors with $\alpha_i > 0$ and $\xi_i = 0$, and solving for b and ρ in

$$y_i\,(\mathbf{x}_i \cdot \mathbf{w} + b) = \rho. \tag{30}$$

3.1 Characteristics of dual *ν* s

Again, we consider a trained SVM, with $\rho > 0$, and by the KKT conditions, $\delta = 0$. This means that constraint (28) reduces to the equality condition

$$\sum_i \alpha_i = 2\sum_{i:y_i=+1}\alpha_i = 2\sum_{i:y_i=-1}\alpha_i = \nu\sum_i C_i \tag{31}$$

Thus we have the bounds (c.f. bounds (15), (16))

$$2N_{\mathrm{BSV}+}C_+ \le \nu\sum_i C_i \le 2N_{\mathrm{SV}+}C_+ \tag{32}$$

$$2N_{\mathrm{BSV}-}C_- \le \nu\sum_i C_i \le 2N_{\mathrm{SV}-}C_-. \tag{33}$$

Dividing (32) by $2C_+l_+$ and substituting

$$\sum_i C_i = C_+ l_+\left(1 + \frac{C_-}{C_+}\frac{1}{R_{+/-}}\right)$$

yields

$$\frac{N_{\mathrm{BSV}+}}{l_+} \le \frac{\nu}{2}\left(1 + \frac{C_-}{C_+}\frac{1}{R_{+/-}}\right) \le \frac{N_{\mathrm{SV}+}}{l_+}$$

or $$\frac{N_{\mathrm{BSV}+}}{l_+} \le \nu_+ \le \frac{N_{\mathrm{SV}+}}{l_+} \tag{34}$$

where

$$\nu_+ = \frac{\nu}{2}\left(1 + \frac{1}{C_{+/-}R_{+/-}}\right) \tag{35}$$

$$C_{+/-} = \frac{C_+}{C_-}. \tag{36}$$

Similarly for the negative vectors

$$\frac{N_{\mathrm{BSV}-}}{l_-} \le \nu_- \le \frac{N_{\mathrm{SV}-}}{l_-} \tag{37}$$

where

$$\nu_- = \frac{\nu}{2}\left(1 + C_{+/-}R_{+/-}\right). \tag{38}$$

The introduction of ν_+ and ν_-, and bounds (35) and (37), provides better control of the error rates and generalisation properties of the SVM required. We obtain the SVM by calculating C_+ and C_-, and solving for the optimisation problem.

It should also be noted that ν_+ (ν_-) set upper bounds to the training error rates for the positive (negative) class, since training errors are all BSVs.

3.2 Biasing for uneven training class sizes

When training sets have different class sizes, we will require the positive and negative bounds to be similar. This is achieved by setting $\nu_+ = \nu_-$ giving

$$\left(1 + \frac{1}{C_{+/-}R_{+/-}}\right) = \left(1 + C_{+/-}R_{+/-}\right)$$

which results in

$$C_{+/-} = \frac{1}{R_{+/-}} = \frac{l_-}{l_+}. \tag{39}$$

This relationship concurs with the results obtained by Chew *et al.* [2] in setting the ratios of Cs in the context of the original SVM. With Dual ν-SVM, the selection of error penalties is based on the training class sizes and does not require a large number of iterations to determine. Thus the effectiveness and efficiency of generating SVMs are improved further by reducing the number of trainings required.

4 CONCLUSION

We have shown that ν-SVM does have undesirable effects with trained with set of uneven class sizes. The effects are similar in the original SVM algorithm with a single error penalty.

We introduced the Dual ν-SVM allowing different bounds for each of the classes, as well as compensating for the uneven training class size effects.

As with the ν-SVM, the reparameterisation of C to ν enables the error factor to be chosen easily as it is just a factor of the number of training data points. This reduces the number of trainings required because there is no need to search for the absolute value of C that works for the problem at hand. The lower bound on the number of support vectors provided by ν ensures the resulting SVM will be able to generalise well.

With the error factor separated to each class, we can vary the performance of the resulting SVM by adjusting relative error factor to account for the costs involved with errors. In particular, in target detection, we would want to set the maximum false alarm rate to a certain value, which is selected by using ν_-.

Again with a different ν for each class, we have shown that we can remove the effects of training sets with adifferent set size for each class. By setting the ratio of Cs to the inverse ratio of training class sizes, the obtained SVM will exhibit the desired performance of similar error rates in both classes.

The modification to obtain the Dual ν-SVM provides a substantial improvement in training an SVM quickly and effectively, while still retaining the ability to fine-tune the required error weights.

ACKNOWLEDGEMENTS

We are pleased to acknowledge stimulating discussions with Chris Burges of Bell Labs, Lucent Technologies, David Crisp of DSTO, Australia, and Steve Gunn of The University of Southampton, UK.

REFERENCES

[1] C.J.C. Burges, "A tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, vol. 2, no. 2, 1998.

[2] H.G. Chew, D.J. Crisp, R.E. Bogner, and C.C. Lim, "Target Detection 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.

[3] C. Cortes, and V. Vapnik, "Support Vector Networks", Machine Learning, vol. 20, pp.1-25, 1995.

[4] R. Fletcher, "Practical Methods of Optimization" John Wiley and Sons, Inc., 2nd edition, 1987.

[5] B. Schölkopf, A. Smola, R. Williamson, and P.L. Bartlett, "New Support Vector Algorithms" NeuroCOLT Technical Report NC-TR-98-031, Royal Holloway College, University of London, UK, 1998.

[6] V. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 1995.